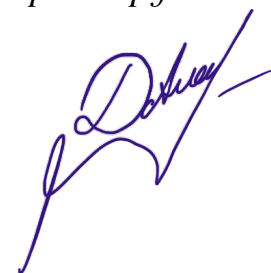


АО «НАУЧНО-ИССЛЕДОВАТЕЛЬСКИЙ ИНСТИТУТ ВЫЧИСЛИТЕЛЬНЫХ
КОМПЛЕКСОВ ИМ. М. А. КАРЦЕВА»

На правах рукописи



АЛАДИН ДМИТРИЙ ВЛАДИМИРОВИЧ

**МИВАРНЫЕ МОДЕЛИ И МЕТОДЫ ИНТЕЛЛЕКТУАЛЬНОГО
ПЛАНИРОВАНИЯ ДЕЙСТВИЙ ПРИ ПРИНЯТИИ УПРАВЛЕНЧЕСКИХ
РЕШЕНИЙ В ТЕХНИЧЕСКИХ СИСТЕМАХ**

2.3.1. «Системный анализ, управление и обработка информации, статистика»

Диссертация на соискание ученой степени
кандидата технических наук

Научный руководитель:
доктор технических наук, профессор
Варламов Олег Олегович

Москва – 2026

ОГЛАВЛЕНИЕ

Введение	5
1 Обзорно-аналитическое исследование подходов автоматического и интеллектуального планирования действий	13
1.1 Постановка задачи исследования	13
1.2 Автоматическое и интеллектуальное планирование.	14
1.2.1 Планирование и диспетчеризация	14
1.2.2 Формальное описание задачи планирования.	16
1.2.3 Краткая хронология развития подходов автоматического и интеллектуального планирования.	19
1.2.4 Краткая хронология развития языков планирования	20
1.2.5 Анализ автоматических и интеллектуальных планировщиков .	23
1.2.6 Анализ диссертационных исследований в области автоматического и интеллектуального планирования.	25
1.3 Миварные технологии логического искусственного интеллекта. . .	27
1.4 Структура проводимых исследований	32
1.5 Выводы по первой главе	34
2 Модели миварной базы знаний и миварной проблемно-ориентированной системы управления и принятия решений	36
2.1 Постановка задач разработки моделей	36
2.2 Миварный подход к накоплению и обработке информации	37
2.3 Модель миварной базы знаний для решения управленческих задач в пространстве состояний	45
2.4 Метаграфовый подход к моделированию сложных интеллектуальных информационных систем.	50
2.5 Модель миварной проблемно-ориентированной системы управления и принятия решений в пространстве состояний	55

2.6	Выводы по второй главе	59
3	Методы автоматической генерации миварных баз знаний и решения задач планирования действий в пространстве состояний	61
3.1	Постановка задач разработки методов	61
3.2	Метод автоматической генерации миварных баз знаний для решения управленческих задач в пространстве состояний	62
3.2.1	Обобщённый процесс автоматической генерации миварной базы знаний для решения задач в пространстве состояний	62
3.2.2	Подготовка исходных данных для автоматической генерации миварных баз знаний	64
3.2.3	Модель автоматического генератора миварных баз знаний	69
3.2.4	Анализ метода автоматической генерации миварных баз знаний для решения управленческих задач в пространстве состояний	72
3.3	Метод решения задач принятия решений и обработки информации для планирования действий в пространстве состояний	73
3.3.1	Описание метода решения задач принятия решений и обработки информации для планирования действий в пространстве состояний	73
3.3.2	Применение метода решения задач принятия решений и обработки информации для решения задач планирования домена «Мир кубиков»	79
3.4	Выводы по третьей главе	88
4	Программное обеспечение на основе моделей и методов планирования действий при принятии управленческих решений	90
4.1	Постановка задач разработки программного обеспечения	90
4.2	Математическое и алгоритмическое обеспечение обработки миварных сетей логических параметров и правил	91
4.2.1	Краткая хронология развития алгоритмов обработки миварных сетей логических параметров и правил.	91
4.2.2	Постановка требований к миварной машине логического вывода для решения задач планирования в пространстве состояний	98
4.2.3	Обобщённый алгоритм обработки миварной сети машиной логического вывода для решения задач планирования в пространстве состояний	102

4.2.4	Матричное представление алгоритма обработки миварной сети машиной логического вывода для решения задач планирования в пространстве состояний	107
4.3	Разработка комплекса программ на основе моделей и методов интеллектуального планирования действий в пространстве состояний . .	115
4.3.1	Логическая интеллектуальная система контроля за соблюдением правил дорожного движения	115
4.3.2	Миварная система планирования действий роботов и роботизированных средств	116
4.3.3	Логическая интеллектуальная система обеспечения ухода за растениями.	117
4.3.4	Программный комплекс на основе информационно-аналитической системы с поддержкой принятия решений об эффективности и безопасности применения термолабильных компонентов крови в медицинской практике	118
4.4	Экспериментальная проверка моделей и методов интеллектуального планирования действий в пространстве состояний	119
4.4.1	Условия эксперимента	119
4.4.2	Численные результаты эксперимента	121
4.4.3	Анализ соответствия расчётных и численных результатов. . .	129
4.5	Выводы по четвёртой главе	131
	Заключение	134
	Список сокращений и условных обозначений	137
	Список литературы	138
	Приложение А Пример применения моделей и методов планирования действий для решения управленческой задачи	161
	Приложение Б Пример обработки миварной сети машиной логического вывода при решении задач в пространстве состояний	178

ВВЕДЕНИЕ

Актуальность темы исследования. Актуальность темы исследования обусловлена стремительным развитием технологий искусственного интеллекта, робототехники и киберфизических систем. В современных условиях, когда технические системы становятся всё более автономными и требуют способности к адаптации и функционированию в реальном времени, проблема автоматического планирования действий при принятии управленческих решений приобретает особую значимость.

Существующие подходы к автоматическому планированию в пространстве состояний сталкиваются с высокой вычислительной сложностью и недостаточной скоростью обработки данных, что ограничивает их применение в автономных системах и системах управления в реальном времени. При построении моделей планирования существенную роль играет формализация знаний о предметной области. Высокая трудоёмкость разработки и сопровождения таких моделей затрудняет создание интеллектуальных систем, способных функционировать в динамичных условиях.

В связи с этим особую актуальность приобретает разработка новых моделей и методов интеллектуального планирования действий, направленных на снижение вычислительной сложности и повышение скорости обработки данных при принятии управленческих решений, обеспечивая тем самым надёжность и оперативность функционирования технических систем.

Перспективным направлением решения обозначенных проблем является использование миварных технологий логического искусственного интеллекта, основанных на линейно-вычислительной логике и продукционных правилах. Данные технологии обеспечивают формализацию знаний для роботизированных и робототехнических комплексов в виде миварных сетей и позволяют достигать линейной вычислительной сложности при построении логического вывода. Это делает воз-

возможным создание гибких и масштабируемых систем планирования, адаптируемых к динамическим изменениям внешней среды.

Таким образом, актуальность исследования обусловлена необходимостью разработки новых моделей и методов интеллектуального планирования, способных обеспечить более высокую производительность при решении задач автоматического планирования в технических системах. Работа выполнена в соответствии с научным направлением разработки математического и алгоритмического обеспечения интеллектуальных систем планирования и управления техническими объектами.

Степень разработанности темы. Исследования в области автоматического планирования имеют фундаментальную теоретическую основу, заложенную в работах основоположников данного направления. Значительный вклад в развитие теории автоматического планирования внесли R.E. Fikes и N.J. Nilsson, разработавшие автоматический планировщик и формальный язык STRIPS, а также J. McCarthy, A. Tate, V. Kumar, A. Blum, M. Furst, H. Kautz и В. Selman, развившие формализмы планирования и языки описания предметных областей. Вопросы вычислительной сложности задач планирования исследовались в трудах Т. Bylander, N. Gupta, D. Nau, S. Chenoweth и G. Sussman, доказавших NP-полноту ряда постановок задач планирования и ограничения STRIPS-подобных алгоритмов. Существенный прогресс в развитии методов планирования и языков описания действий и домена связан с работами E. Pednault, предложившего язык ADL, и D. McDermott с соавторами, разработавшими язык PDDL.

Современные тенденции развития интеллектуальных планировщиков связаны с применением методов машинного обучения и нейронных сетей, исследуемых в работах E. Onaindia, В. Jerbić и S. Thiébaux. Отечественные исследования в области автоматического планирования охватывают широкий спектр направлений и представлены, в частности, трудами Г.С. Осипова, А.И. Панова и Г.В. Рыбиной.

Проблематика автоматического планирования получила развитие в ряде диссертационных исследований И.Н. Габдрахманова, А.А. Алимова, Б.С. Юдинцева, Ю.М. Блохина и Чинь Суан Лонга. Миварным технологиям логического искусственного интеллекта также посвящён ряд диссертационных работ таких авторов, как О.О. Варламов, Р.А. Санду, Д.А. Чувиков и О.В. Кривошеев.

Анализ работ по теме показал, что автоматическое планирование является ключевым элементом при создании робототехнических, транспортных и киберфи-

зических систем. В частности, особый интерес представляет планирование в пространстве состояний, которое применяется в интеллектуальных системах. Обзор позволил выявить ограничения существующих подходов, обусловленные высокой вычислительной сложностью и зависимостью от экспертной формализации знаний. Это подтверждает необходимость поиска новых подходов к решению задач планирования.

Определены направления дальнейших исследований, связанные с автоматизацией генерации баз знаний, снижением вычислительных затрат и повышением адаптивности систем интеллектуального планирования. В настоящем исследовании для решения выявленных проблем применяются миварный и метаграфовый подходы, обеспечивающие формализацию знаний о предметной области, их обработку, а также создание производительных и масштабируемых интеллектуальных систем управления.

Объектом исследования являются интеллектуальные системы планирования действий при принятии управленческих решений в технических системах.

Предметом исследования являются модели и методы интеллектуального планирования действий в пространстве состояний.

Цель и задачи исследования. Целью диссертационного исследования являются снижение вычислительной сложности задач интеллектуального планирования действий в пространстве состояний и ускорение процесса их обработки при принятии управленческих решений в технических системах.

Для достижения поставленной цели решены следующие частные научные задачи:

- 1) анализ существующих подходов автоматического и интеллектуального планирования действий в пространстве состояний и выявление их ограничений и определение направлений их совершенствования;
- 2) разработка и исследование моделей миварной базы знаний и миварной проблемно-ориентированной системы управления и принятия решений;
- 3) разработка методов автоматической генерации миварных баз знаний и решения задач планирования действий в пространстве состояний;
- 4) разработка математического и алгоритмического обеспечения на основе разработанных моделей и методов;
- 5) проведение эксперимента по оценке производительности математического и алгоритмического обеспечения, реализующего предложенные модели и мето-

ды.

Методология и методы исследования. Для решения поставленных задач в исследовании используются методы системного анализа, экспертного моделирования, принятия решений, теории информации, теории самообучающихся технических систем и теории графов. Общей методологической основой является системный подход.

Научная новизна. В диссертации получены следующие результаты, характеризующиеся научной новизной:

1) модель миварной базы знаний для решения управленческих задач в пространстве состояний, характеризующаяся структурной декомпозицией знаний по множествам правил и параметров, обеспечивающей возможность автоматической генерации, масштабирования и параллельной обработки информации;

2) модель миварной проблемно-ориентированной системы управления и принятия решений в пространстве состояний, для описания которой используются миварный и метаграфовый подходы, что делает возможным представление системы в виде сложного графа взаимосвязанных элементов и обеспечивает формализацию интерфейсов её взаимодействия с другими информационными системами;

3) метод автоматической генерации миварных баз знаний для решения управленческих задач в пространстве состояний, основанный на интеграции миварного и метаграфового подходов с целью обеспечения адаптации систем управления и принятия решений к изменениям предметной области посредством динамического масштабирования баз знаний в зависимости от количества управляемых объектов;

4) метод решения задач принятия решений и обработки информации для планирования действий в пространстве состояний, использующий циклическую обработку миварных сетей для автоматической генерации частных планов действий и их адаптации в соответствии с текущим состоянием предметной области;

5) математическое и алгоритмическое обеспечение, реализующее параллельную обработку логических правил в миварной машине логического вывода с механизмом выборочного вычисления выходных параметров при активации правил, что обеспечивает снижение вычислительной сложности и ускорение построения плана действий в интеллектуальных системах управления.

Теоретическая значимость заключается в том, что новые предлагаемые мо-

дели миварной базы знаний и миварной проблемно-ориентированной системы и методы автоматической генерации миварных баз знаний и решения задач планирования действий в пространстве состояний могут быть применены при подготовке информационных моделей систем планирования действий и для исследования и анализа их организационной структуры в рамках концепции гибридных интеллектуальных информационных систем.

Практическая значимость работы заключается в снижении вычислительной сложности задач интеллектуального планирования действий в пространстве состояний и ускорении их обработки при принятии управленческих решений в технических системах. В соответствии с предлагаемыми моделями и методами разработана миварная машина логического вывода в виде программного обеспечения, на основе которой предлагается создавать системы интеллектуального планирования действий. Предложены варианты применения полученных результатов для разработки систем поддержки управления и принятия решений транспортных средств и автономных робототехнических комплексов различного базирования и назначения, а также систем поддержки принятия решений в медицине и сельском хозяйстве.

Достоверность результатов обеспечивается применением известных математических методов при разработке моделей и методов, а также подтверждается результатами аналитических расчётов и вычислительных экспериментов.

Положения, выносимые на защиту:

1) модель миварной базы знаний для решения управленческих задач в пространстве состояний обеспечивает структурную декомпозицию знаний по множествам правил и параметров и предоставляет возможность автоматической генерации баз знаний, их масштабирования и параллельной обработки;

2) модель миварной проблемно-ориентированной системы управления и принятия решений в пространстве состояний позволяет представлять систему в виде сложного графа взаимосвязанных элементов и обеспечивает формализацию интерфейсов её взаимодействия с другими информационными системами;

3) метод автоматической генерации миварных баз знаний для решения управленческих задач в пространстве состояний обеспечивает адаптацию систем управления и принятия решений к изменениям предметной области посредством динамического масштабирования баз знаний в зависимости от количества управляемых объектов;

4) метод решения задач принятия решений и обработки информации для планирования действий в пространстве состояний позволяет применять миварные сети для автоматической генерации частных планов действий и их адаптации к текущему состоянию предметной области;

5) математическое и алгоритмическое обеспечение обеспечивает снижение вычислительной сложности и ускорение построения плана действий в интеллектуальных системах управления.

Апробация работы. Основные результаты диссертационной работы докладывались и обсуждались на следующих конференциях и семинарах: Конгресс по интеллектуальным системам и информационным технологиям «IS&IT» (Дивноморское, 2017 г., 2018 г.); Международный научно-технический конгресс «Интеллектуальные системы и информационные технологии» («ИС & ИТ», «IS&IT») (Дивноморское, 2019 г., 2020 г., 2023 г., 2024 г.); Студенческая научная весна (Москва, 2018 г.); Модели мышления и интеграция информационно-управляющих систем (ММИИУС-2018) (Нальчик, 2018 г.); Международный автомобильный научный форум (МАНФ-2018) «Технологии и компоненты интеллектуальных транспортных систем» (Москва, 2018 г.); XXXI Международная инновационная конференция молодых ученых и студентов по проблемам машиноведения (МИКМУС - 2019) (Москва, 2019 г.); Фундаментальные основы физики, химии и механики наукоёмких технологических систем формообразования и сборки изделий (Ростов-на-Дону, 2020 г.); III Всероссийская научная конференция «Интеллектуальные технологии и проблемы математического моделирования» (Ростов-на-Дону, 2020 г.); Международный автомобильный научный форум (МАНФ-2020) «Наземные интеллектуальные транспортные средства и системы», Автонет-2020 «Форум инновационных транспортных технологий» (Москва, 2020 г.); Искусственный интеллект в автоматизированных системах управления и обработки данных (Москва, 2022 г., 2023 г., 2024 г.); Международная научная конференция молодежной школы «МИВАР» (Москва, 2022 г., 2023 г., 2024 г., 2025 г.); Международная молодёжная конференция по радиоэлектронике, электротехнике и энергетике (REEPE) (Москва, 2025 г.).

Реализация и внедрение результатов работы. Результаты диссертации использованы при реализации гранта на выполнение научно-исследовательских работ и оценку перспектив коммерциализации в рамках инновационного проекта № 13554ГУ/2018 от 23.07.2018 г., предоставленного Федеральным государствен-

ным бюджетным учреждением «Фонд содействия развитию малых форм предприятий в научно-технической сфере» (Фонд содействия инновациям), и внедрены в практическую деятельность ООО «НПИ ФИРМА «ГИПЕРИОН», ООО «Интерпром», ООО «Окологика», а также в образовательный процесс кафедры «Системы обработки информации и управления» МГТУ им. Н.Э. Баумана.

Соответствие паспорту специальности. Содержание диссертации соответствует п. 3 «Разработка критериев и моделей описания и оценки эффективности решения задач системного анализа, оптимизации, управления, принятия решений, обработки информации и искусственного интеллекта», п. 4 «Разработка методов и алгоритмов решения задач системного анализа, оптимизации, управления, принятия решений, обработки информации и искусственного интеллекта», п. 5 «Разработка специального математического и алгоритмического обеспечения систем анализа, оптимизации, управления, принятия решений, обработки информации и искусственного интеллекта» паспорта специальности 2.3.1. «Системный анализ, управление и обработка информации, статистика».

Публикации. По материалам диссертации опубликованы 73 печатные работы [1–73], в том числе: 7 статей [1–7] опубликовано в изданиях, рекомендованных ВАК; 12 публикаций [8–19] в изданиях, индексируемых в базах «Scopus» и «Web of Science»; 1 монография [25] и 5 свидетельств о государственной регистрации программ для ЭВМ [20–24].

Основные публикации по теме диссертации представлены 25 работами [1–25], из которых 4 подготовлены без соавторов [1–3, 20]. В работах, опубликованных в соавторстве, лично соискателем предложены: [4, 9, 10, 12, 13] – формализация знаний в виде миварных сетей в соответствии с моделью миварной базы знаний для решения управленческих задач в пространстве состояний, а также создание систем управления и принятия решений, обеспечивающих их обработку; [7, 18, 21, 22] – использование модели миварной базы знаний для решения управленческих задач в пространстве состояний и создание системы поддержки принятия решений, реализующей обработку баз знаний, построенных на её основе; [5, 6, 8, 11, 15] – метод решения задач принятия решений и обработки информации для планирования действий в пространстве состояний; [14] – архитектура системы управления и принятия решений с применением миварных технологий логического искусственного интеллекта; [16] – способ предварительной подготовки знаний для формализации в виде миварных сетей; [17] – применение модели миварной

базы знаний и метода решения задач принятия решений и обработки информации в контуре управления с циклическим и календарным планированием процессов; [19] – алгоритмы работы машин логического вывода и их архитектуры; [23, 24] – реализация математического и алгоритмического обеспечения для обработки мультимедийных баз знаний; [25] – обоснование и способы применения моделей и методов планирования действий при решении задач в пространстве состояний.

Структура и объем работы. Диссертация состоит из введения, четырёх глав, заключения, списка сокращений и условных обозначений, списка литературы из 179 наименований и 2 приложений. Работа изложена на 183 страницах машинописного текста (основной текст занимает 136 страниц, содержит 44 рисунка и 9 таблиц).

1 ОБЗОРНО-АНАЛИТИЧЕСКОЕ ИССЛЕДОВАНИЕ ПОДХОДОВ АВТОМАТИЧЕСКОГО И ИНТЕЛЛЕКТУАЛЬНОГО ПЛАНИРОВАНИЯ ДЕЙСТВИЙ

1.1 Постановка задачи исследования

Автоматическое планирование – одно из ключевых направлений в области систем управления и автоматизации, в рамках которого исследуются методы формирования последовательности действий для преобразования состояния объектов в требуемое. Каждое действие совершается в условиях ограничений и особенностей предметной области (ПрО), а совокупность знаний о ней – **модель ПрО** – определяет полноту и корректность процесса планирования.

Классические подходы к автоматическому планированию [74–76] продемонстрировали значительные теоретические и практические достижения, обеспечив основу для современных планировщиков. Однако по мере роста сложности задач и появления автономных технических систем, работающих в динамической среде, возникла потребность в подходах, способных адаптироваться к изменяющимся условиям и использовать накопленные знания.

Развитие вычислительных технологий и методов обработки данных способствовало появлению направления планирования, в котором система способна анализировать поступающую информацию и принимать решения на основе технологий искусственного интеллекта (ИИ). Это направление, обозначаемое как **интеллектуальное планирование**, рассматривает процесс построения плана не только как формальное вычисление последовательности действий, но и как деятельность системы, включающая анализ текущей ситуации, выбор стратегии поведения и адаптацию к изменяющимся условиям среды. Это особенно актуально для работо-

технических комплексов (РТК) и других автономных систем, функционирующих в реальном времени, где важна не оптимальность, а своевременность и устойчивость принимаемых решений.

В связи с этим возникает задача поиска путей снижения вычислительных затрат и повышения адаптивности систем интеллектуального планирования, в частности, совершенствования моделей представления знаний и методов организации вычислительного процесса при решении задач в технических системах.

Целью настоящей главы является проведение обзорно-аналитического исследования подходов автоматического и интеллектуального планирования действий, систематизация их основных принципов, формализмов и языков описания, а также выявление ограничений и проблем их применения в автономных технических системах, в том числе в условиях повышенных требований к быстродействию и ограниченных вычислительных ресурсов.

1.2 Автоматическое и интеллектуальное планирование

1.2.1 Планирование и диспетчеризация

При разработке систем управления для таких технических комплексов, как робототехнические средства, автономные роботы и беспилотные транспортные средства, особое внимание уделяется разграничению задач автоматического планирования и диспетчеризации [77]:

- **Планирование** (англ. *planning*) направлено на построение последовательности действий (плана), обеспечивающей достижение поставленной цели задачи в условиях заданного множества возможных операций и ограничений ПрО. Такие задачи отличаются высокой вычислительной сложностью и могут относиться к классу сложности P-SPACE или даже быть алгоритмически неразрешимыми.

- **Диспетчеризация** (англ. *scheduling*) ориентирована на организацию выполнения запланированных действий наиболее эффективным образом (создание расписания) с учётом доступных ресурсов и ограничений ПрО. Как правило, задачи данного типа относятся к классам сложности P и NP.

При решении задач планирования возможна оптимизация по одному или нескольким критериям, что существенно повышает вычислительную сложность и временные затраты. Помимо этого, существует ряд задач, которые требуют высокой скорости формирования решений. В этой связи различают:

- **Осуществимое планирование** – поиск плана без оптимизационных и иных условий.
- **Оптимальное планирование** – выбор наилучшего плана из множества возможных альтернатив с учётом оптимизационных критериев и иных факторов.

Стремление сделать оптимальный план может быть обусловлено экономическими или коммерческими выгодами. В ситуациях, когда ресурсы ограничены, оказывается, что получить приблизительно правильное или неоптимальное решение зачастую значительно эффективнее и быстрее, нежели строго оптимальное [78]. В системах реального времени требуется искать баланс между осуществимостью и оптимальностью процесса планирования. В идеале такие системы должны самостоятельно оценивать собственные ресурсы и ожидаемое время решения задачи, чтобы выбрать наиболее подходящий алгоритм планирования.

Следует отметить, что автоматическое планирование для технических задач может рассматриваться на различных уровнях абстракции. Такое деление позволяет выделить отдельные аспекты процесса планирования в зависимости от его целей и ПрО:

- **Уровень планирования задач и/или действий** – определяет логическую последовательность шагов для достижения целевого состояния системы.
- **Уровень планирования траекторий и/или движения** – построение конкретных пространственно-временных траекторий движения исполнительных органов технической системы в физическом пространстве при учёте геометрических и кинематических ограничений.

В настоящем исследовании рассматривается осуществимое планирование действий.

1.2.2 Формальное описание задачи планирования

Существуют различные подходы к автоматическому планированию, включая планирование в пространстве состояний, планирование в пространстве планов, применение техники прямого распространения ограничений и планирование на основе прецедентов [79]. Первым планировщиком, реализующим планирование в пространстве состояний, был STRIPS (англ. *Stanford Research Institute Problem Solver* – Решатель задач Стэнфордского исследовательского института) [74], разработанный для решения задачи формирования плана поведения робота, перемещающего предметы между помещениями [80]. Впоследствии термин STRIPS стал использоваться также для обозначения формального языка описания моделей мира, с которым работает планировщик. Идея алгоритма работы STRIPS заимствована из системы GPS (англ. *General Problem Solver* – Универсальный решатель задач) [81], в которой применялся метод анализа средств и целей (англ. *means-ends analysis*) [82]. Этот метод предполагает выбор действий, непосредственно связанных с достижением цели. В STRIPS действия выполняются последовательно, что позволяет достигать каждую цель задачи по отдельности.

В рамках системы STRIPS задача планирования определяется тремя сущностями [74]:

- начальное состояние ПрО S_n ;
- конечное множество действий (операторов) G , включая предварительные условия (предусловия) для их активации и описание их эффектов (постусловия) на модель мира;
- целевое состояние ПрО S_c .

Подробнее остановимся на том, с чем взаимодействует планировщик. При описании задач планирования часто упоминается термин **домен планирования** (англ. *planning domain*). В качестве формального определения данного термина можно привести следующую пару [83]:

$$P = \langle S_n, G, L \rangle, \quad (1.1)$$

где L – набор ограничений ПрО.

Иными словами, домен планирования состоит из описаний:

- видов действий, которые допустимы в заданной модели мира;
- динамики и законов мира;
- набора условий, которые необходимо соблюсти;
- возможных видов отношений между объектами.

Описание самих объектов модели мира в данный термин не входит.

Задачу планирования в таком случае можно представить в виде кортежа [79]

$$T = \langle S_n, O, L, S_c \rangle. \quad (1.2)$$

Решением задачи планирования T будет являться последовательность действий (план), выполняя которые можно перевести ПрО из состояния S_n в S_c . Полученный план исполняется активным агентом в ПрО, например, роботом или роботизированной рукой-манипулятором.

Подробнее остановимся на представлении концептуальной модели ПрО (КМПрО). В работе Л.С. Болотовой [84] описывается **КМПрО через знаковую систему** [85]:

$$\text{КМПрО} = \langle X, C, R, G \rangle, \quad (1.3)$$

где X – множество имён (знаков) объектов среды ПрО, C – множество их свойств (признаков), R – множество отношений между ними и G – множество имён (знаков) действий, допустимых над объектами множества X . Эти множества образуют язык описания ПрО.

С учётом (1.3) для решения задач в КМПрО используют понятия состояние ПрО и пространство состояний. Согласно [84], **состояние ПрО** в момент времени t представляется кортежем

$$S_{\text{ПрО}}(t) = \langle X(t), C(t), R(t) \rangle. \quad (1.4)$$

Указание момента времени t подчёркивает динамический характер среды ПрО. При выполнении некоторого действия g_k из множества G осуществляется переход из одного состояния $S_i(t)$ в другое $S_j(t)$:

$$S_i(t) \xrightarrow{g_k} S_j(t). \quad (1.5)$$

Множество всех возможных состояний образует **пространство состояний ПрО**. Переход из одного состояния в другое может осуществляться путём применения нескольких действий из множества G , причём такие переходы зачастую могут быть представлены различными последовательностями действий.

Основываясь на определении (1.2), задачу планирования для КМПрО, рассматриваемой как знаковая система, можно представить кортежем

$$Z = \langle S_H, G, S_C \rangle, \quad (1.6)$$

где S_H – начальное состояние ПрО, S_C – целевое состояние ПрО.

Решением задачи планирования является последовательный набор действий (план действий), применение которого к среде ПрО, начиная с состояния S_H , приводит её в состояние S_C :

$$S_H \xrightarrow{g_1} S_1 \xrightarrow{g_2} \dots \xrightarrow{g_k} S_C, \quad (1.7)$$

где $\{g_1, g_2, \dots, g_k\} \subseteq G$. Следует отметить, что допустимых планов может быть несколько. Если вводится система критериев для сопоставления различных планов, задача рассматривается как задача оптимального планирования; при отсутствии критериев – осуществимое планирование.

Рассмотренное выше формальное описание задачи планирования задаёт теоретическую основу для построения алгоритмов поиска плана, однако оно не определяет конкретные механизмы решения таких задач и не раскрывает всего спектра возникающих при этом вычислительных и методологических трудностей. Появление формализаций типа GPS и STRIPS, а также дальнейшие исследования особенностей и трудоёмкости автоматического планирования продемонстрировали необходимость разнообразных подходов к представлению ПрО и организации процесса поиска решений. В этой связи для обоснования выбора применяемого в данной диссертационной работе направления исследований и понимания его места в контексте существующих разработок целесообразно проследить хронологию развития подходов автоматического и интеллектуального планирования, а также соответствующих языков планирования и планировщиков.

1.2.3 Краткая хронология развития подходов автоматического и интеллектуального планирования

Кратко рассмотрим историю развития подходов автоматического и интеллектуального планирования. Детальные исследования историко-методологического аспекта автоматического и интеллектуального планирования представлены в работах [79, 86].

Начало исследований в области интеллектуального планирования связано с работами, основанными на доказательстве теорем [79]. При таком планировании решение задачи сводилось к процессу поиска доказательства достижимости цели при заданных предпосылках. Подходы, основанные на доказательстве теорем, имеют ряд таких ограничений, как крайне низкая производительность и наличие проблем применения знаний о прошлом для построения выводов о будущем (проблема фрейма). Одним из вариантов преодоления данных ограничений стали подходы, основанные на планировании в пространстве состояний [87], где поиск решения осуществляется через последовательные преобразования исходного состояния ПрО в целевое с использованием набора допустимых операторов. Тем не менее планирование в пространстве состояний во многих случаях оставалось недостаточно гибким, что привело к появлению подходов, в которых задача планирования стала рассматриваться как задача поиска в пространстве частично-упорядоченных планов [88].

Вместе с развитием частично-упорядоченных планов возникло направление иерархического планирования [89]. Этот подход использует иерархию подцелей для упрощения процесса решения задач: сначала подготавливается общий план, затем он детализируется в порядке убывания иерархии. Это позволяет перераспределить вычислительные мощности на решение приоритетных подзадач.

С появлением алгоритмов решения задач удовлетворения ограничений (англ. *constraint satisfaction problem*) и проверки истинности в пропозициональной логике (англ. *propositional satisfiability problem*) подходы планирования также получили своё развитие [90]. Одним из приобретённых качеств планировщиков стало повышение скорости подготовки планов. В то же время появились работы, рассматривающие задачи планирования с точки зрения целочисленного линейно-

го программирования (англ. *inductive logic programming*) [91]. Также развивался подход, основанный на построении бинарных диаграмм решений (англ. *binary decision diagram*) [92].

Эвристические методы также нашли своё применение при создании планировщиков [79]. Данные методы существовали задолго до появления автоматического планирования, однако их использование для решения подобных задач стало возможным благодаря развитию механизмов автоматизированного извлечения эвристики из описаний ПрО. Идейными продолжателями этих механизмов стали методы, основанные на машинном обучении.

Развитие методов и инструментов обучения свёрточных и рекуррентных нейронных сетей вызвало у исследователей интерес к применению глубоких нейронных сетей в задачах интеллектуального планирования [93, 94]. Кроме того, стали активно изучаться возможности и других подходов, включая обучение с подкреплением [95] и использование условных случайных полей CRF (англ. *Conditional Random Fields*) [96]. В работе [97] авторы получили архитектуру STRIPS-HGN для обучения по гиперграфам, которая специализируется на изучении эвристики планирования. Особенно стоит отметить исследования в области алгоритмов планирования, которые применяют модель знакового мира в качестве основы для приобретения и поддержания знаний для будущего использования в планировании поведения [98, 99].

1.2.4 Краткая хронология развития языков планирования

Под STRIPS понимают не только планировщик Р. Файкса и Н. Нильсона (авторы STRIPS [74]), но и язык описания действий и домена. В начале развития STRIPS использовался для задач планирования, в которых ПрО определялась как статическое состояние. Такое состояние изменяется только одним агентом, который посредством каждого действия переводит систему из одного состояния в другое. Такая формулировка задачи с простым состоянием служила основой для исследований автоматического планирования на протяжении продолжительного про-

межутка времени, большая часть из которых была основана именно на структуре представления и методах рассуждения, разработанных для STRIPS [100].

В 1994 году Том Байлендер провёл исследование [101], в котором показал, что для обеспечения полиномиальной или даже NP-полной сложности [102] процесса планирования требуются чрезвычайно жёсткие ограничения как на операторы, так и на формулы, используемые в алгоритмах планирования и описания домена. При этом поиск оптимального плана для задач в доменах Blocks World (с англ. *Мир Блоков*) является NP-полным [103, 104]. Кроме того, разработка планировщика осложняется поиском приемлемого языкового средства описания домена и ПрО. Например, в 1973 году была обнаружена некорректность алгоритмов STRIPS при работе с доменом Blocks World в виде аномалии Сассмана (англ. *Sussman anomaly*) [88].

Развитие работ в области автоматического планирования показало, что STRIPS имеет ряд существенных недостатков, которые не позволяют эффективно описывать сложные ПрО. Тем не менее, имея ограничения, этот язык позволил добиться начального прогресса в решении общей проблемы автоматического планирования.

Следующим важным этапом в развитии формализмов описания ПрО для задач планирования стало появление языка определения действия ADL (англ. *Action Definition Language*) [75]. Данный язык расширил выразительные возможности STRIPS за счёт поддержки более сложных предусловий и состояний. Он некоторое время активно применялся в исследованиях и послужил основой для дальнейшего развития языков представления действий.

По мере роста числа прикладных задач и разработанных формализмов возникла необходимость стандартизации способов описания доменов планирования. В результате был предложен язык определения домена планирования PDDL (англ. *Planning Domain Definition Language*), разработанный группой авторов [76] и впервые представленный на первых международных соревнованиях планировщиков IPC (англ. *International Planning Competition*). Также активно развивались формы планирования на основе декларативного программирования, например, программирование наборов ответов ASP (англ. *Answer Set Programming*) [79].

PDDL – это ориентированный на действие язык, испытавший влияние классических формулировок задач планирования (STRIPS и ADL) [76]. В его основе лежит простая стандартизация синтаксиса для выражения семантики действий с

использованием предварительных и постусловий для описания их применимости и последствий. Язык PDDL имеет модульную структуру. Он разбит на подмножества функций, называемых требованиями. Каждый домен, определенный с использованием PDDL, должен указывать, какие требования он принимает. Планировщик, который не обрабатывает данные требования, может пропустить все определения, связанные с этим доменом.

PPDDL (англ. *Probabilistic PDDL* – вероятностный язык определения домена планирования) стал первым шагом на пути к общему языку для описания вероятностных и теоретико-решающих задач планирования [86]. По своей сути, он является синтаксическим расширением PDDL 2.1. В отличие от PDDL, который задаёт фиксированную структуру плана вывода, PPDDL практически не ограничивает способы представления плана, за исключением требования о выполнении не более одного действия в единицу времени. Проблема представления плана полностью оставлена на усмотрение систем планирования. Более того, системы планирования могут даже отказаться от представления плана вообще.

PPDDL также уточняет интерпретацию целей в задачах планирования. В классической постановке PDDL цель описывается как логическое условие, которое должно быть достигнуто [86]. В PPDDL же цель трактуется как элемент оптимизационного критерия. Для задач, ориентированных на максимизацию вероятности достижения целевого состояния, планировщик стремится выбрать стратегию, обеспечивающую наибольшую вероятность достижения цели. В задачах же, связанных с вознаграждением, оптимизационный критерий определяется максимизацией ожидаемого суммарного вознаграждения, причём цели могут задаваться в виде множества поглощающих состояний (состояния, из которых нельзя выйти после перехода в них), для которых определены значения конечного вознаграждения.

RDDL (англ. *Relational Dynamic Influence Diagram Language* – язык реляционных динамических диаграмм влияния) был разработан для описания таких ПрО, для которых языки семейства (P)PDDL оказываются недостаточно выразительными [86]. Вместо дальнейшего расширения (P)PDDL, исследователи предложили принципиально новый язык, поскольку интеграция стохастических процессов, непрерывных параметров и параллельных действий затруднена в формализмах, основанных исключительно на описании эффектов действий. Центральный принцип RDDL заключается в том, что язык должен быть простым и единообразным,

а его выразительная сила проистекает из состава простых конструкций.

Детальная хронология развития рассмотренных языков планирования представлена в исследовании [86].

1.2.5 Анализ автоматических и интеллектуальных планировщиков

В исследовании [86] представлена хронология развития планировщиков, использующих ранее рассмотренные языки планирования. В работе [105] приведено подробное исследование методов создания современных планировщиков, появившихся благодаря достижениям в области машинного обучения. В частности, рассматриваются ключевые характеристики методов с точки зрения форматов входных и выходных данных, типов генерируемых планов и особенностей их вычислительной сложности. В работе также представлена классификация используемых методов машинного обучения и рассмотрены их преимущества и ограничения. После проведенного анализа делается вывод, что значительное количество методов на основе машинного обучения было применено к строго контролируемым экспериментальным ПрО и стендовым задачам, а для их полноценного применения в разнообразных и неопределенных сценариях реального мира требуется вести дальнейшие исследования. В продолжение данного обзора остановимся на вопросах производительности современных планировщиков, основанных на формализмах PDDL и ASP.

В работе [106] представлены эмпирические сравнения производительности планировщиков, использующих формализмы PDDL и ASP. Авторы отмечают, что PDDL-планировщики лучше работают с задачами, требующими более длинных решений (планы), а ASP-планировщики лучше работают с ПрО, в которых большое количество объектов и требуются сложные рассуждения о предусловиях и эффектах действий. Это подтверждается результатами тестирования планировщиков на домене Blocks World. Авторы работы выбрали планировщики Clingo [107] и FastDownward с настройками LAMA-2011 [108] как представителей формализмов ASP и PDDL соответственно. Эксперименты проводились на двух типах задач,

в которых требовалось собрать башню заданной структуры из набора блоков, расположенных на столе. В первой серии сравнение осуществлялось при варьировании числа объектов планирования; во второй – при различных способах задания условий цели (при фиксированном числе блоков, равном 10). Анализ результатов показал, что время решения в FastDownward увеличивается от приблизительно одной секунды при 15 блоках до порядка 1000 секунд при 60 блоках, однако при этом остаётся практически независимым от длины требуемого плана. В случае Clingo время решения при 60 блоках возрастает от менее чем одной секунды до порядка 1000 секунд при увеличении высоты конечной башни от 2 до 8 блоков. Результаты второй серии демонстрируют преимущества ASP-формализма при работе с ПрО, требующими сложных логических рассуждений на уровне предикатов домена.

Современные автоматические планировщики на основе формализма PDDL или ASP разрабатываются с расчетом на независимость от домена задачи планирования. Есть планировщики, которые нацелены на поиск оптимального плана, а есть те, которые ищут осуществимый план. Необходимо понимать, что в системах реального времени требуется принимать решение максимально быстро, и в некоторых ситуациях совершать управляющее воздействие в процессе планирования (когда полного плана действий еще нет). Увеличение количества оперируемых объектов в процессе планирования также несет потенциальный риск снижения скорости поиска решения. Кроме того, развитие формальных языков планирования происходит с целью достижения универсальности описания ПрО и оптимальности в функционировании автоматизированных планировщиков. Поэтому у таких языков проблема обеспечения простоты трансляции естественно-языковых конструкций на машинный уровень представления знаний не является приоритетной. Впрочем, такие ПрО существуют и требуют активного участия экспертов [109, 110]. Например, в системах беспилотного управления автомобилем совершение маневра должно выполняться согласно правилам дорожного движения (ПДД) [111]. При выработке набора действий для совершения маневра и в оценке действий водителя используются экспертные знания. Экспертные знания определяются умениями участников дорожного движения, законодательством, сводом правил эксплуатации транспортных средств и многими другими факторами. Для того чтобы эффективно представить информацию о такой ПрО, можно воспользоваться миварным подходом [4, 112].

1.2.6 Анализ диссертационных исследований в области автоматического и интеллектуального планирования

Вопросам применения автоматического и интеллектуального планирования в системах реального времени посвящено много научных трудов. Далее дан краткий аналитический обзор этих трудов.

В работе И.Н. Габдрахманова [113] дан подробный обзор классических методов и систем интеллектуального планирования и моделей ПрО планирования. В работе подробно рассматривается вопрос использования иерархических методов синтеза плана, основанных на рекурсивной декомпозиции. Для описания моделей было разработано расширение языка PDDL с целью адаптации синтаксиса под использование шаблонной декомпозиции задач. Это расширение зарекомендовало себя на примерах описания некоторых технологических процессов, однако может столкнуться с проблемой извлечения знаний для специализированных ПрО, характерной для моделей, подготовленных экспертом на PDDL-подобных языках.

В диссертационной работе А.А. Алимова [114] вопрос автоматического планирования рассматривается в рамках обеспечения общего процесса управления поведением многозадачных интеллектуальных агентов в системах реального времени. Разработка автоматического планирования в системе обеспечивает решение задач перемещения агентов в окружающем пространстве, построения маршрута, формирования поведения, ориентированного на цели, приоритизации целей агентов и других задач, направленных на максимизацию полезности агентов. В работе отмечается, что алгоритмы планирования и принятия решений могут работать достаточно продолжительное время, за которое объекты управления и их окружение изменяют свои состояния. Например, может произойти изменение относительного местоположения агентов друг относительно друга. Поэтому для класса методов автоматического планирования необходимо предъявлять требование быстрого отклика и высокой скорости подготовки плана управленческого воздействия. Для рассматриваемого класса мультиагентных систем планирование действий и принятие решений являются ключевыми задачами. Для обработки событий агентом и планирования действий в работе применяется метод динамического изменения структуры дерева поведения, основанный на продукционных правилах. Моде-

лирование ПрО в процессах автоматического планирования на производственных правилах более естественно для понимания, так как производственные правила подобны рассуждениям эксперта, позволяют достичь единообразия структуры знаний и гибкости внесения новых знаний. Вместе с тем производственный способ представления характеризуется рядом недостатков: сложностью логического вывода, обусловленной необходимостью проверки условий применимости правил; высокой трудоёмкостью управления процессом логического вывода; отсутствием наглядного представления иерархии структурных единиц знаний.

В области создания систем принятия решений для мобильных роботов имеется класс задач по планированию траектории движения. В работе Б.С. Юдинцева [115] рассматривается система синтеза плана бесконфликтных траекторий группировки мобильных роботов. Для создания такой системы были использованы нейронные сети Хопфилда. В работе Чинь Суан Лонг [116] рассматриваются методы построения интеллектуальных систем планирования для мобильных роботов в неизвестных средах. Созданные методы основываются на сочетании нечеткой логики и нейронных сетей.

Помимо расширения сфер практического применения методов интеллектуального планирования, также происходит совершенствование инструментов создания интеллектуальных программных средств. Примером такой работы может служить диссертация Ю.М. Блохина [117]. Для создания системы в работе использовался планировщик в пространстве состояний для решения задач планирования действий когнитолога в процессе разработки интеллектуальных экспертных систем. Важно отметить, что инструменты создания интеллектуальных экспертных систем позволяют разрабатывать системы автоматического конструирования алгоритмов функционирования мультиагентных робототехнических систем, иными словами, планировать поведение роботов [118].

1.3 Миварные технологии логического искусственного интеллекта

Наряду с разнообразными подходами автоматического и интеллектуального планирования, рассмотренными выше, особый интерес представляет направление исследований, основанное на миварных технологиях логического искусственного интеллекта (МТ ЛИИ) [119]. В рамках этого направления предложена концепция интеллектуализации РТК [120], в которой процессы управления и планирования представлены единым комплексом логико-вычислительных задач. Решение таких задач основывается на знаниях, формализованных в виде расширенного формализма продукционных правил типа «если . . . , то . . . », и механизме их обработки с линейной вычислительной сложностью [121]. Благодаря такой интеграции планирование действий автономных и групповых РТК трактуется как частный случай логического вывода по миварной базе знаний (МБЗ) [122]. Далее подробнее остановимся на анализе работ, касающихся планирования с использованием МТ ЛИИ.

Основополагающими работами по направлению МТ ЛИИ являются работы О.О. Варламова, посвящённые разработке интерактивных самоорганизующихся баз данных и правил и быстродействующих методов обработки информации [123, 124]. В данных работах было предложено представлять знания о ПрО в виде эволюционирующих структур, которые могут изменяться и дополняться в процессе эксплуатации системы без изменений в уже накопленной информации. Такой подход заложил основу для перехода от статических продукционных баз знаний к адаптивным моделям, ориентированным на решение задач управления в динамических средах.

Дальнейшее развитие этих идей привело к формированию миварной концепции динамического многомерного объектно-системного эволюционного унифицированного представления данных и правил (миварный подход) [123]. В рамках этой концепции информация о вещах, их свойствах и отношениях рассматривается как элементы единого многомерного дискретного пространства, а процедуры вывода представляют собой вычислительные преобразования над этой структурой. Это позволило предложить архитектуру интеллектуальных систем, в которых процессы накопления и обработки данных основываются на едином унифицированном формализме.

В последующем развитие концепции привело к введению в научный оборот **миварных технологий накопления и обработки информации** (МТ ЛИИ) [125]. Эти технологии описывают полный цикл работы со знаниями – от их формализации и структуризации в виде миварных сетей до реализации высокопроизводительных алгоритмов логического вывода с линейной вычислительной сложностью. Разработанный аппарат стал методологической основой для построения **машин логического вывода** (МЛВ) и **миварных экспертных систем** (МЭС), используемых, в том числе, для решения задач интеллектуального планирования действий и поддержки принятия управленческих решений в технических системах [126].

Миварная технология накопления информации представляет собой методологическую основу формирования глобальных эволюционных баз данных и знаний с адаптивно изменяемой структурой, опирающейся на унифицированное миварное представление «Вещь—Свойство—Отношение» и обеспечивающей структурирование знаний в миварном пространстве и миварной сети (в терминах параметров и правил). В свою очередь, **миварная технология обработки информации** предлагает способ построения логического вывода, реализующий автоматическое конструирование алгоритмов из модулей, сервисов или процедур на основе активной обучаемой миварной сети правил с линейной вычислительной сложностью [125].

Обзор возможностей применения МТ ЛИИ для интеллектуализации РТК приведён в монографии [25]. Отдельного внимания заслуживают исследования, демонстрирующие специфику использования данных технологий для решения задач управления и планирования. В работе [127] была предложена система управления офисным сервисным роботом, в которой задача планирования сводилась к автоматической подготовке алгоритма действий для перемещения между произвольными рабочими местами. Карта помещения дискретизируется с учётом габаритов робота, на её основе строится граф допустимых переходов между ячейками и соответствующая МБЗ. МБЗ для конкретной ПрО на момент проведения исследования формализовалась вручную экспертом. План действий представлял собой последовательность элементарных перемещений (двигаться влево, вправо, вверх и вниз) и полностью формировался за одно обращение к МЛВ. При изменении обстановки (появление препятствий, обновление целевой точки) план действий генерировался вновь. Данный пример показывает, что МТ ЛИИ позволяют перейти от фиксированной библиотеки сценариев к построению и перепланированию дей-

ствий в реальном времени в среде ПрО с линейной вычислительной сложностью, что является ключевым требованием к системам интеллектуального планирования для сервисных роботов, работающих в слабо предсказуемой среде.

Исследования в области МТ ЛИИ также затрагивают задачи координации групп роботов. В работе [118] предлагается подход к централизованному формированию плана поведения мультиагентной системы, где миварная модель ПрО позволяет в динамическом режиме вырабатывать последовательность действий – от распределения ролей и выбора исполнителя до подготовки операций и перемещения. Подобно подходу, предложенному в работе [127], МБЗ формировалась экспертом и служила основой для адаптивного, контекстно-зависимого планирования в реальном времени, обеспечивая автоматическое построение алгоритма работы РТК без необходимости вручную задавать порядок действий.

Результаты исследований [118, 127] были реализованы в виде демонстрационной системы, описание которой приведено в работе [49]. Стендовые испытания показали, что система, основанная на МТ ЛИИ, позволяет в автоматическом режиме решать задачи координации и маршрутизации нескольких исполнителей, учитывать появление динамических препятствий и изменение целей, а также оперативно пересчитывать траектории без ручного вмешательства разработчика. Тем самым продемонстрировано, что МТ ЛИИ может использоваться не только для построения планов в статических условиях, но и для реализации адаптивного управления РТК в режиме, близком к реальному времени, что подтверждает применимость данного подхода к задачам автоматического и интеллектуального планирования.

Важным примером применения МТ ЛИИ в системах управления в режиме реального времени является исследование, выполненное научным коллективом под руководством С.С. Шадрина [128]. МТ ЛИИ были использованы для создания подсистемы тактического планирования манёвров экспериментального автомобиля, основанной на МБЗ ПДД, модели рисков и накопленного опыта (на момент реализации ограничивался экспертными знаниями). Эта подсистема была интегрирована с модулем динамического планирования траектории движения. Включение блоков управления рисками и анализа опыта обеспечивало адаптацию к нетипичным ситуациям (например, неисправности светофора) и корректировку движения автомобиля. Логический вывод по МБЗ осуществлялся с учётом текущей дорожной обстановки, возможных манёвров и связанных с ними рисков. На оперативный

уровень управления подсистема передаёт набор допустимых манёвров. Таким образом, подсистема на основе МТ ЛИИ выполняла функцию интеллектуального фильтра и надстройки над траекторным планировщиком, разрешая или запрещая манёвры и обеспечивая планирование в условиях динамично меняющейся дорожной ситуации.

В контексте анализа дорожно-транспортных ситуаций заслуживает внимания диссертационное исследование Д.А. Чувикова [129], посвящённое реконструкции и экспертной оценке аварийных событий при дорожно-транспортных происшествиях (ДТП). В работе реализована интеграция МЭС с системой имитационного моделирования, что обеспечивает более обоснованные решения. С методологической точки зрения этот подход также демонстрирует потенциал МТ ЛИИ для задач планирования действий РТК: связка МЭС и систем имитационного моделирования позволяет верифицировать, уточнять и контролировать возможные варианты поведения управляемых объектов, включая транспортные средства.

Применение МТ ЛИИ выходит за рамки задач планирования и управления РТК. В диссертационном исследовании О.В. Кривошеева [130] МТ ЛИИ используются для преобразования разнородных данных информационных систем в структурированное представление, подходящее для подготовки планов распределения ресурсов при неполноте исходной информации. В работе [131] показано, что формализация задач объёмного, объёмно-календарного и сменно-суточного планирования в виде МБЗ обеспечивает автоматический логический вывод и получение актуальных планов в условиях неопределённости, включая изменения в составе ресурсов и технологических операций. Данные исследования подтверждают, что МТ ЛИИ могут применяться для широкого спектра задач интеллектуального планирования, включая сложные производственные среды, обеспечивая адаптивность и прозрачность принимаемых решений.

Отдельного внимания заслуживает работа Р.А. Санду [132], в которой были предложены модели описания информационных процессов для многомерной эволюционной прикладной автоматизированной информационной системы поддержки принятия решений, ориентированной на управление инновационными ресурсами химической и нефтехимической промышленности России. В рамках одной из моделей описания было предложено развитие продукционных конструкций вида «если . . . , то . . . », применяемых в МТ ЛИИ, за счёт их расширения механизмами явного задания альтернатив и логических развилок. Это позволило формализо-

вать логику выбора типа «если . . . , то . . . , иначе . . . » посредством введения взаимоисключающих правил и параметров-условий, включая признаки невыполнения основного условия и значения по умолчанию. При этом ветвление трактуется как выбор допустимого действия на основе текущего состояния предметной области, а ветвь «иначе» представляется в виде отдельной продукции, активируемой при ложности либо недостижимости основного условия. Такая организация продукционных правил в миварной сети обеспечивает прозрачность механизмов выбора для анализа и повышает масштабируемость модели при увеличении числа условий и альтернатив в динамически изменяющихся управленческих задачах.

Рассмотренные примеры применения МТ ЛИИ демонстрируют, что задачи управления и принятия решений для технических систем фактически реализуют планирование в пространстве состояний. При этом МБЗ может подготавливаться различными способами. В одном случае структура МБЗ сформирована таким образом, чтобы в процессе логического вывода по ней сначала определялось текущее состояние ПрО и на основе этого предлагался конкретный набор действий [128]. В другом – МБЗ описывает допустимые переходы между состояниями ПрО, тогда в результате логического вывода формируется цепочка действий, направленных на переход из начального состояния в целевое через промежуточные состояния [49, 118, 127].

Следует также отметить, что в рассмотренных примерах план действий формировался как полное решение заранее определённой задачи на основе статической, экспертно подготовленной МБЗ. Такая база знаний оставалась неизменной в процессе работы системы, что делало её чувствительной к изменениям среды ПрО или состава объектов управления. Так, в [118] добавление новых сервисных роботов требовало включения сведений о них в МБЗ со стороны эксперта: в [127] появление новых офисных рабочих мест требовало добавления новых знаний, в [49] изменение параметров навигационной сетки – дополнительная формализация возможных переходов между состояниями. Таким образом, МТ ЛИИ использовались преимущественно для получения плана действий в условиях фиксированного набора знаний, что ограничивало гибкость и возможность оперативного перепланирования при изменениях домена планирования.

Несмотря на эти ограничения, результаты показывают, что МТ ЛИИ обеспечивают достижение заданных показателей при решении задач планирования в производственных, навигационных и информационных системах. Однако для

применения в автономных технических системах, прежде всего в РТК, требуется провести дополнительные исследования. В таких системах планирование должно выполняться в условиях ограничений вычислительных ресурсов и времени реакции, обеспечивая устойчивую и предсказуемую работу алгоритмов перепланирования. Перспективным направлением развития МТ ЛИИ для автономных систем является переход от построения полного плана на основе статической МБЗ к динамическому синтезу и актуализации последовательностей действий в реальном времени. Это предполагает автоматическую генерацию МБЗ, а также использование высокопроизводительного логического вывода, способного работать в условиях изменяющейся среды. Указанные соображения определяют направленность и структуру проводимых в данной работе исследований.

1.4 Структура проводимых исследований

Структурная схема проведения исследований представлена на рисунке 1.1. Она отражает логику достижения цели диссертационной работы и включает последовательное решение взаимосвязанных аналитических, методологических, алгоритмических и практических задач.

В работе исследуется осуществимое планирование действий в пространстве состояний на основе МТ ЛИИ при принятии управленческих решений в технических системах. Применение МТ ЛИИ направлено на снижение вычислительной сложности задач интеллектуального планирования и уменьшение трудоёмкости подготовки баз знаний за счёт использования единой КМПрО и механизма высокопроизводительного логического вывода.

Для решения задач планирования разрабатывается модель МБЗ, обеспечивающая структурную декомпозицию знаний и их масштабирование при изменении среды ПрО и числа управляемых объектов. Также подготавливается модель проблемно-ориентированной системы управления и принятия решений на основе МТ ЛИИ, обеспечивающая согласованную работу модуля планирования, основанного на МЛВ, с другими подсистемами управления технических систем.

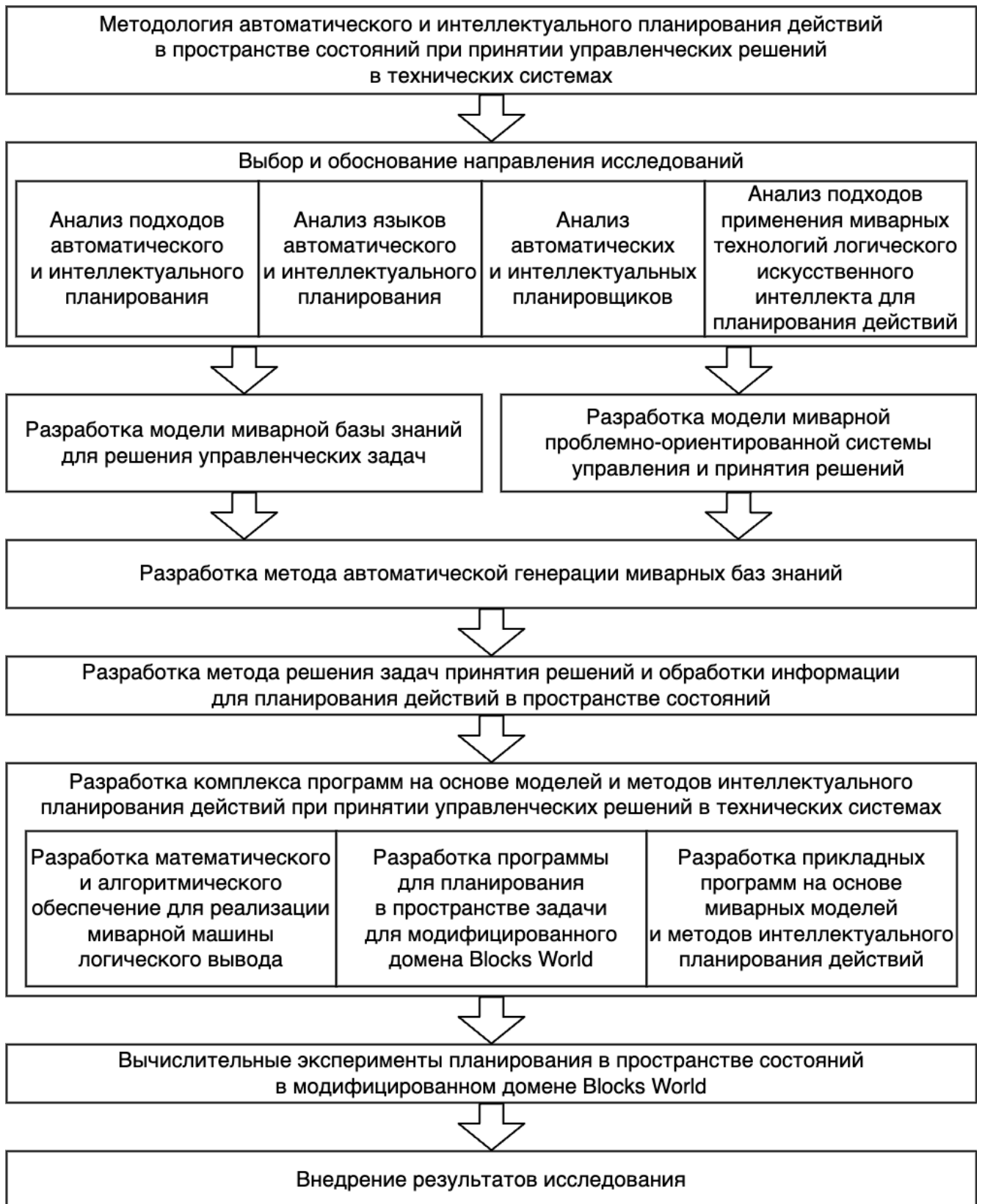


Рисунок 1.1 — Схема проведения исследований

С учётом предложенных моделей разрабатывается метод генерации МБЗ для решения управленческих задач в пространстве состояний. Он обеспечивает автоматическое масштабирование базы знаний под конкретную задачу и текущее состояние ПрО, что исключает участие эксперта и повышает адаптивность систе-

мы. Далее подготавливается метод решения задач принятия решений и обработки информации для планирования действий, реализующий построение частичного плана на основе сгенерированной МБЗ и циклическое взаимодействие с ней. Метод ориентирован на работу в автономных системах в режиме реального времени.

На основе разработанных моделей и методов создаётся комплекс программных средств для планирования действий при принятии управленческих решений в технических системах. Разрабатывается алгоритм параллельной обработки МБЗ и его программная реализация в виде МЛВ. Применение данной МЛВ в задачах планирования обеспечивает требуемые показатели производительности и автономности, поддерживает структурную декомпозицию МБЗ и взаимодействие с другими подсистемами, а также согласуется с механизмами автоматической генерации знаний.

Для проверки разработанных моделей и методов создаётся их программная реализация и проводятся испытания на задачах планирования действий робота в модифицированном домене Blocks World в составе миварной системы планирования действий роботов и роботизированных средств (МИПРА) [6]. Кроме того, полученные результаты используются при создании логической интеллектуальной системы контроля за соблюдением правил дорожного движения (ЛИСК ПДД) [4], логической интеллектуальной системы обеспечения ухода за растениями (ЛИС ОУР) [17] и программного комплекса на основе информационно-аналитической системы с поддержкой принятия решений об эффективности и безопасности применения термолабильных компонентов крови в медицинской практике (ПКИАС) [7].

1.5 Выводы по первой главе

1. Проведённое обзорно-аналитическое исследование показало, что автоматическое планирование является одним из ключевых направлений развития систем управления техническими системами. Классические STRIPS-подобные подходы, основанные на моделях преобразования состояний и формальных

описаниях действий, заложили теоретическую основу современных планировщиков, однако рост сложности задач и предметных областей обусловили развитие интеллектуального планирования, ориентированного на контекстный анализ, адаптацию и обработку динамически изменяющихся сред.

2. Анализ формализаций задач планирования и существующих инструментов показал, что современные универсальные планировщики, использующие формализмы PDDL и ASP, зависят от размера и сложности предметных областей. При увеличении количества объектов и ограничений возрастает вычислительная трудоёмкость, что ограничивает применение таких планировщиков в автономных технических системах, предъявляющих жёсткие требования к быстродействию и функционированию в режиме реального времени.
3. Исследование научных трудов по применению методов планирования в робототехнических комплексах показало, что из-за требований реального времени и ограниченных вычислительных ресурсов бортовых средств ключевым требованием к таким системам является способность быстро адаптироваться к изменениям состояния среды и эффективно обрабатывать знания о предметной области.
4. Анализ работ, посвящённых миварным технологиям логического искусственного интеллекта, показал их перспективность для решения задач планирования действий. Миварный подход предлагает единое унифицированное представление знаний и высокопроизводительный логический вывод с линейной вычислительной сложностью. Показано, что миварные технологии применяются для управления роботами, построения маршрутов, анализа дорожных ситуаций и координации групп мобильных исполнителей, что делает их подходящими для решения задач планирования в пространстве состояний.
5. Полученные результаты обосновывают выбор направления дальнейших исследований, связанного с разработкой миварных моделей и методов, направленных на снижение вычислительной сложности задач планирования и повышение адаптивности систем управления.

2 МОДЕЛИ МИВАРНОЙ БАЗЫ ЗНАНИЙ И МИВАРНОЙ ПРОБЛЕМНО-ОРИЕНТИРОВАННОЙ СИСТЕМЫ УПРАВЛЕНИЯ И ПРИНЯТИЯ РЕШЕНИЙ

2.1 Постановка задач разработки моделей

Представление знаний о ПрО в задачах интеллектуального планирования требует строгой формализации, обеспечивающей структуризацию, масштабирование и адаптацию к изменяющимся условиям среды. МТ ЛИИ предлагают такую формализацию за счёт базы знаний логических правил и параметров, организованных в виде миварных сетей. Однако для успешного применения миварных сетей в задачах планирования и управления необходимо подготовить их обобщённую структуру и способы использования, что формализуется в виде информационных моделей. Данная глава посвящена решению двух ключевых задач – созданию модели МБЗ [1] и модели миварной проблемно-ориентированной системы управления и принятия решений [2].

Первой задачей является разработка модели МБЗ [1], определяющей взаимосвязи между правилами и параметрами, а также их логическую группировку. Такая организация должна обеспечивать анализ текущего состояния ПрО, выявление признаков достижения целей задачи, а также синтез управляющих воздействий, направленных на преобразование среды ПрО. Также модель МБЗ должна корректно отражать структуру знаний о домене задачи, поддерживать масштабируемость при изменении начальных условий и быть применимой к различным ПрО. Её формализация создаёт основу для последующей разработки метода автоматической генерации миварных баз знаний и метода решения задач принятия решений, рассматриваемых в следующей главе.

Второй задачей является разработка модели миварной проблемно-ориентированной системы управления и принятия решений [2]. Решение задач на основе МБЗ осуществляется с помощью МЛВ. МЛВ обеспечивает логический вывод и взаимодействует с другими программными компонентами, например, в составе МЭС. В контексте планирования действий и управления модель системы должна учитывать динамику среды ПрО и изменения условий задачи. Её формализация позволяет интегрировать представление знаний о ПрО, их обработку и механизм принятия решений в единую структуру, повышая тем самым согласованность и предсказуемость функционирования интеллектуальной системы планирования действий в пространстве состояний.

Решение сформулированных выше задач начнём с рассмотрения концепции миварного подхода к накоплению и обработке информации.

2.2 Миварный подход к накоплению и обработке информации

В трудах [25, 119, 121] детально исследован миварный подход к разработке интеллектуальных информационных систем. Также рассматриваются особенности применения миварных технологий накопления и обработки информации (МТ ЛИИ) [125]. Подробнее остановимся на основных компонентах миварного подхода и специфике их использования при решении управленческих задач.

Миварное пространство – это фундаментальная концепция в миварном подходе, представляющая собой многомерное (трёхмерное) дискретное пространство для организации и представления знаний о ПрО [123]. Оно позволяет формализовать и структурировать информацию через три основных компонента: вещь, свойство и отношение. В миварном пространстве каждая **вещь** – это конкретный объект или сущность ПрО, которая обладает определёнными атрибутами и может взаимодействовать с другими компонентами ПрО. **Свойства** описывают характеристики и параметры вещей, позволяя их идентифицировать и отличать друг от друга. **Отношения** определяют связи и взаимодействия между различными компонентами ПрО. Особенность миварного пространства заключается в том, что все

его компоненты – вещи, свойства и отношения – являются неразрывно связанными между собой [123]. Более того, в определённых ситуациях вещь может рассматриваться как свойство или отношение, свойство – как вещь или отношение, а отношение – как вещь или свойство. Таким образом, миварный подход для работы с КМПрО предлагает целостную систему для структурированного представления и обработки знаний в виде миварного пространства. Представление **КМПрО на основе миварного подхода** может быть строго определено аналогично выражению (1.3) и представлено в следующем формализованном виде:

$$\text{КМПрО}_{\text{мивар}} = \langle V, S, O \rangle, \quad (2.1)$$

где V – множество вещей (объектов, сущностей), S – множество свойств (атрибутов, характеристик), O – множество отношений (связей, взаимодействий).

Соответственно, состояние ПрО в момент времени t в рамках данной модели также допускает формальное определение, аналогичное выражению (1.4), и представляется как:

$$S_{\text{ПрО}}(t) = \langle V(t), S(t), O(t) \rangle. \quad (2.2)$$

На основе миварного пространства формируется **миварная сеть** – представление знаний в виде двудольных ориентированных графов [123]. Миварная сеть является одним из вариантов формализации баз знаний, использующих миварный подход. Она состоит из двух видов элементов:

- **Параметры** P , $P \subseteq V$ – измеримые свойства миварного пространства, каждому из которых можно соотнести определённое значение.
- **Правила** R , $R \subseteq O$ – способы получения одних значений параметров при известных других.

Каждому параметру и правилу для их идентификации задают уникальное имя. Значения параметров определяются либо во время подготовки базы знаний, либо в процессе решения задач в рамках ПрО. Каждое правило может быть представлено в виде продукции «если ..., то ...»: *если* известны значения всех параметров из одного набора (**входное множество миварного правила**), *то* можно получить значения параметров из другого набора (**выходное множество миварного правила**). С каждым правилом связано **действие**, которое отвечает за вычисление значений параметров выходного множества и которое может исполь-

зовать значения параметров из входного множества. Это действие может быть реализовано с помощью различных вычислительных методов, например, предикатной логики или машинного обучения. Иными словами, реализация действия правила зависит от функциональных возможностей программного обеспечения для работы с миварной сетью, а именно, редактора базы знаний и машины логического вывода [133]. В качестве примеров такого рода программного обеспечения можно привести КЭСМИ Wi!Mi «Разуматор» версии 2.1 [121] и платформу ЛИСА [23, 24].

Обязательным условием для выполнения действия правила (**условие активации правила**) является известность всех значений параметров из его входного множества. При этом выходное множество может оказаться пустым. Также возможно, что в результате выполнения вычислительного действия будут получены не все значения параметров из его выходного множества.

Миварная технология обработки информации накладывает принципиальное ограничение [125]: правила не могут переопределять значения параметров, которые уже известны в миварной сети. Если в результате активации правила выявляется, что некоторые параметры в ПрО имеют установленные значения, то они сохраняются без изменений. Это обусловлено особенностями механизма логико-вычислительной обработки [123]. Эта особенность также определяет то, как системы, основанные на миварном подходе, решают управленческие задачи в пространстве состояний.

Правила вида *«если известны значения всех параметров из одного набора, то можно получить значения параметров из другого набора»* представляют собой альтернативную реализацию правил выбора *«если . . . , то . . . , иначе . . . »*, рассмотрение которых представлено в [132, 134, 135]. В первом случае формализация условий получения значения того или иного выходного параметра осуществляется через описание алгоритма работы вычислительного средства, реализующего действие правила. Тогда как во втором случае условие задаётся в виде логического предиката над текущим состоянием ПрО и используется для выбора одной из альтернативных ветвей поведения: при истинности условия вычисляются одни выходные параметры, при ложности – другие (либо активируется иной набор правил).

Параметр миварной сети можно представить в виде кортежа:

$$\begin{aligned} P_i &= \langle \text{IDP}_i, \text{VAL}_i \rangle, \\ \forall \text{VAL}_i \exists \text{S}_{\text{PrO}} (\text{VAL}_i = \emptyset), \end{aligned} \quad (2.3)$$

где P_i – параметр, IDP_i – его уникальное имя (идентификатор), VAL_i – значение параметра. Значение параметра может быть задано при описании модели ПрО или определено позже, например, в процессе решения задач в этой ПрО.

Для правила миварной сети также можно составить кортеж:

$$\begin{aligned} R_i &= \langle \text{IDR}_i, \text{IN}_i, \text{OUT}_i, \text{ACT}_i \rangle, \\ \text{IN}_i &= \{P_j\}, \forall P_j \in P, \\ \text{OUT}_i &= \{P_k\}, \forall P_k \in P, \text{IN}_i \cap \text{OUT}_i = \emptyset. \end{aligned} \quad (2.4)$$

В выражении (2.4) для правила R_i определены:

- IDR_i – уникальное имя правила (идентификатор).
- IN_i – входное множество правила. В множество IN_i должен входить как минимум один параметр.
- OUT_i – выходное множество правила. Множество OUT_i может быть пустым. Условие $\text{IN}_i \cap \text{OUT}_i = \emptyset$ является важным условием для представления миварной сети в виде двудольных ориентированных графов и связано с невозможностью переопределить известные значения параметров [123].
- ACT_i – действие правила, задающее способ вычисления значений параметров из множества OUT_i при выполнении условия его активации: для всех параметров $P_j \in \text{IN}_i$ должно существовать непустое значение $\text{VAL}_j \in P_j$ (то есть $\text{VAL}_j \neq \emptyset$).

Анализируя формализованное описание правила миварной сети, можно заметить, что условие активации правила и соответствующее действие полностью удовлетворяют широко известной форме продукции:

$$P(x, y) \rightarrow A \& B, \quad (2.5)$$

где $P(x, y)$ – логическое выражение, A – заключение, B – действие. Иными словами, правило миварной сети представимо в виде пары **антецедент** (перечень параметров, значения которых должны быть известны) и **консеквент** (действие,

связанное с правилом). Особенность миварного подхода заключается в том, что он не ограничивает консеквент только предикатами и логическими выражениями. Это делает возможным использование различных подходов к представлению знаний о способе получения значений параметров и взаимодействие с другими методами искусственного интеллекта.

Подробнее остановимся на рассмотрении миварной сети в виде графа. В общем виде формализованную модель миварной сети можно представить в виде:

$$\text{MIG} = \langle V, E \rangle, \quad (2.6)$$

где: MIG – миварная сеть; $V = P \cup R$ – вершины графа, состоящего из параметров P и правил R ; E – ориентированные ребра графа, которые отображают отношения параметров P к правилам R . Если ребро $E_i \in E$ направлено из вершины $P_j \in P$ в вершину $R_k \in R$, то параметр P_j находится во входном множестве параметра R_k ($P_j \in \text{IN}_k$). Если наоборот, то параметр P_j находится в выходном множестве параметра R_k ($P_j \in \text{OUT}_k$). В миварных сетях не существует рёбер, которые бы связывали параметры или правила напрямую друг с другом. Поэтому переход от одних значений параметров к другим значениям параметров осуществляется только через правила.

Пример отображения миварного правила представлен на рисунке 2.1. Для удобства правила изображают в виде прямоугольников с именем, параметры – в виде круга или овала с именем. Входное множество IN_1 правила R_1 состоит из параметров P_1 и P_2 , а выходное множество OUT_1 – из параметра P_3 .

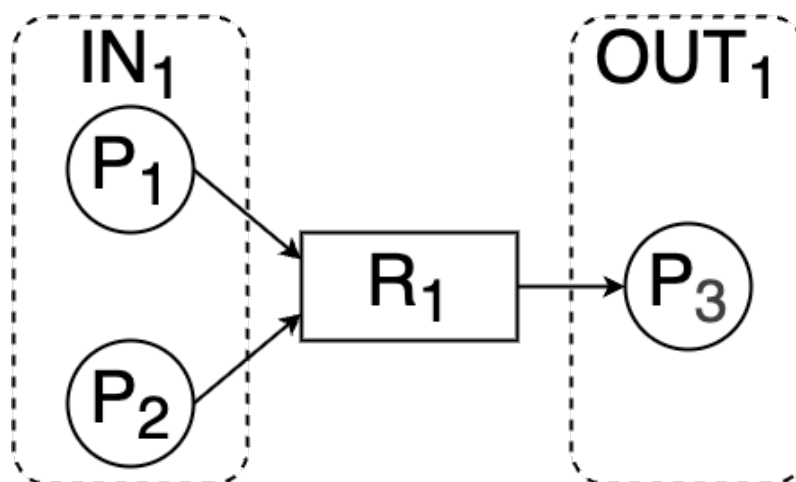


Рисунок 2.1 — Пример правила миварной сети

Пример миварной сети, которая состоит из трёх правил с общими парамет-

рами, представлен на рисунке 2.2.

Задачи, которые решаются с использованием миварных сетей, определяют то, какие значения параметров известны и какие значения параметров требуется получить (искомые). В процессе решения задачи строится **ЛОГИЧЕСКИЙ ВЫВОД** – список параметров, перечень их значений и последовательность правил, которые использовались для вычисления искомых значений.

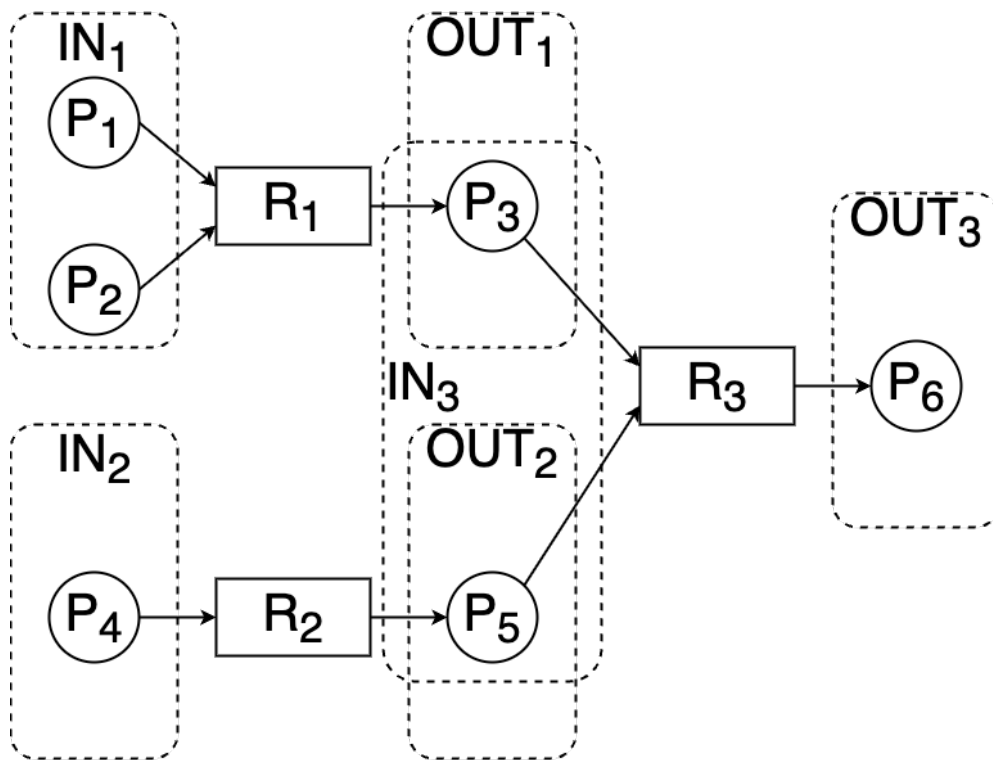


Рисунок 2.2 — Пример миварной сети

Процесс работы с миварной сетью основан на миварной технологии обработки информации и зависит от конкретного алгоритма, например, описанного в патенте [133]. Важно отметить, что миварная технология обработки предлагает метод построения логического вывода без построения полного графа миварной сети [133]. Иными словами, логический вывод строится исходя из текущей задачи ПрО, текущего состояния базы знаний и результатов выполнения действий правил.

Пример логического вывода по миварной сети представлен на рисунке 2.3. Этот вывод был получен по миварной сети, показанной на рисунке 2.2.

В рамках задачи, логический вывод для которой изображён на рисунке 2.3, были определены значения параметров P_1 , P_2 и P_5 , а значение параметра P_6 было искомым. В результате выполнения действий, связанных с правилами R_1 и R_3 , сначала было получено значение параметра P_3 , затем искомое значение параметра

P_6 . Таким образом, логический вывод представляет собой подграф миварной сети, в котором осуществлён переход от известных значений параметров P_1 , P_2 и P_5 к искомому значению параметра P_6 через правила R_1 и R_3 и параметр P_3 .

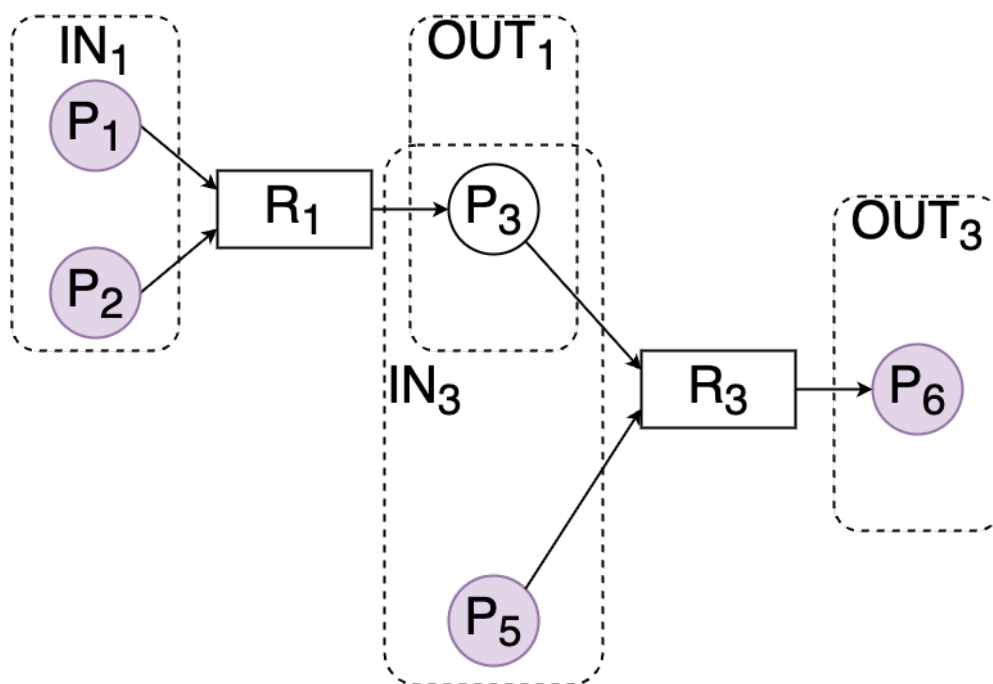


Рисунок 2.3 — Пример логического вывода по миварной сети

Обработка баз знаний МЛВ из КЭСМИ Wi!Mi «Разуматор» версии 2.1 основывается на схеме «запрос – ответ» [133]. Запрос содержит перечень известных значений параметров миварной сети и перечень параметров, которые требуется вычислить. На основе загруженной в МЛВ базы знаний подготавливается логический вывод, и в качестве ответа на запрос выдаются вычисленные значения искомых параметров. После обработки запроса МЛВ сохраняет в своём рабочем пространстве значения параметров, которые имелись в исходной базе знаний. При этом значения параметров, установленные на основе информации из запроса и полученные в процессе подготовки логического вывода, очищаются.

Для представления знаний о ПрО в виде миварной сети существуют различные методологии [16, 136, 137]. Кроме того, для решения задач в конкретных ПрО применяются специализированные методы. Одним из таких методов является разбиение задачи на подзадачи.

Метод разбиения задачи на подзадачи представляет собой способ решения сложных задач путём их разделения на более простые подзадачи. В труде [84] представлено формализованное описание данного метода. Оно основывается на понятии состояния ПрО, которое было рассмотрено в предыдущей главе (вы-

ражение (1.4)). Метод разбиения для решения задачи в пространстве состояний (выражение (1.6)) заключается в том, чтобы представить исходную задачу T в виде подзадач, которые можно решить с помощью одного из операторов из множества G .

В работе [84] отмечается, что между выделенными подзадачами могут устанавливаться два типа логических связей: отношение согласованности (И) и отношение альтернативности (ИЛИ). Отношение И предполагает, что все связанные подзадачи должны быть выполнены для решения родительской задачи, тогда как отношение ИЛИ означает, что достаточно выполнить хотя бы одну из альтернативных подзадач. Для наглядного представления этих связей используется И/ИЛИ-граф. Данный граф представляет собой дерево, в котором вершины соответствуют задачам, а рёбра отражают связи между родительскими и дочерними подзадачами. Связи типа И визуализируются с помощью дуг, объединяющих соответствующие рёбра. Корневая вершина графа представляет исходную задачу планирования. В [84] также упоминается, что И/ИЛИ-граф также является отображением состояний ПрО: корневая вершина графа соответствует исходному состоянию S_n , а концевые вершины – промежуточные состояния.

Ранее отмечалось, что МБЗ для решения задачи планирования в пространстве состояний может формализоваться двумя разными способами. С точки зрения метода разбиения задачи на подзадачи в обоих случаях МБЗ включает описание всех подзадач, однако стратегия их использования определяется выбранным подходом к решению:

- Ищется и решается одна подзадача, соответствующая текущему состоянию ПрО. При этом каждая подзадача задаётся подграфом миварной сети, а в процессе построения логического вывода активизируется лишь тот фрагмент, который релевантен текущей ситуации. Такая стратегия используется в системе определения допустимых манёвров транспортного средства в соответствии с ПДД [128].
- Решаются все подзадачи сразу, поскольку их совокупное выполнение необходимо для достижения цели исходной задачи. Подграфы, соответствующие отдельным подзадачам, обрабатываются совместно, а логический вывод нацелен на согласованное выполнение всей совокупности действий. При этом подзадачи, как правило, не являются независимыми: результат решения одной из них может определять достижимость или порядок решения других.

Подобный подход реализован, например, в системе управления офисным сервисным роботом [127].

Данные стратегии определяют характер работы МЛВ с МБЗ. При одном подходе для полного решения задачи требуется несколько раз получить логический вывод по МБЗ. МЛВ в цикле обрабатывает МБЗ для анализа текущего состояния ПрО и решения релевантной подзадачи. Сформированное управленческое решение переводит среду в новое промежуточное состояние. МЛВ повторяет обработку МБЗ до тех пор, пока исходная задача не будет решена. Работа МЛВ в цикле характерна при управлении транспортным средством [128]. При другом подходе МЛВ строит логический вывод один раз и формирует полный план действий, опираясь на единое, заранее сформированное описание подзадач в МБЗ. Такой режим характерен для рассмотренных ранее систем координации и управления роботами [49, 118, 127].

Таким образом, при поэтапном выборе и решении одной подзадачи МЛВ работает с локально отдельным взятым фрагментом МБЗ и ориентирована на быстрый пересчёт решения в ответ на изменения состояния ПрО, тогда как при одновременном решении всех подзадач используется вся МБЗ целиком, обеспечивая глобальную согласованность плана и учёт взаимозависимостей между подзадачами.

С учётом рассмотренных особенностей миварного подхода к накоплению и обработке информации, а также специфики применения метода разбиения задачи на подзадачи для миварных систем, перейдём к определению модели МБЗ для решения управленческих задач в пространстве состояний.

2.3 Модель миварной базы знаний для решения управленческих задач в пространстве состояний

Для решения задач с использованием миварного подхода требуется участие когнитолога (инженера по знаниям), который отвечает за формализацию экспертных знаний о ПрО и выбор способа решения задач. На основе этого для создания

МБЗ предварительно подготавливается обобщённая структура базы знаний [16]. Её можно визуализировать в виде графа миварной сети, в котором отдельные блоки знаний логически группируются. Группировка может осуществляться по разным признакам: связь с определенным этапом анализа задачи и синтеза его решения, отношение к конкретному объекту ПрО, роль в процессе формирования управляющих воздействий и т.д. Такая визуализация помогает когнитологу заранее спланировать шаги по формализации знаний, а также обсуждать и разрабатывать алгоритм решения задач совместно с экспертами в ПрО, не знакомыми с инструментарием разработки МБЗ.

Практика разработки миварных систем управления [49, 118, 127, 128] показывает, что при работе с типовыми объектами, имеющими однотипное описание и поддерживающими одинаковые действия, в базе знаний можно выделить несколько устойчивых групп параметров и правил: отвечающих за анализ текущего состояния ПрО и формирование управляющих воздействий, а также за оценку достижимости целей задачи. Подобная организация МБЗ не зависит ни от количества управляемых объектов, ни от конкретного начального состояния ПрО и согласуется с методом разбиения задачи на подзадачи. Между выделенными группами параметров и правил возникают характерные взаимосвязи, обусловленные формализацией знаний в виде миварных сетей, методом разбиения задачи на подзадачи и используемым алгоритмом решения задач. Для дальнейшего анализа этих взаимосвязей предлагается использовать модель МБЗ для решения управленческих задач в пространстве состояний, представленную в виде ориентированного графа на рисунке 2.4.

Модель МБЗ опирается на метод решения задач посредством разбиения их на подзадачи. Соответственно, структура ориентированного графа организована таким образом, чтобы явно представлять знания о подзадачах и их взаимосвязях. Узлы графа и соответствующие им фрагменты МБЗ отражают типовые шаги анализа текущего состояния, выбора вариантов управляющих воздействий и оценки достижимости целевых состояний.

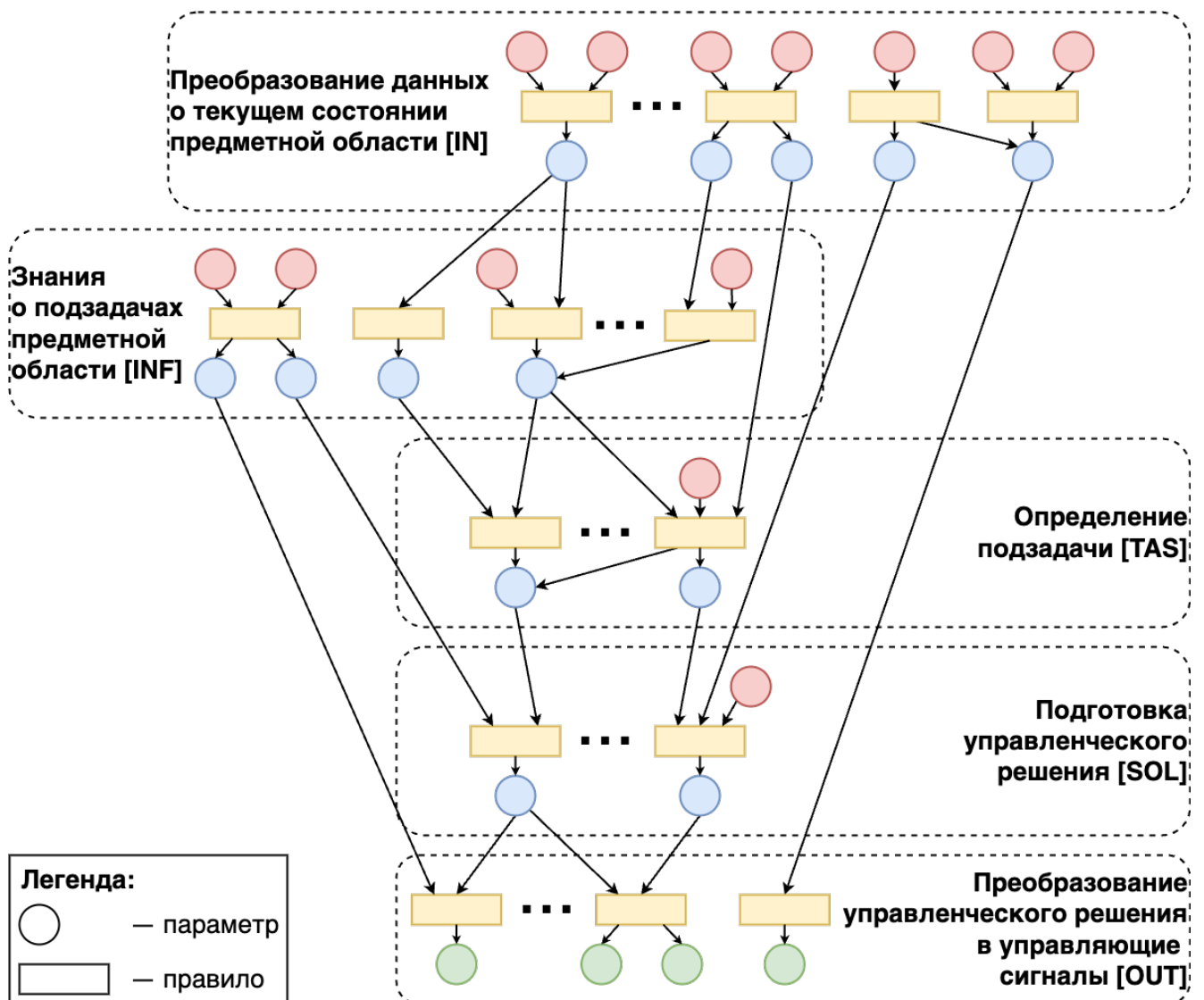


Рисунок 2.4 — Модель миварной базы знаний для решения управленческих задач в пространстве состояний

На рисунке 2.4 представлены следующие множества правил и параметров:

- IN – преобразование данных о текущем состоянии ПрО. Данные, поступающие на вход в МЛВ, могут требовать дополнительной обработки и преобразования. Поэтому в базах знаний выделяют отдельное множество правил и параметров для подготовки данных к дальнейшему анализу. Например, это может включать преобразование числовых значений или приведение данных к единому формату.
- INF – знания о подзадачах ПрО. В этом множестве содержатся правила и параметры, отвечающие за организацию процесса решения задачи методом разбиения на подзадачи. Помимо этого, сюда включают знания, которые связаны с фактами о ПрО, что позволяет учитывать особенности конкретной задачи и среды при подготовке решения. Результаты работы правил также

могут использоваться для подготовки управленческого решения и управляющих сигналов.

- *TAS* – определение подзадачи. В нём содержатся правила и параметры, позволяющие анализировать текущее состояние ПрО и устанавливать, какие подзадачи уже решены, а какие решаются в настоящий момент.
- *SOL* – подготовка управленческого решения. В этом множестве содержатся правила и параметры, необходимые для анализа текущей подзадачи и подготовки управленческого решения.
- *OUT* – преобразование управленческого решения в управляющие сигналы. После того как решение подготовлено, оно должно быть преобразовано в формат, который может быть использован для передачи в другие информационные системы. Например, в системы управления действиями робота.

Необходимо отметить, что выявленные множества могут быть представлены как отдельные подграфы миварной сети. При этом подграфы внутри одного множества могут быть не связаны друг с другом. Каждый из подграфов внутри множества может содержать взаимосвязанные правила, у которых есть общие входные и выходные параметры.

Каждое из рассмотренных множеств можно масштабировать под решение конкретной задачи ПрО. Логический вывод, получаемый в процессе решения задачи, представляется в виде графа, включающего правила и параметры из описанных выше множеств. Если происходит выдача сигнала управления, то логический вывод обязательно будет содержать правила и параметры из множеств *IN* и *OUT*.

Представленная модель МБЗ для решения управленческих задач в пространстве состояний обладает следующими преимуществами:

1. **Гибкость использования.** Предложенную модель можно применять при разработке миварных систем, которые используют метод разбиения задачи на подзадачи. В частности, её можно использовать при разработке систем, где процессы мониторинга и управления реализуются на основе **цикло-календарного планирования** [70] в соответствии с заранее разработанными технологическими программами. Решение задач цикло-календарного планирования характерно для систем управления производственными процессами [138, 139], роботизированными комплексами [140] и сельскохозяйственными объектами [44]. В таких системах обработка поступающих данных осуществляется с помощью правил и параметров множества *IN*,

а управляющие сигналы формируются на основе множества *OUT*. Множество *INF* содержит знания о требуемых показателях среды ПрО в заданные периоды времени и соответствующие циклы технологической программы. Подзадачи отвечают за управление подсистемами, изменяющими состояние среды ПрО. Для решения этих подзадач используются правила и параметры из множеств *TAS* и *SOL*.

2. **Структурная декомпозиция.** База знаний, построенная на основе предложенной модели, может быть представлена в виде отдельных компонентов – подграфов миварных сетей правил и параметров. Это позволяет организовать и представить в структурированном виде информацию о сложных ПрО, разделяя её на более простые и удобные для формализации элементы.
3. **Масштабируемость.** Базы знаний, созданные на основе предложенной модели, могут адаптироваться к изменению количества управляемых объектов. Их структура позволяет повторно использовать ранее формализованные знания [125]. Поэтому источником новых знаний об объектах управления может выступать не только когнитолог, но и генератор-программа, работающая на основе результатов системного анализа ПрО. Пример возможности генерации МБЗ представлен в работе [141].
4. **Параллельная обработка.** В работах [50, 71] обосновывается использование параллельного алгоритма выполнения действий правил миварной сети для решения задач планирования действий в технических системах. Особенность предлагаемой модели заключается в том, что базы знаний строятся с учётом возможности параллельной обработки. Знания о подзадачах и условиях их выполнения выделяются в отдельные подграфы, которые могут обрабатываться независимо друг от друга с помощью МЛВ. Это позволяет повысить производительность подготовки решения и работать с большим объёмом знаний.

Предложенная модель задаёт обобщённую структуру МБЗ и рассматривает взаимосвязи между группами правил и параметров. Однако она не раскрывает вопросов использования МБЗ в составе информационной системы и роли МЛВ в этом процессе. Для этого необходимо учитывать архитектуру всей системы, распределение функций между её компонентами, а также взаимодействие с внешними информационными системами. В этой связи дальнейшее изложение будет посвящено разработке модели миварной проблемно-ориентированной системы

управления и принятия решений в пространстве состояний.

2.4 Метаграфовый подход к моделированию сложных интеллектуальных информационных систем

Изменение среды ПрО на основе логического вывода осуществляется в результате взаимодействия МЛВ с другими элементами технической системы. Как показано в модели МБЗ, такие элементы могут подготавливать данные для МЛВ, а также преобразовывать результаты её работы в конкретные управляющие сигналы на органы исполнения технической системы. Кроме того, они могут отвечать за дополнительную обработку поступающей информации о среде ПрО. Для такой обработки могут применяться различные методы искусственного интеллекта.

Рассмотрим примеры совместного использования МТ ЛИИ и других методов искусственного интеллекта для накопления и обработки информации о ПрО. В исследовании [142] рассматривается подготовка МБЗ из обучающих текстов, инструкций или множества правил, а также анализируются возможности больших языковых моделей (англ. *Large Language Model*, LLM). В работе [143] предложен подход комплексного взаимодействия моделей классификации с МЭС для повышения точности определения пород деревьев на основе данных лидара (англ. *Light Detection and Ranging*, LiDAR). В труде [144] представлена модель, объединяющая нейронные сети и МЭС, для семантической сегментации растительности в плотном облаке точек, полученном при сканировании лесного полигона.

Применение МТ ЛИИ с другими методами искусственного интеллекта можно рассматривать в рамках концепции гибридной интеллектуальной информационной системы (ГИИС) [145, 146]. В такой системе главные компоненты ГИИС – это модули сознания (связанные с логическим искусственным интеллектом) и подсознания (ассоциируемые с методами мягких вычислений). В работе [147] рассматривается вариант ГИИС, в котором модуль сознания реализуется через миварный подход. Для организации взаимодействия между различными интеллектуальными методами в рамках ГИИС применяют подход, основанный на мно-

гоагентных системах [148]. Для описания структуры ГИИС успешно применяется **метаграфовый подход** [149, 150].

Метаграфовый подход находит широкое применение для моделирования различных областей знаний [151]. В статье [152] предложен метод преобразования нотаций описания бизнес-процессов в единую форму – метаграф. Такая форма применима для анализа и оптимизации бизнес-процессов. В работе [153] рассматривается применение метаграфовых баз данных для управления проектами, подчёркивая их эффективность в организации и анализе сложных взаимосвязей между элементами проекта. В исследовании [154] описан метод определения смыслового содержания изображений и способы представления полученного семантического анализа с помощью метаграфового подхода.

Метаграфовый подход предлагает инструментарий для представления ПрО в виде сложного графа [155–157]. Например, в статье [158] предлагается использовать метаграфовый подход для семантической обработки сложных событий с реализацией агента на основе функционального реактивного программирования. В труде [159] предложен подход, объединяющий глубокое обучение и самоорганизующиеся нейронные сети для построения адаптивных моделей описания потока данных в реальном времени, а также проведён эксперимент по кластеризации данных с использованием метаграфового подхода для программной реализации. В исследованиях [160, 161] рассматривается интеграция метаграфовой модели: в первом случае – с алгоритмами нечёткого логического вывода, во втором – с алгоритмами продукционного логического вывода.

ГИИС может использоваться в качестве архитектурного шаблона, упрощающего процесс разработки интеллектуальных систем. Миварный подход к созданию систем может быть применён для создания ГИИС и гармонично сочетается с другими подходами создания интеллектуальных систем [145]. Метаграфовый подход, в свою очередь, может применяться для описания разнородных сложных структур, входящих в состав ГИИС. Это позволяет использовать его как основу для моделирования взаимодействия миварных проблемно-ориентированных систем с другими информационными системами в рамках ГИИС.

Для построения модели миварной проблемно-ориентированной системы необходимо опираться на формализованное представление КМПРО. Миварный и метаграфовый подходы предлагают свои собственные пространства для работы с КМПРО. Эти пространства позволяют преобразовывать данные между собой,

что было показано в работах [162, 163]. Такая совместимость открывает путь к интеграции их выразительных возможностей при проектировании интеллектуальных систем. Однако для эффективного использования этих подходов в рамках единой архитектуры требуется их строгая формализация. Для миварного подхода такая формализация была рассмотрена выше (выражение (2.1)). Далее рассмотрим метаграфовый подход к представлению КМПрО.

Формализованную **модель метаграфа** MG в соответствии с [148, 151, 164] представляют в виде кортежа:

$$MG = \langle V, MV, E, ME \rangle, \quad (2.7)$$

где V – множество вершин, каждое из которых характеризуется множеством атрибутов, MV – множество метавершин, E – множество рёбер, ME – множество метарёбер.

Каждая **вершина** v_i характеризуется множеством атрибутов $\{atr_k\}$ (выражение (2.8)). Каждое **ребро** e_i характеризуется исходной вершиной v_S , конечной вершиной v_E , признаком направленности $eo = true|false$ и множеством атрибутов $\{atr_k\}$ (выражение (2.9)).

$$v_i = \{atr_k\}, v_i \in V, \quad (2.8)$$

$$e_i = \langle v_S, v_E, eo, \{atr_k\} \rangle, e_i \in E, eo = true|false. \quad (2.9)$$

Как указано в [148], фрагмент метаграфа MG_i в общем виде можно представить как множество элементов, включающее произвольные вершины (метавершины) и рёбра (метарёбра):

$$MG_i = \{ev_j\}, ev_j \in (V \cup E \cup MV \cup ME). \quad (2.10)$$

Метавершина $mv_i \in MV$ обладает свойствами вершины $\{atr_k\}$ и включает фрагмент метаграфа $MG_j = \{ev_j\}$, у которого рёбра и метарёбра могут быть только ненаправленными:

$$mv_i = \langle \{atr_k\}, \{ev_j\} \rangle, \quad (2.11)$$

$$ev_j \in (V \cup E^{eo=false} \cup MV \cup ME^{eo=false})$$

Метаребро $me_i \in ME$ обладает свойствами ребра и включает фрагмент метаграфа $MG_j = \{ev_j\}$, у которого рёбра и метаребра могут быть только направленными:

$$\begin{aligned} me_i &= \langle v_S, v_E, eo, \{atr_k\}, \{ev_j\} \rangle, \\ ev_j &\in (V \cup E^{eo=true} \cup MV \cup ME^{eo=true}) \end{aligned} \quad (2.12)$$

ГИИС в рамках метаграфового подхода рассматривают как многоагентную структуру. Совокупность агентов такой системы обозначают следующим образом [161]:

$$AG = \{ag_i\}, \quad (2.13)$$

где ag_i соответствует отдельному агенту.

Метаграфовый подход предлагает к рассмотрению агент-функцию и метаграфовый агент. Кроме того, в работе [148] дополнительно определяются контейнерные и динамические агенты, использующие базовые конструкции метаграфового агента и отображающие статические и динамические свойства ГИИС.

Агент-функция ag^F определяется как кортеж, включающий входной метаграф MG_{IN} , выходной метаграф MG_{OUT} и абстрактное синтаксическое дерево AST , которое может быть выражено в виде фрагмента метаграфа:

$$ag^F = \langle MG_{IN}, MG_{OUT}, AST \rangle. \quad (2.14)$$

Метаграфовый агент ag^M представим кортежем из метаграфа данных и знаний MG_D , на основе которого выполняются правила из множества $\{r_i\}$ и стартовое условие выполнения агента AG^{ST} , который может быть представлен фрагментом метаграфа:

$$ag^M = \langle MG_D, \{r_i\}, AG^{ST} \rangle. \quad (2.15)$$

Правило метаграфового агента r_i имеет структуру

$$MG_j \rightarrow OP^{MG}, \quad (2.16)$$

где: MG_j – фрагмент метаграфа (антецедент), на основе которого выпол-

няется правило; OP^{MG} – множество операций, выполняемых над метаграфом (консеквент).

Пример метаграфового агента изображен на рисунке 2.5.

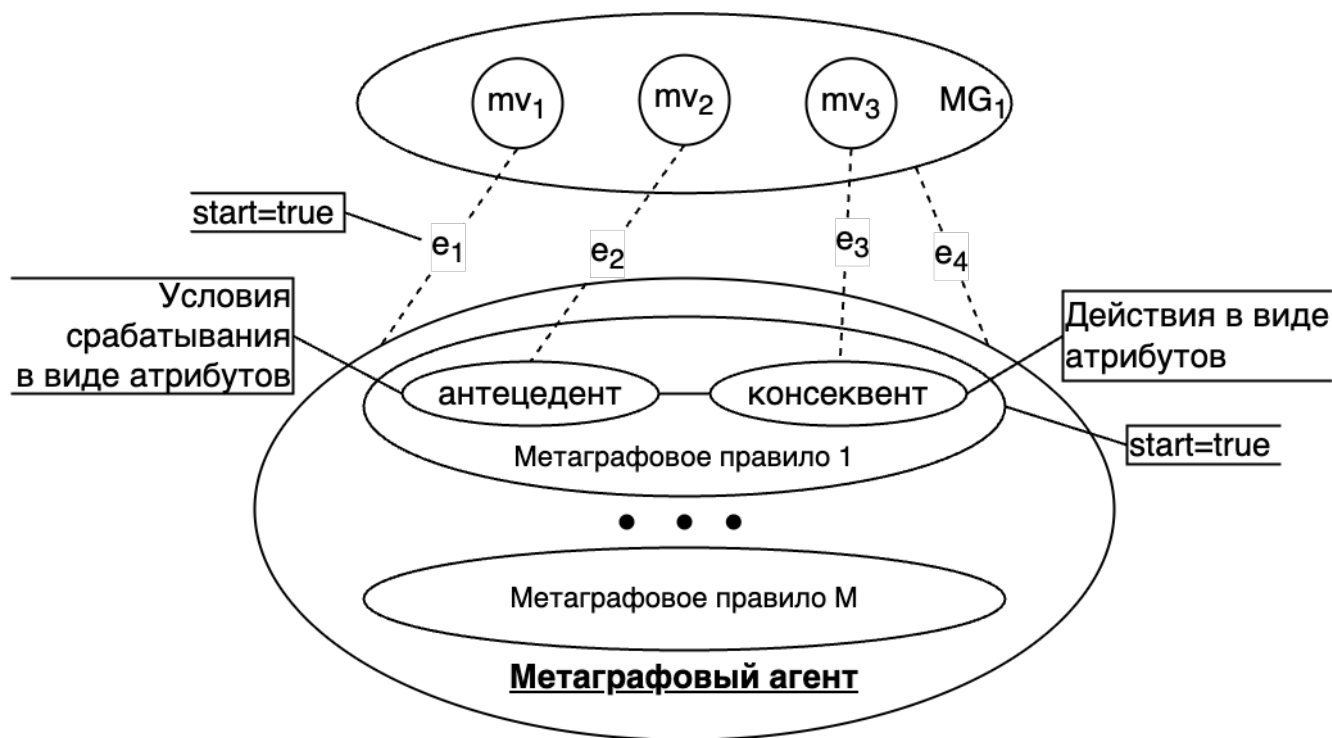


Рисунок 2.5 — Модель метаграфового агента в виде фрагмента метаграфа

Для наглядности на рисунке 2.5 детально рассмотрено лишь одно метаграфовое правило (метаграфовое правило 1). Всего количество правил в агенте: $M \in \mathbb{N}$. Метаграфовый агент отображается в виде вершины метаграфа, внутри которой размещается множество метаграфовых правил ($\{r_i\}$), представленных в форме метавершин. Каждая такая метавершина содержит элементы, соответствующие антецеденту и консеквенту правила. Условие активации правила и выполняемое действие задаются через атрибуты антецедента и консеквента. Агент функционирует на основе метаграфа данных и знаний MG_1 , с которым он связан посредством ребра e_4 . В метаграфе MG_1 выделены фрагменты mv_1 , mv_2 и mv_3 . Стартовое условие работы агента (AG^{ST}) задаётся фрагментом mv_1 , который через ребро e_1 с атрибутом $start=true$ связывается с метаграфовым агентом. Первое к выполнению правило также обозначается атрибутом $start=true$. Фрагмент mv_2 соединяется ребром e_2 с антецедентом метаграфового правила 1, а фрагмент mv_3 – ребром e_3 с его консеквентом.

Метаграфовые агенты могут работать с двумя типами правил [161]: разомкнутыми и замкнутыми. Разомкнутые правила позволяют агенту генерировать один метаграф на основе другого, замкнутые правила – модифицировать метаграф.

Эта особенность метаграфового агента влияет на то, как миварная проблемно-ориентированная система управления и принятия решений в пространстве состояний может быть представлена в виде сложного графа.

2.5 Модель миварной проблемно-ориентированной системы управления и принятия решений в пространстве состояний

Для построения модели миварной проблемно-ориентированной системы управления и принятия решений в пространстве состояний предлагается применить метаграфовый подход. Данный подход обеспечивает представление компонентов сложных систем в виде взаимодействующих агентов, формализуемых с помощью метаграфов. Агенты могут реализовывать различные методы искусственного интеллекта. Для описания миварной системы предлагается ввести понятие миварного агента, определяемого на основе метаграфового агента. Это позволяет интерпретировать взаимодействие миварной системы с другими компонентами как взаимодействие миварного агента с метаграфовым агентом в рамках ГИИС.

Перед тем как дать формальное определение миварного агента, необходимо определить способ отображения миварной сети в метаграф. Один из возможных вариантов такого отображения представлен в работах [162, 163]. Вариант отображения далее строится на основе ранее введённых формализаций миварной КМПРО (см. выражение (2.1)) и модели метаграфа (см. выражение (2.7)):

- Параметр миварной сети (выражение (2.3)) может быть представлен вершиной метаграфа (выражение (2.8)). В этом случае идентификатор параметра и его значение фиксируются в атрибутах соответствующей вершины.
- Правило миварной сети (выражение (2.4)) может быть формализовано в виде метаграфа вместе с множеством рёбер (выражение (2.9)). Атрибуты вершины правила определяют его идентификатор и действие, а рёбра задают связь правила с параметрами и указывают отношение к правилу (входной или выходной параметр).

Формализм миварных сетей учитывает требования, предъявляемые к накоп-

лению и обработке эволюционных баз знаний [165]. Это обеспечивает согласованность формализма миварной сети с МТ ЛИИ [123]. Использование метаграфов в качестве модели представления миварных сетей позволяет отразить архитектуру системы, основанной на миварном подходе, в виде сложного графа, что необходимо для дальнейшего формального описания модели.

Отображение миварной сети в метаграф создаёт основу для формализованного определения миварного агента. Миварный агент ag^{MI} может быть представлен аналогичным кортежем метаграфового агента (выражение (2.15)):

$$ag^{MI} = \langle MG_D, \{r_i\}, AG^{ST} \rangle, \quad (2.17)$$

где: MG_D – фрагмент метаграфа данных и знаний (параметры миварной сети), на основе которого выполняются правила из множества $\{r_i\}$ (правила миварной сети); AG^{ST} – стартовое условие выполнения агента, которое может быть представлено фрагментом метаграфа.

Отличия миварного агента от метаграфового агента проявляются в следующем:

- **Миварный агент оперирует исключительно замкнутыми правилами.** Такие правила допускают лишь установление атрибутов вершин метаграфа, связанных с параметрами миварной сети. Данная особенность связана со спецификой реализации миварной технологии обработки информации и обеспечивает отсутствие заикливания при построении логического вывода (бесконечное переопределение ранее известных значений параметров) [165].
- **Миварному агенту не требуется указывать первое правило для запуска.** Миварная технология обработки информации [165] предлагает механизм, при котором последовательность выполнения правил определяется автоматически на основе набора известных параметров, без явного указания первого выполняемого правила.

Как отмечалось ранее, обработку миварной сети осуществляет МЛВ. Если интерпретировать процедуру обработки в МЛВ в терминах миварного агента, то установку значений параметров в соответствии с текущим состоянием ПрО можно рассматривать как результат совместного функционирования метаграфовых агентов. В данной интерпретации миварный и метаграфовый агенты взаимодействуют посредством общего фрагмента метаграфа, что обеспечивает согласованность об-

работки данных и подготовки логического вывода. Такое взаимодействие можно исследовать с помощью модели, схематичный вид которой представлен на рисунке 2.6.

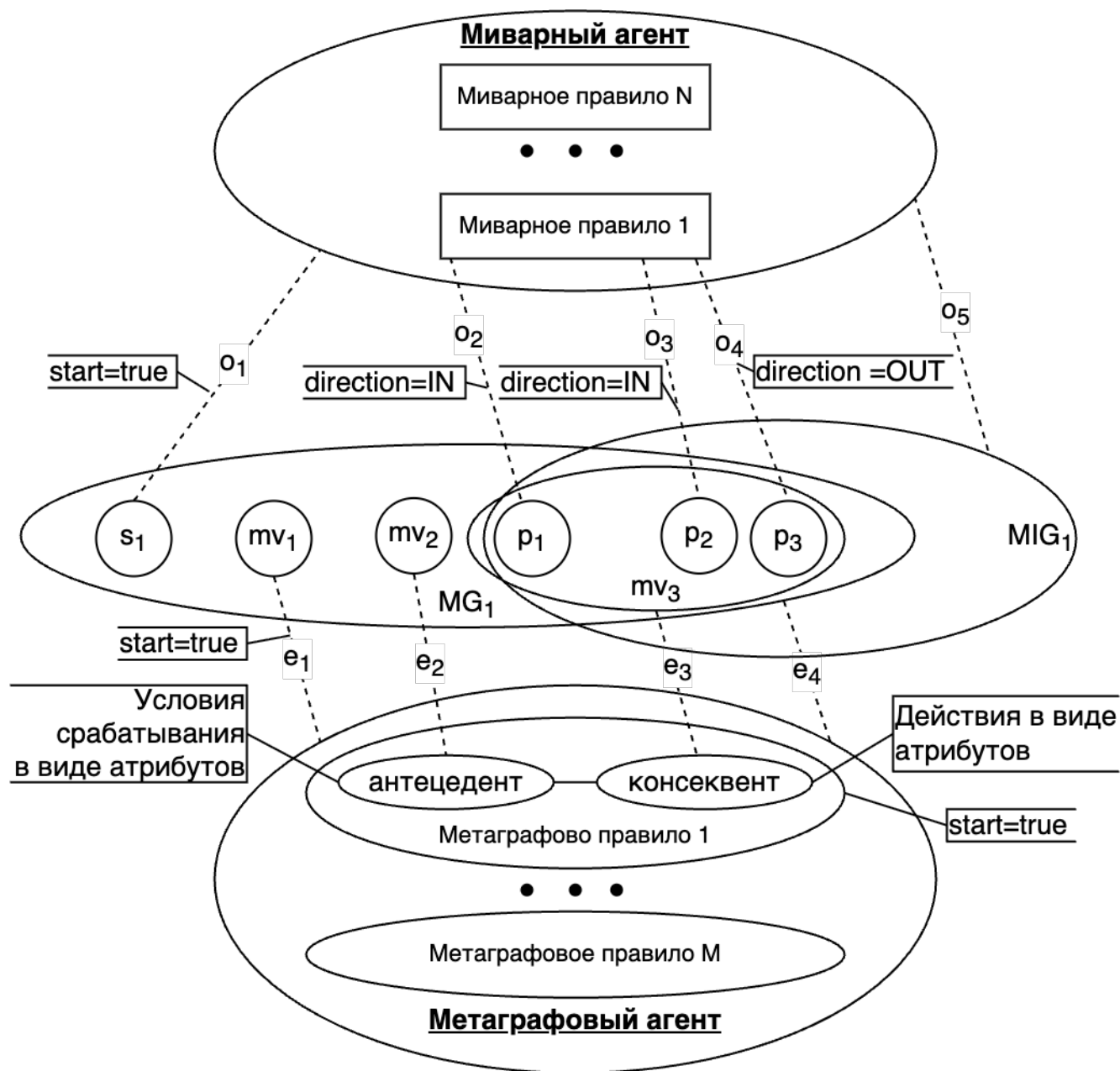


Рисунок 2.6 — Модель миварной проблемно-ориентированной системы управления и принятия решений в пространстве состояний

На рисунке 2.6 для наглядности вершины метаграфа, соответствующие правилам миварной сети, представлены прямоугольниками. Количество правил у метаграфового агента и миварного агента равно $M \in \mathbb{N}$ и $N \in \mathbb{N}$ соответственно. По аналогии с изображением на рисунке 2.5, подробно рассматриваются одно правило метаграфового агента и одно правило миварного агента. С рёбрами o_2 , o_3 и o_4 связаны атрибуты, указывающие на принадлежность входному или выход-

ному множеству миварного правила 1 миварных параметров (метавершин) p_1 , p_2 и p_3 соответственно: атрибут `direction=IN` указывает на принадлежность параметра входному множеству, а `direction=OUT` – выходному множеству. Фрагмент метаграфа s_1 определяет условие активации миварного агента и соединён с ним посредством ребра o_1 с атрибутом `start=true`. Фрагмент метаграфа o_1 может быть обработан метаграфовым агентом, что позволяет управлять моментом запуска миварного агента. На рисунке также изображён фрагмент метаграфа MIG_1 и его связь с миварным агентом через ребро o_5 . Фрагмент MIG_1 соответствует знаниям и данным, на основании которых выполняются правила миварного агента. Данный фрагмент не обязательно входит в состав MG_1 , поскольку информация о текущем состоянии ПрО может поступать от других метаграфовых и миварных агентов.

Предлагаемая модель системы интегрирует миварный и метаграфовый подходы и предоставляет инструментарий для проектирования и анализа ГИИС, ориентированных на решение циклических управленческих задач в пространстве состояний. Основные возможности модели выглядят следующим образом:

1. **Сегментация знаний посредством метаграфового представления миварной сети.** Модель поддерживает явное распределение знаний, используемых для различных задач ПрО или этапов обработки информации, что повышает выразительную мощность КМПрО.
2. **Формализация интерфейсов обмена информацией между миварными и другими информационными системами.** Применение метаграфовых и миварных агентов для описания ГИИС позволяет строго задавать структуру данных и знаний на границах взаимодействия различных подсистем, что упрощает проектирование и анализ интеграционных связей.
3. **Поддержка различных способов построения КМПрО при решении задач методом разбиения на подзадачи.** Если модель ПрО ориентирована на решение задачи целиком, то миварная система может быть описана одним миварным агентом. В случаях, когда предполагается разбиение задачи на подзадачи и использование отдельных МБЗ для их решения, представление миварной системы может включать несколько миварных агентов, каждый из которых отвечает за свою часть процесса.
4. **Формализация цикло-календарного планирования на основе миварного подхода.** Возможность задания стартовых условий миварных и метагра-

фовых агентов в виде фрагментов метаграфа позволяет использовать предложенную модель для описания систем с циклическими процессами, в том числе календарными и производственными циклами. Циклический вызов МЛВ инициируется другим агентом ГИИС в результате изменения стартовых условий рассматриваемого миварного агента.

Разработанная модель миварной системы представляет собой полезный инструмент для проектирования и исследовательского анализа ГИИС. Применение данной модели способствует повышению архитектурной согласованности МЛВ и других информационных компонентов в составе технической системы.

2.6 Выводы по второй главе

1. Разработана обобщённая модель миварной базы знаний, ориентированная на решение управленческих задач в пространстве состояний. Модель строится на основе миварного подхода и метода разбиения задачи на подзадачи, обеспечивая структурированное представление знаний через логически сгруппированные множества параметров и правил. Такая структура способствует декомпозиции сложных задач, облегчает масштабирование базы знаний и поддерживает её адаптацию к изменению среды ПрО.
2. Предложенная модель миварной базы знаний обладает важными свойствами, такими как гибкость, масштабируемость, структурная декомпозиция и поддержка параллельной обработки логического вывода. Эти свойства позволяют применять модель в различных прикладных областях – от систем управления производственными процессами до координации роботизированных комплексов и сельскохозяйственных объектов. Благодаря модульной организации, фрагменты миварной базы знаний могут быть многократно использованы и автоматически генерированы на основе результатов системного анализа предметной области, что снижает трудозатраты когнитолога и повышает адаптацию технической системы в целом.
3. Для исследования архитектуры миварной проблемно-ориентированной си-

стемы управления и принятия решений применён метаграфовый подход, позволяющий формализовать взаимодействие миварной машины логического вывода с другими интеллектуальными и программными модулями в составе гибридной интеллектуальной информационной системы. В результате была получена модель, для описания которой используются метаграфовые и миварные агенты. Миварный агент представляет собой специализированный метаграфовый агент, работающий с замкнутыми правилами и автоматически определяющий порядок активации правил на основе текущих значений параметров, что согласуется с миварной технологией обработки информации.

4. Модель миварной системы обеспечивает строгую формализацию интерфейсов взаимодействия между подсистемами, поддерживает сегментацию знаний по функциональным задачам и этапам обработки, а также позволяет описывать как одноэтапные, так и циклические сценарии планирования и управления.
5. Интеграция миварного и метаграфового подходов создаёт единую методологическую основу для проектирования, разработки и анализа сложных интеллектуальных систем управления. Такой синтез обеспечивает согласованность между представлением знаний, механизмом логического вывода и программными решениями, реализующими методы машинного обучения, большие языковые модели и иные технологии искусственного интеллекта.

3 МЕТОДЫ АВТОМАТИЧЕСКОЙ ГЕНЕРАЦИИ МИВАРНЫХ БАЗ ЗНАНИЙ И РЕШЕНИЯ ЗАДАЧ ПЛАНИРОВАНИЯ ДЕЙСТВИЙ В ПРОСТРАНСТВЕ СОСТОЯНИЙ

3.1 Постановка задач разработки методов

В условиях, когда объекты управления и их количество могут изменяться как в ходе выполнения задачи, так и от задачи к задаче, возникает необходимость в механизмах динамической адаптации базы знаний. При традиционном подходе к формированию МБЗ [125] когнитолог вручную формализует экспертные знания с использованием специализированного программного инструментария. В результате полученная МБЗ остаётся статичной на всём протяжении решения задачи, а адаптация к изменениям ПрО обеспечивается исключительно за счёт гибкости технологии обработки информации – миварной технологии логического вывода. Однако такой подход не позволяет масштабировать структуру знаний под изменяющуюся конфигурацию объектов управления. Это обуславливает первую задачу разработки метода автоматической генерации МБЗ [3], обеспечивающего динамическое масштабирование структуры знаний в зависимости от текущей задачи и состояния ПрО. Разрабатываемый метод должен генерировать базу знаний под задачу, используя заранее подготовленные шаблоны и прототипы правил и параметров, что позволит системе управления адаптироваться к новым условиям без участия когнитолога и в режиме реального времени.

Вторая задача заключается в разработке метода решения задач планирования действий в пространстве состояний на основе автоматически сгенерированной МБЗ [8, 60]. Одним из ключевых требований к такому методу является поддержка циклической обработки миварных сетей для автоматической генерации частных

(промежуточных) планов и их оперативной корректировки в соответствии с текущим состоянием ПрО. Особое значение приобретает возможность пошагового планирования: система должна формировать и уточнять план по мере поступления новой информации и изменения внешних условий, сохраняя при этом работоспособность в режиме реального времени. Разрабатываемый метод должен быть согласован с моделью миварной проблемно-ориентированной системы управления и принятия решений, обеспечивая согласованное взаимодействие подсистем наблюдения, анализа, планирования и исполнения.

Оба метода должны быть тесно интегрированы в рамках миварной интеллектуальной системы планирования и опираться на формализованную модель МБЗ, предложенную во второй главе данной работы. Метод автоматической генерации должен поддерживать структурную декомпозицию знаний по подзадачам и обеспечивать возможность параллельной обработки фрагментов базы знаний. Метод принятия решений, в свою очередь, должен реализовывать итеративный цикл «анализ состояния – планирование шага – выполнение – обратная связь», что позволит системе оперативно реагировать на изменения внешней среды и корректировать поведение технической системы в реальном времени.

3.2 Метод автоматической генерации миварных баз знаний для решения управленческих задач в пространстве состояний

3.2.1 Обобщённый процесс автоматической генерации миварной базы знаний для решения задач в пространстве состояний

Для подготовки баз знаний на основе МТ ЛИИ необходимо учитывать совокупность факторов, связанных с особенностями формализации знаний и их структурной организации [16, 136]. В частности, в работе [137] предложена методика подготовки исходных данных для создания МБЗ, предназначенных для реше-

ния задач принятия решений в роботизированных средствах и РТК. Отдельного внимания заслуживают исследования, посвящённые автоматической и автоматизированной подготовке данных для формализации знаний на основе миварного подхода. Так, в работах [142, 166] показана возможность применения больших генеративных моделей для извлечения и формализации знаний из текстовых инструкций. В исследовании [167] рассматривается алгоритм автоматизированного формирования МБЗ на основе данных, хранящихся в специально организованных реляционных базах данных.

Несмотря на достигнутые результаты в области формализации и генерации МБЗ, сохраняется актуальность задачи обеспечения их динамической адаптации в условиях изменения конфигурации управляемых объектов и состояния ПрО. В связи с этим предлагается разработка метода автоматической генерации МБЗ [3], предназначенных для решения задач планирования и управления в пространстве состояний. В основу данного метода положена модель МБЗ, учитывающая особенности автоматического накопления знаний и поддерживающая применение метода разбиения задачи на подзадачи. Данная модель обеспечивает систематизацию формализованных знаний и позволяет выделять области, в которых генерация баз знаний осуществляется на основе заранее подготовленных шаблонов и прототипов. Процесс генерации МБЗ реализуется в рамках миварной проблемно-ориентированной системы, модель которой рассмотрена в предыдущей главе. В такой системе генератор базы знаний представляется в виде специализированного агента. Таким образом, предлагаемый метод представляет собой совокупность взаимосвязанных процессов и моделей, включающая обобщённый процесс автоматической генерации, процесс подготовки исходных данных и модель представления генератора базы знаний.

На рисунке 3.1 представлено обобщённое представление процесса автоматической генерации МБЗ. В общем случае решаемая задача заключается в переводе среды ПрО из начального состояния в заданное целевое с учётом существующих ограничений и допустимых действий. Структура МБЗ, на основе которой осуществляется синтез решения, строится в соответствии с методом разбиения задачи на подзадачи: в рамках миварной сети выделяются подграфы, каждый из которых отвечает за решение конкретных подзадач. Поскольку число объектов управления в ПрО не является фиксированным и может изменяться в ходе функционирования системы, для решения задач применяется механизм автома-

тической генерации баз знаний. В результате миварная сеть формируется таким образом, чтобы обеспечивать достижение всех целевых состояний подзадач. При изменении целей или состава объектов управления МБЗ повторно генерируется с учётом актуального состояния среды ПрО и параметров текущей задачи.

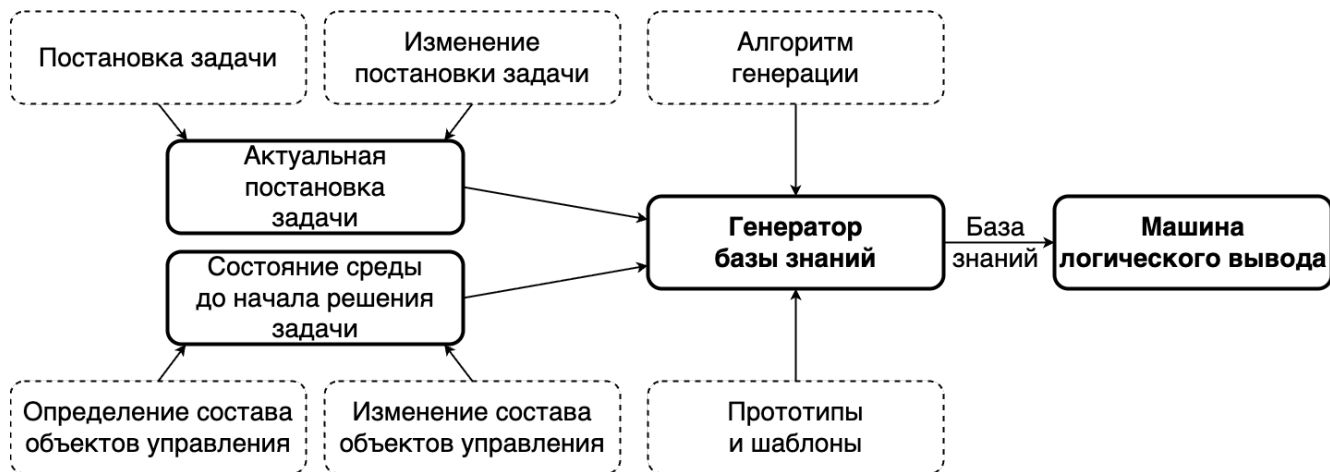


Рисунок 3.1 — Обобщённый процесс автоматической генерации миварной базы знаний для решения задач в пространстве состояний

3.2.2 Подготовка исходных данных для автоматической генерации миварных баз знаний

На этапе проектирования системы когнитолог проводит системный анализ ПрО, по результатам которого формируется обобщённый алгоритм решения задач. В ходе анализа также выполняется предварительная формализация знаний и подготовка отдельных фрагментов миварной сети. Подготовленные фрагменты совместно с алгоритмом генерации МБЗ, также разработанным когнитологом, используются для синтеза базы знаний для решения конкретной задачи. Фрагменты миварной сети могут включать:

- **Прототип параметра** – параметр, формально представляемый в виде кортежа (2.3), уникальное имя IDP_i которого определяется в процессе генерации. У такого параметра определено только значение VAL_i . Такой параметр используется для задания характеристики, присущей определённому типу

объектов. Имя параметра даётся при соотнесении данной характеристики с конкретным объектом среды ПрО. В качестве примера такой характеристики можно привести максимальную грузоподъёмность объекта управления определенного типа.

- **Прототип правила** – правило, формально представляемое в виде кортежа (2.4), у которого уникальное имя IDR_i , параметры IN_i и параметры OUT_i определяются в процессе генерации. У такого правила конкретно определено только действие ACT_i . Такие прототипы применяются, например, для создания правил, используемых при анализе состояния объектов управления.
- **Шаблон параметра** – параметр, у которого все элементы кортежа (2.3) определяются в процессе генерации. На их основе, например, создают параметры, которые характеризуют текущее состояние объектов управления.
- **Шаблон правила** – правило, у которого все элементы кортежа (2.4) определяются в процессе генерации. Шаблоны правил используются тогда, когда заранее неизвестно, со сколькими параметрами работает правило. Например, для создания правил, которые обеспечивают проверку достижимости целей подзадач, связанных между собой.

Реализация прототипов и шаблонов параметров и правил определяется конкретным используемым инструментарием формализации и обработки МБЗ. Рассмотрим особенности их реализации при использовании КЭСМИ Wi!Mi «Разуматор» версии 2.1 [133].

Параметры и правила в КЭСМИ Wi!Mi «Разуматор» версии 2.1 целесообразно рассматривать с позиций объектно-ориентированного подхода. В этом случае можно ввести класс «Параметр», характеризуемый следующими атрибутами:

- *Идентификатор параметра* – уникальная идентифицирующая строка, реализующая элемент IDP_i кортежа (2.3). Идентификатор параметра не повторяется в пределах базы знаний.
- *Значение параметра*, представляющее собой реализацию элемента VAL_i кортежа (2.3).

Класс «Правило» включает следующие атрибуты:

- *Идентификатор правила* – уникальная идентифицирующая строка, реализующая элемент IDR_i кортежа (2.4). Идентификатор правила также является уникальным в рамках базы знаний.

- *Тело правила*, представляющее реализацию действия правила АСТ_i из кортежа (2.4). Тело правила задаётся в виде программы (скрипта) на языке JavaScript [168]. Для тела правила определяются два набора переменных: входные переменные, значения которых инициализируются до выполнения программы, и выходные переменные, значения которых извлекаются из среды выполнения после завершения её работы.
- *Связи входных переменных тела правила с параметрами базы знаний*. Реализуется множеством IN_i кортежа (2.4). Данные связи представляются в виде структуры типа «словарь», в которой ключ соответствует имени входной переменной тела правила, а значение – идентификатору параметра базы знаний.
- *Связи выходных переменных тела правила с параметрами базы знаний*. Реализуется множеством OUT_i кортежа (2.4). Аналогично входным связям, они также задаются в виде словаря, где ключом является имя выходной переменной тела правила, а значением – идентификатор параметра базы знаний.

При активации правила в КЭСМИ Wi!Mi «Разуматор» версии 2.1 значения параметров, входящих во входное множество правила, передаются в среду выполнения программы в соответствии с заданными связями: для каждого элемента словаря создаётся переменная с именем, соответствующим ключу, и значением параметра базы знаний, идентификатор которого указан в значении данного элемента. После выполнения тела правила значения выходных переменных извлекаются из среды выполнения и присваиваются параметрам выходного множества по идентификаторам, заданным в соответствующем словаре. Таким образом, обеспечивается двусторонний перенос значений параметров миварной базы знаний между формализованной моделью и программной реализацией действия правила на языке JavaScript.

Для рассмотренной реализации параметров и правил в КЭСМИ Wi!Mi «Разуматор» версии 2.1 определения составляющих прототипов и шаблонов приведены в виде, представленном в таблице 3.1. Таблица фиксирует, какие атрибуты экземпляров классов «Параметр» и «Шаблон» задаются на этапе системного анализа предметной области (до генерации), а какие определяются непосредственно в процессе автоматической генерации МБЗ.

Для генерации фрагментов миварной сети на основе шаблонов параметров

и правил описываются специальные алгоритмы. Для шаблонов правил эти алгоритмы определяют порядок формирования словарей, соответствующих связям переменных тела правил с параметрами базы знаний. Генерация программ тел шаблонов правил может осуществляться с использованием специализированных программных средств – текстовых шаблонизаторов.

Как отмечалось ранее, подготовка прототипов и шаблонов параметров и правил осуществляется на основе результатов системного анализа ПрО, выполняемого когнитологом. Процедуры работы с ПрО могут реализовываться в соответствии с выбранной методикой. Так, в работе [137] представлена методика подготовки баз знаний для роботов, робототехнических средств и РТК. Данный подход возможно использовать в качестве основы при разработке методики подготовки знаний для их автоматической генерации в процессе решения управленческих задач в пространстве состояний с применением метода разбиения задач на подзадачи.

Таблица 3.1 — Реализация прототипов и шаблонов параметров и правил КЭСМИ Wi!Mi «Разуматор» версии 2.1

Фрагмент	До генерации	В процессе генерации
Прототип параметра	Значение параметра	Идентификатор параметра
Прототип правила	Тело правила	Идентификатор правила
	Связи переменных (входных и выходных) тела правила с параметрами базы знаний: до генерации определяются только ключи соответствующих словарей, а их значения (идентификаторы параметров) – в процессе генерации	
Шаблон параметра	—	Идентификатор и значение параметра
Шаблон правила	—	Идентификатор правила, тело правила, связи переменных (входных и выходных) тела правила с параметрами базы знаний

Для успешного применения выбранной методики когнитологу необходимо

дополнительно сформулировать требования к исходным данным с учётом специфики решаемых задач и особенностей автоматической генерации МБЗ. В качестве основы для формирования таких требований можно рассматривать следующие положения [8]:

1. Алгоритм решения задачи, реализуемый на основе МБЗ, должен обладать свойствами обобщаемости и масштабируемости по отношению к числу объектов, участвующих в решении. Это обеспечивает возможность предварительного задания прототипов и шаблонов параметров и правил, на базе которых формируется конкретная МБЗ, адаптированная к заданной задаче и текущему состоянию ПрО.
2. Задача, решаемая с использованием МБЗ, должна допускать декомпозицию с учётом структуры и множества оперируемых объектов. В рамках такого подхода исходная задача представляется в виде совокупности подзадач, каждая из которых ориентирована на управление или преобразование состояния отдельного объекта в пространстве состояний.
3. ПрО, для которой осуществляется автоматическая генерация МБЗ, должна включать множество однотипных объектов, взаимодействие с которыми осуществляется по унифицированным правилам, а их состояния допускают формализованное описание в единой модели. В этих условиях автоматическая генерация базы знаний применяется для учёта изменений в количестве объектов и допустимых состояний среды. На основе заранее подготовленных прототипов и шаблонов в базу знаний добавляются параметры и правила, обеспечивающие анализ текущего состояния конкретного объекта и выбор последующего действия с учётом состояния ПрО.

Для применения автоматической генерации миварных баз знаний при решении управленческих задач в пространстве состояний на этапе разработки и проектирования системы необходимо реализовать следующую последовательность шагов:

1. Проведение системного анализа ПрО:

- Определить класс и специфику задач, предполагаемых к решению с применением автоматически генерируемых баз знаний. Например, задачи перестановки объектов в пространстве при заданных ограничениях и доступной совокупности действий.
- Подготовить обобщённый алгоритм решения задач, для которых пре-

дусматривается использование автоматической генерации миварных баз знаний.

- Идентифицировать и формализовать объекты управления, участвующие в процессе решения задач.
- Описать способ декомпозиции исходных задач на подзадачи.

2. Формирование основы для генерации базы знаний:

- Выделить и формализовать совокупность знаний, инвариантных по отношению к изменениям среды ПрО.
- Подготовить прототипы и шаблоны параметров и правил, ассоциированных с объектами управления, которые будут использоваться для построения конкретных структур базы знаний под заданную задачу.
- Сформировать прототипы и шаблоны параметров и правил, предназначенные для генерации структур, обеспечивающих решение отдельных подзадач.

3. Разработка генератора базы знаний:

- Разработать алгоритм автоматической генерации базы знаний с учётом класса решаемых задач и результатов предварительной формализации знаний о ПрО.
- Реализовать программный инструмент, обеспечивающий автоматическое формирование миварных баз знаний с учётом конкретной управленческой задачи и состояния ПрО.

3.2.3 Модель автоматического генератора миварных баз знаний

В программном продукте КЭСМИ Wi!Mi «Разуматор» версии 2.1 для описания и долговременного хранения баз знаний используется текстовый формат документов XML (англ. *eXtensible Markup Language* – расширяемый язык разметки) [125]. Взаимодействие с внешними информационными системами осуществляется посредством программного интерфейса. Поскольку загрузка МБЗ в МЛВ осуществляется в формате XML, при проектировании систем на основе КЭСМИ Wi!Mi

«Разуматор» с автоматической генерацией баз знаний необходимо использовать специализированные средства генерации, обеспечивающие подготовку миварной сети в требуемом формате. В простейшем случае такой генератор может быть реализован в виде программы текстового шаблонизатора, предназначенной для формирования фрагментов миварной сети в виде XML-элементов и их последующей компоновки в единый документ, загружаемый в МЛВ.

Следует отметить, что автоматический генератор МБЗ может быть реализован различными способами [142, 166]. Выбор конкретной реализации определяется спецификой ПрО, особенностями решаемых управленческих задач, требованиями к масштабируемости и адаптивности системы, а также принятыми архитектурными решениями и используемым инструментарием. В связи с этим для изучения генератора в составе интеллектуальной системы необходимо рассматривать его в обобщённом виде независимо от конкретной программной реализации.

В рамках интеллектуальной системы генератор МБЗ предлагается рассматривать как самостоятельный функциональный компонент и описывать его с использованием формальной модели. В частности, представляется возможным рассматривать генератор в виде метаграфового агента и исследовать его функционирование в контексте ГИИС (рисунок 2.6), что позволяет формализовать его взаимодействие с другими компонентами системы. Возможный вариант модели метаграфового агента генератора баз знаний представлен на рисунке 3.2. Данное представление основывается на методе описания метаграфовых агентов, предложенном в [146, 148], и модели, формализация которой рассмотрена в предыдущей главе (рисунок 2.5).

Метаграфовый агент генератора МБЗ формализуется в виде отдельной вершины метаграфа, формализация которого изложена в работе Ю. Е. Гапанюка [148]. В состав данной метавершины входят несколько вложенных метавершин, каждая из которых соответствует отдельному метаграфовому правилу. Для примера на рисунке 3.2 так же, как на рисунке 2.5, подробно демонстрируется одно метаграфовое правило (метаграфовое правило 1). Метаграфовое правило 1 включает метавершины, соответствующие антецеденту и консеквенту, при этом условия активации правила и выполняемые им действия задаются соответствующими атрибутами. С метаграфовым агентом ассоциирован метаграф MG_1 , на основе которого осуществляется выполнение вложенных правил в метаграфовом агенте. Связь метаграфового агента с MG_1 отображается ребром e_4 . В структуре метаграфа MG_1

выделяются фрагменты mv_1 , mv_2 и mv_3 . Начальное условие функционирования метаграфового агента задаётся фрагментом mv_1 (связь через ребро e_1 с атрибутом $start=true$). Атрибутом $start=true$ помечается правило, выполняемое первым (метаграфовое правило 1). Фрагмент mv_2 посредством ребра e_2 связывается с антецедентом метаграфового правила 1, тогда как фрагмент mv_3 – с его консеквентом через ребро e_3 .

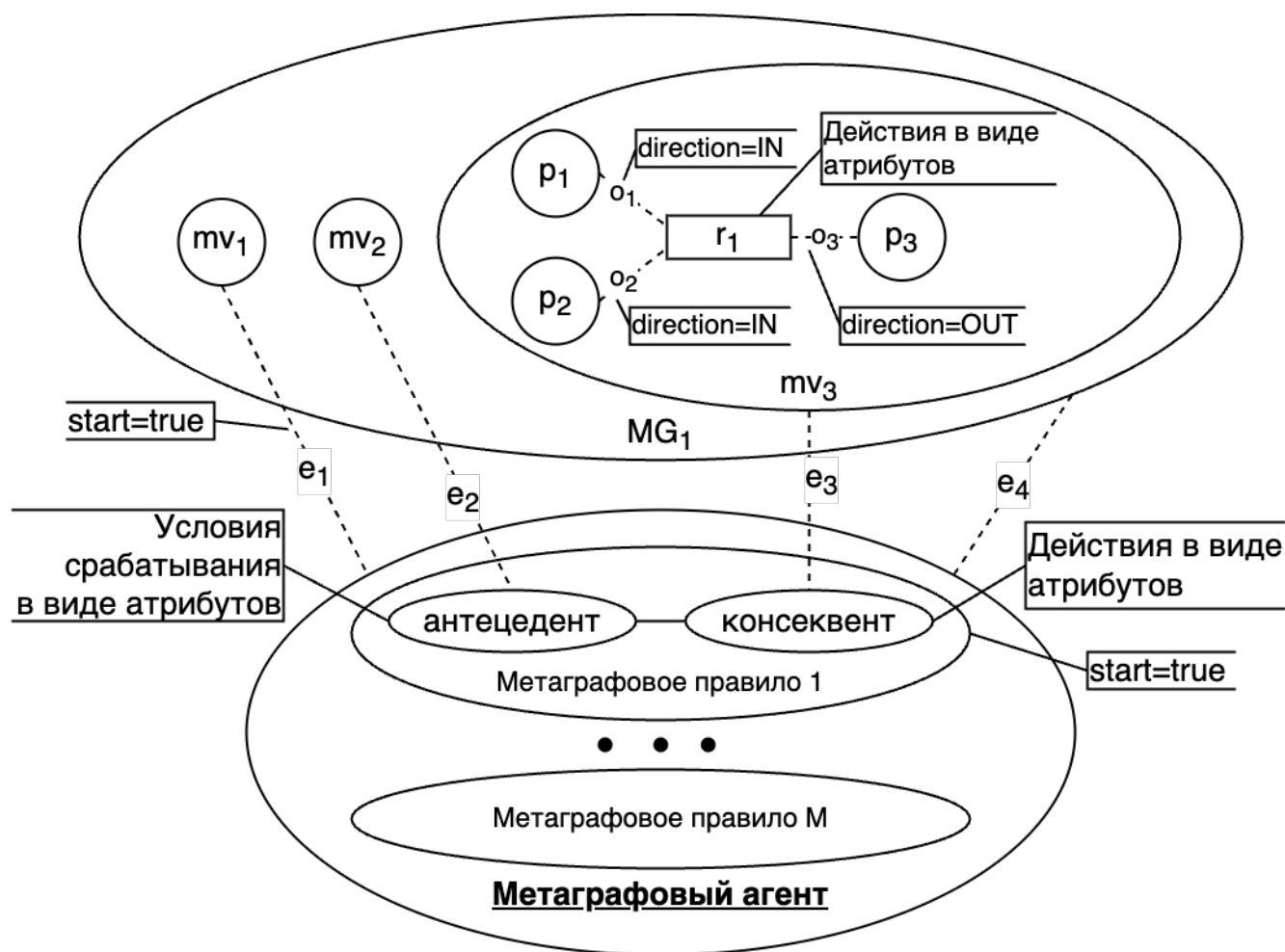


Рисунок 3.2 — Модель генератора миварных сетей для решения управленческих задач в пространстве состояний

Как отмечалось ранее, метаграфовые правила делятся на разомкнутые и замкнутые. Метаграфовый агент, отображающий генератор МБЗ, порождает новые фрагменты метаграфа и модифицирует их. Иными словами, метаграфовый агент генератора МБЗ работает с разомкнутыми и замкнутыми правилами. При выполнении определённых условий фрагмент метаграфа может быть преобразован в миварную сеть и обратно, что показано в работах [163] и [162] соответственно. На рисунке 3.2 показано, что в результате срабатывания метаграфового правила 1 формируется фрагмент метаграфа, включающий рёбра o_1 , o_2 и o_3 , а также вер-

шины r_1 , p_1 , p_2 и p_3 . Вложенный граф фрагмента mv_3 является ненаправленным и соответствует отображению подграфа миварной сети на метаграф. Для указания принадлежности миварных параметров входному или выходному множеству правила r_1 используются атрибуты: атрибут $direction=IN$ обозначает отнесение параметра ко входному множеству, тогда как атрибут $direction=OUT$ – к выходному. Значения параметров, а также ссылки на действия правил могут дополнительно задаваться в виде атрибутов соответствующих вершин.

3.2.4 Анализ метода автоматической генерации миварных баз знаний для решения управленческих задач в пространстве состояний

Использование метода автоматической генерации МБЗ при решении управленческих задач в пространстве состояний обеспечивает ряд возможностей при проектировании и разработке интеллектуальных систем:

1. **Адаптивность миварной системы к изменениям среды ПрО.** Разработанный метод позволяет создавать системы, способные к автоматическому масштабированию своих баз знаний в зависимости от количества и состава управляемых объектов. Это достигается за счёт применения заранее формализованных фрагментов знаний, которые используются в качестве основы для генерации. В результате когнитолог получает возможность описывать алгоритм решения задачи в обобщённом виде, тогда как сама система адаптирует данный алгоритм к конкретному состоянию ПрО и текущей задаче. Такой подход обеспечивает оперативную реакцию системы на изменения числа управляемых объектов и их состояний.
2. **Формализованное описание интерфейсов взаимодействия генератора баз знаний.** Формализация взаимодействия генератора баз знаний с другими компонентами интеллектуальной системы реализуется посредством преобразований между метаграфовым и миварным концептуальными моделями ПрО. Это позволяет унифицировать и структурировать представление знаний и данных, передаваемых между компонентами системы, а также обес-

печивать их корректную интерпретацию при смене формата представления. Примеры таких преобразований представлены в работах [162, 163].

3. **Поддержка различных подходов к генерации баз знаний.** С метаграфовым подходом обеспечивается возможность описания агентов, функционирующих на основе различных методов искусственного интеллекта, включая продукционные правила, нейронные сети, методы нечёткой логики, эволюционные алгоритмы и другие [146]. Использование метаграфового агента в качестве модели генератора миварных баз знаний позволяет абстрагироваться от конкретных технических реализаций и сосредоточиться на анализе архитектурных свойств миварных систем. В результате система представляется в виде сложного графа взаимосвязанных компонентов, каждый из которых может быть модифицирован или заменён без нарушения целостности общей структуры.

Таким образом, предложенный метод является полезным инструментом для проектирования и анализа миварных проблемно-ориентированных систем, решающих управленческие задачи в пространстве состояний с использованием автоматической генерации баз знаний. Его применение направлено на совершенствование архитектуры миварной системы, повышение её гибкости и адаптивности к изменениям ПрО.

3.3 Метод решения задач принятия решений и обработки информации для планирования действий в пространстве состояний

3.3.1 Описание метода решения задач принятия решений и обработки информации для планирования действий в пространстве состояний

Для решения задач планирования в пространстве состояний с использованием МТ ЛИИ в условиях динамически изменяющейся среды ПрО необходимо при-

менять методы, обеспечивающие автоматическую подготовку и адаптацию баз знаний непосредственно в процессе функционирования системы. Такие методы должны поддерживать итеративное, пошаговое формирование планов действий, при котором построение частных (промежуточных) планов осуществляется с учётом текущего состояния ПрО и результатов исполнения предыдущих шагов. В данных условиях возникает потребность в многократных обращениях к МЛВ для получения логического вывода по миварной сети при формировании очередного набора действий, направленных на преобразование среды ПрО. Разработке соответствующего метода посвящено дальнейшее изложение.

Все элементы, связанные с реализацией планирования действий на основе МТ ЛИИ, можно структурно выделить в отдельный компонент технической системы, далее обозначаемый как **миварный планировщик**. Данный компонент реализует **вычислительный цикл планирования**, включающий формирование запросов к МЛВ, интерпретацию результатов логического вывода и выработку согласованных управляющих воздействий для исполнительных подсистем. Получение информации о состоянии окружающей среды осуществляется за счёт взаимодействия миварного планировщика с подсистемами предобработки данных, предназначенными для приведения входной информации к формату, совместимому с формализацией в виде миварных сетей.

Помимо миварного планировщика, в составе технической системы могут присутствовать и другие компоненты, обеспечивающие обработку поступающей информации и подготовку управляющих сигналов. Совокупность компонентов, реализующих МТ ЛИИ, образует миварную проблемно-ориентированную систему управления. В свою очередь, совокупность всех информационных систем, функционирующих в рамках технической системы, можно рассматривать как ГИИС. На рисунке 3.3 представлена схема включения миварного планировщика в структуру технической системы. Данная схема иллюстрирует взаимодействие планировщика с внешней средой, МЛВ и исполнительными модулями технической системы.

На рисунке 3.3 планировщик представлен совокупностью следующих функциональных элементов: МЛВ, генератор базы знаний, рабочая память, координатор и компоновщик плана действий. Структурный элемент «координатор» представляет собой программный модуль, обеспечивающий управление потоками данных и запуск процессов обработки между остальными компонентами планировщика. Его функционирование определяется алгоритмами, разработанными для реше-

ния заданного класса задач планирования и подготовленными на основе результатов системного анализа ПрО. В состав планировщика включена «рабочая память», предназначенная для хранения данных о текущем состоянии ПрО, актуальной задаче планирования, выполненных действиях, текущих решаемых подзадачах, а также другой вспомогательной служебной информации. Элемент «компоновщик плана действий» предназначен для интерпретации результатов логического вывода по миварной сети и формирования последовательности действий, подлежащих исполнению.

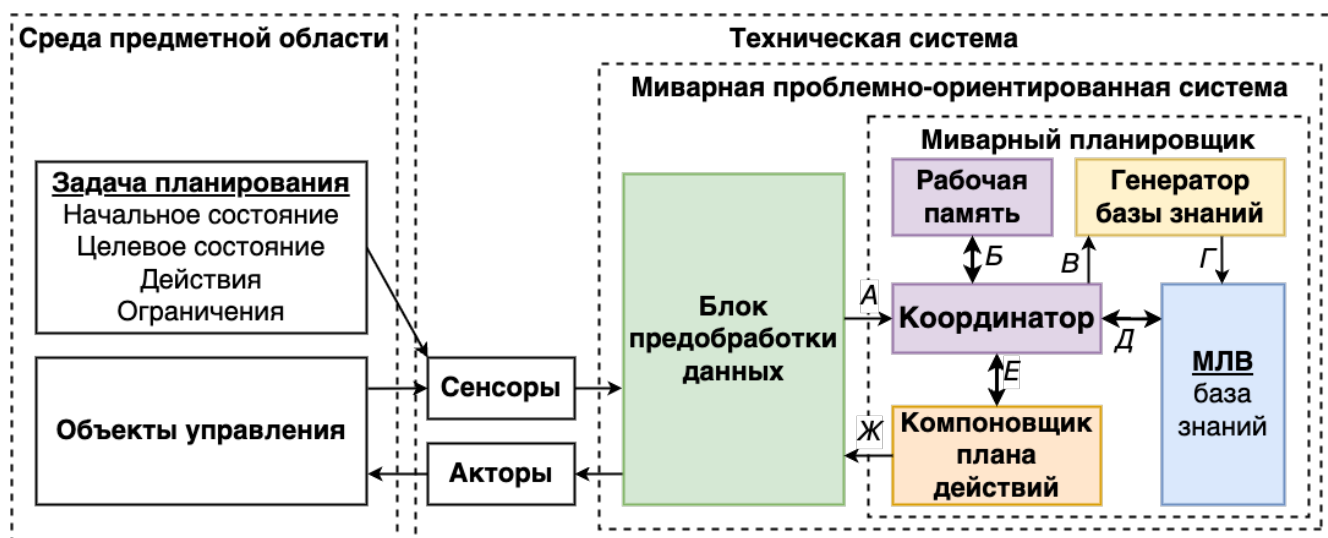


Рисунок 3.3 — Миварный планировщик в пространстве состояний в составе технической системы

Для построения базы знаний необходимо определить набор действий, доступных для взаимодействия со средой ПрО. Указанные действия задаются доменом задачи и составом исполнительных средств технической системы. В рамках рассматриваемого метода решения задач каждому действию может быть сопоставлен параметр миварной сети. В случае если параметр, ассоциированный с действием, в результате логического вывода по миварной сети правил получает определённое значение, техническая система при текущем состоянии среды должна выполнить соответствующее действие.

Рассмотрим подробнее поток данных, представленный на рисунке 3.3. Задача планирования, которая в формализованном виде представляется кортежем (1.2), а также информация о текущем состоянии ПрО поступают в техническую систему через сенсоры. Данные, получаемые с сенсоров и необходимые для функционирования планировщика, проходят этап предварительной обработки и преобразуются в формат, совместимый с представлением знаний в виде миварных

сетей. Далее данные от блока предобработки поступают в координатор планировщика (стрелка А). Рабочая память и координатор связаны между собой напрямую (стрелка Б). На начальном этапе решения задачи планирования данные из рабочей памяти передаются в генератор базы знаний (стрелка В). Генератор на основе поступивших данных готовит МБЗ. Сгенерированная МБЗ загружается в МЛВ (стрелка Г). Для получения плана действий, соответствующего текущему состоянию ПрО, координатор передаёт актуальное представление о среде в МЛВ и иницирует процесс построения логического вывода (стрелка Д). В результате МЛВ возвращает вычисленные данные в виде набора параметров миварной сети с их значениями. Полученные значения фиксируются в рабочей памяти и передаются в компоновщик плана действий (стрелка Е). Сформированная последовательность действий после преобразования в управляющие сигналы передаётся на исполнение (стрелка Ж). В результате выполнения действий состояние ПрО изменяется. Описанный цикл движения данных повторяется до тех пор, пока состояние ПрО не достигнет целевого состояния задачи планирования.

Если состав объектов управления остаётся неизменным и объекты управления корректно изменяют своё состояние в соответствии с результатами исполнения плана действий, МБЗ генерируется однократно на начальном этапе решения задачи и не изменяется в процессе планирования. В случае изменения целевого состояния, утраты достигнутых целевых состояний или изменения состава объектов управления дальнейшее планирование рассматривается в рамках новой задачи, для которой формируется новая МБЗ с учётом актуального состояния ПрО.

МБЗ планировщика формируется генератором с учётом модели, иллюстрация которой представлена на рисунке 2.4. Применение данной модели обеспечивает выполнение процесса, при котором при решении каждой подзадачи достигается по меньшей мере одна промежуточная цель общей задачи планирования, при этом количество подзадач и промежуточных целей совпадает. Общая стратегия и алгоритм решения задач планирования формируются когнитологом и экспертом в предметной области с учётом метода генерации МБЗ. Следует отметить, что в ряде случаев отдельные подзадачи не могут быть решены изолированно и требуют предварительного достижения определённого набора промежуточных целей. В результате достижение целевого состояния задачи планирования представляет собой последовательный процесс, реализуемый через поэтапное достижение промежуточных целей за счёт решения соответствующих подзадач.

В качестве примера поэтапного достижения промежуточных целей можно рассмотреть задачу планирования действий по перемещению объектов в среде. Алгоритм действий, реализуемый в соответствии с планом, представляет собой последовательную обработку каждого объекта, в рамках которой выполняются три шага (рисунок 3.4):

1. *подготовить среду* – подготовка целевой площадки для размещения объекта;
2. *подготовить текущий объект* – устранение препятствий для перемещения объекта;
3. *достигнуть текущую цель* – непосредственное перемещение объекта в целевую позицию.

Три шага обработки рассматриваются в рамках подзадачи, решение которой соответствует достижению промежуточной цели задачи планирования. При этом размещение отдельных объектов среды может быть невозможным без предварительного перемещения других объектов, например при необходимости размещения одного объекта над другим. В связи с этим подзадачи решаются последовательно, а соответствующие им промежуточные цели рассматриваются как взаимосвязанные. Как следствие, в любой момент планирования однозначно известно то, какая подзадача решается и с каким объектом происходит взаимодействие.

Для решения задачи планирования перемещения объектов в МБЗ необходимо включить правила, отражающие пространственное положение объектов управления. Каждому объекту сопоставляется фрагмент миварной сети, на основе которого анализируется его текущее положение в пространстве и определяется наличие признаков размещения объекта в положении, предписанном целевым состоянием задачи планирования. Генерация данных фрагментов для МБЗ происходит автоматически под конкретную задачу планирования.

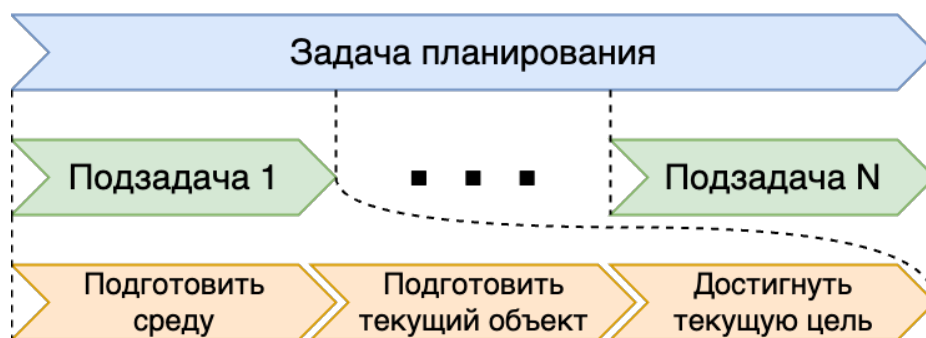


Рисунок 3.4 — Декомпозиция задачи планирования перемещения объектов в пространстве

Важно отметить, что для решения отдельной подзадачи не требуется априорного знания полного плана действий, что позволяет реализовать два режима работы миварной проблемно-ориентированной системы:

- Поэтапное формирование плана действий для достижения очередной промежуточной цели с немедленной передачей сформированного промежуточного плана на исполнение и последующим повторением данного цикла для всех промежуточных целей. В этом режиме компоновщик плана действий передаёт промежуточные планы по мере их формирования на исполнительные органы. В результате миварная проблемно-ориентированная система работает **как система управления и принятия решений**, а полное решение задачи планирования формируется путём последовательного объединения всех промежуточных планов, переданных на исполнение акторам технической системы.
- Предварительное построение полного плана решения задачи с последующей передачей его на акторы технической системы. В этом случае в компоновщик загружается полное представление о начальном состоянии ПрО и задаче планирования. Вместо непосредственной передачи планов действий на исполнение (стрелка Ж на рисунке 3.3), компоновщик воспроизводит изменения состояния ПрО в собственной рабочей памяти. Результат преобразования состояния ПрО, с которым работает компоновщик, передаётся координатору планировщика в качестве актуального состояния внешней среды. После достижения в воспроизводимой модели целевого состояния задачи планирования частичные планы, сформированные МЛВ, агрегируются в единый план и передаются на акторы технической системы. Таким образом, полное решение задачи планирования формируется и выдаётся после завершения внутренней обработки данных миварным планировщиком. Миварная проблемно-ориентированная система работает **как планировщик действий в пространстве состояний**.

Далее предлагается рассмотреть конкретную реализацию метода решения задач принятия решений и обработки информации для планирования действий в пространстве состояний, а именно результаты разработки системы планирования МИПРА (англ. *Mivar-based Intelligent Planner of Robot Actions*, MIPRA) [8].

3.3.2 Применение метода решения задач принятия решений и обработки информации для решения задач планирования домена «Мир кубиков»

Формализация задачи миварного планирования действий робота для домена «Мир кубиков»

Домен задачи планирования, с которым работает система МИПРА, представляет собой модификацию классического домена Blocks World [169], далее обозначаемую как «Мир кубиков» (англ. *Cubes World*) [6]. В рамках данного домена осуществляется планирование перемещения объектов в пространстве. Объектами управления являются кубики в количестве N штук ($N \in \mathbb{N}$), которые могут устанавливаться друг на друга, образуя башни различной высоты. Башни размещаются на поверхности стола на специально выделенных площадках. Размещение кубиков на столе допускается только на этих площадках. На каждой площадке может быть размещена не более чем одна башня минимальной высотой в один кубик. Перемещение кубиков осуществляется роботом с рукой-манипулятором. Максимальное количество площадок составляет M штук ($M \in \mathbb{N}, M \geq 3$). Площадки пронумерованы, при этом нумерация начинается с нуля, а номер последней площадки равен $M - 1$. Кубики также пронумерованы от 0 до $N - 1$. Для удобства описания башни могут быть пронумерованы, при этом их номера совпадают с номерами площадок, на которых они расположены. Пример состояния ПрО задачи планирования в домене «Мир кубиков» представлен на рисунке 3.5.

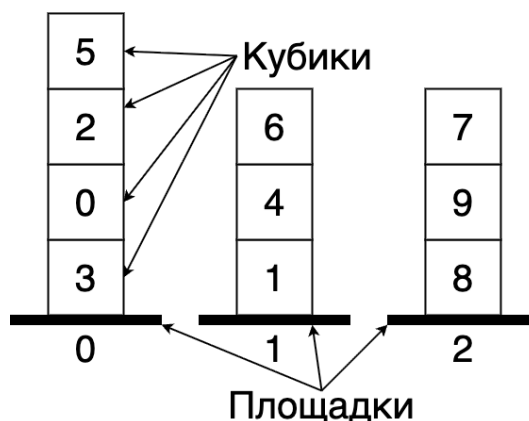


Рисунок 3.5 — Пример текущего состояния предметной области задачи планирования в домене «Мир кубиков»

Для исследуемого домена «Мир кубиков» введём формализованное описа-

ние состояния ПрО. Обозначим через States множество всех возможных состояний задачи. Для описания состояния S ($S \in \text{States}$) можно использовать упорядоченное множество вида

$$S = (P_0, P_1, \dots, P_{M-1}), \quad (3.1)$$

где элемент P_i соответствует конфигурации башни из кубиков, размещённой на площадке с номером i ($i \in [0, M - 1]$). Описание P_i представляет собой упорядоченное множество, элементами которого являются номера кубиков. В случае отсутствия кубиков на площадке i элемент P_i задаётся пустым множеством, то есть $P_i = ()$. Если на площадке размещены кубики, то в P_i последовательно перечисляются номера кубиков в порядке их расположения – от нижнего кубика к верхнему.

Рассмотрим пример преобразования состояния ПрО. На рисунке 3.6 состояние S_2 получено из состояния S_1 в результате выполнения роботом некоторого плана действий. Состоянию S_1 соответствует множество

$$S_1 = ((0, 1, 2), (3), (4, 5, 6)), \quad (3.2)$$

тогда как состоянию S_2 соответствует множество

$$S_2 = ((0, 1, 2), (), (4, 5, 6, 3)). \quad (3.3)$$

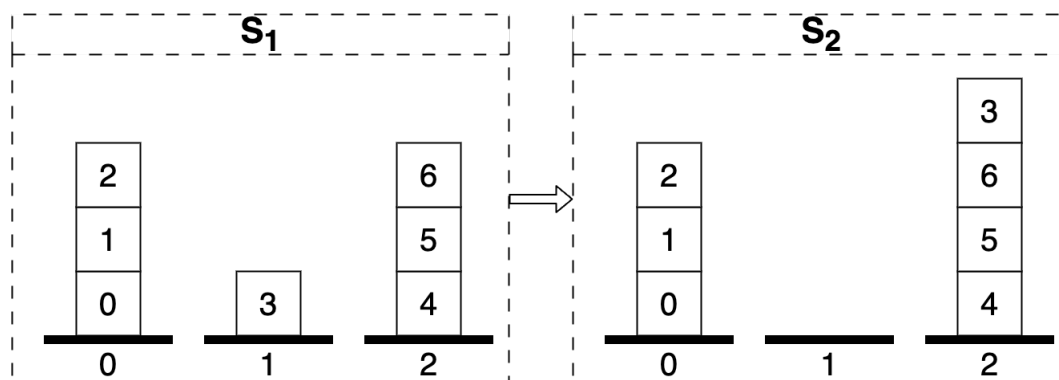


Рисунок 3.6 — Пример преобразования состояния предметной области задачи планирования в домене «Мир кубиков»

Робот оснащён одной механической рукой-манипулятором, обеспечивающей выполнение действий по перестановке кубиков. Для восприятия окружающей среды используется система технического зрения, предоставляющая информацию

о пространственном расположении объектов. Система технического зрения является частью технической системы робота, в состав которой также входит система МИПРА. Наличие перечисленных акторов и сенсоров технической системы позволяет роботу выполнять действия из множества Actions, направленные на изменение состояния ПрО. Множество действий Actions определим по аналогии с действиями, приведёнными в исследовании [170]:

- $MoveCube(i, j)$ – взять один кубик с вершины башни № i и переместить его на площадку № j . В случае наличия башни на площадке № j кубик устанавливается на вершину башни № j .
- $MoveCubes(i, j, k)$ – k раз подряд выполнить действие $MoveCube(i, j)$.
- $FindTargetCube(i)$ – определить номер площадки, на которой размещена башня, содержащая кубик № i .
- $CubeOnTarget(i)$ – зафиксировать в памяти робота факт установки кубика № i в целевое положение.
- $WaitNextRequest()$ – ожидать поступления следующего частного (промежуточного) плана действий от планировщика.
- $Stop()$ – остановить процесс решения задачи планирования.

Следует отметить, что манипулирование кубиками допускается только при условии снятия кубика с вершины башни и его установки либо на вершину другой башни, либо на свободную площадку. Система технического зрения распознаёт положение кубиков в пространстве и для каждого из них определяет следующие характеристики:

- $CurrentTower$ – номер башни, в составе которой находится кубик.
- On – номер кубика, на котором расположен рассматриваемый кубик. Если кубик находится непосредственно на поверхности стола, данному параметру присваивается значение -1 .
- $OverFree$ – флаг, отражающий наличие или отсутствие кубиков, расположенных выше рассматриваемого кубика в текущей башне.

Обозначим $InitialState$ ($InitialState \in States$) как состояние ПрО, в котором кубики находились до начала манипуляций робота (начальное состояние). Под $TargetState$ ($TargetState \in States$) будем понимать состояние ПрО, в которое необходимо перевести кубики посредством выполнения действий робота из множества Actions. Решением задачи планирования T будет являться последовательность действий робота, обеспечивающих преобразование состояния предметной обла-

сти из InitialState в TargetState:

$$T = \text{InitialState} \rightarrow \text{TargetState}. \quad (3.4)$$

Обобщённый алгоритм решения задачи планирования с использованием миварных баз знаний

В МИПРА задача планирования декомпозируется в соответствии с представлением, приведённым на рисунке 3.4. На основе данной декомпозиции формируется обобщённый алгоритм действий робота, который реализуется как последовательность *итераций*, каждая из которых включает три *шага*: подготовка среды, подготовка текущего объекта и достижение текущей цели. Каждая итерация соответствует решению отдельной подзадачи. Последовательное выполнение шагов, подготовленных МИПРА, обеспечивает полный переход среды ПрО из начального состояния в требуемое целевое состояние.

В результате выполнения итерации робот должен установить как минимум один кубик в положение, заданное целевым состоянием задачи планирования TargetState, что соответствует достижению одной *промежуточной цели* (подцели). Формирование каждой башни осуществляется последовательно, начиная с нижних кубиков, которые в целевом состоянии располагаются непосредственно на поверхности стола. Кубики, установленные на *целевые позиции*, исключаются из дальнейшего рассмотрения и в последующем не перемещаются в пространстве. Построение башен выполняется в порядке возрастания нумерации площадок (от 0 до $M - 1$).

В начальном состоянии кубики, как правило, не находятся в целевых положениях. В связи с этим для построения очередной башни роботу необходимо выполнить совокупность вспомогательных действий, включающих поиск кубика, подлежащего перемещению в рамках текущей подзадачи, подготовку соответствующей целевой площадки и последующее перемещение выбранного кубика в требуемое положение. Такие действия совершаются в рамках шагов итерации. В рамках одного шага может выполняться несколько последовательных действий робота из множества Actions.

В качестве примера итерации рассмотрим преобразование состояния S_1 в S_2 , изображённое на рисунке 3.6. Предположим, что роботу требуется разместить кубик № 3 на площадку № 2. Тогда роботу потребуется предпринять несколько

действий: определить положение кубика № 3 ($\text{FindTargetCube}(3)$) и переставить кубик с площадки № 1 на площадку № 2 ($\text{MoveCube}(1, 2)$).

В процессе выполнения шагов может возникнуть ситуация, при которой некоторая промежуточная цель оказывается достигнутой ранее запланированного момента. В этом случае планировщик исключает связанный с данной целью кубик из дальнейшего рассмотрения, и робот не выполняет с ним последующих манипуляций.

Следует отметить, что для подготовки плана действий одного шага система МИПРА, а именно координатор миварного планировщика (рисунок 3.4), вызывает МЛВ для построения логического вывода. Поскольку кубики, установленные в целевые положения, не перемещаются в пространстве, на каждом последующем шаге объём логического вывода сокращается за счёт уменьшения числа рассматриваемых кубиков. Это обеспечивается тем, что в рабочей памяти планировщика фиксируется факт достижения промежуточных целей.

В соответствии с разработанным методом принятия решений и обработки информации, при изменении количества кубиков в среде, числа доступных площадок для размещения, утрате целевой позиции каким-либо кубиком или обновлении конечной цели система МИПРА инициирует процесс решения задачи планирования заново. В этом случае на основе текущего положения кубиков и актуальной формулировки задачи планирования в МИПРА генерируется МБЗ и запускается процесс подготовки плана.

Для каждого кубика № a формулируется подцель T_i ($a, i \in [0, N - 1]$). В случае если в целевом состоянии кубик в башне должен располагаться первым, соответствующая подцель описывается структурой вида

$$T_i = \langle \text{Кубик: } a, \text{ Башня: } b, \text{ На: } -1, \text{ ЯвляетсяИстинной: } - \rangle, \\ \forall \begin{cases} i, a \in [0, N - 1], \\ b \in [0, M - 1], \end{cases} \quad (3.5)$$

где: N – количество кубиков; M – количество площадок; i – номер промежуточной цели; a – номер кубика, с которым связана подцель № i ; b – номер площадки, где должен располагаться кубик № a .

Каждой подцели T_i сопоставляется флаг TF_i , в случае достижения этой подцели принимающий значение ИСТИНА, в противном случае – ЛОЖЬ. Если в целевом

состоянии кубик стоит на другом кубике, то соответствующая промежуточная цель описывается структурой вида

$$T_i = \langle \text{Кубик: } a, \text{ Башня: } b, \text{ На: } c, \text{ ЯвляетсяИстинной: } TF_{i-1} \rangle, \quad \forall \begin{cases} i, a, c \in [0, N - 1], \\ b \in [0, M - 1], \end{cases} \quad (3.6)$$

где TF_{i-1} – флаг подцели № $i - 1$, достижение которой является обязательным условием перехода к реализации подцели № i ; c – номер кубика, на котором должен быть размещён кубик № a . Если под кубиком № a нет других кубиков, то $c = -1$ и вместо TF_{i-1} ставится прочерк (выражение (3.5)). Зависимость между целями T_i и флагами TF_{i-1} обеспечивает корректный порядок размещения кубиков в любой башне на столе в состоянии TargetState. Кроме того, подобная декомпозиция цели на подцели исключает возникновение аномалий Сассмана [88].

Для иллюстрации способа описания промежуточных целей рассмотрим конкретный пример. Пусть целевое состояние соответствует расположению кубиков, представленному на рисунке 3.5. В формализованном виде оно задаётся в виде

$$S_{\text{ц}} = ((3, 0, 2, 5), (1, 4, 6), (8, 9, 7)). \quad (3.7)$$

Нумерация промежуточных целей соответствует фактическому порядку следования кубиков в упорядоченном множестве $S_{\text{ц}}$ (рисунок 3.7). Так, размещение кубика № 3 в целевом положении соответствует цели № 0, кубика № 5 – цели № 3, кубика № 1 – цели № 4, кубика № 7 – цели № 9 и т.д. Расстановка кубиков по целевым местам начинается с площадки № 0 и завершается на площадке № 2. Достижение подцелей осуществляется в порядке их нумерации.

Промежуточная цель № 0 выглядит следующим образом:

$$T_0 = \langle \text{Кубик: } 3, \text{ Башня: } 0, \text{ На: } -1, \text{ ЯвляетсяИстинной: } - \rangle. \quad (3.8)$$

Промежуточная цель № 5:

$$T_5 = \langle \text{Кубик: } 4, \text{ Башня: } 1, \text{ На: } 1, \text{ ЯвляетсяИстинной: } TF_4 \rangle. \quad (3.9)$$

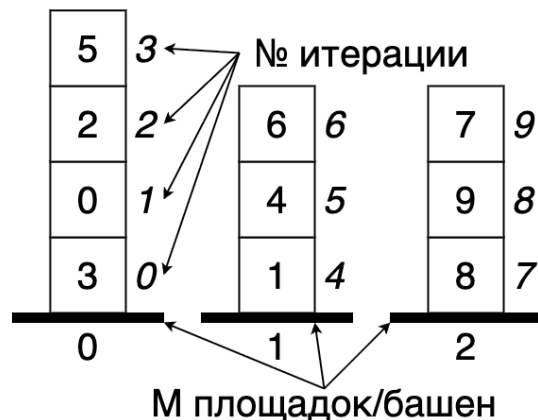


Рисунок 3.7 — Пример нумерации промежуточных целей по целевому состоянию предметной области задачи планирования в домене «Мир кубиков»

Обобщённый алгоритм действий, реализуемый в соответствии с планом, генерируемым МИПРА при работе с доменом «Мир кубиков», можно представить в виде псевдокода (алгоритм 3.1).

Алгоритм 3.1 — Обобщённый алгоритм действий технической системы при работе с доменом «Мир кубиков»

Вход: состояния InitialState и TargetState задачи планирования T

1. определить количество кубиков N на основе InitialState и TargetState
2. определить количество площадок M на основе InitialState и TargetState
3. **для** каждого кубика i ($i \in [0, N - 1]$) **выполнить**
4. инициализировать подцели T_i на основе TargetState
5. **конец для**
6. **для** каждой подцели T_i ($i \in [0, N - 1]$) **выполнить**
7. CurrentState \leftarrow текущее состояние среды
8. $j \leftarrow$ номер площадки, где находится кубик № a в CurrentState
9. **если** площадка № b не пуста в CurrentState **то**
10. **если** $c = -1$ **то**
11. освободить площадку № b , переместив все кубики на площадку/башню k , при этом $k \in [0, M - 1]$, $k \neq j$, $k \neq b$
12. **иначе**
13. снять все кубики с площадки № b до тех пор, пока кубик № c не станет свободным, перемещая при этом все снятые кубики на площадку/башню k , где $k \in [0, M - 1]$, $k \neq j$, $k \neq b$
14. **конец если**
15. **конец если**

16. **если** над кубиком № a стоят другие кубики **то**
17. снять все кубики с кубика № a , переместив их на площадку/башню k ,
при этом $k \in [0, M - 1], k \neq j, k \neq b$
18. **конец если**
19. переместить кубик № a с площадки j на площадку/башню b
20. **конец для**

Подробный пример решения задачи планирования в соответствии с представленным алгоритмом приведён в работе [8].

Генерация миварной базы знаний для решения задачи планирования

Для решения любой задачи планирования домена «Мир кубиков» система МИПРА генерирует МБЗ, структура которой соответствует модели, представленной на рисунке 2.4. С учётом особенностей обобщённого алгоритма решения задач для данного домена в структуре этой МБЗ могут быть выделены собственные множества и уровни вложенности, иерархия которых проиллюстрирована на рисунке 3.8.

На рисунке 3.8 представлены следующие множества:

- Через параметры множества *Current state* в МЛВ поступает информация о текущем и целевом состояниях кубиков, о состоянии площадок/башен и об итерациях алгоритма.
- Множество *Cubes* описывает текущее состояние совокупности кубиков.
- Через параметры множества *Cube [i]* в МЛВ попадает информация о текущем положении кубика № i в пространстве (параметры *CurrentTower*, *On* и *OverFree*).
- Параметры множества *Iterations* содержат информацию об итерациях алгоритма решения задачи планирования. Параметры данного множества предварительно вычисляются на основе *TargetState* задачи планирования T . Они передаются в МЛВ при каждом запросе и участвуют в формировании логического вывода.
- В множестве *Iteration [i]* содержатся параметры, описывающие промежуточную цель № i .
- С помощью множества *InitialActions [i]* описываются доступные действия робота, с которых начинается итерация № i .
- Параметры множества *Targets [i]* описывают подцель № i .

- В параметрах множества Towers хранится информация о текущем состоянии площадок/башен.
- С помощью множества Management МЛВ управляет процессом планирования.
- Параметры множества Actions определяют действия, подлежащие выполнению на текущем шаге.
- Параметры множества Processing отвечают за управление ходом процесса планирования.
- Параметры множества SetValue обеспечивают изменение переменных окружения в рабочей памяти планировщика.
- Флаги достижения промежуточных целей представлены множеством IntermediateTargets.

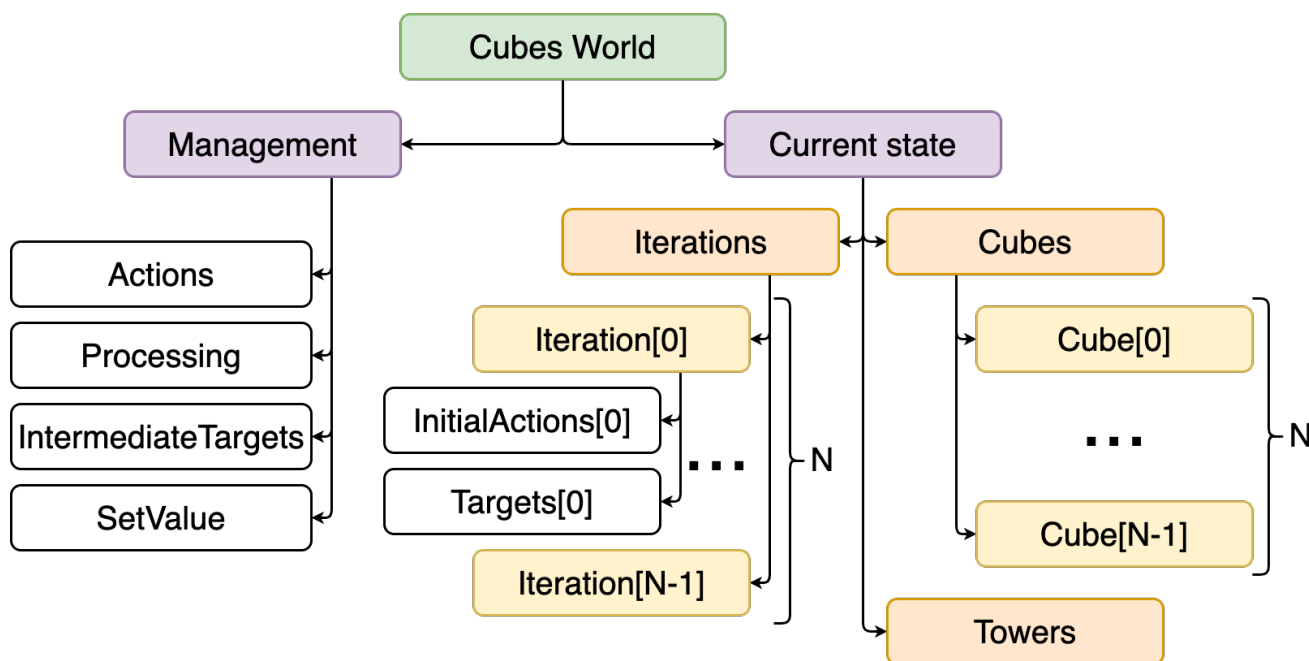


Рисунок 3.8 — Иерархия множеств параметров миварной базы знаний для решения задач планирования в домене «Мир кубиков»

Правила, связывающие параметры МБЗ из множеств, представленных на рисунке 3.8, генерируются на основе следующих прототипов и шаблонов:

- PCheckTarget – проверка выполнения условий промежуточной цели.
- PStartWorking – инициализация процесса планирования.
- PIterationInProgress – освобождение текущего целевого кубика путём снятия объектов, препятствующих его перемещению.
- PIterationBeforeCompleting – установка текущего целевого кубика на целевую площадку/башню при отсутствии мешающих объектов сверху.

- PIterationComplete – завершение текущей итерации и переход к следующей.
- PFinish – завершение процесса планирования при достижении целевого состояния ПрО.

Подробный пример генерации базы знаний в МИПРА приведён в работе [8]. Кроме того, пример решения задачи рассмотрен в приложении А.

В следующей главе подробно рассматриваются особенности программной реализации МИПРА, в частности вопросы создания МЛВ, обеспечивающей её функционирование. Кроме того, в данной главе представлены результаты вычислительных экспериментов с МИПРА, а также анализ разработанного программного обеспечения на основе предложенных моделей и методов планирования действий при принятии управленческих решений.

3.4 Выводы по третьей главе

1. Обоснована необходимость разработки методов динамической адаптации миварных баз знаний в условиях изменяющейся конфигурации объектов управления и состояния предметной области. Это обусловило постановку двух взаимосвязанных задач: автоматической генерации миварной базы знаний и планирования действий в пространстве состояний на основе автоматически сформированной базы знаний.
2. Предложен метод автоматической генерации миварной базы знаний, основанный на использовании заранее подготовленных прототипов и шаблонов параметров и правил. Метод обеспечивает масштабируемое и гибкое формирование структуры знаний под конкретную задачу без участия когнитолога в режиме реального времени. Определены требования к исходным данным и регламентирована последовательность проектных этапов, включающая системный анализ предметной области, формализацию инвариантных знаний и подготовку шаблонов, а также разработку и реализацию генератора миварных баз знаний.

3. Создана модель генератора миварной базы знаний в виде метаграфового агента, обеспечивающая унифицированное описание интерфейсов взаимодействия компонентов проблемно-ориентированной системы и формальное представление преобразований между метаграфовым и миварным уровнями. Применение данной модели поддерживает архитектурную целостность миварной системы, обеспечивает корректность межуровневых преобразований представления знаний и допускает использование различных методов искусственного интеллекта.
4. Сформулирован и реализован метод планирования действий в пространстве состояний, функционирующий в виде итеративного цикла «анализ состояния – формирование шага плана – исполнение – обратная связь» и реализуемый как миварный планировщик. Метод поддерживает как пошаговое построение частных планов, так и генерацию полного плана, а также предусматривает повторное планирование и повторную генерацию миварных баз знаний при изменении целей, состояния или конфигурации среды.
5. Применимость предложенных методов подтверждена на примере домена «Мир кубиков», в рамках которого выполнена формализация предметной области и алгоритма решения задач. Целевое состояние предметной области представлено в виде упорядоченной системы промежуточных целей с флагами достижимости, что обеспечивает корректную последовательность манипуляций и исключает возникновение аномалий Сассмана.

4 ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ НА ОСНОВЕ МОДЕЛЕЙ И МЕТОДОВ ПЛАНИРОВАНИЯ ДЕЙСТВИЙ ПРИ ПРИНЯТИИ УПРАВЛЕНЧЕСКИХ РЕШЕНИЙ

4.1 Постановка задач разработки программного обеспечения

Разработка программного обеспечения в рамках диссертационного исследования направлена на практическую реализацию предложенных моделей и методов интеллектуального планирования действий при принятии управленческих решений, основанных на МТ ЛИИ. Ключевой задачей является создание программной реализации МЛВ, отвечающей следующим требованиям: возможность функционирования на бортовых вычислительных платформах роботов, поддержка автономного режима работы, задействование доступного числа вычислительных потоков, возможность выборочного вычисления выходных переменных правил (без обязательного расчёта всех выходных параметров) и обеспечение параллельной обработки независимых правил. Данная реализация призвана преодолеть ограничения предыдущих версий МЛВ, обусловленные последовательной обработкой правил и жёсткой привязкой к обязательному вычислению всех выходных параметров. В основе разрабатываемой МЛВ лежит математическое и алгоритмическое обеспечение обработки миварных сетей логических параметров и правил, позволяющее эффективно работать с МБЗ, построенными в соответствии с моделью для решения управленческих задач в пространстве состояний.

Разработанная МЛВ используется для программной реализации миварного планировщика системы МИПРА, предназначенной для решения задач в домене «Мир кубиков». Программная реализация планировщика поддерживает как пошаговое формирование и немедленное исполнение частичных планов, так и

предварительное построение полного плана с последующей передачей на исполнение, а также обеспечивает взаимодействие с подсистемами предобработки сенсорных данных и управления исполнительными устройствами. При этом структура МИПРА полностью соответствует модели миварной проблемно-ориентированной системы управления и принятия решений в пространстве состояний, а процесс планирования реализуется на основе методов автоматической генерации МБЗ и решения задач принятия решений, адаптированных к специфике домена «Мир кубиков».

Для демонстрации универсальности и гибкости предложенных решений создан комплекс программных систем, включающий ЛИСК ПДД, ЛИС ОУР и ПКИАС. Эти системы подтверждают применимость предложенных моделей и методов к широкому спектру прикладных задач, выходящих за рамки сферы РТК.

Завершающей задачей разработки является экспериментальная проверка работоспособности и эффективности реализации. В качестве объекта испытаний выступает система МИПРА. Результаты вычислительных экспериментов используются для обоснования достигнутых показателей производительности, масштабируемости и применимости моделей и методов в задачах интеллектуального планирования действий в пространстве состояний.

4.2 Математическое и алгоритмическое обеспечение обработки миварных сетей логических параметров и правил

4.2.1 Краткая хронология развития алгоритмов обработки миварных сетей логических параметров и правил

Обзор подходов к созданию МЛВ, основанных на МТ ЛИИ, а также их классификация представлены в работе [19]. На основе результатов данного обзора далее анализируются особенности алгоритмов обработки миварных сетей и формулиру-

ются требования к разрабатываемой МЛВ, предназначенной для интеллектуального планирования действий в пространстве состояний.

Одной из ранних практических реализаций миварных МЛВ является программный комплекс «УДАВ» [171]. Условно данное МЛВ можно отнести к *первому типу*. МЛВ данного типа работают с правилами вида «если известны все значения входных параметров, то рассчитать значения всех выходных параметров» [172]. Данное представление отличается от того, что рассматривалось в параграфе 2.2. Требование вычисления всех значений выходных параметров правила при его активации связано со следующей особенностью обработки МБЗ: для построения логического вывода по миварной сети правил сначала выполняется поиск последовательности правил, активация которых потенциально приводит к вычислению искомого значения параметров задачи по известным параметрам, затем производится расчёт найденной последовательности. Обнаруженная последовательность правил задаёт **маршрут логического вывода по миварной сети параметров и правил** [123].

Псевдокод обобщённого алгоритма обработки миварной сети в программном комплексе «УДАВ» приведён в виде алгоритма 4.2 [171]. В рамках данного алгоритма можно выделить ряд особенностей, используемых в процессе поиска маршрута логического вывода:

1. Если в ходе поиска найдено правило, позволяющее вычислить значение искомого параметра (из множества find), но для его активации отсутствуют значения всех входных параметров, неизвестные входные параметры данного правила включаются в перечень искомого параметров (множество find).
2. Если в ходе поиска найдено правило, позволяющее вычислить значение искомого параметра (из множества find), и при этом значения всех его входных параметров известны (включены в множество known), такое правило добавляется в маршрут (упорядоченное множество route) и исключается из поиска. Сам искомый параметр помечается как известный (множество known).

Процесс поиска завершается, когда все искомые параметры задачи можно вычислить. Далее происходит последовательная активация правил по обнаруженному маршруту. Завершение поиска также наступает в ситуации, когда построить алгоритм решения невозможно. В этом случае для МЛВ либо отсутствуют значения части параметров миварной сети, либо отсутствует часть знаний для формирования маршрута логического вывода.

Алгоритм 4.2 — Обобщённый алгоритм обработки миварной сети машиной логического вывода программного комплекса «УДАВ»

Вход: множества параметров P и правил R миварной сети, множества известных I и искомых O параметров задачи

Выход: маршрут логического вывода route и искомые значения параметров O

1. route \leftarrow () ▷ упорядоченное множество
2. check $\leftarrow R$
3. known \leftarrow известные параметры из множества P
4. known $\leftarrow I \cup (\text{known} \setminus I)$
5. find $\leftarrow O$
6. *начало:*
7. repeat \leftarrow false
8. cycle \leftarrow find
9. **для** каждого параметра i из множества cycle **выполнить**
10. **для** каждого правила j из множества check **выполнить**
11. output \leftarrow выходные параметры правила j
12. **если** $i \in \text{output}$ **то**
13. input \leftarrow входные параметры правила j
14. **если** input \subseteq known **то**
15. добавить j в начало множества route
16. known \leftarrow known \cup output
17. check \leftarrow check $\setminus \{j\}$
18. **иначе**
19. **если** (input \setminus known) \setminus find $\neq \emptyset$ **то**
20. find \leftarrow (input \setminus known) \cup find
21. repeat \leftarrow true
22. **конец если**
23. **конец если**
24. **конец для**
25. **конец для**
26. **конец для**
27. **если** $O \subseteq$ known **то**
28. последовательно активировать правила из route и зафиксировать результаты вычислений в known

29. **иначе**
30. **если** repeat = true **то** ▷ если было добавление в find
31. **перейти к началу**
32. **иначе**
33. **сообщить о нехватке знаний для решения задачи**
34. **конец если**
35. **конец если**
36. **вернуть** множества route и known

По результатам работы над программным комплексом «УДАВ» были предложены различные способы усовершенствовать МЛВ первого типа [132, 134, 135]. Одно из предложений воплотилось в программном комплексе “MivarLogicParallel” [132, 172, 173]. Используемый МЛВ в этом комплексе условно можно отнести ко *второму типу*.

МЛВ данного типа для построения логического вывода использует кластеризацию вычислений. Вычислительный процесс реализуется на нескольких аппаратных и/или программных узлах, взаимодействующих по схеме «ведущий—ведомые»: ведущий узел отвечает за хранение и актуализацию множества известных и искомых параметров, а ведомые узлы – за обработку правил МБЗ. На этапе анализа задачи построения логического вывода ведущий узел осуществляет назначение ведомых узлов и распределяет между ними правила МБЗ. В процессе решения задачи ведущий модуль информирует ведомые узлы о текущем состоянии множества известных параметров. Ведомые узлы выполняют отбор правил, которые могут быть активированы в текущий момент, после чего отобранные правила активируются. Вычисленные значения ранее неизвестных параметров передаются ведущему узлу, а активированные правила исключаются из дальнейшего рассмотрения. Ведущий модуль агрегирует поступившие результаты, актуализирует множество известных параметров и, при необходимости, осуществляет повторную рассылку обновлённого состояния на ведомые узлы. Работа МЛВ завершается либо после вычисления значений всех искомых параметров задачи, либо при отсутствии новых результатов активации правил со стороны ведомых узлов.

Отметим принципиальные различия МЛВ первого и второго типа:

- *Направление построения маршрута логического вывода.* Алгоритм МЛВ первого типа осуществляет построение маршрута от множества искомых

параметров (условие «найти» задачи, **обратный поиск правил**), тогда как алгоритм МЛВ второго типа – от множества известных параметров (условие «дано» задачи, **прямой поиск правил**).

- *Параллельность обработки правил.* В МЛВ второго типа обработка правил всегда осуществляется параллельно. В МЛВ первого типа можно параллельно обрабатывать множества искомых параметров find (cycle) при условии корректной реализации синхронизации множеств check, known и find между вычислительными потоками.
- *Активация правил и построение маршрута логического вывода.* В МЛВ второго типа отсутствует этап предварительного построения маршрута логического вывода. Формирование маршрута осуществляется постфактум на основе совокупности правил, активированных в процессе решения задачи. В МЛВ первого типа, напротив, сначала формируется маршрут логического вывода, после чего осуществляется активация правил. Алгоритм МЛВ первого типа можно модифицировать таким образом, чтобы активация правила выполнялась непосредственно после его обнаружения. В этом случае после добавления готового к активации правила в множество route (алгоритм 4.2), соответствующее маршруту логического вывода, необходимо выполнить действие, ассоциированное с данным правилом. После этого зафиксировать результаты вычисления выходных параметров в множестве известных параметров known. Такая модификация также позволяет сделать необязательным требование на вычисление всех выходных параметров правила при его активации.

Следует отметить, что МЛВ второго типа ориентированы на реализацию логического вывода по сверхбольшим базам знаний. Данная возможность обеспечивается за счёт поддержки распределённых вычислений и организации работы МЛВ в составе кластерных вычислительных систем.

Третий тип МЛВ был предложен и подробно рассмотрен в ряде работ, в частности в [174]. Его практической реализацией является программный комплекс КЭСМИ Wi!Mi «Разуматор» версии 2.1 [175]. Блок-схема алгоритма работы МЛВ подробно рассмотрена в [133]. Кроме того, алгоритм работы МЛВ может быть представлен в виде псевдокода (алгоритм 4.3).

В отличие от МЛВ второго типа, МЛВ третьего типа не использует модель параллельной обработки правил по схеме «ведущий—ведомые». Вместо этого

применяется иной механизм поиска правил. В МЛВ фиксируются сведения об известных и искомым параметрах задачи, при этом перечень известных параметров помещается в специальное множество *queue* (алгоритм 4.3). Данное множество дополняется параметрами, значение которых вычисляется впервые. До начала процесса обработки миварной сети формируется структура связей между параметрами и правилами МБЗ, для которых данные параметры являются входными. На основе этой структуры осуществляется поиск правил по параметру, извлечённого из множества *queue*. Далее выполняется проверка возможности активации обнаруженного правила, и в случае её выполнения производится вычислительное действие для расчёта значений выходных параметров правила. Таким образом, построение маршрута логического вывода в МЛВ третьего типа осуществляется от известных параметров миварной сети (условие «Дано», прямой поиск).

Алгоритм 4.3 — Обобщённый алгоритм обработки миварной сети машиной логического вывода программного комплекса КЭСМИ Wi!Mi «Разуматор»

Вход: множества параметров P и правил R миварной сети, множества известных I и искомым O параметров задачи

Выход: маршрут логического вывода *route* и искомые значения параметров O

1. $route \leftarrow ()$ ▷ упорядоченное множество
2. $known \leftarrow$ известные параметры из множества P
3. $known \leftarrow I \cup (known \setminus I)$
4. $queue \leftarrow known$
5. *начало:*
6. **если** $O \subseteq known$ **то**
7. **перейти к концу**
8. **конец если**
9. **если** $queue = \emptyset$ **то**
10. сообщить о нехватке знаний для решения задачи
11. **перейти к концу**
12. **конец если**
13. $check \leftarrow false$
14. **для** каждого параметра i из множества *queue* **выполнить**
15. $queue \leftarrow queue \setminus \{i\}$ ▷ извлечение параметра из множества
16. $rules \leftarrow$ правила у которых параметр i входной

17. для каждого правила j из множества rules **выполнить**
18. input \leftarrow входные параметры правила j
19. **если** input \subseteq known **то**
20. output \leftarrow выходные параметры правила j
21. добавить j в конец множества route
22. выполнить действие правила j и зафиксировать результаты вычисления в output
23. queue \leftarrow queue \cup (output \setminus known)
24. known \leftarrow known \cup (output \setminus known)
25. check \leftarrow true
26. **конец если**
27. **конец для**
28. **если** check = true **то** ▷ если была активация правил
29. **перейти к началу**
30. **конец если**
31. **конец для**
32. **перейти к началу**
33. *конец;*
34. **вернуть** множества route и known

На основе псевдокода, представленного алгоритмом 4.3, можно отметить, что распараллеливание реализовать возможно только для обработки множества правил rules, обнаруженных при рассмотрении текущего известного параметра из очереди queue. Кроме того, в отличие от МЛВ первого типа, требование обязательного вычисления всех выходных параметров при выполнении вычислительного действия правила не используется в МЛВ третьего типа, поскольку построение маршрута логического вывода и активация правил выполняются синхронно. Однако, в реализации МЛВ КЭСМИ Wi!Mi «Разуматор» версии 2.1 данное требование применяется в принудительном порядке, аналогично реализации, используемой в программном комплексе «УДАВ».

В работе [176] представлена система «Миварная активная энциклопедия», для которой была разработана миварная МЛВ. МЛВ данной системы не использует специальных структур для поиска правил, в отличие от МЛВ третьего типа. Алгоритм работы МЛВ «Миварной активной энциклопедии» в обобщённом виде

похож на алгоритм МЛВ второго типа, однако в качестве «ведущего» и «ведомого» выступает один вычислительный узел. После каждого изменения множества известных параметров выполняется полный перебор правил миварной сети с целью проверки возможности их активации. В случае активации правило исключается из дальнейшего рассмотрения. Если в ходе очередного перебора не происходит активации ни одного правила или становятся известны значения всех искомым параметров, то процесс решения задачи завершается.

4.2.2 Постановка требований к миварной машине логического вывода для решения задач планирования в пространстве состояний

Первые результаты разработки системы МИПРА были представлены в работе [6]. На момент публикации данной работы в МИПРА использовался МЛВ из программного комплекса КЭСМИ Wi!Mi «Разуматор» версии 2.1. В последующих исследованиях [32, 50] были выявлены и обоснованы предпосылки, определившие направления развития системы МИПРА и применяемой в ней МЛВ. В частности, были сформулированы требования к МЛВ, предназначенной для решения задач в пространстве состояний:

1. МЛВ должна обеспечивать возможность автономной работы на аппаратной платформе робота или роботизированного средства.
2. МЛВ должна поддерживать параллельную обработку и активацию правил миварной сети и задействовать всё доступное число вычислительных потоков.
3. МЛВ должна поддерживать механизм опционального вычисления всех выходных параметров правила при его активации. Необходимость вычисления конкретных выходных параметров должна определяться действием правила.

Первое требование исключает возможность применения методов кластеризации на разных аппаратных вычислительных узлах, как это предлагается для МЛВ второго типа.

Второе требование определяется способом представления знаний о ПрО при

решении задач планирования в пространстве состояний. Данное представление опирается на выбранный метод решения задач и обработки информации, а именно на использование промежуточных целей при декомпозиции задачи планирования [50]. Далее рассмотрим пример миварной сети, формируемой с учётом данного метода.

В работе [6] исследуется решение задачи планирования системой МИПРА при наличии трёх доступных площадок для размещения башен из кубиков. Предположим, что в соответствии с целевым состоянием задачи в пространстве состояний кубики необходимо разместить таким образом, как показано на рисунке 3.6 в состоянии $S_2 = ((0, 1, 2), (), (4, 5, 6, 3))$. В этом случае фрагмент миварной сети, предназначенной для определения достижения промежуточных целей может быть схематично представлен в виде, показанном на рисунке 4.1. Схематичность отображения заключается в том, что для проверки достижения промежуточных целей используется больше параметров, связанных с описанием кубиков в пространстве.

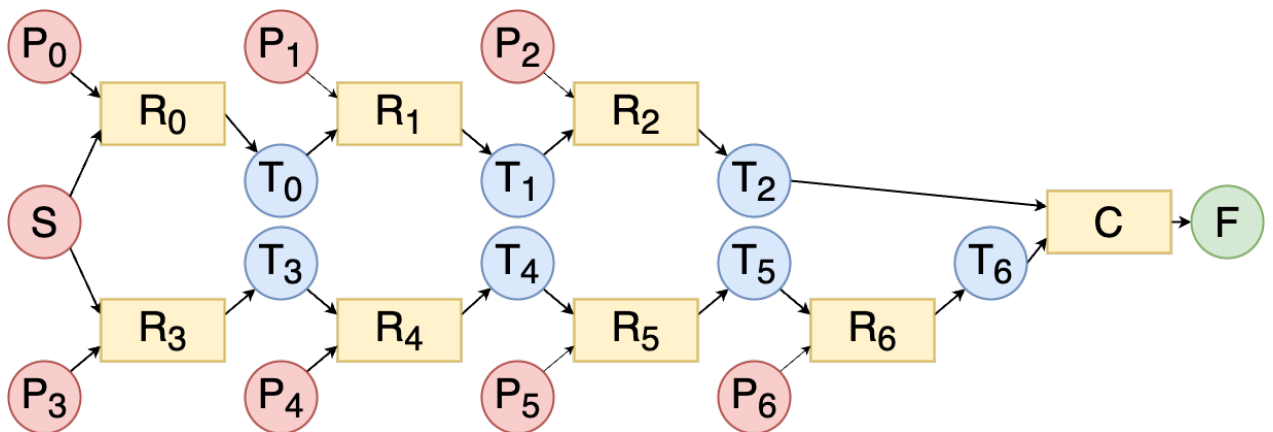


Рисунок 4.1 — Схематичное отображение фрагмента миварной сети, отвечающего за проверку достижимости промежуточных целей при решении задачи планирования в домене «Мир кубиков»

На рисунке 4.1 правила $R_0—R_6$ предназначены для проверки достижения промежуточных целей задачи планирования. Напомним, что в соответствии с алгоритмом решения задачи планирования, изложенным в параграфе 3.3.2, нумерация промежуточных целей соответствует фактическому порядку следования кубиков в упорядоченном множестве S_2 .

Параметры $T_0—T_6$ отражают состояние достижения промежуточных целей, то есть факт размещения кубиков в положениях, предписанных целевым состоянием задачи планирования. Параметры $P_0—P_6$ связаны с описанием пространственного положения кубиков. Правило R_0 отвечает за проверку достижения кубиком

№ 0 целевого положения на площадке № 0, а правило R_3 – за достижение кубиком № 4 целевого положения на площадке № 2.

Цепочка правил $R_0—R_2$ используется для проверки размещения кубиков на площадке № 0, тогда как цепочка правил $R_3—R_6$ – для проверки размещения кубиков на площадке № 2. Параметр S фиксирует факт запуска задачи планирования на вычисление. Правило C предназначено для проверки достижения всех промежуточных целей задачи планирования, а параметр F является индикатором достижения целевого состояния среды ПрО.

Вычисление логического вывода по миварной сети, представленной на рисунке 4.1, может быть частично выполнено параллельно в двух потоках, а именно за счёт разнесения вычислений по цепочкам правил $R_0—R_2$ и $R_3—R_6$. Однако МЛВ третьего типа обеспечивает параллельную активацию лишь для правил R_0 и R_3 . Далее, в соответствии с алгоритмом 4.3, вычисленные параметры T_0 и T_3 добавляются в множество *queue*, из которого они последовательно извлекаются для поиска правил, связанных с данными параметрами. В результате параллельная обработка и активация правил R_1 и R_4 не осуществляется.

Для реализации параллельных вычислений потенциально может быть использован второй тип МЛВ, однако наибольшую эффективность данный тип демонстрирует в условиях кластерных вычислений при обработке сверхбольших баз знаний. При этом требуется заранее определить такое распределение правил между ведомыми вычислительными узлами, при котором, например, цепочки правил $R_0—R_2$ и $R_3—R_6$ размещаются на различных узлах. Следовательно, для эффективного использования доступных вычислительных потоков требуется введение дополнительных механизмов загрузки правил в МЛВ ведомые узлы с учётом информации о составе каждой башни в целевом состоянии задачи планирования.

Третье требование по своей сути исключает использование МЛВ первого типа, а также МЛВ, реализованных в программном комплексе КЭСМИ Wi!Mi «Разуматор» версии 2.1. Вместе с тем формализация правила миварной сети, рассмотренная в параграфе 2.2, полностью удовлетворяет данному требованию.

Рассмотрим далее аспект, иллюстрирующий причину формулирования третьего требования. В работе [6] для вычисления части плана действий работа используются фрагменты миварной сети, схематическое представление которых приведено на рисунке 4.2. На данном рисунке параметры I_0 и I_1 связаны с описанием состояния кубиков. Параметр O при присвоении ему определённого значения

соответствует конкретному действию робота. Если рассматриваемый фрагмент миварной сети обрабатывается МЛВ первого типа, то при известности значений параметров I_0 и I_1 правила R_0 и R_1 подлежат активации, в результате чего значение параметра O должно быть установлено в обязательном порядке.

В работе [133] отмечается, что во избежание «зацикливания» логического вывода при присвоении значения параметру в процессе построения логического вывода данное значение в дальнейшем не изменяется. Поэтому если правило R_0 при текущем состоянии ПрО указывает на недопустимость выполнения действия, предписываемого параметром O , тогда как правило R_1 формирует противоположное заключение, возникает конфликт фиксации значения параметра O .

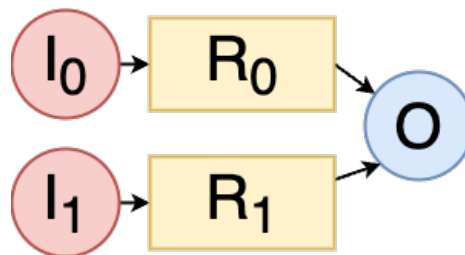


Рисунок 4.2 — Схематичное отображение фрагмента миварной сети, отвечающего за определение части плана действий при решении задачи планирования в домене «Мир кубиков»

Указанный конфликт может быть разрешён различными способами, однако в данном случае невозможно гарантировать корректность формируемого управляющего воздействия, обеспечивающего перевод среды ПрО в иное состояние. Для его устранения требуется явное задание порядка активации правил R_0 и R_1 . В противном случае результат активации правила R_0 может зафиксировать значение параметра O , что приведёт к невозможности его изменения при последующей активации правила R_1 . В результате роботу может быть передано некорректное действие на исполнение.

В ранних исследованиях, посвящённых планированию в пространстве состояний [177], для обхода данного ограничения параметры I_0 и I_1 связывались с промежуточными целями. Поскольку в процессе планирования однозначно определялась актуальная на текущем этапе промежуточная цель, в МЛВ не передавались параметры, не относящиеся к формированию действий, необходимых для её достижения. Таким образом, необходимость выполнения конкретного действия определялась текущей промежуточной целью. Реализация данного подхода потребовала разработки дополнительных механизмов учёта текущей промежуточ-

ной цели в системе МИПРА, а также проведения дополнительного анализа ПрО с целью адаптации метода к решению определённого класса задач.

Следует отметить, что введение указанных механизмов не требуется при отсутствии ограничения на обязательность вычисления значений выходных параметров при активации правил. В этом случае в МЛВ могут передаваться все параметры, описывающие текущее состояние ПрО, без предварительного отбора параметров, относящихся исключительно к достижению текущей промежуточной цели.

На основе результатов анализа требований к МЛВ для задач планирования в пространстве состояний разработан программный комплекс “Balabza.Razumator” версии 5.1 [20]. Разработанную МЛВ можно отнести к *четвёртому типу*.

4.2.3 Обобщённый алгоритм обработки миварной сети машиной логического вывода для решения задач планирования в пространстве состояний

Обобщённый процесс обработки миварной сети в МЛВ программного комплекса “Balabza.Razumator” версии 5.1 представлен в виде псевдокода алгоритма 4.4, а также в виде блок-схемы на рисунке 4.3.

Алгоритм 4.4 — Обобщённый алгоритм обработки миварной сети машиной логического вывода программного комплекса “Balabza.Razumator”

Вход: множества параметров P и правил R миварной сети, множества известных I и искомых O параметров задачи

Выход: маршрут логического вывода $route$ и искомые значения параметров O

1. $route \leftarrow ()$ ▷ упорядоченное множество
2. $known \leftarrow$ известные параметры из множества P
3. $known \leftarrow I \cup (known \setminus I)$
4. $notProcessed \leftarrow P$
5. $set1 \leftarrow known$

6. $set2 \leftarrow ()$ ▷ неупорядоченное множество
7. *начало:*
8. **если** $set1 = \emptyset$ **то**
9. сообщить о нехватке знаний для решения задачи
10. **перейти к концу**
11. **конец если**
12. **для** каждого параметра i из множества $set1$ **выполнить**
13. **если** $i \in notProcessed$ **то**
14. $notProcessed \leftarrow notProcessed \setminus \{i\}$
15. $known \leftarrow known \cup (\{i\} \setminus known)$
16. $rules \leftarrow$ правила у которых параметр i входной
17. **для** каждого правила j из множества $rules$ **выполнить**
18. $input \leftarrow$ входные параметры правила j
19. $output \leftarrow$ выходные параметры правила j
20. $check1 \leftarrow false$
21. **если** $input \subseteq known$ **то**
22. $check1 \leftarrow true$ ▷ для активации правила всё известно
23. **конец если**
24. $check2 \leftarrow false$
25. **если** $j \notin route$ **то**
26. $check2 \leftarrow true$ ▷ правило не было активировано
27. **конец если**
28. $check3 \leftarrow false$
29. **если** $output \setminus known \neq \emptyset$ **то**
30. $check3 \leftarrow true$ ▷ не все значения выходных параметров известны
31. **конец если**
32. **если** $check1 \wedge check2 \wedge check3$ **то**
33. $set2 \leftarrow set2 \cup \{j\}$ ▷ правило можно активировать
34. **конец если**
35. **конец для**
36. **конец если**
37. **конец для**
38. **если** $O \subseteq known$ **то**

39. **перейти к концу**
40. **конец если**
41. $set1 \leftarrow ()$
42. **для** каждого правила j из множества $set2$ **выполнить**
43. $output \leftarrow$ выходные параметры правила j
44. добавить j в конец множества $route$
45. выполнить действие правила j и зафиксировать результаты вычисления в $output$
46. $set1 \leftarrow set1 \cup output$
47. **конец для**
48. $set2 \leftarrow ()$
49. **перейти к началу**
50. *конец:*
51. **вернуть** множества $route$ и $known$

В псевдокоде алгоритма 4.4 множество $set1$ представляет собой совокупность известных и вычисленных параметров миварной сети, а множество $set2$ – совокупность правил, готовых к активации. Существенным отличием МЛВ четвёртого типа от МЛВ третьего типа является разделение процессов обработки множества известных и вычисленных параметров и активации правил. Кроме того, в отличие от МЛВ первого типа, МЛВ четвёртого типа осуществляет прямой поиск правил по миварной сети, при котором правила отбираются исходя из текущего состояния множества известных и вычисленных параметров.

Для работы МЛВ четвёртого типа, аналогично МЛВ третьего типа, перед началом обработки миварной сети требуется подготовить информацию о входных параметрах правил. При анализе множества $set1$ для каждого известного параметра миварной сети формируется множество правил $rules$, для которых данный параметр является входным.

Для каждого правила из множества $rules$ выполняются три проверки, фиксируемые в виде логических флагов $check1$, $check2$ и $check3$. Флаг $check1$ принимает значение ИСТИНА, если известны значения всех входных параметров правила. Флаг $check2$ принимает значение ИСТИНА, если правило ранее не было активировано. Флаг $check3$ принимает значение ИСТИНА, если не все выходные параметры правила были вычислены.

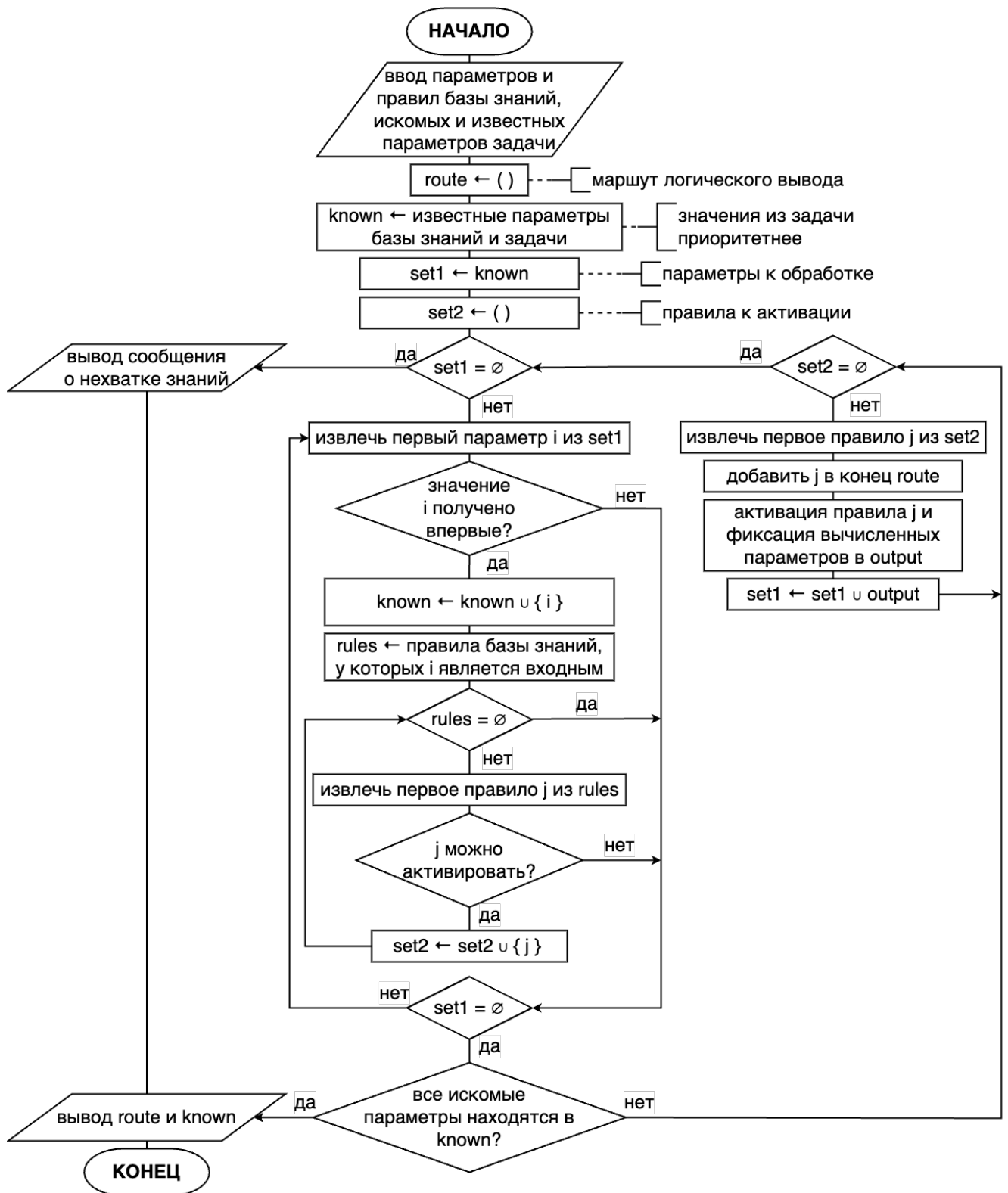


Рисунок 4.3 — Блок-схема обобщённого алгоритма обработки миварной сети машиной логического вывода программного комплекса “Balabza.Razumator”

Демонстрация проверки check2 в псевдокоде является избыточной с точки зрения алгоритмической реализации, однако она необходима для последующей интерпретации алгоритма работы МЛВ в виде математической обработки матриц [64], которая будет подробно рассмотрена далее. Проверка check3 может быть

исключена в случае, если действие правила представляет собой выполнение программы, результат которой находится за пределами МЛВ.

Процесс поиска и отбора правил, готовых к активации, может зависеть от конкретной реализации МЛВ. В частности, для ускорения данного процесса могут использоваться специализированные индексы и связанные списки. Вместе с тем формирование подобных структур требует дополнительной обработки МБЗ при её загрузке в память МЛВ, а также может приводить к увеличению объёма используемой оперативной памяти в процессе подготовки логического вывода.

Выделение этапа активации правил в отдельный этап обработки миварной сети позволяет реализовать их параллельную активацию. При этом количество параллельно обрабатываемых правил может соответствовать числу вычислительных потоков, доступных на аппаратной платформе. Подобная организация обработки близка к схеме «ведущий—ведомые», используемой в МЛВ второго типа. Однако в рассматриваемом варианте построение логического вывода осуществляется в рамках одной аппаратной платформы, что соответствует первому требованию к МЛВ, сформулированному в параграфе 4.2.2. Распределение активации правил из множества $set2$ по всем доступным вычислительным потокам также удовлетворяет второму требованию.

Третье требование к МЛВ обеспечивается за счёт использования формализма, рассмотренного в параграфе 2.2, а также за счёт построения маршрута логического вывода на основе фактической активации правил.

Таким образом, программный комплекс “Balabza.Razumator” версии 5.1 удовлетворяет всем требованиям к МЛВ для решения задач планирования в пространстве состояний. Опыт эксплуатации данного комплекса лёг в основу разработки платформы ЛИСА [23, 24]. Алгоритм обработки миварной сети МЛВ, реализованный в данной платформе, полностью соответствует четвёртому типу.

Далее приводится интерпретация алгоритма обработки миварной сети МЛВ четвёртого типа в виде матричных преобразований [64].

4.2.4 Матричное представление алгоритма обработки миварной сети машиной логического вывода для решения задач планирования в пространстве состояний

Разбор задачи построения логического вывода

Пусть количество параметров базы знаний равно N , а количество правил – M . Каждый параметр базы знаний имеет уникальный числовой идентификатор. Обозначим упорядоченное неповторяющееся множество идентификаторов параметров как $ParametersIDs = \{1, 2, \dots, N\}$. Аналогично, множество идентификаторов правил обозначим как $RulesIDs = \{1, 2, \dots, M\}$.

Сформулируем задачу построения логического вывода T следующим образом: даны МБЗ и множество $Input$ известных значений параметров, требуется найти значения параметров из множества $Output$.

Определим матрицу $RulesInputs$ размерности $(M \times N)$, где M – количество правил, а N – количество параметров базы знаний. Строки матрицы соответствуют правилам базы знаний, а столбцы – параметрам базы знаний. В процессе построения логического вывода данная матрица остаётся неизменной. Значения элементов матрицы отражают факт принадлежности параметра к входному множеству соответствующего правила.

Матрица $RulesInputs$ имеет следующий вид:

$$RulesInputs_{(M \times N)} = (x_{ij}) = \begin{pmatrix} x_{11} & x_{12} & x_{13} & \dots & x_{1j} & \dots & x_{1N} \\ x_{21} & x_{22} & x_{23} & \dots & x_{2j} & \dots & x_{2N} \\ x_{31} & x_{32} & x_{33} & \dots & x_{3j} & \dots & x_{3N} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ x_{M1} & x_{M2} & x_{M3} & \dots & x_{Mj} & \dots & x_{MN} \end{pmatrix}. \quad (4.1)$$

Номер строки i ($i \in \{1, \dots, M\}$) соответствует порядковому номеру правила, а номер столбца j ($j \in \{1, \dots, N\}$) – порядковому номеру параметра. Если параметр под номером j является входным по отношению к правилу с номером i , то в ячейку матрицы $RulesInputs$ со строкой i и столбцом j заносится значение 1, в противном случае значение 0.

Аналогичным образом формируется матрица $RulesOutputs$, элементы которой отражают принадлежность параметров к выходному множеству соответствующих правил:

$$RulesOutputs_{(M \times N)} = (y_{ij}) = \begin{pmatrix} y_{11} & y_{12} & y_{13} & \dots & y_{1j} & \dots & y_{1N} \\ y_{21} & y_{22} & y_{23} & \dots & y_{2j} & \dots & y_{2N} \\ y_{31} & y_{32} & y_{33} & \dots & y_{3j} & \dots & y_{3N} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ y_{M1} & y_{M2} & y_{M3} & \dots & y_{Mj} & \dots & y_{MN} \end{pmatrix}. \quad (4.2)$$

Также рассчитываются матрицы $NumRulesInputs$ и $NumRulesOutputs$ размерности $(M \times 1)$, строки которых соответствуют правилам, а значения элементов – количеству входных и выходных параметров соответственно.

Матрица $NumRulesInputs$ имеет следующий вид:

$$NumRulesInputs_{(M \times 1)} = (a_{i1}) = \begin{pmatrix} a_{11} & a_{21} & a_{31} & \dots & a_{M1} \end{pmatrix}^T. \quad (4.3)$$

Матрица $NumRulesOutputs$:

$$NumRulesOutputs_{(M \times 1)} = (b_{i1}) = \begin{pmatrix} b_{11} & b_{21} & b_{31} & \dots & b_{M1} \end{pmatrix}^T. \quad (4.4)$$

Пример матрицы правил $RulesInputs_{(4 \times 5)}$, составленной из 5 параметров и 4 правил, представлен ниже:

$$RulesInputs_{(4 \times 5)} = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{pmatrix}. \quad (4.5)$$

Соответствующая матрица $NumRulesInputs_{(4 \times 1)}$:

$$NumRulesInputs_{(4 \times 1)} = \begin{pmatrix} 3 & 1 & 2 & 2 \end{pmatrix}^T. \quad (4.6)$$

$Found_{(1 \times N)} = (c_{1j})$ – матрица известных значений параметров, размерность

которой $(1 \times N)$:

$$Found_{(1 \times N)} = (c_{1j}) = \begin{pmatrix} c_{11} & c_{12} & c_{13} & \dots & c_{1N} \end{pmatrix}. \quad (4.7)$$

Элементы матрицы $Found$ формируются на основе множества $Input$ задачи T , а также известных констант МБЗ. Номер столбца j соответствует порядковому номеру параметра. Значения ячеек матрицы $Found$ соответствуют типам данных параметров, поддерживаемым МЛВ, и могут быть пустыми.

$FlagsFound_{(N \times 1)} = (g_{j1})$ – матрица признаков известности значений параметров, размерность которой $(N \times 1)$:

$$FlagsFound_{(N \times 1)} = (g_{j1}) = \begin{pmatrix} g_{11} & g_{21} & g_{31} & \dots & g_{N1} \end{pmatrix}^T. \quad (4.8)$$

Элементы матрицы $FlagsFound$ отражают факт того, известно ли значение соответствующего параметра. Номер строки j соответствует порядковому номеру параметра. Элемент матрицы $FlagsFound$ принимает значение 1, если значение параметра с номером j известно, в противном случае ему присваивается значение 0.

$FlagsFind_{(N \times 1)} = (s_{j1})$ – матрица признаков необходимости вычисления значений параметров, размерность которой $(N \times 1)$:

$$FlagsFind_{(N \times 1)} = (s_{j1}) = \begin{pmatrix} s_{11} & s_{21} & s_{31} & \dots & s_{N1} \end{pmatrix}^T. \quad (4.9)$$

Элементы матрицы $FlagsFind$ формируются на основе множества $Output$ задачи T . Номер строки j соответствует порядковому номеру параметра. Элемент матрицы $FlagsFind$ принимает значение 1, если значение параметра с номером j требуется найти, в противном случае ему присваивается значение 0.

Отметим, что значения элементов матриц $Found$ и $FlagsFound$ могут изменяться в процессе построения логического вывода, тогда как матрица $FlagsFind$ остаётся неизменной на всём протяжении данного процесса.

При разборе задачи T также задаётся константа $NumFind$ – количество элементов в множестве $Output$.

$FlagsActivatedRules_{(M \times 1)} = (e_{i1})$ – матрица признаков активации правил,

размерность которой $(M \times 1)$:

$$FlagsActivatedRules_{(M \times 1)} = (e_{i1}) = \left(e_{11} \ e_{21} \ e_{31} \ \dots \ e_{M1} \right)^T. \quad (4.10)$$

Номер строки i матрицы $FlagsActivatedRules$ соответствует порядковому номеру правила. Элемент матрицы $FlagsActivatedRules$ принимает значение 1, если соответствующее правило было активировано, и значение 0 – в противном случае.

Определим матрицу $WhoSetValue$ размерности $(1 \times N)$, номер столбца которой соответствует порядковому номеру параметра. В ячейку данной матрицы заносится порядковый номер правила, в результате активации которого было вычислено значение соответствующего параметра. Если значение параметра не было вычислено и не было известно на момент постановки задачи T , то соответствующий элемент матрицы принимает значение 0.

Введём счётчик $Stage$, определяющий номер этапа построения логического вывода. Начальное значение счётчика устанавливается равным 1.

Определим матрицу $WhenSetValue$ размерности $(1 \times N)$, номер столбца которой соответствует порядковому номеру параметра. В элемент данной матрицы заносится значение счётчика $Stage$, соответствующее этапу, на котором было вычислено значение рассматриваемого параметра. Если значение параметра не было вычислено и не было известно на момент постановки задачи T , то соответствующий элемент матрицы принимает значение 0.

Для формирования отчёта о логическом выводе используется стек $Route$, в который заносятся порядковые номера правил в последовательности их активации.

Поиск правил, готовых к активации

Поиск правил, готовых к активации при текущем состоянии множества известных значений параметров, осуществляется поэтапно, при этом номер текущего этапа фиксируется счётчиком $Stage$.

В начале каждого этапа вычисляется матрица $NeedActivationX$ размерно-

сти $(M \times 1)$:

$$\begin{aligned} NeedActivationX_{(M \times 1)} = RulesInputs_{(M \times N)} \times FlagsFound_{(N \times 1)} - \\ - NumRulesInputs_{(M \times 1)} + FlagsActivatedRules_{(M \times 1)}. \end{aligned} \quad (4.11)$$

Строки матрицы $NeedActivationX$ соответствуют правилам базы знаний. В результате умножения матрицы $RulesInputs$ на матрицу $FlagsFound$ получается информация о количестве параметров с известными значениями во входном множестве каждого правила. Вычитание матрицы $NumRulesInputs$ позволяет определить, достаточно ли известных значений для активации соответствующего правила, то есть известны ли все входные параметры данного правила. Далее, с учётом сложения с матрицей $FlagsActivatedRules$, значения элементов матрицы $NeedActivationX$ интерпретируются следующим образом:

- 1 – правило было активировано на предыдущем этапе;
- 0 – правило можно активировать на текущем этапе;
- отрицательное значение – для активации рассматриваемого правила не все значения входных параметров известны.

Также вычисляется матрица $NeedActivationY$ размерности $(M \times 1)$:

$$\begin{aligned} NeedActivationY_{(M \times 1)} = RulesOutputs_{(M \times N)} \times FlagsFound_{(N \times 1)} - \\ - NumRulesOutputs_{(M \times 1)}. \end{aligned} \quad (4.12)$$

Строки матрицы $NeedActivationY$ соответствуют правилам базы знаний. В результате умножения матрицы $RulesOutputs$ на матрицу $FlagsFound$ получается информация о количестве параметров с известными значениями в выходном множестве каждого правила. После вычитания матрицы $NumRulesOutputs$ значения элементов матрицы $NeedActivationY$ интерпретируются следующим образом:

- 0 – активация правила не требуется, поскольку значения всех его выходных параметров уже известны;
- отрицательное значение – действие правила необходимо выполнить, так как могут быть получены значения некоторых выходных параметров правила.

Затем формируется временная матрица флагов $FlagsFoundOnStage$ размерности $(N \times 1)$, все элементы которой инициализируются значением 0. Строки матрицы соответствуют параметрам базы знаний. Если на текущем этапе зна-

чение параметра было получено впервые, то соответствующий элемент матрицы *FlagsFoundOnStage* устанавливается в значение 1. По завершении этапа построения логического вывода матрица *FlagsFoundOnStage* обнуляется.

Активация правил

По строкам матриц *NeedActivationX* и *NeedActivationY* выполняется синхронный построчный обход со следующей проверкой:

- если в строке i матрицы *NeedActivationX* содержится значение 0, а в соответствующей строке матрицы *NeedActivationY* – значение отрицательное, то выполняется действие правила, и правило считается активированным;
- если в строке i матрицы *NeedActivationX* содержится значение 0, а в соответствующей строке матрицы *NeedActivationY* – значение 0, то правило считается активированным, при этом выполнение его действия является опциональным и зависит от настроек МЛВ.

В случае активации правила в матрице *FlagsActivatedRules* в строке, соответствующей данному правилу, устанавливается значение 1, а его порядковый номер добавляется в стек *Route*.

Вычисленные в результате выполнения действия правила значения параметров помещаются в множество *Calculated*, элементы которого представляют собой кортежи вида $\langle i, Value, WhoSet \rangle$, где: i – порядковый номер параметра; $Value$ – его вычисленное значение; $WhoSet$ – порядковый номер правила, в результате активации которого было вычислено значение. Множество *Calculated* очищается по завершению этапа построения логического вывода.

Фиксация вычисленных значений параметров

Для каждого вычисленного значения параметра из множества *Calculated* выполняется следующая последовательность действий.

Сначала по порядковому номеру параметра в строке матрицы *FlagsFound* проверяется значение соответствующего флага:

- Если 1, то рассматриваемый параметр пропускается и осуществляется переход к обработке следующего вычисленного параметра. То же самое справедливо в случае, если значение флага равно 0, а значение параметра является пустым.

- Если 0 и значение параметра является непустым, то происходит проверка матрицы $FlagsFoundOnStage$.

Проверка матрицы $FlagsFoundOnStage$ по порядковому номеру параметра в строке осуществляется следующим образом:

1. Если значение флага 0, то оно меняется на 1. Вычисленное значение параметра фиксируется в матрице $Found$, при этом в матрице $WhoSetValue$ сохраняется номер правила, установившего значение параметра, а в матрице $WhenSetValue$ – номер текущего этапа, определяемый счётчиком $Stage$.
2. Если значение флага 1, то возможны два варианта разрешения конфликта: **приоритет первой активации**, при котором фиксация вычисленного значения параметра не производится; **приоритет последней активации**, при котором значение флага 1 игнорируется и выполняются все действия, предусмотренные в пункте 1. Вариант разрешения конфликта зависит от настройки МЛВ.

Проверка условий достижения целей задачи

После обработки всех элементов из множества $Calculated$ происходит обновление матрицы $FlagsFound$:

$$FlagsFound_{(N \times 1)} = FlagsFound_{(N \times 1)} + FlagsFoundOnStage_{(N \times 1)}. \quad (4.13)$$

Далее осуществляется проверка условий достижения целей задачи T . Проверка представляет собой следующее условное выражение:

$$FlagsFind_{(N \times 1)}^T \times FlagsFound_{(N \times 1)} == NumFind. \quad (4.14)$$

Если условное выражение принимает значение ИСТИНА, то процесс построения логического вывода завершается и выполняется формирование отчёта. В противном случае решение задачи продолжается, и производится вычисление значения элемента $Next$. Для этого матрица $FlagsFoundOnStage_{(N \times 1)}$ перемножается с единичной матрицей $1(1 \times N)$:

$$Next = 1_{(1 \times N)} \times FlagsFoundOnStage_{(N \times 1)}. \quad (4.15)$$

Если значение $Next$ отлично от нуля, то текущий этап считается завершённым.

ным, а счётчик *Stage* инкриминируется на 1. Новый этап начинается с повторного поиска правил, готовых к активации. В противном случае процесс подготовки логического вывода прекращается с ошибкой, обусловленной недостаточностью информации в разделе «дано» (отсутствием необходимых знаний или значений параметров).

Подготовка отчёта о решённой задаче

Если по результатам решения задачи T требуется предоставить только значения параметров из множества *Output*, то соответствующие значения извлекаются из матрицы *Found*.

В случае необходимости формирования полного логического вывода с представлением всех утверждений, для подготовки отчёта используются следующие структуры данных:

- матрица *Found* – для отображения всех значений параметров, которые были заданы или вычислены в процессе решения задачи T ;
- стек *Route* – для демонстрации последовательности правил, на основе которых был получен результат решения задачи;
- матрица *WhoSetValue* – для указания правил, в результате активации которых были установлены значения параметров;
- матрица *WhenSetValue* – для отображения этапов, на которых были вычислены значения соответствующих параметров;
- матрицы *RulesInputs* и *RulesOutputs* – для представления связей между правилами и параметрами миварной сети.

С примером применения разработанного алгоритма обработки миварной сети машиной логического вывода при решении задач планирования в пространстве состояний можно ознакомиться в приложении Б.

С результатами реализации матричной интерпретации в виде программного кода, использующего графический процессор, можно ознакомиться в публикации [64].

4.3 Разработка комплекса программ на основе моделей и методов интеллектуального планирования действий в пространстве состояний

4.3.1 Логическая интеллектуальная система контроля за соблюдением правил дорожного движения

Одной из первых прикладных систем, реализовавших модели МБЗ и миварной проблемно-ориентированной системы управления и принятия решений, стала ЛИСК ПДД [4, 28, 30, 31, 45]. Система предназначена для формирования экспертного заключения о действиях, совершённых водителем транспортного средства: на основе базы знаний Про ПДД и текущей дорожной ситуации строится логический вывод, формируется рекомендуемый план действий, после чего он сопоставляется с фактически выполненным манёвром. Итогом работы системы является оценка согласованности действий водителя с требованиями ПДД.

Первоначально архитектура ЛИСК ПДД включала систему верификации миварных моделей (СВММ) [20, 112] и систему анализа дорожной ситуации, а в качестве средства интеллектуализации планировалось использовать КЭСМИ Wi!Mi «Разуматор» версии 2.1. В ходе разработки была выявлена необходимость совершенствования алгоритма обработки миварных сетей. В результате состав системы был уточнён [45]: ключевым компонентом интеллектуализации стал программный комплекс «Balabza.Razumator», дополненный СВММ и редактором МБЗ из КЭСМИ Wi!Mi «Разуматор» версии 2.1.

СВММ предназначена для проверки корректности и непротиворечивости формализованных знаний в виде миварных сетей. Верификация осуществляется путём сопоставления контрольных значений параметров с результатами решения задач построения логического вывода.

Для предметной области ПДД каждая МБЗ содержит входные параметры, описывающие дорожную ситуацию и планируемые действия водителя (в том числе «задание навигатора»), и выходные параметры, определяющие разрешённые действия и рекомендации. Важным проектным решением стала группировка парамет-

ров и правил миварной сети по подзадачам (определение направления движения, сигналов светофора, наличия помех и т.д.). Такая декомпозиция упрощает как разработку и обновление базы знаний несколькими экспертами, так и последующую интеграцию с внешними системами.

Следует подчеркнуть, что область применения ЛИСК ПДД не ограничивается задачами классификации и выявления нарушений. На её основе может быть реализован процесс принятия решений в дорожной ситуации для систем помощи водителю (англ. Advanced Driver-Assistance System, ADAS – усовершенствованная система помощи водителю): по входным фактам система синтезирует алгоритм выполнения манёвра (план действий) и выдаёт его в качестве рекомендации, а затем оценивает соответствие фактических действий полученному плану. Проектирование такого решения возможно благодаря использованию модели миварной проблемно-ориентированной системы управления и принятия решений.

ЛИСК ПДД также может интегрироваться с телематическими системами автомобиля и системами анализа дорожно-транспортных происшествий [178]. При этом предполагается, что на текущем этапе развития данные о состоянии ПрО поступают из внешних систем, распознающих дорожную ситуацию и выполненный манёвр, и передаются в ЛИСК ПДД в виде значений входных параметров. Компонент информационного сопровождения водителя также рассматривается как внешний по отношению к ЛИСК ПДД, что подчёркивает модульность архитектуры и возможность использования системы в качестве интеллектуального агента в составе более широкой киберфизической платформы или РТК.

4.3.2 Миварная система планирования действий роботов и роботизированных средств

Для исследования предложенных моделей и методов интеллектуального планирования действий в пространстве состояний была разработана система МИПРА [6, 8]. МИПРА представляет собой планировщик действий по перестановке кубиков в пространстве и предназначена для взаимодействия с роботами как в реальных

условиях, так и в виртуальной среде. В реальной среде план формируется и уточняется по мере выполнения действий роботом: состояние среды изменяется в ходе планирования и последовательно приближается к целевому. В виртуальной среде сначала строится полный план действий, после чего он передаётся роботу на исполнение.

Создание МИПРА позволило экспериментально оценить производительность систем, основанных на предложенных моделях и методах интеллектуального планирования действий. Результаты экспериментальной проверки моделей и методов представлены далее.

4.3.3 Логическая интеллектуальная система обеспечения ухода за растениями

Система ЛИС ОУР [17, 44] является прикладной реализацией модели МБЗ и миварной проблемно-ориентированной системы управления и принятия решений. ЛИС ОУР осуществляет мониторинг и корректировку микроклимата в теплице на основе данных, поступающих с различных датчиков. МБЗ системы подготавливается когнитологами и экспертами ПрО на основе технологических карт роста растений и с учётом технического оснащения теплицы. Каждая подзадача описывается в виде подграфа миварной сети и соответствует управлению конкретным оборудованием теплицы или отдельной фазе роста растений.

ЛИС ОУР относится к системам с цикло-календарным планированием и управлением [44]. Структура и периодичность цикла определяются технологическими картами и соотносятся с фазами дня и ночи, а также с этапом роста растений. Цикл управления выполняется с заданной периодичностью, при изменении данных, поступающих от систем мониторинга, либо по сигналу таймера, настроенного в соответствии с технологической картой роста растения.

4.3.4 Программный комплекс на основе информационно-аналитической системы с поддержкой принятия решений об эффективности и безопасности применения термолабильных компонентов крови в медицинской практике

Модели МБЗ и миварной проблемно-ориентированной системы управления применены при разработке ПКИАС [21, 22]. ПКИАС предназначен для контроля качества и оценки безопасности использования термолабильных компонентов крови. Программный комплекс реализован в виде информационно-аналитической системы поддержки принятия решений: пользователь (медицинский специалист) задаёт характеристики компонента крови, после чего система формирует решение о возможности и безопасности его использования, а также предоставляет развернутый протокол обоснования принятого решения.

Принятие решений в ПКИАС основывается на МБЗ: экспертные знания о ПрО (показатели качества и безопасности, допустимые диапазоны, логические зависимости) формализованы и декомпозированы в соответствии с решаемыми задачами оценки пригодности конкретного компонента крови. Такая декомпозиция обеспечивает масштабируемость и гибкость ПКИАС, поскольку добавление новых фрагментов для решения дополнительных задач осуществляется без изменения ранее формализованных знаний.

В состав ПКИАС [21, 22] входят подсистемы логического вывода и объяснений, осуществляющие обработку локального репозитория баз знаний и решение задач на её основе. Для интеграции с внешними системами предусмотрены программные и графические интерфейсы, что позволяет разрабатывать прикладные решения на базе ПКИАС с использованием модели миварной проблемно-ориентированной системы управления и принятия решений.

4.4 Экспериментальная проверка моделей и методов интеллектуального планирования действий в пространстве состояний

4.4.1 Условия эксперимента

Для оценки работоспособности моделей и методов, предлагаемых в рамках данного диссертационного исследования, была разработана библиотека “MIPRA.Cubes” версии 5.0, входящая в состав системы МИПРА. Библиотека включает в себя эмулятор среды ПрО и планировщик, предназначенный для решения задач в рамках домена «Мир кубиков». В качестве среды для разработки библиотеки использовалась платформа .NET 6. Разработка системы осуществлялась на языке программирования C#. В качестве МЛВ использовался программный комплекс “Balabza.Razumator” версии 5.1 [20].

Одной из целей проведённого эксперимента являлось получение показателей, характеризующих скорость и вычислительную сложность процесса планирования при использовании МТ ЛИИ. Тестирование МИПРА выполнялось на стенде, представляющем собой персональный компьютер разработчика, характеристики которого представлены в таблице 4.1.

Для получения представления о зависимости производительности МИПРА от характеристик аппаратной платформы было принято решение провести тестирование на трёх различных вычислительных стендах. В качестве первого тестового стенда использовался персональный компьютер, технические характеристики которого приведены в таблице 4.1. Второй тестовый стенд был реализован на базе одноплатного микрокомпьютера Raspberry Pi 4 Model B (4 GB). Его характеристики представлены в таблице 4.2. Третий тестовый стенд представлял собой ноутбук Apple MacBook Pro (14-дюймовый, 2021 года выпуска). Характеристики данного стенда приведены в таблице 4.3.

Тестирование МИПРА проводилось на задачах планирования, в которых количество кубиков в среде варьировалось от 5 до 1000. Шаг изменения числа кубиков составлял: от 10 до 50 – 5 единиц (при этом добавление одной единицы

соответствовало добавлению одного кубика в среду), от 50 до 150 – 10 единиц, и свыше 150 – 25 единиц. Для каждого выбранного значения числа кубиков было сгенерировано по 10 различных задач планирования. В результате на каждом тестовом стенде было решено 530 задач.

Для проведения эксперимента использовались три площадки на столе, что позволяло продемонстрировать работоспособность планировщика при минимально допустимом количестве площадок для размещения башен. Для генерации описаний начального и целевого состояний задачи планирования применялся следующий алгоритм. На первом этапе формировалось пустое упорядоченное множество (выражение (3.1)), состоящее из M (M – число площадок) пустых упорядоченных подмножеств. Затем последовательно рассматривались целые числа от 0 до $N - 1$, соответствующие номерам кубиков. Каждое выбранное число случайным образом добавлялось в конец одного из подмножеств. В результате все числа распределялись между M подмножествами, которые задавали порядок расположения кубиков в башнях.

Для оценки зависимости производительности МИПРА от числа доступных вычислительных потоков на стендах № 1 и № 2 были сгенерированы задачи планирования с 60 площадками при сохранении ранее описанной схемы варьирования количества кубиков в среде. Для стенда № 1 выполнены измерения при 1 и 12 доступных вычислительных потоках, для стенда № 2 – при 1 и 4 потоках.

Таблица 4.1 — Стенд № 1 для эксперимента

Компонент	Описание
Центральный процессор	AMD Ryzen 5 9600X, 5445 MHz
Материнская плата	Gigabyte B850M Gaming X WiFi6E
Оперативная память	ADATA XPG LANCER Blade RGB 32GB (2x16Gb) (AX5U6400C3216G-DTLABRBK) (dual-channel mode)
Жёсткий диск	ADATA LEGEND 960 (1907 GB) (SLEG-900-2TCS) (PCI-E 4.0 x4)
Операционная система	Windows 11 Pro x64 version 26200.7623

Таблица 4.2 — Стенд № 2 для эксперимента

Компонент	Описание
Центральный процессор	Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC 1.5GHz
Оперативная память	4GB LPDDR4-3200 SDRAM
Жёсткий диск	Samsung MB-MC32GA (32 GB)
Операционная система	Raspberry Pi OS (64-bit) Lite (Release date: 4 Dec 2025, Kernel version: 6.12, Debian version: 13)

Таблица 4.3 — Стенд № 3 для эксперимента

Компонент	Описание
Центральный процессор	Apple M1 Pro with 10-core CPU and 14-core GPU
Оперативная память	32 GB
Жёсткий диск	512 GB SSD
Операционная система	MacOS Tahoe 26.2 (25C56)

4.4.2 Численные результаты эксперимента

В данном параграфе представлена часть показателей производительности МИП-РА, полученных в ходе экспериментов на тестовом стенде № 1 (таблица 4.1):

1. **Общее время решения задачи планирования** (рисунок 4.4).
2. **Общее количество перестановок кубиков** (рисунок 4.5) – суммарное число действий вида $\text{MoveCube}(i, j)$, выполненных роботом при реализации плана. В случае присутствия в плане действия $\text{MoveCubes}(i, j, k)$ оно учитывается как k повторений действия $\text{MoveCube}(i, j)$.

3. **Время генерации базы знаний** (рисунок 4.6) – время, затраченное на генерацию МБЗ в соответствии с условиями задачи планирования.
4. На рисунке 4.7 представлена зависимость **количества параметров**, включённых в МБЗ, от числа кубиков.
5. На рисунке 4.8 представлена зависимость **количества правил**, включённых в МБЗ, от числа кубиков.
6. **Общее время работы МЛВ** (рисунок 4.9) – суммарное время, затраченное МЛВ на построение логических выводов при подготовке промежуточных планов.
7. **Количество запросов к МЛВ** (рисунок 4.10) – число обращений к МЛВ для построения логического вывода в процессе решения задачи планирования.

Для демонстрации зависимости производительности МИПРА от характеристик аппаратной платформы были собраны метрики **общего времени решения задачи планирования** на различных тестовых стендах (рисунок 4.11). Для повышения наглядности на рисунке 4.11 ось ординат ограничена значением 75, а результаты эксперимента для стенда № 3 представлены только для задач с количеством кубиков не более 500.

Для демонстрации зависимости производительности МИПРА от числа доступных вычислительных потоков были получены метрики **общего времени решения задачи планирования** на стендах № 1 (рисунок 4.12) и № 2 (рисунок 4.13). В целях повышения наглядности на рисунках представлены результаты только для задач с числом кубиков не более 400.

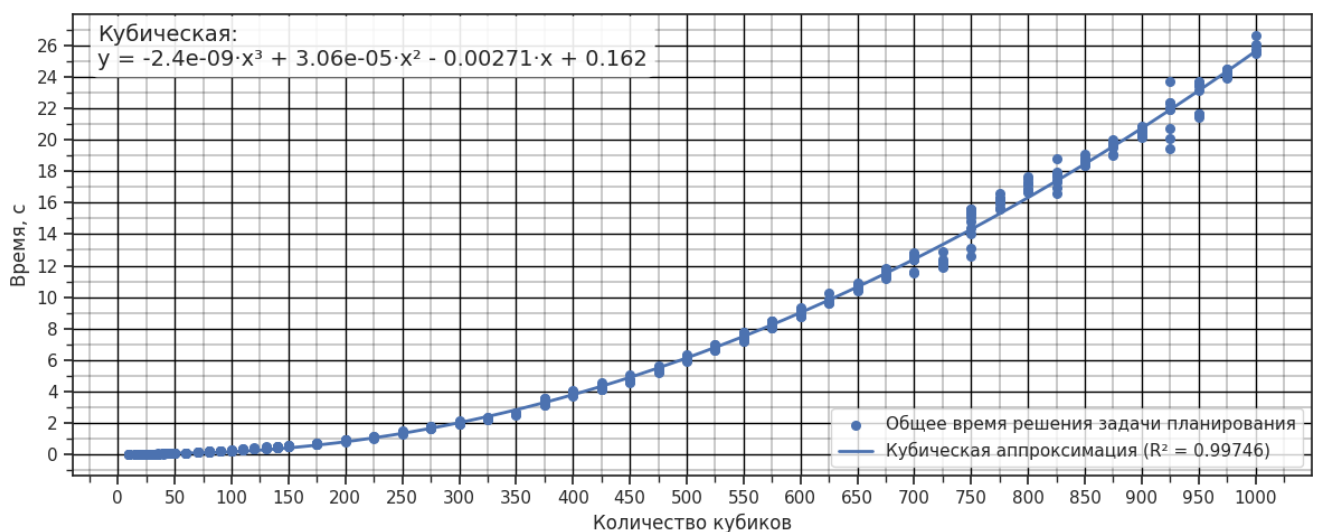


Рисунок 4.4 — Общее время решения задачи планирования

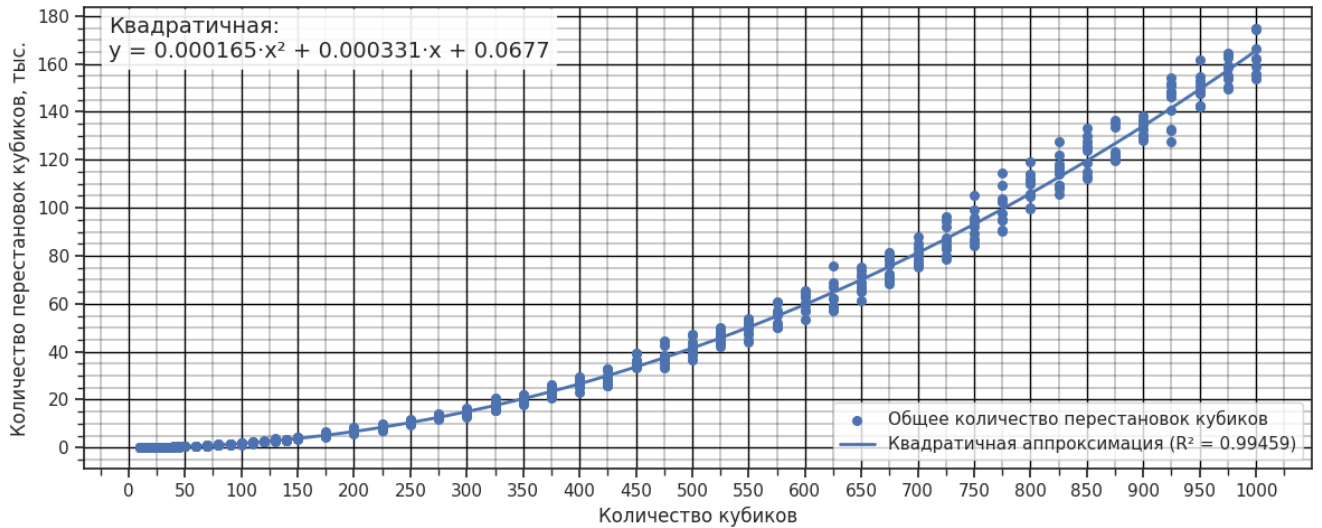


Рисунок 4.5 — Общее количество перестановок кубиков

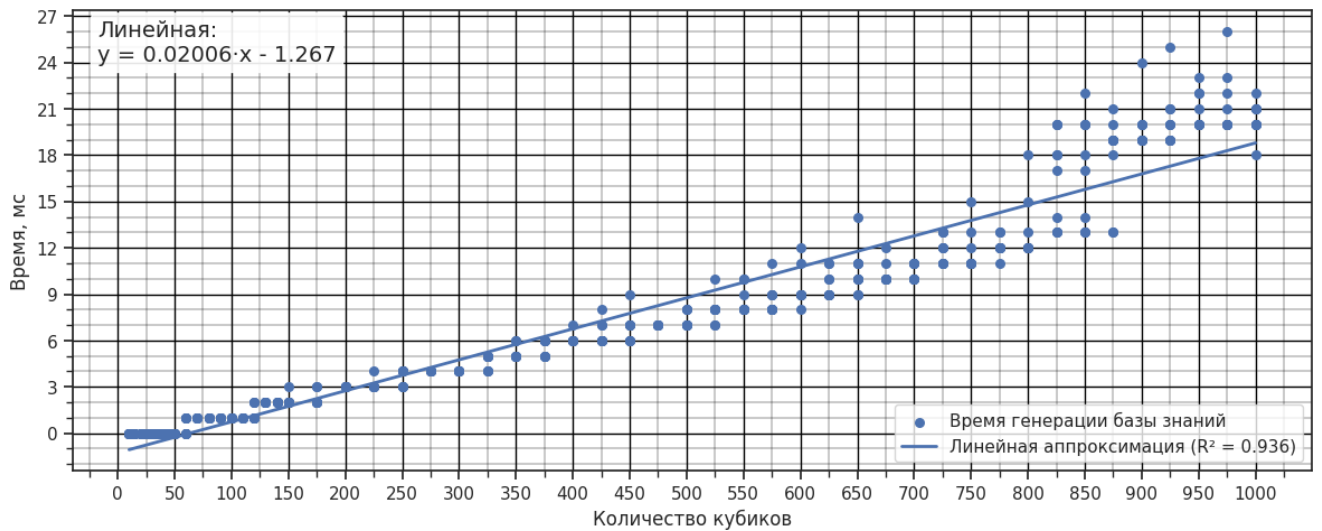


Рисунок 4.6 — Время генерации базы знаний

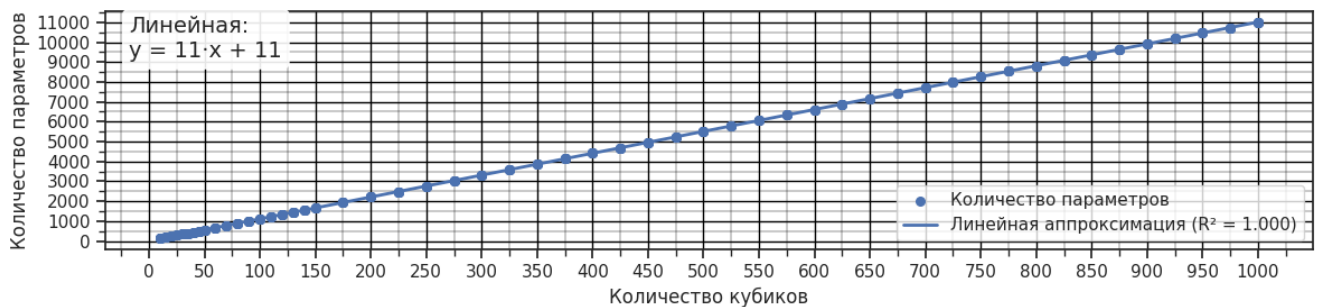


Рисунок 4.7 — Зависимость количества параметров от количества кубиков

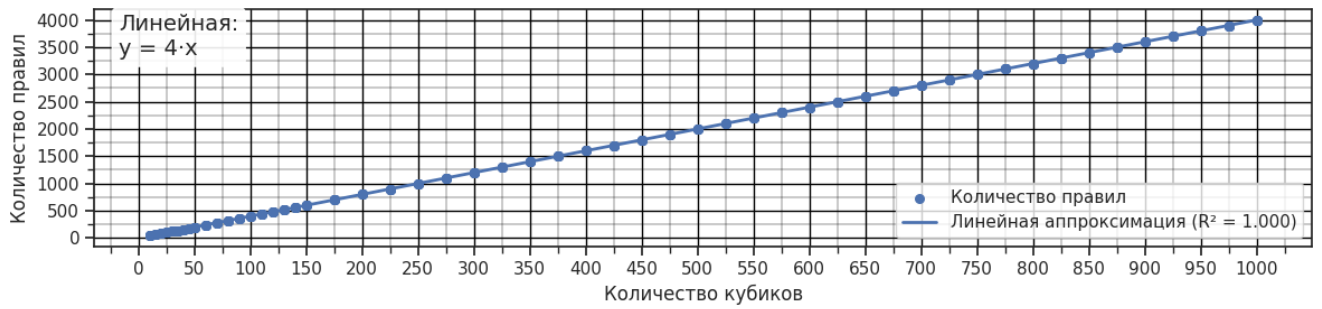


Рисунок 4.8 — Зависимость количества правил от количества кубиков

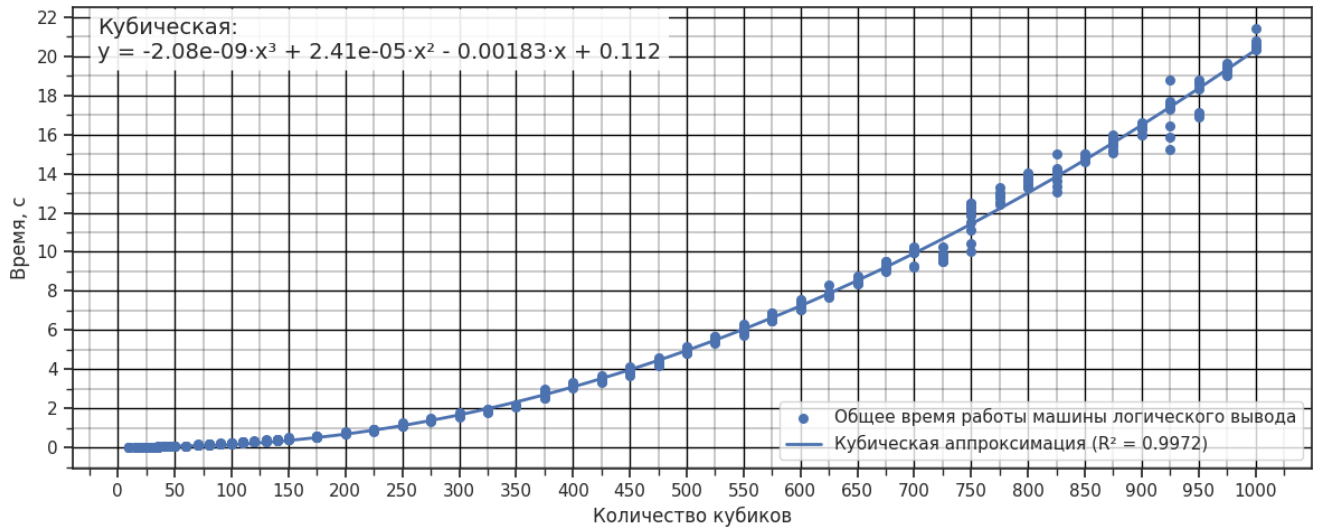


Рисунок 4.9 — Общее время работы машины логического вывода

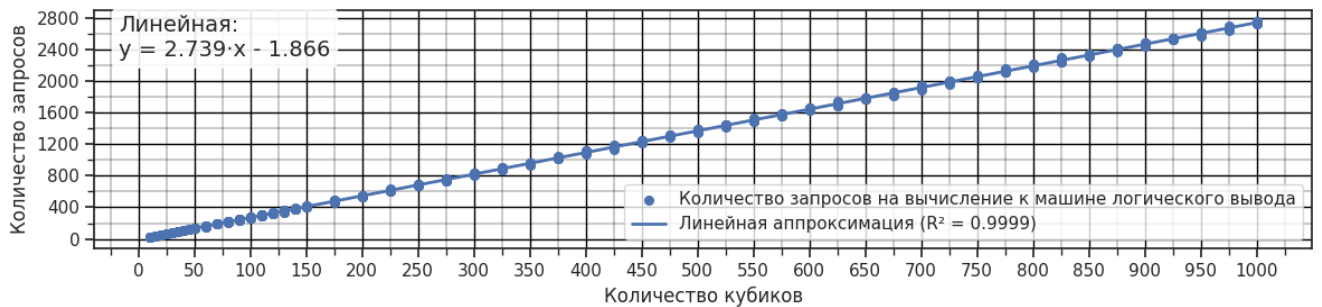


Рисунок 4.10 — Количество запросов на вычисление к машине логического вывода

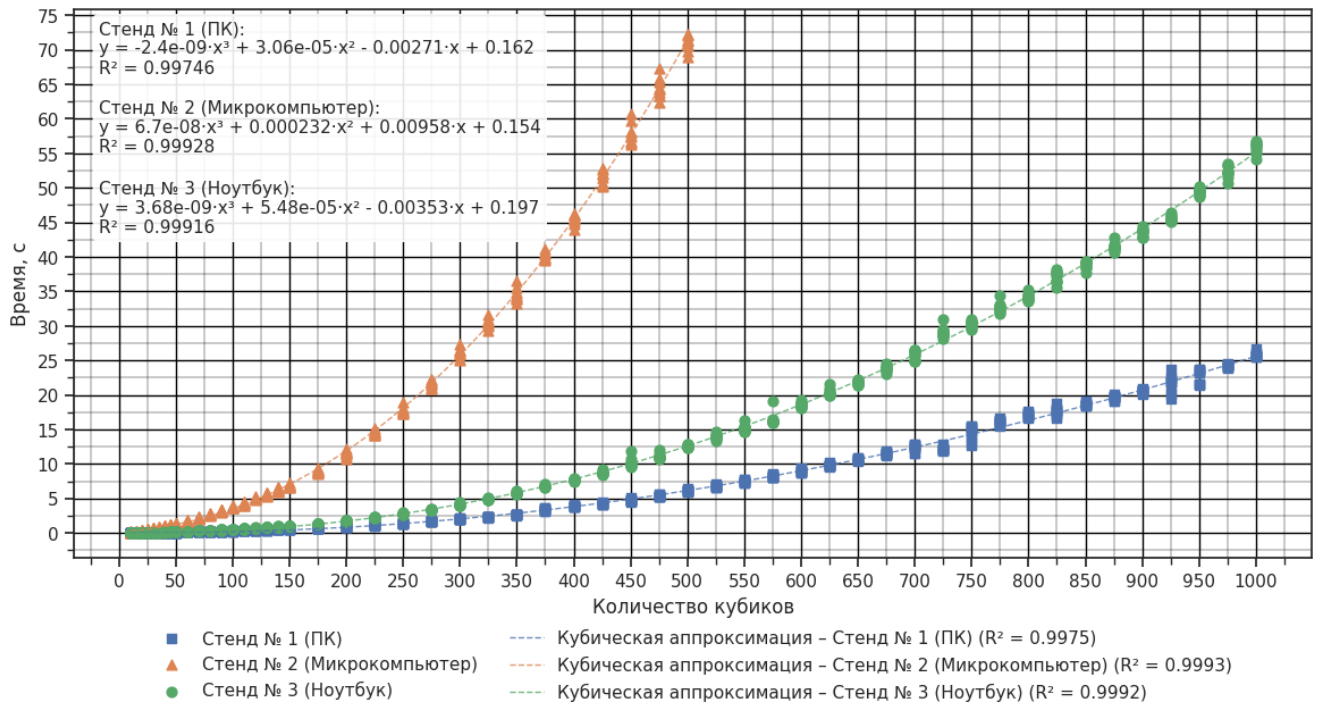


Рисунок 4.11 — Сравнение производительности на разных стандах (общее время решения задачи планирования)

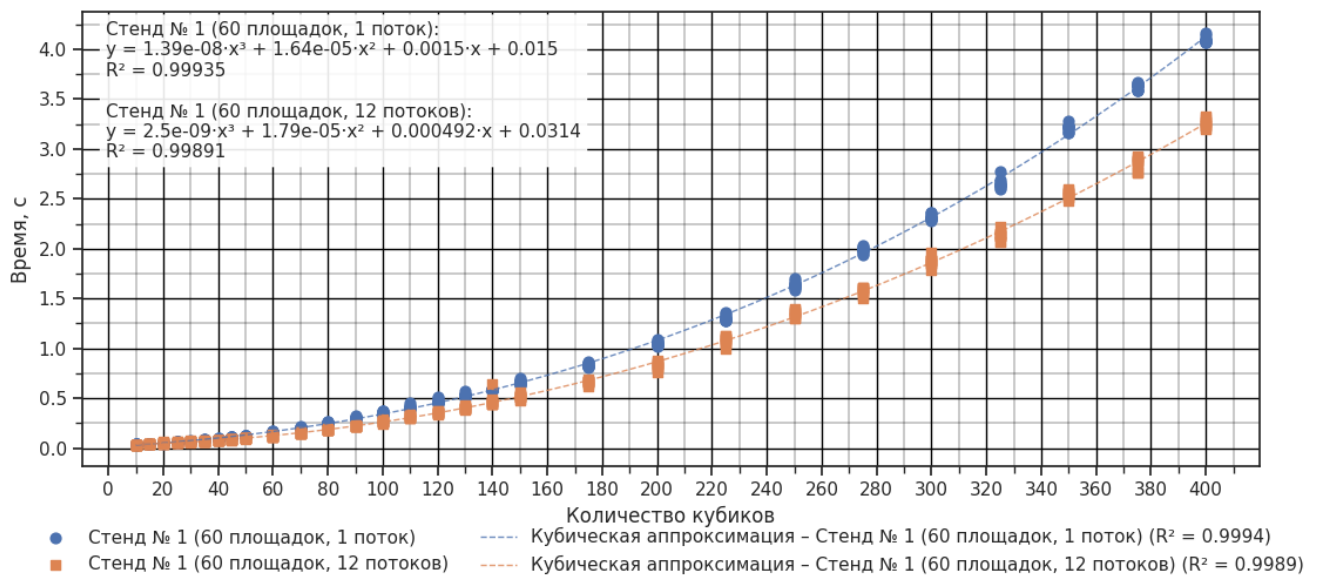


Рисунок 4.12 — Сравнение производительности при различном количестве доступных вычислительных потоков на станде № 1 при решении задач с 60 площадками (общее время решения задачи планирования)

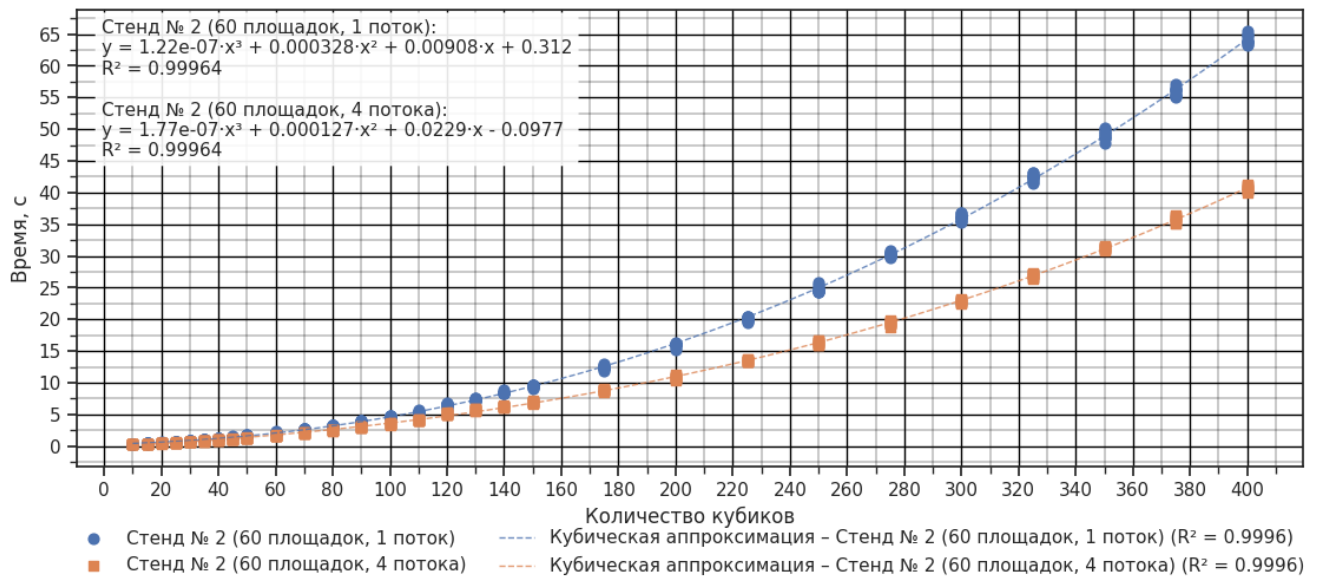


Рисунок 4.13 — Сравнение производительности при различном количестве доступных вычислительных потоков на стенде № 2 при решении задач с 60 площадками (общее время решения задачи планирования)

Проведём оценку полученных результатов измерения общего времени решения задачи планирования (рисунок 4.4) для тестового стенда № 1.

Оценим однородность полученного набора экспериментальных данных. Для этого сформируем подвыборки на основе исходного набора данных, сгруппированные по количеству кубиков. Для каждой подвыборки вычислим коэффициент вариации c_v :

$$c_v = \frac{\sigma}{\bar{x}} \times 100\%, \quad (4.16)$$

где $\sigma = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2}$ – стандартное (среднеквадратическое) отклонение, $\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$ – среднее арифметическое (математическое ожидание), $n = 10$ – число наблюдений в подвыборке, x_i – значение из подвыборки.

Результаты вычисления коэффициента вариации для сформированных подвыборок представлены на рисунке 4.14.

Из рисунка 4.14 следует, что значение коэффициента вариации не превышает 15%, что позволяет считать полученные данные однородными [179].

Коэффициент детерминации R^2 вычисляется по формуле:

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}, \quad (4.17)$$

где: y_i – значения, полученные в результате эксперимента; \hat{y}_i — значения, вычисленные с использованием аппроксимирующей функции; $\bar{y} = \frac{1}{n} \sum_{i=1}^n y_i$ – среднее значение, полученное по результатам эксперимента; n – количество решённых задач на данном тестовом стенде.

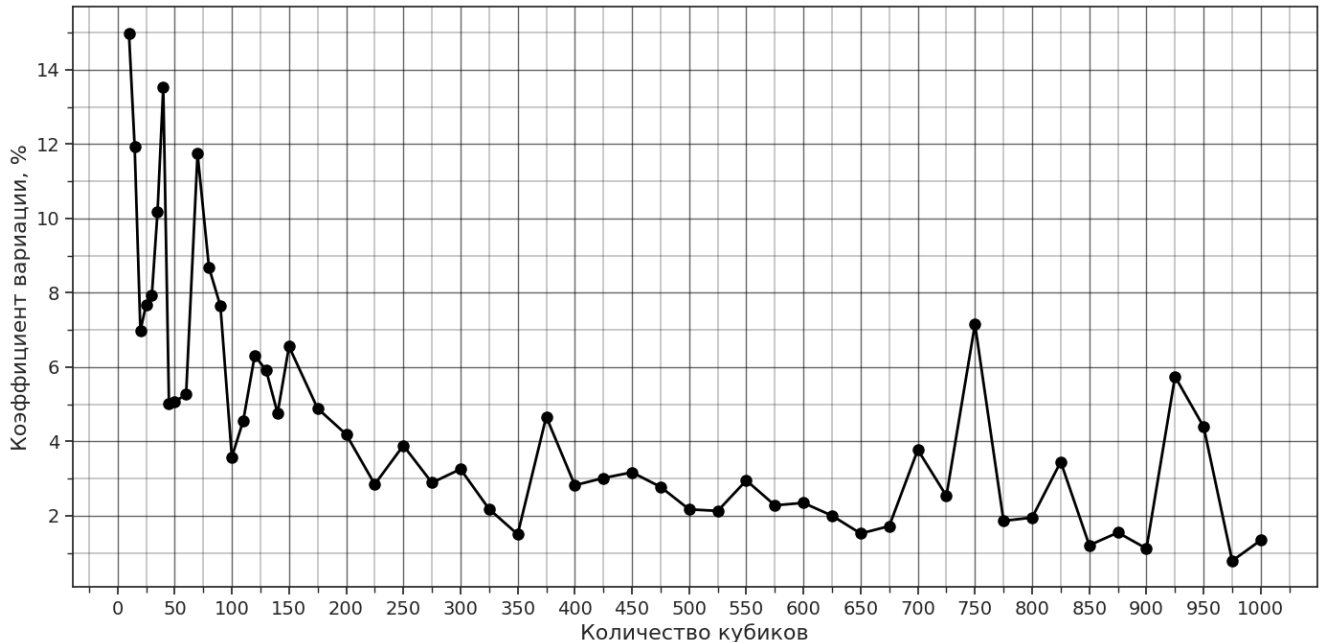


Рисунок 4.14 — Сравнение производительности на разных стендах

Для стенда № 1 квадратичная аппроксимация:

$$y = 2.71 \cdot 10^{-5} \cdot x^2 - 0.00142 \cdot x + 0.083, \quad (4.18)$$

$$R^2 = 0.99742$$

Для стенда № 1 кубическая аппроксимация:

$$y = -2.4 \cdot 10^{-9} \cdot x^3 + 3.06 \cdot 10^{-5} \cdot x^2 - 0.00271 \cdot x + 0.162, \quad (4.19)$$

$$R^2 = 0.99746$$

При сравнении кубической аппроксимации (формула (4.19)) с квадратичной (формула (4.18)) установлено, что коэффициент детерминации для кубической зависимости ($R^2 = 0.99746$) больше соответствующего значения квадратичной аппроксимации ($R^2 = 0.99742$). Следовательно, кубическая зависимость точнее описывает зависимость общего времени решения задачи планирования от количества кубиков.

Тогда для тестового стенда № 2 кубическая аппроксимация зависимости общего времени решения задачи планирования от количества кубиков имеет вид

(рисунок 4.11):

$$y = 6.7 \cdot 10^{-8} \cdot x^3 + 0.000232 \cdot x^2 + 0.00958 \cdot x + 0.154, \quad (4.20)$$

$$R^2 = 0.99928.$$

Для стенда № 3 кубическая аппроксимация (рисунок 4.11):

$$y = 3.68 \cdot 10^{-9} \cdot x^3 + 5.48 \cdot 10^{-5} \cdot x^2 - 0.00353 \cdot x + 0.197, \quad (4.21)$$

$$R^2 = 0.99916.$$

Таким образом, результаты эксперимента позволили определить следующие значения среднего времени, затрачиваемого на решение одной задачи планирования на стенде № 1 (рисунок 4.4):

- 100 кубиков – 400 правил – 0.195 секунд;
- 300 кубиков – 1200 правил – 2.038 секунды;
- 600 кубиков – 2400 правил – 9.035 секунды;
- 1000 кубиков – 4000 правил – 25.652 секунды.

Результаты эксперимента с иной конфигурацией тестового стенда № 1 представлены в публикации [8]. Результаты эксперимента с более ранней версией системы МИПРА опубликованы в работе [6].

Для стенда № 1 (рисунок 4.12) при увеличении числа доступных вычислительных потоков с 1 до 12 для задач с 400 кубиками и 60 площадками общее время решения сократилось с 4.1286 секунд (кубическая аппроксимация: $1.39 \times 10^{-8} \cdot x^3 + 1.64 \times 10^{-5} \cdot x^2 + 0.0015 \cdot x + 0.015$) до 3.2522 секунд (кубическая аппроксимация: $2.5 \times 10^{-9} \cdot x^3 + 1.79 \times 10^{-5} \cdot x^2 + 0.000492 \cdot x + 0.0314$), что соответствует ускорению в 1.27 раза ($4.1286/3.2522 \approx 1.27$).

Для стенда № 2 (рисунок 4.13) при увеличении числа доступных вычислительных потоков с 1 до 4 для задач с 400 кубиками и 60 площадками общее время решения сократилось с 64.232 секунд (кубическая аппроксимация: $1.22 \times 10^{-7} \cdot x^3 + 0.000328 \cdot x^2 + 0.00908 \cdot x + 0.312$) до 40.7103 секунд (кубическая аппроксимация: $1.77 \times 10^{-7} \cdot x^3 + 0.000127 \cdot x^2 + 0.0229 \cdot x - 0.0977$), что соответствует ускорению в 1.58 раза ($64.232/40.7103 \approx 1.58$).

Сравнительный анализ производительности при различном количестве доступных вычислительных потоков на стендах № 1 и № 2 при решении задач с 60 площадками показывает, что увеличение числа потоков приводит к снижению об-

щего времени решения задач планирования. При малых значениях числа кубиков различия во времени решения незначительны. С ростом количества кубиков, а следовательно, увеличением высоты башен, различия становятся статистически значимыми.

4.4.3 Анализ соответствия расчётных и численных результатов

В работе [141] приведены теоретические положения, обосновывающие вычислительную сложность процесса построения логического вывода, основанного на использовании МТ ЛИИ. Опираясь на данные положения, в публикации [8] была получена оценка теоретической вычислительной сложности алгоритма планировщика, применяемого в системе МИПРА.

Рассмотрим задачу планирования, для решения которой сформирована миварная сеть, включающая m правил и n параметров. При использовании представления миварной сети в виде табличных структур, применяемых при исследовании миварной технологии обработки информации в работах [133, 141], а также в приложении Б, строится миварная матрица V размерности $(m \times n)$.

В работе [141] показано, что в случае отсутствия возможности применения модификаций алгоритма обработки миварной сети МЛВ решает задачу построения маршрута логического вывода в миварной матрице V с вычислительной сложностью

$$KD = O(m \times n), \quad (4.22)$$

где KD обозначает количество элементарных действий, выполняемых при использовании метода миварной обработки.

В частных случаях может наблюдаться линейная зависимость либо от количества правил при фиксированном числе параметров ($n = \text{const}$) (выражение (4.23)), либо от количества параметров при фиксированном числе правил ($m = \text{const}$) (выражение (4.24)). Если количество правил и параметров не изменя-

ется во времени, то имеет место зависимость, представленная выражением (4.25).

$$KD = O(m \times Const). \quad (4.23)$$

$$KD = O(Const \times n). \quad (4.24)$$

$$KD = O(Const \times Const). \quad (4.25)$$

Зависимость количества правил сгенерированной МБЗ от числа кубиков x представлена на рисунке 4.8 и имеет следующий вид:

$$m = 4 \cdot x. \quad (4.26)$$

Зависимость количества параметров сгенерированной миварной сети от числа кубиков x представлена на рисунке 4.7 и имеет следующий вид:

$$n = 11 \cdot x + 11. \quad (4.27)$$

Для планирования одного шага миварный планировщик формирует запрос к МЛВ на построение логического вывода. Опираясь на выражения (4.26) и (4.27), получаем, что вычислительная сложность построения логического вывода для одного шага имеет следующий вид:

$$KD = O(x^2). \quad (4.28)$$

В соответствии с обобщённым алгоритмом решения задачи планирования, рассмотренным в параграфе 3.3.2, миварный планировщик формирует план таким образом, что для достижения одной промежуточной цели выполняется не более трёх шагов (подготовка среды, подготовка текущего объекта, достижение текущей цели). Для формирования промежуточного плана действий в рамках каждого шага требуется обращение к МЛВ с целью построения логического вывода по миварной сети. Число промежуточных целей совпадает с количеством кубиков в среде ПрО. Следовательно, зависимость количества обращений к МЛВ от числа кубиков x имеет следующий вид:

$$y = 3 \cdot x. \quad (4.29)$$

С учётом выражения (4.28) вычислительная сложность функционирования МЛВ при построении полного плана действий определяется следующим выражением:

$$KD = O(x^3). \quad (4.30)$$

Построение логического вывода для миварного планировщика является наиболее ресурсоёмким процессом (рисунок 4.9). Следовательно, в условиях реальной эксплуатации МЛВ оказывает существенное влияние на функционирование планировщика и определяет основные временные затраты при решении задач планирования.

Из рисунка 4.11 следует, что характеристики аппаратной платформы оказывают влияние на время решения задачи планирования: чем выше вычислительные возможности платформы, тем быстрее миварная проблемно-ориентированная система решает задачи планирования. При этом важно подчеркнуть, что аппаратная платформа не влияет на тип зависимости общего времени решения задачи планирования от количества кубиков. Для всех рассмотренных конфигураций аппаратных платформ данная зависимость аппроксимируется полиномом третьей степени.

В параграфе 1.2.5, а также в публикации [8], приведены показатели производительности других планировщиков. Анализ представленных данных позволяет сделать вывод о том, что МИПРА демонстрирует существенно более высокую скорость подготовки планов по сравнению с рассмотренными планировщиками.

4.5 Выводы по четвёртой главе

1. Разработана программная реализация миварной машины логического вывода для задач интеллектуального планирования действий при принятии управленческих решений. Реализация обеспечивает автономную работу на бортовых вычислительных платформах, использование доступных вычисли-

тельных потоков и поддерживает выборочное вычисление выходных параметров правил при их активации.

2. Программная реализация миварной машины логического вывода опирается на математическое и алгоритмическое обеспечение обработки миварных сетей. Алгоритм обработки представлен в виде блок-схем, псевдокода и матричных преобразований, что обеспечивает комплексный подход к проектированию, разработке и исследованию миварных проблемно-ориентированных систем планирования действий в пространстве состояний.
3. Разработанная миварная машина логического вывода используется в качестве вычислительного ядра миварной системы планирования действий роботов и роботизированных средств для решения задач в модифицированном домене Blocks World. Система поддерживает как пошаговое формирование и немедленное исполнение частичных планов, так и предварительное построение полного плана с последующей передачей на исполнение. Интеграция с подсистемами предобработки сенсорных данных и управления исполнительными устройствами обеспечивается за счёт применения разработанных моделей и методов интеллектуального планирования действий в пространстве состояний.
4. Продемонстрирована переносимость предложенных моделей и методов планирования действий при принятии управленческих решений. Разработан комплекс программ, включающий логическую интеллектуальную систему контроля за соблюдением правил дорожного движения, логическую интеллектуальную систему обеспечения ухода за растениями, программный комплекс на основе информационно-аналитической системы с поддержкой принятия решений об эффективности и безопасности применения термолабильных компонентов крови в медицинской практике. Результаты разработки подтверждают гибкость и адаптивность разработанных моделей и методов к широкому спектру прикладных задач управления и поддержки принятия решений.
5. Производительность математического и алгоритмического обеспечения миварной системы планирования действий роботов и роботизированных средств подтверждена результатами вычислительных экспериментов, показавшими высокую скорость решения задач. Продемонстрирована переносимость разработанного решения на различные аппаратные платформы, вклю-

чая одноплатные микрокомпьютеры. Сравнение с альтернативными планировщиками показало более высокую скорость подготовки планов и снижение вычислительной сложности за счёт применения разработанных моделей и методов.

ЗАКЛЮЧЕНИЕ

В ходе проведенного исследования были достигнуты поставленные цели и решены все задачи диссертационной работы. Получены следующие основные результаты:

1. Выполнено обзорно-аналитическое исследование подходов автоматического и интеллектуального планирования действий в пространстве состояний. Показано, что универсальные планировщики, основанные на PDDL и ASP, существенно зависят от размера и сложности предметной области, что приводит к росту вычислительной трудоёмкости и ограничивает применение в автономных технических системах реального времени.
2. Разработана модель миварной базы знаний для решения управленческих задач в пространстве состояний, обеспечивающая структурную декомпозицию знаний по множествам параметров и правил, а также возможность автоматической генерации, масштабирования и параллельной обработки логического вывода.
3. Разработана модель миварной проблемно-ориентированной системы управления и принятия решений в пространстве состояний, позволяющая представлять систему в виде сложного графа взаимосвязанных элементов и формализовать интерфейсы взаимодействия с внешними информационными системами в рамках гибридных интеллектуальных информационных систем.
4. Предложен метод автоматической генерации миварных баз знаний для решения управленческих задач в пространстве состояний, основанный на использовании заранее подготовленных прототипов и шаблонов параметров и правил. Метод обеспечивает динамическое масштабирование базы знаний при изменении количества объектов управления и состояния предметной области без участия когнитолога или эксперта в режиме реального времени.
5. Создана модель генератора миварной базы знаний в виде метаграфового агента, обеспечивающая унифицированное описание интерфейсов взаимо-

действия компонентов проблемно-ориентированной системы и корректные преобразования представления знаний между метаграфовой и миварной концептуальными моделями предметной области. Это обеспечивает архитектурную целостность миварной системы и допускает применение различных методов искусственного интеллекта при генерации баз знаний.

6. Разработан метод решения задач принятия решений и обработки информации для планирования действий в пространстве состояний, реализующий итеративный цикл «анализ состояния – формирование шага плана – исполнение – обратная связь», обеспечивающий автоматическую генерацию частных планов действий и их адаптацию к текущему состоянию и целям, включая повторное планирование при изменениях среды.
7. Разработано математическое и алгоритмическое обеспечение, реализующее параллельную обработку логических правил в миварной машине логического вывода, что обеспечивает снижение вычислительной сложности и ускорение построения планов действий в интеллектуальных системах управления.
8. Создана и реализована программная миварная машина логического вывода для задач интеллектуального планирования действий при принятии управленческих решений, обеспечивающая автономную работу на бортовых вычислительных платформах, использование доступных вычислительных потоков и выборочное вычисление выходных параметров при активации правил.
9. Подтверждена применимость предложенных моделей и методов на примере решения задач модифицированного домена Blocks World: выполнена формализация состояния среды, действий и целей, целевое состояние представлено системой упорядоченных промежуточных целей, что обеспечивает корректную последовательность манипуляций и повышает устойчивость планирования в динамически изменяющихся условиях.
10. Продемонстрирована переносимость разработанных моделей и методов на широкий спектр прикладных задач управления и поддержки принятия решений (в том числе контроль соблюдения правил дорожного движения, обеспечение ухода за растениями, поддержка решений по применению термолабильных компонентов крови), что подтверждает их гибкость и адаптивность.
11. Работоспособность разработанного математического и алгоритмического обеспечения подтверждена вычислительными экспериментами на миварной

системе планирования действий роботов и роботизированных средств: показаны высокая скорость решения задач, масштабируемость и переносимость на различные аппаратные платформы, а сравнение с альтернативными планировщиками выявило преимущество по скорости подготовки планов и снижению вычислительной сложности.

СПИСОК СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ

ADAS (англ. *Advanced Driver-Assistance System*) – усовершенствованная система помощи водителю

GPS (англ. *General Problem Solver*) – универсальный решатель задач

STRIPS (англ. *Stanford Research Institute Problem Solver*) – Решатель задач Стэнфордского исследовательского института

ГИИС – гибридные интеллектуальные информационные системы

ДТП – дорожно-транспортное происшествие

ИИ – искусственный интеллект

КМПрО – концептуальная модель предметной области

ЛИС ОУР – логическая интеллектуальная система обеспечения ухода за растениями

ЛИСК ПДД – логическая интеллектуальная система контроля за соблюдением правил дорожного движения

МБЗ – миварная база знаний

МИПРА – миварная система планирования действий роботов и роботизированных средств

МЛВ – машина логического вывода

МТ ЛИИ – миварные технологии логического искусственного интеллекта

МЭС – миварные экспертные системы

ПДД – правила дорожного движения

ПКИАС – программный комплекс на основе информационно-аналитической системы с поддержкой принятия решений об эффективности и безопасности применения термолабильных компонентов крови в медицинской практике

ПрО – предметная область

РТК – робототехнический комплекс

СВММ – система верификации миварных моделей

СПИСОК ЛИТЕРАТУРЫ

1. Аладин, Д. В. Модель миварной базы знаний для решения управленческих задач в пространстве состояний / Д. В. Аладин // Нейрокомпьютеры: разработка, применение. – 2025. – Т. 27, № 4. – С. 66–71.
2. Аладин, Д. В. Модель миварной проблемно-ориентированной системы управления и принятия решений в пространстве состояний / Д. В. Аладин. – Текст : электронный // Моделирование, оптимизация и информационные технологии. – 2025. – Т. 13, № 4. – URL: <https://moitvvt.ru/ru/journal/pdf?id=2083>. – Дата публикации: 20.10.2025.
3. Аладин, Д. В. Метод автоматической генерации миварных баз знаний для решения управленческих задач в пространстве состояний / Д. В. Аладин // Проблемы искусственного интеллекта. – 2025. – № 3(38). – С. 88–99.
4. Варламов, О. О. О создании миварных систем контроля за соблюдением правил дорожного движения на основе “Разуматоров” и экспертных систем / О. О. Варламов, Д. В. Аладин // Радиопромышленность. – 2018. – № 2. – С. 25–35.
5. Варламов, О. О. О применении миварных сетей для интеллектуального планирования поведения роботов в пространстве состояний / О. О. Варламов, Д. В. Аладин // Известия Кабардино-Балкарского научного центра РАН. – 2018. – № 6-2(86). – С. 75–82.
6. Варламов, О. О. Успешное применение миварных экспертных систем для MIPRA - решения задач планирования действий робототехнических комплексов в реальном времени / О. О. Варламов, Д. В. Аладин // Радиопромышленность. – 2019. – № 3. – С. 15–25.
7. Программный комплекс с поддержкой принятия решений о безопасности применения термолабильных компонентов крови / О. О. Варламов, Д. А. Чувиков, В. Н. Лемонджава, А. Г. Гудков, Д. В. Аладин, Л. Е. Адамова, В. Г. Оси-

- пов, А. В. Чечеткин, В. Ю. Леушин, А. Д. Касьянов, Н. А. Ветрова // Медицинская техника. – 2021. – № 5(329). – С. 40–43.
8. Varlamov, O. A New Generation of Rules-based Approach: Mivar-based Intelligent Planning of Robot Actions (MIPRA) and Brains for Autonomous Robots / O. Varlamov, D. Aladin // Machine Intelligence Research. – 2024. – Vol. 21, No. 5. – P. 919–940.
 9. Logical artificial intelligence Mivar technologies for autonomous road vehicles / O. O. Varlamov, D. A. Chuvikov, D. V. Aladin [et al.] // IOP Conference Series: Materials Science and Engineering. – 2019. – Vol. 534. – P. 012015.
 10. Logic-based artificial intelligence in systems for monitoring the enforcing traffic regulations / D. V. Aladin, O. O. Varlamov, D. A. Chuvikov [et al.] // IOP Conference Series: Materials Science and Engineering. – 2019. – Vol. 534. – P. 012025.
 11. Control of vehicles and robots: Creation of planning systems in the state space (MIPRA) / D. V. Aladin, O. O. Varlamov, D. A. Fedoseev [et al.] // IOP Conference Series: Materials Science and Engineering. – 2020. – Vol. 747. – P. 012097.
 12. Control of machines and robots: Creation of mivar decision-making systems for controlling autonomous tractors and special vehicles of the ministry of emergencies / D. V. Aladin, O. O. Varlamov, D. A. Chuvikov [et al.] // IOP Conference Series: Materials Science and Engineering. – 2020. – Vol. 747. – P. 012098.
 13. Control of vehicles and robots: Creating of knowledge bases for mivar decision making systems robots and vehicles / D. V. Aladin, O. O. Varlamov, L. E. Adamova [et al.] // IOP Conference Series: Materials Science and Engineering. – 2020. – Vol. 747. – P. 012099.
 14. Creation of autonomous groups of combine harvesters and tractors for agriculture based on the Mivar decision-making systems “ROBO!RAZUM” / O. O. Varlamov, D. V. Aladin, L. E. Adamova [et al.] // IOP Conference Series: Materials Science and Engineering. – 2020. – Vol. 819. – P. 012002.
 15. About the project developing “MIPRA” – the intelligent planner in the state space for vehicles, tractors, and robots based on the architectural solutions of the Mivar systems for traffic enforcement / D. V. Aladin, O. O. Varlamov, D. A. Fedoseev

- [et al.] // IOP Conference Series: Materials Science and Engineering. – 2020. – Vol. 819. – P. 012006.
16. A new method for creating Mivar knowledge bases in tabular-matrix form for ground intelligent vehicle control systems / D. A. Chuvikov, D. V. Aladin, L. E. Adamova [et al.] // Journal of Physics: Conference Series. – 2021. – Vol. 2061. – P. 012123.
 17. Creating a “Logical intelligent plant care system” in digital agriculture based on Mivar approach / D. V. Aladin, E. V. Aladina, D. A. Chuvikov [et al.] // IOP Conference Series: Earth and Environmental Science. – 2022. – Vol. 9541. – P. 012004.
 18. A Software Package Supporting Decision Making on the Safety of Thermolabile Blood Components / O. O. Varlamov, D. A. Chuvikov, V. N. Lemondzhava, A. G. Gudkov, D. V. Aladin, L. E. Adamova, V. G. Osipov, A. V. Chechetkin, V. Yu. Leushin, A. D. Kasyanov, N. A. Vetrova // Biomedical Engineering. – 2022. – Vol. 55, No. 5. – P. 355–359.
 19. Aladin, D. V. On Approaches to Creating the Logical Inference Machines for Ensuring Compatibility and Integration of Systems and Controls / D. V. Aladin, A. K. Bilalov, L. I. Ilinov // 7th International Youth Conference on Radio Electronics, Electrical and Power Engineering (REEPE). – [б. м.] : Institute of Electrical and Electronics Engineers, 2025.
 20. Свидетельство о государственной регистрации программы для ЭВМ № 2020618424 Российская Федерация. Система верификации миварных моделей : № 2020617729 : заявл. 14.07.2020 : опубл. 28.07.2020 / Д. В. Аладин.
 21. Свидетельство о государственной регистрации программы для ЭВМ № 2022610851 Российская Федерация. Серверная часть программного комплекса на основе информационно-аналитической системы с поддержкой принятия решений об эффективности и безопасности применения термолабильных компонентов крови в медицинской практике : № 2021669016 : заявл. 24.11.2021 : опубл. 17.01.2022 / Д. В. Аладин, Д. А. Чувииков, А. А. Кононенко [и др.] ; заявитель Общество с ограниченной ответственностью “Научно-производственная инновационная фирма “ГИПЕРИОН”.
 22. Свидетельство о государственной регистрации программы для ЭВМ № 2021680245 Российская Федерация. Клиентская часть программного комплекса на основе информационно-аналитической системы с поддержкой

- принятия решений об эффективности и безопасности применения термолабильных компонентов крови в медицинской практике : № 2021668891 : заявл. 24.11.2021 : опубл. 08.12.2021 / Д. В. Аладин, Д. А. Чувиков, А. А. Кононенко [и др.] ; заявитель Общество с ограниченной ответственностью “Научно-производственная инновационная фирма “ТИПЕРИОН”.
23. Свидетельство о государственной регистрации программы для ЭВМ № 2023684948 Российская Федерация. ЛИСА МЛВ : № 2023683700 : заявл. 09.11.2023 : опубл. 21.11.2023 / Д. В. Аладин, А. М. Балашов, Ю. Е. Гапанюк [и др.].
24. Свидетельство о государственной регистрации программы для ЭВМ № 2025616642 Российская Федерация. ЛИСА Редактор : заявл. 01.03.2025 : опубл. 18.03.2025 / Д. В. Аладин, Е. В. Аладина, А. К. Билалов [и др.].
25. Миварные системы принятия решений роботов. РобоРазум : монография / О. О. Варламов, А. А. Коценко, Д. В. Аладин [и др.]. – Москва : ИНФРА-М, 2024. – 549 с.
26. Коваленко, А. А. О системе адаптивного управления движением мобильными роботизированными устройствами / А. А. Коваленко, Д. В. Аладин // Труды Конгресса по интеллектуальным системам и информационным технологиям «IS&IT’ 17» : сб. трудов конф. : в 3 т. – Таганрог: Изд-во Ступина С.А., 2017. – Т. 1. – С. 213–218.
27. Аладин, Д. В. О миварной системе помощи водителю по применению правил дорожного движения / Д. В. Аладин, А. А. Миядин // Труды Конгресса по интеллектуальным системам и информационным технологиям «IS&IT’ 18» : сб. трудов конф. : в 3 т. – Таганрог: Изд-во Ступина С.А., 2018. – Т. 1. – С. 338–342.
28. Аладин, Д. В. Принципы создания системы контроля за соблюдением правил дорожного движения на базе миварных технологий / Д. В. Аладин // Студенческая научная весна : сб. тезисов конф. – Москва: ООО “Издательский дом “Научная библиотека”, 2018. – С. 260–261.
29. Аладин, Д. В. О разработке миварной системы контроля и помощи водителю по соблюдению правил дорожного движения / Д. В. Аладин, И. Г. Булатова, А. А. Миядин // Модели мышления и интеграция информационно-управляющих систем (ММИИУС-2018) : материалы второй международной научной конференции – Нальчик: Изд-во КБНЦ РАН, 2018. – С. 145–147.

30. О перспективах использования миварного подхода в системах контроля за соблюдением правил дорожного движения / Д. В. Аладин, О. О. Варламов, Д. А. Чувилов [и др.] // Технологии и компоненты интеллектуальных транспортных систем : материалы международного автомобильного научного форума (МАНФ-2018). – Москва: Центральный ордена Трудового Красного Знамени научно-исследовательский автомобильный и автомоторный институт “НАМИ”, 2018. – С. 127–140.
31. Аладин, Д. В. О применении принципов создания миварной системы контроля за соблюдением правил дорожного движения в интеллектуальных планировщиках / Д. В. Аладин, О. О. Варламов // Труды Международного научно-технического конгресса «Интеллектуальные системы и информационные технологии - 2019» («ИС & ИТ-2019», «IS&IT' 19») : сб. трудов конф. : в 2 т. – Таганрог: Изд-во Ступина С.А., 2019. – Т. 1. – С. 301–306.
32. Аладин, Д. В. Об архитектурных решениях для создания миварных систем интеллектуального планирования реального времени / Д. В. Аладин // Труды Международного научно-технического конгресса «Интеллектуальные системы и информационные технологии - 2019» («ИС & ИТ-2019», «IS&IT' 19») : сб. трудов конф. : в 2 т. – Таганрог: Изд-во Ступина С.А., 2019. – Т. 1. – С. 306–313.
33. Варламов, О. О. О миварном подходе к созданию логически рассуждающих систем ситуационного планирования / О. О. Варламов, Д. В. Аладин // Труды Международного научно-технического конгресса «Интеллектуальные системы и информационные технологии - 2019» («ИС & ИТ-2019», «IS&IT' 19») : сб. трудов конф. : в 2 т. – Таганрог: Изд-во Ступина С.А., 2019. – Т. 1. – С. 313–320.
34. Управление машинами и роботами: создание миварных систем принятия решений для управления автономными тракторами и спецмашинами МЧС / Д. В. Аладин, О. О. Варламов, Д. А. Чувилов [и др.] // XXXI Международная инновационная конференция молодых ученых и студентов по проблемам машиноведения (МИКМУС - 2019) : сб. трудов конф. – Москва: Федеральное государственное бюджетное учреждение науки Институт машиноведения им. А.А. Благонравова Российской академии наук, 2020. – С. 716–719.

35. Управление машинами и роботами: создание баз знаний для миварных систем принятия решений роботов и автомобилей / Д. В. Аладин, О. О. Варламов, Л. Е. Адамова [и др.] // XXXI Международная инновационная конференция молодых ученых и студентов по проблемам машиноведения (МИКМУС - 2019) : сб. трудов конф. – Москва: Федеральное государственное бюджетное учреждение науки Институт машиноведения им. А.А. Благонравова Российской академии наук, 2020. – С. 720–723.
36. Управление машинами и роботами: создание систем планирования в пространстве состояний (МИПРА) / Д. В. Аладин, О. О. Варламов, Д. А. Чувилов [и др.] // XXXI Международная инновационная конференция молодых ученых и студентов по проблемам машиноведения (МИКМУС - 2019) : сб. трудов конф. – Москва: Федеральное государственное бюджетное учреждение науки Институт машиноведения им. А.А. Благонравова Российской академии наук, 2020. – С. 724–727.
37. Аладин, Д. В. Применение миварной экспертной системы для решения задач в предметной области “правила дорожного движения” / Д. В. Аладин, Е. В. Ромичева, М. Ю. Чекулина // Труды Международного научно-технического конгресса «Интеллектуальные системы и информационные технологии - 2020» («ИС & ИТ-2020», «IS&IT'20») : сб. трудов конф. : в 2 т. – Таганрог: Изд-во Ступина С.А., 2020. – Т. 2. – С. 285–293.
38. Аладина, Е. В. Создание логической интеллектуальной системы обеспечения ухода за растениями на основе миварных экспертных систем / Е. В. Аладина, Д. В. Аладин // Труды Международного научно-технического конгресса «Интеллектуальные системы и информационные технологии - 2020» («ИС & ИТ-2020», «IS&IT'20») : сб. трудов конф. : в 2 т. – Таганрог: Изд-во Ступина С.А., 2020. – Т. 2. – С. 304–312.
39. Аладин, Д. В. Миварная экспертная система для контроля соблюдения правил дорожного движения / Д. В. Аладин, Е. В. Ромичева, М. Ю. Чекулина // Интеллектуальные технологии и проблемы математического моделирования : тезисы докладов III Всероссийской научной конференции – Ростов-на-Дону: Донской государственный технический университет, 2020. – С. 28.
40. Аладина, Е. В. Миварная логическая интеллектуальная система обеспечения ухода за растениями на фермах / Е. В. Аладина, Д. В. Аладин // Ин-

- теллектуальные технологии и проблемы математического моделирования : тезисы докладов III Всероссийской научной конференции, Ростов-на-Дону – Ростов-на-Дону: Донской государственный технический университет, 2020. – С. 29.
41. Аладин, Д. В. Об использовании миварной экспертной системы для предметной области “правила дорожного движения” / Д. В. Аладин, Е. В. Ромичева, М. Ю. Чекулина // *Фундаментальные основы физики, химии и механики наукоёмких технологических систем формообразования и сборки изделий* : сб. трудов научного симпозиума технологов-машиностроителей. – Ростов-на-Дону: Донской государственный технический университет, 2020. – С. 190–198.
42. Аладина, Е. В. О возможности создания логической интеллектуальной системы обеспечения ухода за растениями на основе миварных экспертных систем / Е. В. Аладина, Д. В. Аладин // *Фундаментальные основы физики, химии и механики наукоёмких технологических систем формообразования и сборки изделий* : сб. трудов научного симпозиума технологов-машиностроителей. – Ростов-на-Дону: Донской государственный технический университет, 2020. – С. 230–237.
43. Разработка методики создания миварных баз знаний в таблично-матричном виде на примере систем управления наземных интеллектуальных транспортных средств / Д. А. Чувиков, Д. В. Аладин, Л. Е. Адамова [и др.] // *Форум инновационных транспортных технологий, наземные интеллектуальные транспортные средства и системы* : материалы объединённого международного онлайн форума. – Москва: Центральный ордена Трудового Красного Знамени научно-исследовательский автомобильный и автомоторный институт “НАМИ”, 2020. – С. 1140–1152.
44. О проекте создания в цифровом сельском хозяйстве “логической интеллектуальной системы обеспечения ухода за растениями” на основе миварного подхода / Д. В. Аладин, Е. В. Аладина, Д. А. Чувиков [и др.] // *Форум инновационных транспортных технологий, наземные интеллектуальные транспортные средства и системы* : материалы объединённого международного онлайн форума. – Москва: Центральный ордена Трудового Красного Знамени научно-исследовательский автомобильный и автомоторный институт “НАМИ”, 2020. – С. 1156–1164.

45. Аладин, Д. В. Об архитектурных особенностях систем интеллектуального планирования и систем контроля за соблюдением правил дорожного движения / Д. В. Аладин, Д. А. Чувиков, О. О. Варламов // Искусственный интеллект в автоматизированных системах управления и обработки данных : сб. статей Всероссийской научной конф. : в 2 т. – Москва : Изд-во МГТУ им. Н.Э. Баумана, 2022. – Т. 2. – С. 27–34.
46. Чувиков, Д. А. Создание автономных групп роботов на основе системы принятия решений / Д. А. Чувиков, Д. В. Аладин, О. О. Варламов // Искусственный интеллект в автоматизированных системах управления и обработки данных : сб. статей Всероссийской научной конф. : в 2 т. – Москва : Изд-во МГТУ им. Н.Э. Баумана, 2022. – Т. 2. – С. 212–219.
47. Машиностроительный искусственный интеллект: создание автономных групп роботов на основе миварных систем принятия решений / Д. А. Чувиков, Д. В. Аладин, А. А. Коценко [и др.] // Машиностроительные технологические системы : сб. трудов международной научно-технической конф. – Ростов-на-Дону: Донской государственный технический университет, 2022. – С. 416–422.
48. Аладин, Д. В. Миварные системы интеллектуального планирования и контроля за соблюдением ПДД / Д. В. Аладин, Д. А. Чувиков // Мивар'22 : сб. научных статей. – Москва : ИНФРА-М, 2022. – С. 344–350.
49. Чувиков, Д. А. Автономные группы роботов с миварными системами принятия решений / Д. А. Чувиков, Д. В. Аладин, О. О. Варламов // Мивар'22 : сб. научных статей. – Москва : ИНФРА-М, 2022. – С. 356–362.
50. Аладин, Д. В. Об использовании параллельной активации логических правил при решении управленческих задач в технических системах / Д. В. Аладин // Труды Международного научно-технического конгресса «Интеллектуальные системы и информационные технологии - 2023» («ИС & ИТ-2023», «IS&IT'23») : сб. трудов конф. : в 2 т. – Таганрог: Изд-во Ступина С.А., 2023. – Т. 1. – С. 348–354.
51. Сеницын, Л. С. Разработка платформы для системы поддержки принятия решения робота на базе гибридной интеллектуальной системы / Л. С. Сеницын, Д. В. Аладин // Труды Международного научно-технического конгресса «Интеллектуальные системы и информационные технологии - 2023»

- («ИС & ИТ-2023», «IS&IT'23»): сб. трудов конф. : в 2 т. – Таганрог: Изд-во Ступина С.А., 2023. – Т. 1. – С. 361–367.
52. Аладин, Д. В. Об автоматическом построении миварных баз знаний для решения управленческих задач в технических системах / Д. В. Аладин // Труды Международного научно-технического конгресса «Интеллектуальные системы и информационные технологии - 2023» («ИС & ИТ-2023», «IS&IT'23»): сб. трудов конф. : в 2 т. – Таганрог: Изд-во Ступина С.А., 2023. – Т. 2. – С. 148–154.
53. Плешаков, В. И. О выборе средств разработки миварной машины логического вывода для процессора Эльбрус / В. И. Плешаков, Д. В. Аладин // Труды Международного научно-технического конгресса «Интеллектуальные системы и информационные технологии - 2023» («ИС & ИТ-2023», «IS&IT'23»): сб. трудов конф. : в 2 т. – Таганрог: Изд-во Ступина С.А., 2023. – Т. 2. – С. 350–356.
54. Аладин, Д. В. Об одном подходе автоматического построения миварных баз знаний для решения управленческих задач в технических системах / Д. В. Аладин // Искусственный интеллект в автоматизированных системах управления и обработки данных : сб. статей II Всероссийской научной конференции : в 5 т. – Москва: Издательский дом КДУ, Добросвет, 2023. – Т. 3. – С. 521–526.
55. Аладин, Д. В. Об одном подходе использования параллельной активации логических правил при решении управленческих задач в технических системах / Д. В. Аладин // Искусственный интеллект в автоматизированных системах управления и обработки данных : сб. статей II Всероссийской научной конференции : в 5 т. – Москва: Издательский дом КДУ, Добросвет, 2023. – Т. 3. – С. 532–536.
56. Плешаков, В. И. Обоснование выбора средств разработки миварной машины логического вывода для процессора Эльбрус / В. И. Плешаков, Д. В. Аладин // Искусственный интеллект в автоматизированных системах управления и обработки данных : сб. статей II Всероссийской научной конференции : в 5 т. – Москва: Издательский дом КДУ, Добросвет, 2023. – Т. 3. – С. 542–546.
57. Синицын, Л. С. О разработке платформы для системы поддержки принятия решения робота на базе гибридной интеллектуальной системы / Л. С. Синицын, Д. В. Аладин // Искусственный интеллект в автоматизированных

- системах управления и обработки данных : сб. статей II Всероссийской научной конференции : в 5 т. – Москва: Издательский дом КДУ, Добросвет, 2023. – Т. 3. – С. 547–551.
58. Варламов, О. О. О подходе автоматического построения миварных баз знаний для решения управленческих задач в технических системах / О. О. Варламов, Д. В. Аладин // Перспективные направления развития отделочно-упрочняющих и виброволновых технологий : сб. трудов научного семинара. – Ростов-на-Дону: Донской государственный технический университет, 2023. – С. 208–216.
59. Варламов, О. О. О необходимости использования параллельной активации логических правил при решении управленческих задач в технических системах / О. О. Варламов, Д. В. Аладин // Перспективные направления развития отделочно-упрочняющих и виброволновых технологий : сб. трудов научного семинара. – Ростов-на-Дону: Донской государственный технический университет, 2023. – С. 216–223.
60. Аладин, Д. В. Подход автоматического построения миварных баз знаний для решения управленческих задач в технических системах / Д. В. Аладин // Мивар'23 : сб. студенческих статей. – Москва : ИНФРА-М, 2023. – С. 527–532.
61. Аладин, Д. В. Необходимость использования параллельной активации логических правил при решении управленческих задач в технических системах / Д. В. Аладин // Мивар'23 : сб. студенческих статей. – Москва : ИНФРА-М, 2023. – С. 533–538.
62. Синицын, Л. С. Платформа для СПР работа на базе гибридной интеллектуальной системы / Л. С. Синицын, Д. В. Аладин // МИВАР'24 : сб. научных статей. – Москва: ИНФРА-М, 2024. – С. 384–388.
63. Плешаков, В. И. Разработка миварной машины логического вывода для процессора Эльбрус / В. И. Плешаков, Д. В. Аладин // МИВАР'24 : сб. научных статей. – Москва: ИНФРА-М, 2024. – С. 450–454.
64. Билалов, А. К. О методе создания матричной машины логического вывода для баз знаний, основанных на графе правил и параметров / А. К. Билалов, Д. В. Аладин // ИИАСУ'24 – Искусственный интеллект в автоматизированных системах управления и обработки данных : сб. статей III Всероссийской

- научной конференции : в 3 т. – Москва: Издательский дом КДУ, 2025. – Т. 2. – С. 114–119.
65. Аладин, Д. В. О подходе к моделированию миварных экспертных систем для решения задач управления / Д. В. Аладин // Мивар '25 : сб. трудов конф. – Москва: ИНФРА-М, 2025. – С. 297–300.
66. Аладин, Д. В. О подходе к моделированию автоматически генерируемых миварных баз знаний для решения задач управления / Д. В. Аладин // Мивар '25 : сб. трудов конф. – Москва: ИНФРА-М, 2025. – С. 406–408.
67. Аладин, Д. В. О подходе к автоматической генерации миварных баз знаний для решения задач управления / Д. В. Аладин // Мивар '25 : сб. трудов конф. – Москва: ИНФРА-М, 2025. – С. 409–411.
68. Об опыте обучения студентов созданию продукционных баз знаний для миварных систем принятия решений беспилотных автомобилей / О. О. Варламов, Л. Е. Адамова, И. Г. Булатова, Д. В. Аладин // Информация и образование: границы коммуникаций. – 2019. – № 11(19). – С. 171–172.
69. Аладин, Д. В. Миварная система контроля за соблюдением правил дорожного движения для роботизированных устройств и транспортных средств / Д. В. Аладин, И. Г. Булатова, А. А. Миядин // Информация и образование: границы коммуникаций. – 2019. – № 11(19). – С. 151–152.
70. Аладина, Е. В. О применении миварных технологий для создания логической интеллектуальной системы обеспечения ухода за растениями / Е. В. Аладина, Д. В. Аладин // Информация и образование: границы коммуникаций. – 2020. – № 12(20). – С. 169–173.
71. Аладин, Д. В. О необходимости использования параллельной активации логических правил при решении управленческих задач в технических системах / Д. В. Аладин // Информация и образование: границы коммуникаций. – 2023. – № 15(23). – С. 205–208.
72. Чувиков, Д. А. О создании автономных групп роботов на основе миварной системы принятия решений / Д. А. Чувиков, Д. В. Аладин, О. О. Варламов // Информация и образование: границы коммуникаций. – 2022. – № 14(22). – С. 217–220.
73. Аладин, Д. В. Применение автоматического построения миварных баз знаний при решении управленческих задач в технических системах / Д. В. Ала-

- дин // Информация и образование: границы коммуникаций. – 2023. – № 15(23). – С. 201–204.
74. Fikes, R. E. STRIPS: A new approach to the application of theorem proving to problem solving / R. E. Fikes, N. J. Nilsson // Artificial Intelligence. – 1971. – Vol. 2, № 3–4. – P. 189–208.
75. Pednault, E. P. D. ADL: Exploring the Middle Ground between STRIPS and the Situation Calculus / E. P. D. Pednault // International Conference on Principles of Knowledge Representation and Reasoning. – San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1989. – P. 324–332.
76. PDDL - the Planning Domain Definition Language : Tech Report CVC TR-98-003/DCS TR-1165 / D. McDermott, M. Ghallab, A. Howe [et al.]. – 1998.
77. Ghallab, M. Automated Planning and Acting / M. Ghallab, D. Nau, P. Traverso. – 1st ed. – Cambridge University Press, 2016.
78. The Challenges of Real-Time AI / D. J. Musliner, J. A. Hendler, A. K. Agrawala [et al.] // Computer. – 1995. – Vol. 28, № 1. – P. 58–66.
79. Осипов, Г. С. Методы искусственного интеллекта. – Москва: Физматлит, 2011. – 295 с.
80. Sabbadin, R. Planning in Artificial Intelligence / R. Sabbadin, F. Teichteil-Königsbuch, V. Vidal // A Guided Tour of Artificial Intelligence Research / ed. by P. Marquis, O. Papini, H. Prade. – Cham: Springer International Publishing, 2020. – P. 285–312.
81. Hunt, E. GPS: A Case Study in Generality and Problem Solving / E. Hunt, G. W. Ernst // The American Mathematical Monthly. – 1971. – Vol. 78, № 8. – P. 923.
82. Simon, H. A. The Sciences of the Artificial / H. A. Simon. – MIT Press, 1969.
83. Hendler, J. Artificial Intelligence Planning Systems / J. Hendler. – Elsevier, 1992.
84. Болотова, Л. С. Системы искусственного интеллекта: модели и технологии, основанные на знаниях / Л. С. Болотова. – Москва: Финансы и статистика, 2012. – 663 с.
85. Аладин, Д. В. Понятие знака и его использование при построении онтологий / Д. В. Аладин // Молодежный научно-технический вестник. – 2016. – № 7. – С. 12.
86. Рыбина, Г. В. Методы и средства интеллектуального планирования: применение для управления процессами построения интегрированных эксперт-

- ных систем / Г. В. Рыбина, Ю. М. Блохин // Искусственный интеллект и принятие решений. – 2015. – № 1. – С. 75–93.
87. McCarthy, J. Formalization of STRIPS in Situation Calculus / J. McCarthy // Citeseer, Tech. Rep. – Citeseer, 1985.
 88. Sussman, G. J. A Computational Model of Skill Acquisition / G. J. Sussman. – 1973.
 89. Tate, A. Generating Project Networks / A. Tate // International Joint Conference on Artificial Intelligence. – 1977. – P. 888–893.
 90. Kumar, V. Algorithms for Constraint-Satisfaction Problems: A Survey / V. Kumar // AI Magazine. – 1992. – Vol. 13, № 1. – P. 32.
 91. Blum, A. L. Fast Planning through Planning Graph Analysis / A. L. Blum, M. L. Furst // Artificial intelligence. – Elsevier, 1997. – Vol. 90, № 1–2. – P. 281–300.
 92. Kautz, H. Planning as Satisfiability / H. Kautz, B. Selman // Proceedings of the 10th European Conference on Artificial Intelligence : ECAI '92. – USA: John Wiley & Sons, Inc., 1992. – Vol. 359–363.
 93. Ayunts, E. Task Planning in «Block World» with Deep Reinforcement Learning / E. Ayunts, A. I. Panov // Biologically Inspired Cognitive Architectures (BICA) for Young Scientists / ed. by A. V. Samsonovich, V. V. Klimov. – Cham: Springer International Publishing, 2018. – Vol. 636. – P. 3–9.
 94. Aineto, D. Learning STRIPS Action Models with Classical Planning / D. Aineto, S. Jiménez, E. Onaindia // ICAPS. – 2018. – Vol. 28. – P. 399–407.
 95. Švaco, M. A Reinforcement Learning Based Algorithm for Robot Action Planning / M. Švaco, B. Jerbić, M. Polančec, F. Šuligoj // Advances in Service and Industrial Robotics / ed. by N. A. Aspragathos, P. N. Koustoumpardis, V. C. Moulianitis. – Cham: Springer International Publishing, 2019. – Vol. 67. – P. 493–503.
 96. Plan Explicability and Predictability for Robot Task Planning / Y. Zhang, S. Sreedharan A. Kulkarni [et al.] // 2017 IEEE International Conference on Robotics and Automation (ICRA). – Singapore: IEEE, 2017. – P. 1313–1320.
 97. Shen, W. Learning Domain-Independent Planning Heuristics with Hypergraph Networks / W. Shen, F. Trevizan, S. Thiébaux // Proceedings International Conference on Automated Planning and Scheduling, ICAPS / ed. by J. C. Beck, O. Buffet, J. Hoffmann [et al.]. – AAAI press, 2020. – Vol. 30(1). – P. 574–584.

98. Osipov, G. S. Behavior Control as a Function of Consciousness. II. Synthesis of a Behavior Plan / G. S. Osipov, A. I. Panov, N. V. Chudova // *Journal of Computer and Systems Sciences International*. – 2015. – Vol. 54, № 6. – P. 882–896.
99. Panov, A. I. Behavior Planning of Intelligent Agent with Sign World Model / A. I. Panov // *Biologically Inspired Cognitive Architectures*. – 2017. – Vol. 19. – P. 21–31.
100. Sousa, A. R. Toward Automated Planning Algorithms Applied to Production and Logistics / A. R. Sousa, J. J. P. Z. S. Tavares // *IFAC Proceedings Volumes*. – 2013. Vol. 46, № 24. – P. 165–170.
101. Bylander, T. The Computational Complexity of Propositional STRIPS Planning / T. Bylander // *Artificial Intelligence*. – 1994. – Vol. 69, № 1–2. – P. 165–204.
102. Gupta, N. On the Complexity of Blocks-World Planning / N. Gupta, D. S. Nau // *Artificial Intelligence*. – 1992. – Vol. 56, № 2–3. – P. 223–254.
103. Chenoweth, S. V. On the NP-Hardness of Blocks World / S. V. Chenoweth // *AAAI Conference on Artificial Intelligence*. – California: The AAAI Press, 1991. – P. 623–628.
104. Трофимов, И. В. Анализ сложности задач для классического домена планирования Blocks World / И. В. Трофимов // *Труды VIII Международной Конференции «Интеллектуальный Анализ Информации» ИАИ-2008*. – Киев: Просвіта, 2008. – С. 492–501.
105. A Review of Learning Planning Action Models / A. Arora, H. Fiorino, D. Pellier [et al.] // *The Knowledge Engineering Review*. – 2018. – Vol. 33. – P. e20.
106. Task Planning in Robotics: An Empirical Comparison of PDDL- and ASP-based Systems / Y. Jiang, S. Zhang, P. Khandelwal, P. Stone // *Frontiers of Information Technology & Electronic Engineering*. – 2019. – Vol. 20, № 3. – P. 363–373.
107. Clingo = ASP + Control: Preliminary Report / M. Gebser, R. Kaminski, B. Kaufmann, T. Schaub // *ArXiv*. – 2014. – Vol. abs/1405.3694.
108. Richter, S. LAMA 2008 and 2011 / S. Richter, M. Westphal, M. Helmert. – 2011.
109. Чуви́ков, Д. А. Имитационное моделирование как средство представления дополнительной информации для принятия решений человеком / Д. А. Чуви́ков // *Промышленные АСУ и контроллеры*. – 2017. – № 4. – С. 44–53.
110. Варламов, О. О. Миварные технологии как средство создания систем автоматизации разумной деятельности человека / О. О. Варламов, Д. А. Чуви́ков

- // Автоматизация и управление в технических системах. – 2016. – № 1(18). – С. 13.
111. A Methodology for Encoding Regulatory Rules / Н. Bhuiyan, F. Olivieri, G. Governatori // CEUR Workshop Proceedings / ed. by G. Casini, G. Casini, L. Di Caro [et al.]. – CEUR-WS, 2020. – Vol. 2632.
112. Аладин, Д. В. О создании системы верификации миварных моделей знаний / Д. В. Аладин // Интеллектуальные технологии и проблемы математического моделирования / под ред. Б. В. Соболев, Е.В. Рашидова, Н. Н. Венцов и др. – Ростов-на-Дону: Донской государственный технический университет, 2019. – С. 48–49.
113. Габдрахманов, И. Н. Планирование задач в сложноструктурированных ситуациях : специальность 05.13.01 “Системный анализ, управление и обработка информации (по отраслям)” : диссертация на соискание ученой степени кандидата технических наук / Габдрахманов Ильшат Накипович. – Ижевск, 2006. – 180 с.
114. Алимов, А. А. Управление поведением многозадачных интеллектуальных агентов в системах реального времени : специальность 05.13.01 “Системный анализ, управление и обработка информации (по отраслям)” : диссертация на соискание ученой степени кандидата технических наук / Алимов Александр Александрович. – Волгоград, 2017. – 111 с.
115. Юдинцев, Б. С. Нейросетевая система планирования траекторий для группы мобильных роботов : специальность 05.13.01 “Системный анализ, управление и обработка информации (по отраслям)” : диссертация на соискание ученой степени кандидата технических наук / Юдинцев Богдан Сергеевич, 2020. – 192 с.
116. Чинь Суан Лонг. Методы построения интеллектуальных систем планирования перемещения мобильного робота в неизвестной среде : специальность 05.02.05 “Роботы, мехатроника и робототехнические системы” : диссертация на соискание ученой степени кандидата технических наук / Чинь Суан Лонг. – Новочеркасск, 2010. – 139 с.
117. Блохин, Ю. М. Разработка интеллектуальной программной среды для построения интегрированных экспертных систем : специальность 05.13.11 “Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей” : диссертация на соискание ученой степени

- кандидата технических наук / Блохин Юрий Михайлович. – Москва, 2018. – 164 с.
118. Панферов, А. А. Автоматическое конструирование алгоритмов функционирования мультиагентной робототехнической системы в программной среде Wi!Mi / А. А. Панферов, Е. А. Жданович // Big Data and Advanced Analytics. – 2016. – № 2. – С. 212–217.
119. Варламов, О. О. Основы создания миварных экспертных систем / О. О. Варламов. – Москва : ИНФРА-М, 2021. – 267 с.
120. Варламов, О. О. Создание Больших Знаний и расширение областей применения миварных технологий логического искусственного интеллекта / О. О. Варламов // Информационные и математические технологии в науке и управлении. – 2023. – № 4(32). – С. 30–41.
121. Варламов, О. О. Миварные базы данных и правил / О. О. Варламов. – Москва : ИНФРА-М, 2021. – 351 с.
122. Васюгова, С. А. О возможностях использования миварных технологий представления знаний и обработки данных для групп роботов и гетерогенных мультиагентных систем и сред / С. А. Васюгова, О. О. Варламов // Известия Кабардино-Балкарского научного центра РАН. – 2011. – № 1-1(39). – С. 65–70.
123. Варламов, О. О. Эволюционные базы данных и знаний для адаптивного синтеза интеллектуальных систем. Миварное информационное пространство / О. О. Варламов. – Москва : Радио и связь, 2002. – 286 с.
124. Варламов, О. О. Системный анализ и синтез моделей данных и методы обработки информации в самоорганизующихся комплексах оперативной диагностики : специальность 05.13.01 “Системный анализ, управление и обработка информации (по отраслям)” : диссертация на соискание ученой степени доктора технических наук / Варламов Олег Олегович. – Москва, 2003. – 307 с.
125. Практикум по созданию миварных экспертных систем / О. О. Варламов, М. О. Чибирова, А. М. Хадиев [и др.]. – Москва : МИВАР, 2016. – 184 с.
126. Панферов, А. А. Применение миварного подхода при создании системы управления автономными интеллектуальными роботами / А. А. Панферов, Е. А. Жданович, К. А. Юфимычев // XXVII Международная инновационно-ориентированная конференция молодых ученых и студентов (МИКМУС -

- 2015) : сб. трудов конф. – Москва: Федеральное государственное бюджетное учреждение науки Институт машиноведения им. А.А. Благонравова Российской академии наук, 2015. – С. 337–340.
127. Вычисление произвольных алгоритмов функционирования сервисных роботов на основе миварного подхода / Е. А. Жданович, П. К. Чернышев, К. А. Юфимычев [и др.] // Радиопромышленность. – 2015. – № 3. – С. 226–242.
128. Shadrin, S. S. Experimental autonomous road vehicle with logical artificial intelligence / S. S. Shadrin, A. M. Ivanov, O. O. Varlamov // Journal of Advanced Transportation. – 2017. – Vol. 2017. – P. 2492765.
129. Чувиков, Д. А. Модели и алгоритмы реконструкции и экспертизы аварийных событий дорожно-транспортных происшествий : специальность 05.13.01 “Системный анализ, управление и обработка информации (по отраслям)” : диссертация на соискание ученой степени кандидата технических наук / Чувиков Дмитрий Алексеевич. – Москва, 2017. – 318 с.
130. Кривошеев, О. В. Технология распределения ресурсов производственных систем в условиях неполноты данных для высокотехнологичных отраслей промышленности : специальность 05.13.01 “Системный анализ, управление и обработка информации (по отраслям)” : диссертация на соискание ученой степени кандидата технических наук / Кривошеев Олег Викторович. – Саров, 2022. – 168 с.
131. Кривошеев, О. В. Применение МЭС в условиях неполноты данных “без предыстории” для решения задач объемного, объемно календарного и сменно-суточного планирования / О. В. Кривошеев // Мивар’22 : сб. научных статей. – Москва: ИНФРА-М, 2022. – С. 406–413.
132. Санду, Р. А. Многомерная эволюционная прикладная автоматизированная информационная система поддержки принятия решений для управления инновационными ресурсами химической и нефтехимической промышленности России : специальность 05.25.05 “Информационные системы и процессы” : диссертация на соискание ученой степени доктора технических наук / Санду Роман Александрович. – Москва, 2011. – 314 с.
133. Патент № 2607995 С Российская Федерация, МПК G06F 17/20. Автоматизированное построение маршрута логического вывода в миварной базе знаний : № 2015104624 : заявл. 11.02.2015 : опубл. 11.01.2017 / О. О. Вар-

- ламов, А. М. Хадиев, М. О. Чибирова [и др.] ; заявитель ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ “МИВАР”.
134. Владимиров, А. Н. Развитие на основе многодольных графов миварных логико-вычислительных сетей для реализации правил выбора “если.., то.., иначе...” / А. Н. Владимиров // Труды Научно-исследовательского института радио. – 2010. – № 3. – С. 35–44.
 135. Владимиров, А. Н. О возможности создания активной миварной интернет-энциклопедии и развития миварных сетей на основе многомерных бинарных матриц для одновременной эволюционной обработки более 10 000 правил в реальном времени / А. Н. Владимиров // Труды Научно-исследовательского института радио. – 2010. – № 3. – С. 45–52.
 136. Варламов, О. О. Большие Знания: анализ расширения областей применения миварных технологий логического ИИ / О. О. Варламов // Искусственный интеллект в автоматизированных системах управления и обработки данных : сб. статей II Всероссийской научной конференции. – Москва: Издательский дом КДУ, Добросвет, 2023. – Т. 3 – С. 10–14.
 137. Варламов, О. О. О требованиях к подготовке исходных данных для миварных систем принятия решений роботов / О. О. Варламов // Искусственный интеллект в автоматизированных системах управления и обработки данных : сб. статей II Всероссийской научной конференции. – Москва: Издательский дом КДУ, Добросвет, 2023. – Т. 3. – С. 15–20.
 138. Машиностроительный искусственный интеллект: разработка методик логического искусственного интеллекта для систем полного жизненного цикла изделий / А. В. Трищенко, В. Г. Осипов, Е. С. Лялин [и др.] // Машиностроительные технологические системы : сб. трудов Международной научно-технической конференции. – Ростов-на-Дону: Донской государственный технический университет, 2022. – С. 403–409.
 139. Машиностроительный искусственный интеллект: разработка миварной методики для поиска траекторий робота и решения оптимизационных задач / А. А. Коценко, А. В. Герасименко, А. В. Калашникова [и др.] // Машиностроительные технологические системы : сб. трудов Международной научно-технической конференции. – Ростов-на-Дону: Донской государственный технический университет, 2022. – С. 363–370.

140. Алгоритм планирования траектории робота на основе технологии обнаружения объектов с использованием МЭС / Ц. Шэнь, Ш. Гун, Ч. Лю [и др.] // МИВАР'24 : сб. научных статей. – Москва: ИНФРА-М, 2024. – С. 367–373.
141. Варламов, О. О. Практическая реализация линейной вычислительной сложности логического вывода на правилах “ЕСЛИ-ТО” в миварных сетях и обработка более трех миллионов правил / О. О. Варламов // Автоматизация и управление в технических системах. – 2013. – № 1(3). – С. 60–97.
142. Обработка текста с помощью LLM для автоматической генерации МБЗ / А. В. Андреев, Г. В. Тураев, А. А. Коценко [и др.] // МИВАР'24 : сб. научных статей. – Москва: ИНФРА-М, 2024. – С. 389–396.
143. Гришин, И. А. Многоклассовая классификация видов деревьев по лидарным данным с использованием МЭС / И. А. Гришин, О. О. Варламов, В. И. Терехов // Интеллектуальные системы и информационные технологии - 2024 : труды Международного научно-технического конгресса. – Таганрог: Ступина С.А., 2024. – Т. 2. – С. 20–25.
144. Комплексный ИИ: семантическая сегментация растительности в плотном облаке точек / Н. К. Демирев, М. Р. Камалов, Д. А. Васильев, О. О. Варламов // МИВАР'24 : сб. научных статей. – Москва: ИНФРА-М, 2024. – С. 287–292.
145. Использование многомерной открытой гносеологической активной сети (MOGAN) в гибридных интеллектуальных информационных системах / О. О. Варламов, В. И. Терехов, А. А. Сухобоков, Ю. Е. Гапанюк // Гибридные и синергетические интеллектуальные системы : материалы VI Всероссийской Поспеловской конференции с международным участием. – Калининград: Балтийский федеральный университет имени Иммануила Канта, 2022. – С. 151–158.
146. Метаграфовый подход для описания гибридных интеллектуальных информационных систем / В. М. Черненький, Ю. Е. Гапанюк, Г. И. Ревунков [и др.] // Прикладная информатика. – 2017. – Т. 12, № 3(69). – С. 57–79.
147. Подход к разработке модели цифрового двойника водителя высокоавтоматизированного транспортного средства на основе гибридной интеллектуальной информационной системы / Д. В. Аладин, Б. С. Горячкин, О. О. Варламов [и др.] // Современная наука: актуальные проблемы теории и практики. Серия: Естественные и технические науки. – 2022. – № 2-2. – С. 34–40.

148. Гапанюк, Ю. Е. Конспект лекций по спецкурсу “Гибридные интеллектуальные информационные системы на основе метаграфового подхода” / Ю. Е. Гапанюк. – Москва: ООО “Издательство “Спутник+”, 2018. – 56 с.
149. Гапанюк, Ю. Е. Описание базовых элементов метаграфовой модели с использованием гранулярно-протографового подхода / Ю. Е. Гапанюк // Двадцать первая Национальная конференция по искусственному интеллекту с международным участием (КИИ-2023) : труды конференции. – Смоленск: Принт-Экспресс, 2023. – С. 68–80.
150. Представление метаграфовой модели в виде категории / С. С. Винников, А. Н. Нардид, Г. И. Ревунков, Ю. Е. Гапанюк // Динамика сложных систем - XXI век. – 2023. – Т. 17, № 3. – С. 72–77.
151. Гапанюк, Ю. Е. Метаграфовое описание синергетических холархий / Ю. Е. Гапанюк, Ю. Т. Каганов, М. Г. Кузьмина // Гибридные и синергетические интеллектуальные системы : сб. статей по материалам научной VII Всероссийской Поспеловской конференции. – Калининград, Санкт-Петербург: Русская христианская гуманитарная академия им. Ф.М. Достоевского, 2024. – С. 351–360.
152. Федюкин, Д. А. Отображение наиболее популярных нотаций бизнес-процессов в метаграф / Д. А. Федюкин, С. Д. Кудрявцев, Ю. Е. Гапанюк // Искусственный интеллект в автоматизированных системах управления и обработки данных : сб. статей II Всероссийской научной конференции. – Москва: Издательский дом КДУ, “Добросвет”, 2024. – Т. 4. – С. 127–131.
153. Зонова, А. В. Управление проектами на новом уровне: возможности метаграфовых баз данных и VI-инструментов / А. В. Зонова, Ю. Е. Гапанюк, В. Л. Зонов // Управление проектами : сб. статей по материалам III Всероссийской научной конференции. – Магнитогорск: Магнитогорский государственный технический университет им. Г.И. Носова, 2025. – С. 73–80.
154. Метаграфовое описание семантики изображения / И. А. Ерохин, Е. А. Белоусов, Р. Р. Сафин [и др.] // Естественные и технические науки. – 2021. – № 7(158). – С. 179–182.
155. Каганов, Ю. Т. Символическая динамика, метаграфы и интеллект / Ю. Т. Каганов, Ю. Е. Гапанюк // Нелинейная динамика в когнитивных исследованиях - 2019 : труды Шестой Всероссийской конференции. – Нижний Новгород: Институт прикладной физики, 2019. – С. 78–79.

156. Модуль перевода знаний из плоских графов в метаграфовое пространство / Д. А. Федюкин, С. Д. Кудрявцев, Д. А. Храмцов, Ю. Е. Гапанюк // Естественные и технические науки. – 2021. – № 6(157). – С. 125–128.
157. Дунин, И. В. Реализация метаграфовых агентов на основе технологий больших данных / И. В. Дунин, Ю. Е. Гапанюк // Искусственный интеллект в автоматизированных системах управления и обработки данных : Сборник статей Всероссийской научной конференции. – Москва: Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет), 2022. – Т. 1. – С. 337–347.
158. The Implementation of Metagraph Agents Based on Functional Reactive Programming / V. Chernenkiy, Yu. Gapanjuk, A. Nardid, N. Todosiev // Conference of Open Innovations Association, FRUCT. – 2020. – № 26. – P. 74–81.
159. Федоренко, Ю. С. Построение адаптивных моделей на основе многоуровневой нейронной сети с использованием метаграфового подхода / Ю. С. Федоренко, Ю. Е. Гапанюк // НЕЙРОИНФОРМАТИКА-2016 : сб. научных трудов. – Москва: Национальный исследовательский ядерный университет “МИФИ”, 2016. – Т. Часть 1 – С. 214–224.
160. Особенности реализации метаграфовых агентов на основе нечеткого логического вывода / А. А. Ветошкин, Н. С. Грунин, А. А. Тахтамышева [и др.] // Динамика сложных систем - XXI век. – 2023. – Т. 17, № 3. – С. 40–50.
161. Особенности реализации метаграфовых агентов на основе продукционного подхода / Ю. А. Слободчикова, А. В. Молчанов, А. В. Зонова [и др.] // Динамика сложных систем - XXI век. – 2023. – Т. 17, № 3. – С. 78–84.
162. Методика преобразования из миварной модели представления знаний в метаграфовую / Е. Д. Соболева, И. А. Попова, Д. А. Макаров [и др.] // Мивар’22 : сб. научных статей. – Москва : Инфра-М, 2022. – С. 228–235.
163. Методика преобразования из метаграфовой модели представления знаний в миварную / Е. Д. Соболева, И. А. Попова, Д. А. Макаров [и др.] // Мивар’22 : сб. научных статей. – Москва : Инфра-М, 2022. – С. 220–227.
164. Каганов, Ю. Т. Гибридизация метаграфового подхода и подхода на основе символической динамики для разработки интеллектуальных информационных систем / Ю. Т. Каганов, Г. И. Ревунков, Ю. Е. Гапанюк // Гибридные и синергетические интеллектуальные системы : материалы IV Всероссийской

- Поспеловской конференции с международным участием. – Калининград: Издательство Балтийского федерального университета им. Иммануила Канта, 2018. – С. 192–199.
165. The multidimensional open gnoseological active network (MOGAN) approach for creating an artificial general intelligence / O. O. Varlamov, L. E. Adamova, Y. E. Garanyuk [et al.] // *Нейроинформатика-2021*. – 2021. – Р. 11–20.
166. Автоматическое создание миварных баз знаний с помощью выборки правил из текстовых инструкций с применением нейросетей / А. Желтова, И. Р. Ваксина, Д. Ю. Уткин [и др.] // *Информация и образование: границы коммуникаций*. – 2024. – № 16(24). – С. 216–219.
167. Автоматизированное создание миварных баз знаний на основе наборов формул по информатике / А. Д. Пасатюк, Ф. А. Аникин, А. Р. Якубов [и др.] // *Информация и образование: границы коммуникаций*. – 2024. – № 16(24). – С. 238–242.
168. ECMA-262, 16th edition, June 2025 ECMAScript® 2025 Language Specification [Электронный ресурс] // ECMA International. URL: <https://262.ecma-international.org/16.0/index.html> (дата обращения: 15.12.2025).
169. Nilsson, N. J. Principles of Artificial Intelligence / N. J. Nilsson // *Symbolic Computation*. – 1980.
170. Estlin, T. A. Integrating Explanation-Based and Inductive Learning Techniques to Acquire Search-Control for Planning: AI96-250 / T. A. Estlin. – Austin, TX: University of Texas, 1996.
171. Программа “УДАВ”: реализация линейной вычислительной сложности матричного метода поиска маршрута логического вывода на основе миварной сети правил / А. В. Носов, А. Н. Владимиров, Т. С. Потапова [и др.] // *Искусственный интеллект*. – 2009. – № 3. – С. 443–448.
172. Программный комплекс “УДАВ”: практическая реализация активного обучаемого логического вывода с линейной вычислительной сложностью на основе миварной сети правил / А. Н. Владимиров, О. О. Варламов, А. В. Носов, Т. С. Потапова // *Труды Научно-исследовательского института радио*. – 2010. – № 1. – С. 108–116.
173. Варламов, О. О. Мивары: 25 Лет Создания Искусственного Интеллекта / О. О. Варламов. – Москва: Aegitas, 2016. – 355 с.

174. Хадиев, А. М. Разработка и практическая реализация миварной машины логического вывода / А. М. Хадиев // Радиопромышленность. – 2015. – № 3. – С. 79–89.
175. Varlamov, O. O. Wi!Mi expert system shell as the novel tool for building knowledge-based systems with linear computational complexity / O. O. Varlamov // International Review of Automatic Control. – 2018. – Vol. 11, № 6. – P. 314–325.
176. Проект “Миварная Активная Энциклопедия” / Р. Г. Байбарин, М. А. Кучеренко, Н. В. Тюлькина [и др.] // Мивар’22 : сб. научных статей. – Москва : ИНФРА-М, 2022. – С. 178–186.
177. Варламов, О. О. О применении миварных технологий для интеллектуального планирования в пространстве состояний / О. О. Варламов // Модели мышления и интеграция информационно-управляющих систем (ММИИУС-2018) : материалы второй Международной научной конференции, посвящённой 25-летию юбилею Кабардино-Балкарского научного центра Российской академии наук. – Нальчик: Издательство КБНЦ РАН, 2018. – С. 36–41.
178. Чувиков, Д. А. Создание платформы реконструкции и экспертизы аварийных событий дорожно-транспортных происшествий на базе логического искусственного интеллекта / Д. А. Чувиков // Труды международного научно-технического конгресса “Интеллектуальные системы и информационные технологии - 2024” (“ИС & ИТ-2024”, “IS&IT’24”) : научное издание в 2-х томах. – Таганрог: Ступина С.А., 2024. – Т. 1 – С. 67–73.
179. Бережная, Л. И. Систематизация и анализ статистической информации с помощью аналитической группировки. Использование показателей вариации в анализе взаимосвязей признаков социально-экономических явлений / Л. И. Бережная, Л. В. Колядов, П. Ф. Тарасенко. – Москва : Нефть и газ, 2014. – 30 с.

ПРИЛОЖЕНИЕ А

ПРИМЕР ПРИМЕНЕНИЯ МОДЕЛЕЙ И МЕТОДОВ ПЛАНИРОВАНИЯ ДЕЙСТВИЙ ДЛЯ РЕШЕНИЯ УПРАВЛЕНЧЕСКОЙ ЗАДАЧИ

С целью демонстрации работы МИПРА рассмотрим пример решения задачи планирования в пространстве состояний в домене «Мир кубиков». Пусть задача планирования T формулируется следующим образом: начальное положение кубиков – $S_H = ((), (), (0, 1))$, целевое положение – $S_C = ((), (0, 1), ())$.

Система МИПРА оснащена графическим пользовательским интерфейсом. Пример ввода задачи планирования представлен на рисунке А.1.

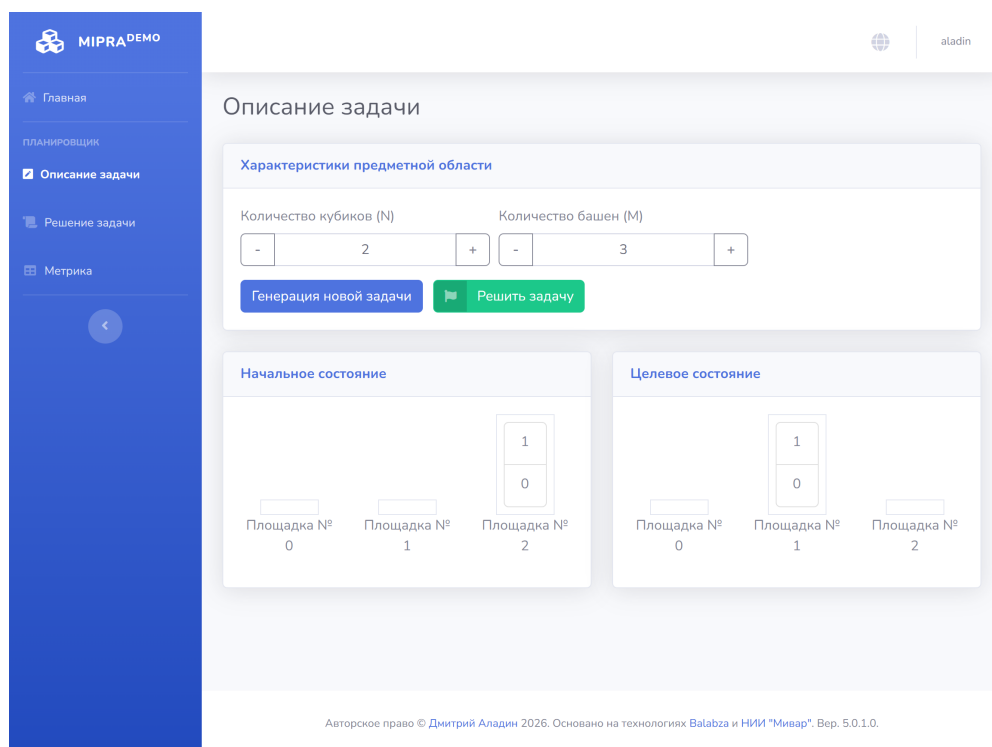


Рисунок А.1 — Ввод задачи планирования в домене «Мир кубиков» с использованием графического интерфейса

На основе сформулированной задачи T генерируется МБЗ для работы с двумя кубиками и тремя доступными площадками для их размещения. Результат

генерации представлен в листинге А.1. Представленный в листинге код является XML-документом, сформированным в соответствии с форматом хранения МБЗ, используемым в программном комплексе КЭСМИ Wi!Mi «Разуматор» версии 2.1. Описание данного формата рассматривается в работе [125].

Листинг А.1 — Пример генерации миварной базы знаний для решения задач в пространстве состояний в домене «Мир кубиков»

```

1 <model id="{u1}"
2   shortName="Cubes World" description="This model was generated automatically."
3   formatXmlVersion="2.0">
4   <class id="u2" shortName="Cubes World" description="">
5     <classes>
6       <class id="u3" shortName="Current state"
7         description="">
8         <classes>
9           <class id="u4" shortName="Cubes"
10            description="">
11            <classes>
12              <class id="u5" shortName="Cube[0]"
13                description="">
14                <parameters>
15                  <parameter id="u6"
16                    shortName="Cube[0].On" description="" type="double"
17                    defaultValue="" />
18                  <parameter id="u7"
19                    shortName="Cube[0].OverFree" description="" type="string"
20                    defaultValue="" />
21                  <parameter id="u8"
22                    shortName="Cube[0].CurrentTower" description=""
23                    type="double" defaultValue="" />
24                  <parameter id="u9"
25                    shortName="Cube[0].IsCurrentTarget" description=""
26                    type="string" defaultValue="" />
27                  <parameter id="u10"
28                    shortName="Cube[0].TargetOn" description="" type="double"
29                    defaultValue="" />
30                  <parameter id="u11"
31                    shortName="Cube[0].TargetTower" description="" type="double"
32                    defaultValue="" />
33                </parameters>
34            </class>
35            <class id="u12" shortName="Cube[1]"
36              description="">
37              <parameters>
38                <parameter id="u13"
39                  shortName="Cube[1].On" description="" type="double"
40                  defaultValue="" />

```

```

41     <parameter id="u14"
42         shortName="Cube[1].OverFree" description="" type="string"
43         defaultValue="" />
44     <parameter id="u15"
45         shortName="Cube[1].CurrentTower" description=""
46         type="double" defaultValue="" />
47     <parameter id="u16"
48         shortName="Cube[1].IsCurrentTarget" description=""
49         type="string" defaultValue="" />
50     <parameter id="u17"
51         shortName="Cube[1].TargetOn" description="" type="double"
52         defaultValue="" />
53     <parameter id="u18"
54         shortName="Cube[1].TargetTower" description="" type="double"
55         defaultValue="" />
56     </parameters>
57 </class>
58 </classes>
59 </class>
60 <class id="u19" shortName="Iterations"
61     description="">
62     <parameters />
63     <classes>
64         <class id="u20"
65             shortName="Iteration[0]" description="">
66             <parameters />
67             <classes>
68                 <class id="u21"
69                     shortName="InitialActions[0]" description="">
70                     <parameters>
71                         <parameter id="u22"
72                             shortName="InitialActions[0].FindTargetCube"
73                             description="" type="double" defaultValue="" />
74                         <parameter id="u23"
75                             shortName="InitialActions[0].PrepareTargetTower"
76                             description="" type="double" defaultValue="" />
77                     </parameters>
78                 </class>
79                 <class id="u24"
80                     shortName="Targets[0]" description="">
81                     <parameters>
82                         <parameter id="u25"
83                             shortName="Targets[0].TargetCube" description=""
84                             type="double" defaultValue="" />
85                         <parameter id="u26"
86                             shortName="Targets[0].TargetTower" description=""
87                             type="double" defaultValue="" />
88                     </parameters>

```

```

89         </class>
90     </classes>
91 </class>
92 <class id="u27"
93     shortName="Iteration[1]" description="">
94     <parameters />
95     <classes>
96         <class id="u28"
97             shortName="InitialActions[1]" description="">
98             <parameters>
99                 <parameter id="u29"
100                     shortName="InitialActions[1].FindTargetCube"
101                     description="" type="double" defaultValue="" />
102                 <parameter id="u30"
103                     shortName="InitialActions[1].PrepareTargetTower"
104                     description="" type="double" defaultValue="" />
105             </parameters>
106         </class>
107         <class id="u31"
108             shortName="Targets[1]" description="">
109             <parameters>
110                 <parameter id="u32"
111                     shortName="Targets[1].TargetCube" description=""
112                     type="double" defaultValue="" />
113                 <parameter id="u33"
114                     shortName="Targets[1].TargetTower" description=""
115                     type="double" defaultValue="" />
116             </parameters>
117         </class>
118     </classes>
119 </class>
120 </classes>
121 </class>
122 <class id="u34" shortName="Towers"
123     description="">
124     <parameters>
125         <parameter id="u35" shortName="M"
126             description="" type="double" defaultValue="" />
127         <parameter id="u36"
128             shortName="CurrentTargetTower" description="" type="double"
129             defaultValue="" />
130     </parameters>
131 </class>
132 </classes>
133 </class>
134 <class id="u37" shortName="Management" description="">
135     <parameters />
136 </classes>

```

```
137 <class id="u38" shortName="Actions"
138   description="">
139   <parameters>
140     <parameter id="u39"
141       shortName="PutCubeOnTargetPlace" description="" type="double"
142       defaultValue="" />
143     <parameter id="u40"
144       shortName="PrepareTargetTower" description="" type="double"
145       defaultValue="" />
146     <parameter id="u41"
147       shortName="FindTargetCube" description="" type="double"
148       defaultValue="" />
149     <parameter id="u42"
150       shortName="MoveCubesFromCurrentTargetCube" description=""
151       type="double" defaultValue="" />
152   </parameters>
153 </class>
154 <class id="u43" shortName="SetValue"
155   description="">
156   <parameters>
157     <parameter id="u44"
158       shortName="CurrentTargetCube" description="" type="double"
159       defaultValue="" />
160     <parameter id="u45"
161       shortName="CurrentTargetTower" description="" type="double"
162       defaultValue="" />
163   </parameters>
164 </class>
165 <class id="u46" shortName="Processing"
166   description="">
167   <parameters>
168     <parameter id="u47" shortName="Start"
169       description="" type="string" defaultValue="" />
170     <parameter id="u48"
171       shortName="Continue" description="" type="string" defaultValue="" />
172     <parameter id="u49" shortName="Stop"
173       description="" type="string" defaultValue="" />
174   </parameters>
175 </class>
176 <class id="u50" shortName="IntermediateTargets"
177   description="">
178   <parameters>
179     <parameter id="u51"
180       shortName="Target[0]" description="" type="string" defaultValue="" />
181     <parameter id="u52"
182       shortName="Target[1]" description="" type="string" defaultValue="" />
183   </parameters>
184 </class>
```

```

185     </classes>
186   </class>
187 </classes>
188 <rules>
189   <rule id="u53" shortName="StartWorking" description=""
190     relation="u54"
191     initId="Start:u47;TargetCube:u25;TargetTower:u26"
192     resultId="CurrentTargetCube:u44;PrepareTargetTower:u40;FindTargetCube:u41;
      CurrentTargetTower:u45" />
193   <rule id="u55" shortName="IterationBeforeCompleting[0]"
194     description="" relation="u56"
195     initId="IsCurrentTarget:u9;CurrentTargetTower:u36;CurrentTower:u8;OverFree:u7"
196     resultId="PutCubeOnTargetPlace:u39" />
197   <rule id="u57" shortName="IterationComplete[0]"
198     description="" relation="u58"
199     initId="IsCurrentTarget:u9;Target:u51;NextTargetCube:u32;NextTargetTower:u33;
      NextActionPrepareTargetTower:u30;NextActionFindTargetCube:u29"
200     resultId="FindTargetCube:u41;SetCurrentTargetTower:u45;SetCurrentTargetCube:u44;
      PrepareTargetTower:u40" />
201   <rule id="u59" shortName="IterationInProgress[0]"
202     description="" relation="u60"
203     initId="IsCurrentTarget:u9;CurrentTargetTower:u36;CurrentTower:u8;OverFree:u7"
204     resultId="MoveCubes:u42" />
205   <rule id="u61" shortName="IterationBeforeCompleting[1]"
206     description="" relation="u56"
207     initId="IsCurrentTarget:u16;CurrentTargetTower:u36;CurrentTower:u15;OverFree:u14"
208     resultId="PutCubeOnTargetPlace:u39" />
209   <rule id="u62" shortName="Finish" description=""
210     relation="u63"
211     initId="TowerComplete1:u52"
212     resultId="Stop:u49" />
213   <rule id="u64" shortName="CheckCubeOnTarget[0]"
214     description="" relation="u65"
215     initId="ControlVariable:u48;On:u6;TargetOn:u10;CurrentTower:u8;TargetTower:u11"
216     resultId="CubeOnTarget:u51" />
217   <rule id="u66" shortName="CheckCubeOnTarget[1]"
218     description="" relation="u65"
219     initId="ControlVariable:u51;On:u13;TargetOn:u17;CurrentTower:u15;TargetTower:u18"
220     resultId="CubeOnTarget:u52" />
221 </rules>
222 </class>
223 <relations>
224   <relation id="u54" shortName="PStartWorking" description=""
225     relationType="prog" inObj="Start:string;TargetCube:double;TargetTower:double"
226     outObj="CurrentTargetCube:double;PrepareTargetTower:double;FindTargetCube:double;
      CurrentTargetTower:double">var
227     Start,
228     TargetCube, TargetTower,

```

```

229     CurrentTargetCube, CurrentTargetTower,
230     PrepareTargetTower, FindTargetCube;
231     if ((Start == "true"))
232     {
233         CurrentTargetCube = TargetCube;
234         CurrentTargetTower = TargetTower;
235         PrepareTargetTower = 0;
236         FindTargetCube = 1;
237     }</relation>
238 <relation id="u56" shortName="PIterationBeforeCompleting"
239     description="" relationType="ifclause"
240     inObj="IsCurrentTarget:string;OverFree:string;CurrentTargetTower:double;
           CurrentTower:double"
241     outObj="PutCubeOnTargetPlace:double">if ((IsCurrentTarget ==
           "true")&&(OverFree
242     == "true")&&(CurrentTargetTower != CurrentTower)) {PutCubeOnTargetPlace = 0;}
243     else {PutCubeOnTargetPlace = -1;}</relation>
244 <relation id="u58" shortName="PIterationComplete"
245     description="" relationType="prog"
246     inObj="IsCurrentTarget:string;Target:string;NextTargetCube:double;
           NextActionPrepareTargetTower:double;
           NextTargetTower:double;NextActionFindTargetCube:double"
247     outObj="FindTargetCube:double;SetCurrentTargetTower:double;SetCurrentTargetCube:
           double;PrepareTargetTower:double">var
248     IsCurrentTarget, Target,
249     NextActionFindTargetCube, NextActionPrepareTargetTower,
250     NextTargetCube, NextTargetTower,
251     SetCurrentTargetCube, SetCurrentTargetTower,
252     FindTargetCube, PrepareTargetTower;
253     if ((IsCurrentTarget == "true")&&(Target == "true"))
254     {
255         SetCurrentTargetCube = NextTargetCube;
256         SetCurrentTargetTower = NextTargetTower;
257         FindTargetCube = NextActionFindTargetCube;
258         PrepareTargetTower = NextActionPrepareTargetTower;
259     }
260     else {
261         SetCurrentTargetCube = -1;
262         SetCurrentTargetTower = -1;
263         FindTargetCube = -1;
264         PrepareTargetTower = -1;
265     }
266 </relation>
267 <relation id="u60" shortName="PIterationInProgress"
268     description="" relationType="ifclause"
269     inObj="IsCurrentTarget:string;OverFree:string;CurrentTargetTower:double;
           CurrentTower:double"
270     outObj="MoveCubes:double">if ((IsCurrentTarget == "true")&&(OverFree ==

```

```

271     "false")&&(CurrentTargetTower != CurrentTower)) {MoveCubes = 0;} else
      {MoveCubes
272     = -1;}</relation>
273 <relation id="u63" shortName="PFinish" description=""
274     relationType="ifclause" inObj="TowerComplete1:string" outObj="Stop:string">if
275     ((TowerComplete1 == "true")) {Stop = "true";} else {Stop = "false";}</relation>
276 <relation id="u65" shortName="PCheckTarget" description=""
277     relationType="ifclause"
278     inObj="ControlVariable:string;On:double;TargetOn:double;CurrentTower:double;
      TargetTower:double"
279     outObj="CubeOnTarget:string">if ((ControlVariable == "true") && (On ==
      TargetOn)
280     && (CurrentTower == TargetTower)) {CubeOnTarget = "true";} else
      {CubeOnTarget =
281     "false";}</relation>
282 </relations>
283 </model>

```

Из листинга А.1 видно, что МБЗ содержит 33 параметра и 8 правил. Элементы class XML-документа образуют множества МБЗ, описание которых приведено в параграфе 3.1. Визуализация иерархии данных множеств представлена на рисунке А.2.

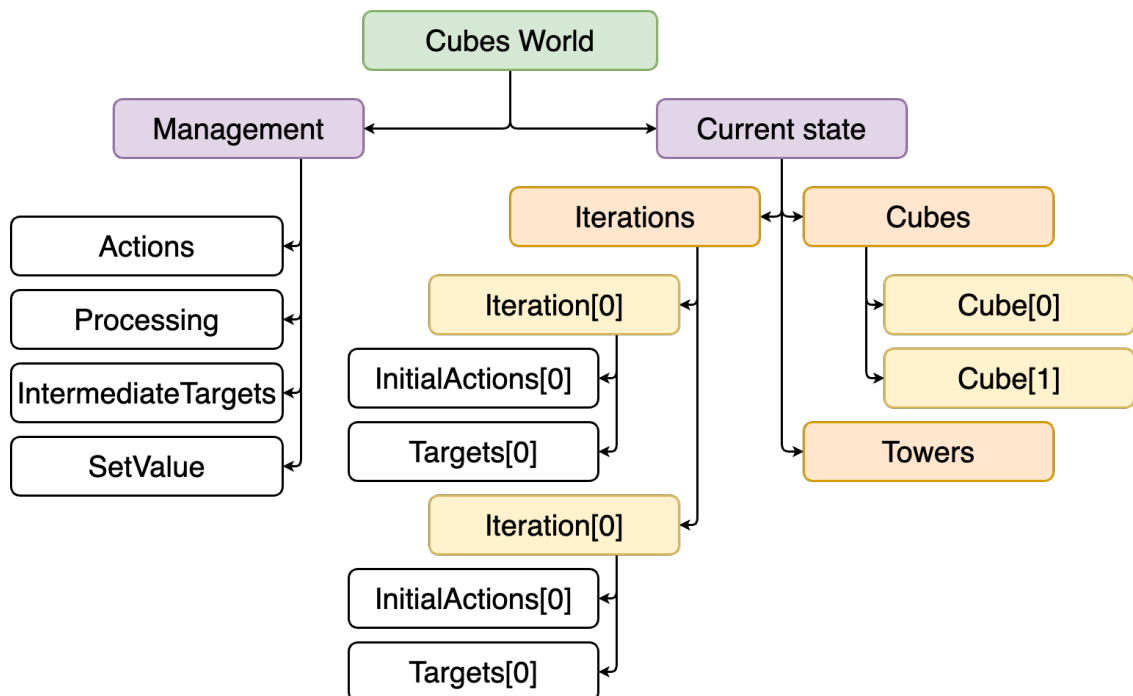


Рисунок А.2 — Иерархия множеств параметров миварной базы знаний для решения задач планирования в домене «Мир кубиков» при наличии двух кубиков в среде

Для каждого кубика среды в миварной сети имеется его пространственное представление. Рассмотрим в качестве примера кубик № 0. Описание его поло-

жения в пространстве задаётся множеством `Cube [0]`, включающим следующие параметры:

- `Cube [0] .On` – номер кубика (значение -1 соответствует размещению непосредственно на площадке стола), на котором расположен кубик № 0 в текущем состоянии;
- `Cube [0] .OverFree` – флаг, указывающий на отсутствие других кубиков над кубиком № 0 в текущем состоянии;
- `Cube [0] .CurrentTower` – номер башни, в которой в текущий момент размещён кубик № 0;
- `Cube [0] .IsCurrentTarget` – флаг, указывающий на то, что кубик № 0 является целевым на текущем этапе планирования;
- `Cube [0] .TargetOn` – номер кубика (значение -1 соответствует размещению непосредственно на площадке стола), на котором кубик № 0 должен быть размещён в целевом состоянии;
- `Cube [0] .TargetTower` – номер башни, в которой кубик № 0 должен находиться в целевом состоянии.

Исходя из определений прототипов и шаблонов параметров и правил миварной сети, описание которых представлено в таблице 3.1, рассмотрим их конкретные реализации на основе листинга А.1:

- В системе МИПРА не используются прототипы параметров, что подтверждается наличием пустых значений атрибута `defaultValue` у элементов `parameter`. Все параметры рассматриваемой МБЗ являются шаблонными.
- Элемент `rule` с атрибутом `shortName="CheckCubeOnTarget [1]"` представляет собой правило, предназначенное для проверки достижения целевого положения кубиком № 1. Поскольку описание каждого кубика осуществляется единообразно, алгоритм проверки достижения целевого положения для кубика является заранее определённым. В связи с этим тело правила (элемент `relation` с атрибутом `shortName="PCheckTarget"`) известно до этапа генерации МБЗ.
- Элемент `rule` с атрибутом `shortName="Finish"` представляет собой правило, предназначенное для определения признака достижения цели задачи T . Данное правило является шаблонным, поскольку его тело (элемент `relation` с атрибутом `shortName="PFinish"`) и набор входных параметров формируются в процессе генерации МБЗ. На вход правила подаются призна-

ки достижения промежуточных целей. Поскольку подцели взаимосвязаны, в рамках данного правила осуществляется проверка достижения целевого состояния для тех кубиков, которые в целевом состоянии должны располагаться на вершинах башен. Так как количество башен является переменным, число входных параметров данного правила определяется на основе конкретной задачи планирования. Алгоритм действия правила также зависит от входных параметров.

В соответствии с алгоритмом, рассмотренным в параграфе 3.3.2, МИПРА пошагово осуществляет решение задачи планирования. Рассмотрим эти шаги подробнее.

На шаге № 0 система на основе логического вывода по МБЗ формирует промежуточный план, состоящий из действия $\text{FindTargetCube}(0)$ (поиск текущего целевого кубика № 0). При этом состояние ПрО остаётся неизменным:

$$S_0 = ((), (), (0, 1)). \quad (\text{A.1})$$

После выполнения шага № 0 состояние параметров в рабочей памяти будет выглядеть так, как показано в таблице А.1.

На шаге № 1 к исполнению предлагается действие $\text{MoveCubes}(2, 0, 1)$ (перемещение одного кубика с вершины башни № 2 на площадку № 0). Состояние ПрО после выполнения данного шага принимает следующий вид:

$$S_1 = ((1), (), (0)). \quad (\text{A.2})$$

Состояние рабочей памяти планировщика после выполнения шага № 1 представлено в таблице А.2.

На шаге № 2 к исполнению предлагается действие $\text{MoveCube}(2, 1)$ (перемещение кубика с площадки № 2 на площадку № 1). Состояние ПрО после выполнения данного шага принимает следующий вид:

$$S_2 = ((1), (0), ()). \quad (\text{A.3})$$

Состояние рабочей памяти планировщика после выполнения шага № 2 представлено в таблице А.3.

На шаге № 3 формируется план, состоящий из последовательности дей-

ствий $\text{CubeOnTarget}(0)$ (фиксация достижения кубиком № 0 целевого положения) и $\text{FindTargetCube}(1)$ (поиск целевого кубика № 1). Состояние ПрО после выполнения данного шага остаётся неизменным:

$$S_3 = ((1), (0), ()). \quad (\text{A.4})$$

Состояние рабочей памяти планировщика после выполнения шага № 3 представлено в таблице А.4.

На шаге № 4 к исполнению предлагается действие $\text{MoveCube}(0, 1)$ (перемещение кубика с площадки № 0 на площадку № 1). Состояние ПрО после выполнения данного шага принимает следующий вид:

$$S_4 = ((), (0, 1), ()). \quad (\text{A.5})$$

Состояние рабочей памяти планировщика после выполнения шага № 4 представлено в таблице А.5.

На заключительном шаге № 5 планировщик сообщает о завершении процесса решения задачи, передавая на исполнение действие $\text{Stop}()$. ПрО достигает целевого состояния, заданного в постановке задачи T :

$$S_5 = S_{\text{ц}} = ((), (0, 1), ()). \quad (\text{A.6})$$

Состояние рабочей памяти планировщика после выполнения шага № 5 представлено в таблице А.6.

Таблица А.1 — Параметры рабочей памяти планировщика после шага № 0

Параметр	Значение
Текущий целевой кубик	0
Текущая башня целевого кубика	2
Целевая башня текущего кубика	1
Обработанные кубики	()
Необработанные кубики	(0, 1)
Достигнутые цели	()

Таблица А.2 — Параметры рабочей памяти планировщика после шага № 1

Параметр	Значение
Текущий целевой кубик	0
Текущая башня целевого кубика	2
Целевая башня текущего кубика	1
Обработанные кубики	()
Необработанные кубики	(0, 1)
Достигнутые цели	()

Таблица А.3 — Параметры рабочей памяти планировщика после шага № 2

Параметр	Значение
Текущий целевой кубик	0
Текущая башня целевого кубика	1
Целевая башня текущего кубика	1
Обработанные кубики	()
Необработанные кубики	(0, 1)
Достигнутые цели	()

Таблица А.4 — Параметры рабочей памяти планировщика после шага № 3

Параметр	Значение
Текущий целевой кубик	1
Текущая башня целевого кубика	0
Целевая башня текущего кубика	1
Обработанные кубики	(0)
Необработанные кубики	(1)
Достигнутые цели	(0)

Таблица А.5 — Параметры рабочей памяти планировщика после шага № 4

Параметр	Значение
Текущий целевой кубик	1
Текущая башня целевого кубика	1
Целевая башня текущего кубика	1
Обработанные кубики	(0)
Необработанные кубики	(1)
Достигнутые цели	(0)

Таблица А.6 — Параметры рабочей памяти планировщика после шага № 5

Параметр	Значение
Текущий целевой кубик	—
Текущая башня целевого кубика	—
Целевая башня текущего кубика	—
Обработанные кубики	(0, 1)
Необработанные кубики	()
Достигнутые цели	(0, 1)

Вывод плана действий в графическом интерфейсе системы МИПРА представлен на рисунке А.3.

MIPRA DEMO aladin

Главная | Планировщик | Описание задачи | Решение задачи | Метрика

Решение задачи

[Скачать модель](#)

Шаги

- Шаг № 0
- Шаг № 1
- Шаг № 2
- Шаг № 3
- Шаг № 4
- Шаг № 5

ТЕКУЩИЙ ЦЕЛЕВОЙ КУБ
0

КОЛИЧЕСТВО ПЕРЕСТАНОВОК
0

РАЗУМАТОР: ВРЕМЯ ГЕНЕРАЦИИ ЛОГИЧЕСКОГО ВЫВОДА
154,3928 мс

Действия шага

0. FindTargetCube(0)

РАЗУМАТОР: Граф логического вывода

Состояние перед выполнением действий

Площадка № 0: [] Площадка № 1: [] Площадка № 2: [1 / 0]

Состояние после выполнения действий

Площадка № 0: [] Площадка № 1: [] Площадка № 2: [1 / 0]

Авторское право © Дмитрий Аладин 2026. Основано на технологиях Balabza и НИИ "Мивар". Вер. 5.0.1.0.

Рисунок А.3 — Вывод результата решения задачи планирования в домене «Мир кубиков» с использованием графического интерфейса

Вывод метрик производительности в графическом интерфейсе представлен на рисунке А.4.

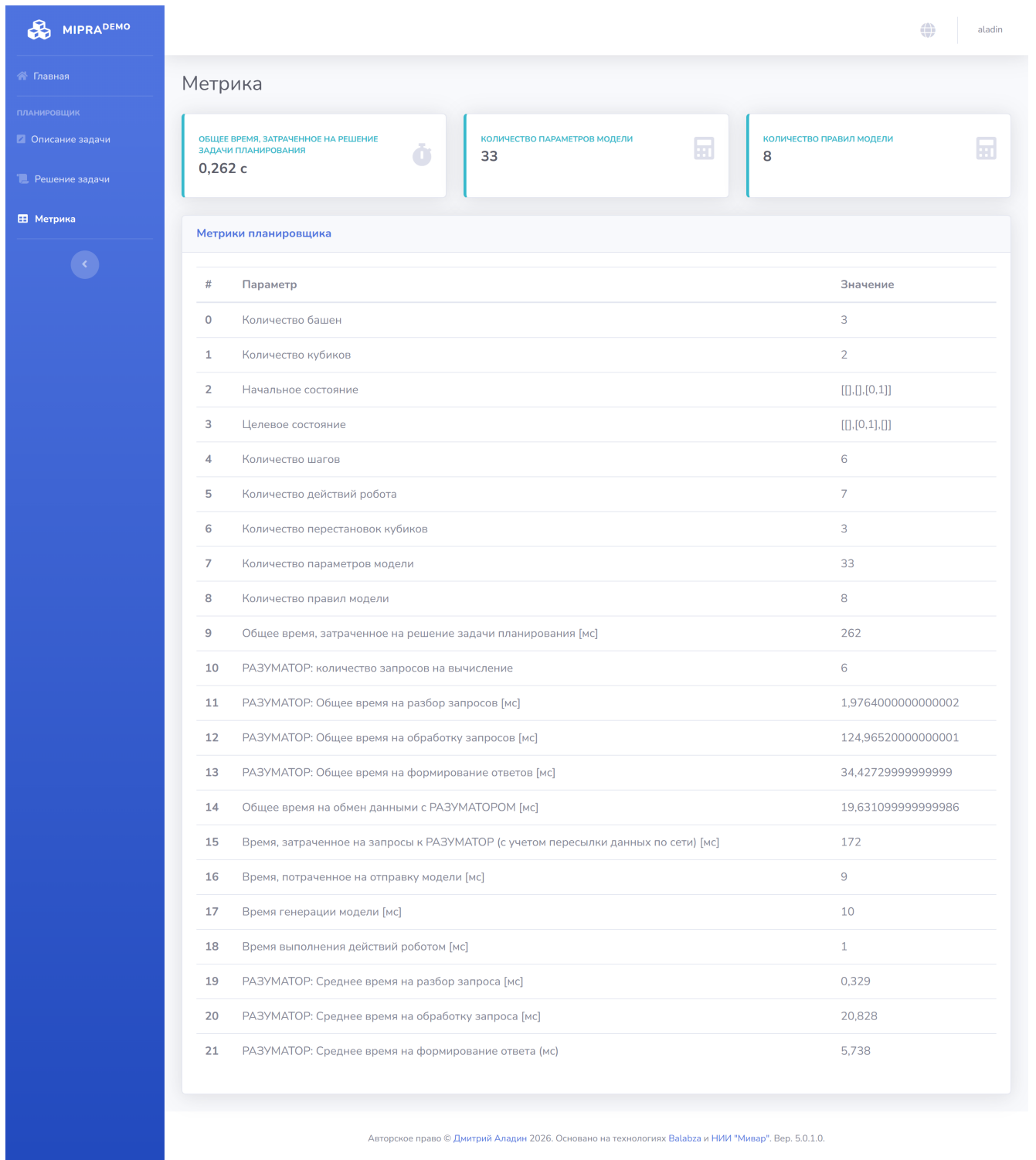


Рисунок А.4 — Вывод метрик производительности в графическом интерфейсе, полученных при решении задачи планирования в домене «Мир кубиков»

Логический вывод по МБЗ на шаге № 1 представлен на рисунке А.5.

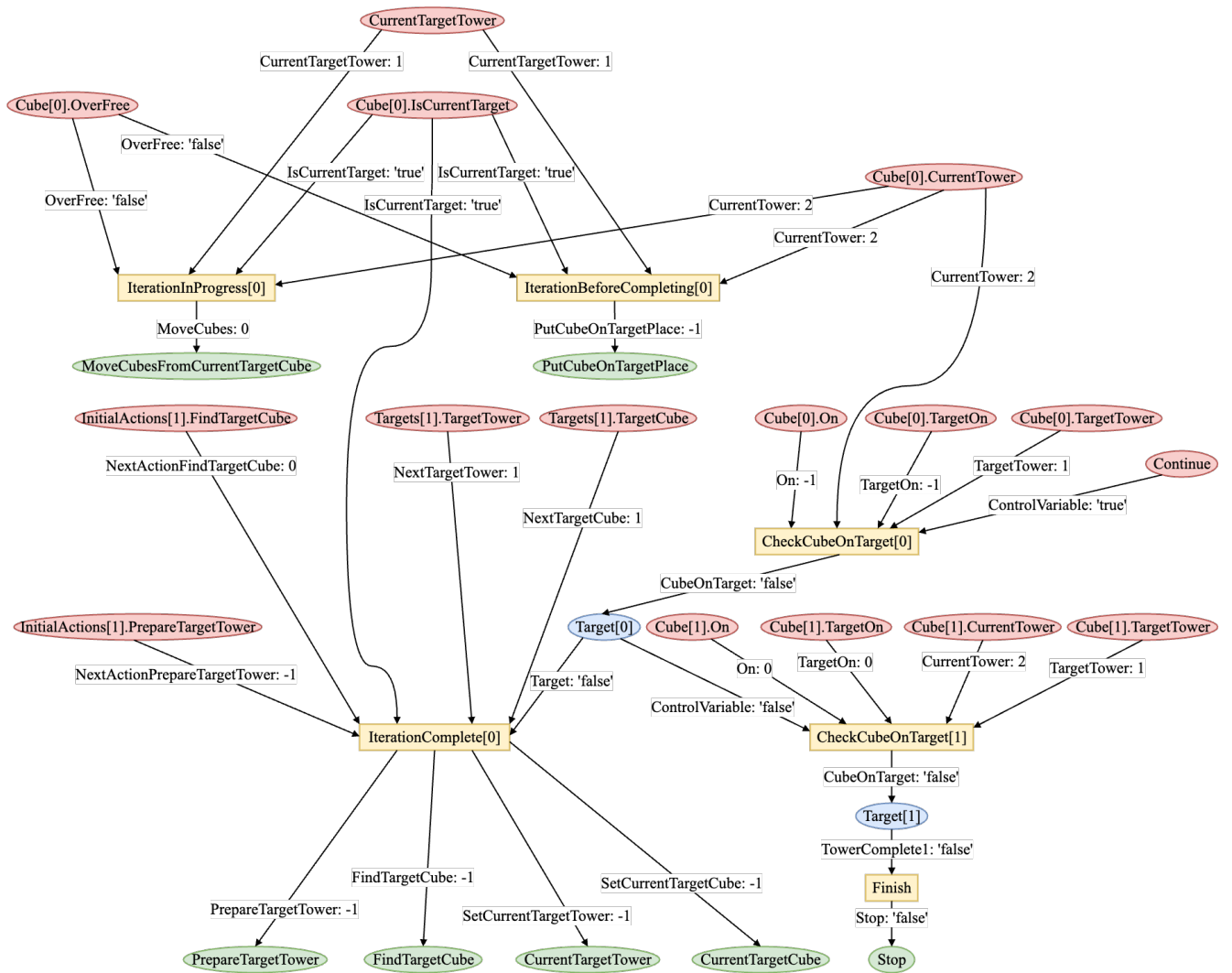


Рисунок А.5 — Пример № 1 логического вывода по миварной сети при решении задачи планирования в домене «Мир кубиков»

На схеме логического вывода представлены параметры и правила, на основе которых был сформирован промежуточный план действий для рассматриваемого шага. Для повышения информативности графического представления рёбра аннотированы парами, содержащими обозначение переменной, входящей в тело правила, и значение, которое присваивается данной переменной в момент активации правила. Если ребро направлено к параметру, то значение переменной присваивается данному параметру в результате активации правила.

Действие, переданное на исполнение, было определено в результате активации правила `IterationInProgress[0]`.

Связанность подцелей проявляется в виде последовательной активации правил `CheckCubeOnTarget[0]` и `CheckCubeOnTarget[1]`.

Логический вывод по МБЗ на шаге № 4 представлен на рисунке А.6.

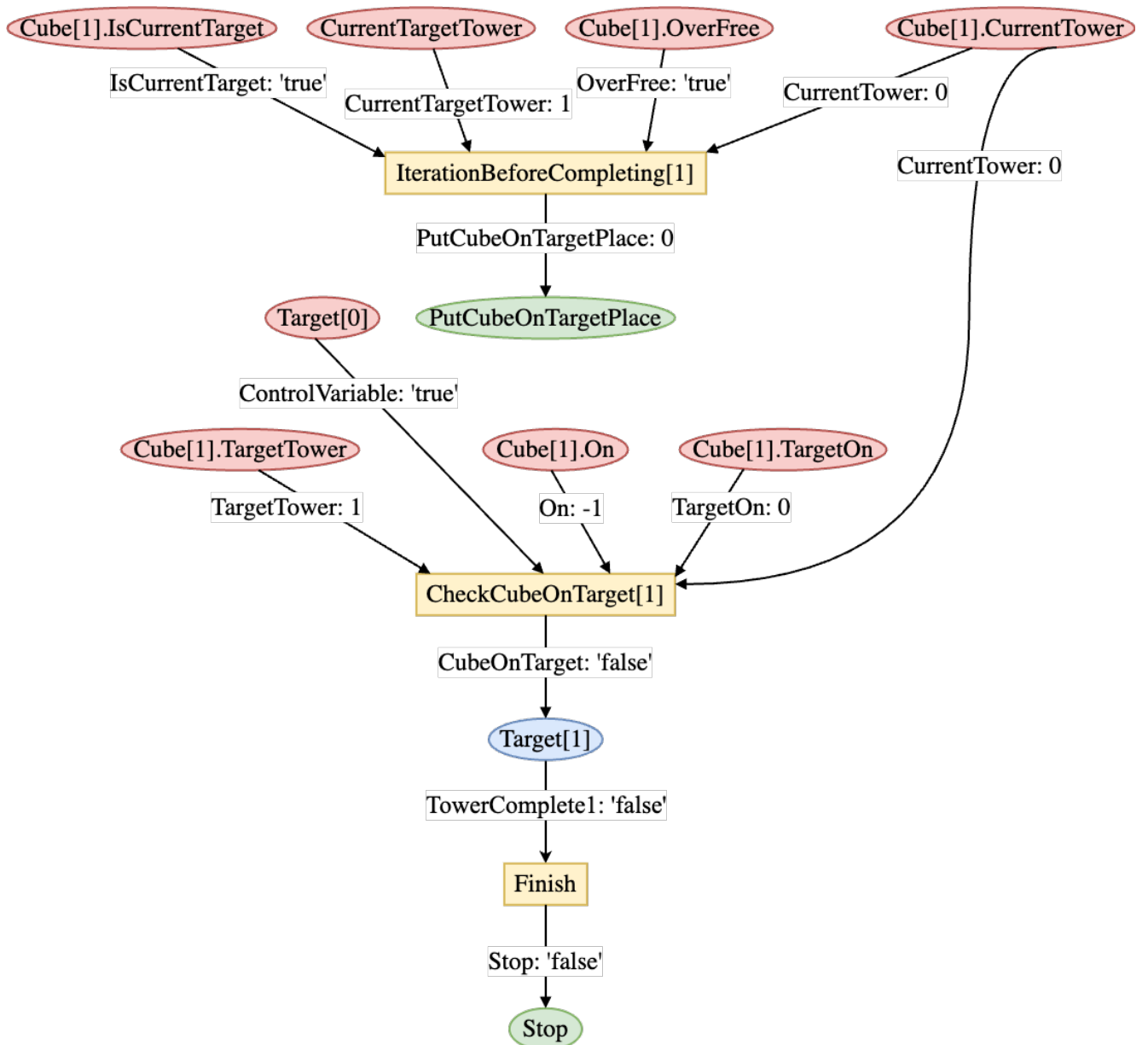


Рисунок А.6 — Пример № 2 логического вывода по миварной сети при решении задачи планирования в домене «Мир кубиков»

На рисунке А.6 виден результат применения дополнительных механизмов планировщика, предназначенных для обхода требования обязательного вычисления всех выходных параметров при активации правила в МЛВ программного комплекса КЭСМИ Wi!Mi «Разуматор» версии 2.1 (параграф 4.2.2). Расчёт признака достижения промежуточной цели № 0 не осуществляется непосредственно в МЛВ, а передаётся в неё в составе запроса на построение логического вывода.

ПРИЛОЖЕНИЕ Б

ПРИМЕР ОБРАБОТКИ МИВАРНОЙ СЕТИ МАШИНОЙ ЛОГИЧЕСКОГО ВЫВОДА ПРИ РЕШЕНИИ ЗАДАЧ В ПРОСТРАНСТВЕ СОСТОЯНИЙ

В работе [125] для описания миварной технологии логико-вычислительной обработки данных используется представление миварной сети в табличном виде. Далее данное представление будем называть *миварной таблицей*.

Предположим, что задана миварная сеть правил, отображение которой в виде двудольного графа представлено на рисунке Б.1.

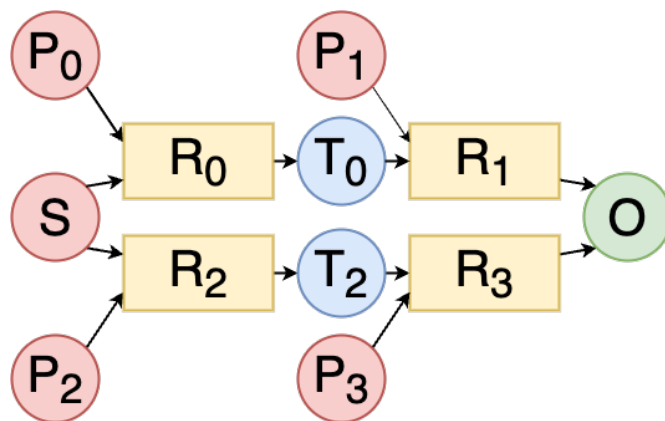


Рисунок Б.1 — Пример миварной сети в виде двудольного графа

Табличное представление миварной сети представлено на рисунке Б.2: слева показано табличное представление миварной сети, справа – объекты миварной сети без указания связей между ними.

Столбцы таблицы соответствуют переменным миварной сети, за исключением столбца А. Столбец А отражает признак активации правил: отсутствие значения указывает на то, что правило не активировано; значение 1 – правило готово к активации; значение 2 – правило было активировано. В последней строке столбца А указывается номер текущего этапа обработки миварной сети.

	S	P ₀	P ₁	P ₂	P ₃	T ₀	T ₂	O	A
R ₀	X	X				Y			
R ₁			X			X		Y	
R ₂	X			X			Y		
R ₃					X		X	Y	
F									1

Рисунок Б.2 — Миварная таблица и объекты миварной сети на этапе № 1

Строки таблицы, за исключением последней, соответствуют правилам миварной сети. В ячейках на пересечении строк правил и столбцов параметров указывается тип связи: отсутствие значения означает, что правило не связано с параметром; значение X – параметр является входным для правила; значение Y – параметр является выходным для правила. В последней строке F отмечается, какие параметры являются известными (значение Z), а какие подлежат вычислению (значение W).

Сформулируем задачу поиска логического вывода по миварной сети, представленной на рисунке Б.1, следующим образом: при известных значениях параметров S , P_0 , P_1 , P_2 и P_3 требуется вычислить значение параметра O .

В этом случае миварная таблица принимает вид, представленный на рисунке Б.3. Справа от миварной таблицы показан процесс построения логического вывода по миварной сети на текущем этапе обработки.

	S	P ₀	P ₁	P ₂	P ₃	T ₀	T ₂	O	A
R ₀	X	X				Y			
R ₁			X			X		Y	
R ₂	X			X			Y		
R ₃					X		X	Y	
F	Z	Z	Z	Z	Z			W	2

Рисунок Б.3 — Миварная таблица, а также известные и искомые параметры задачи построения логического вывода на этапе № 2

Рассмотрим процесс решения задачи построения логического вывода при использовании МЛВ программного комплекса КЭСМИ Wi!Mi «Разуматор».

Пусть упорядоченное множество известных параметров queue на этапе № 2 имеет вид (S, P_0, P_1, P_2, P_3) . В соответствии с алгоритмом 4.3 элементы данно-

го множества извлекаются последовательно для последующей обработки. Пусть первым извлечённым элементом является S . Данный параметр является входным для правил R_0 и R_2 . Для этих правил имеются все необходимые значения входных параметров, что позволяет выполнить их активацию после завершения этапа № 3 (рисунок Б.4).

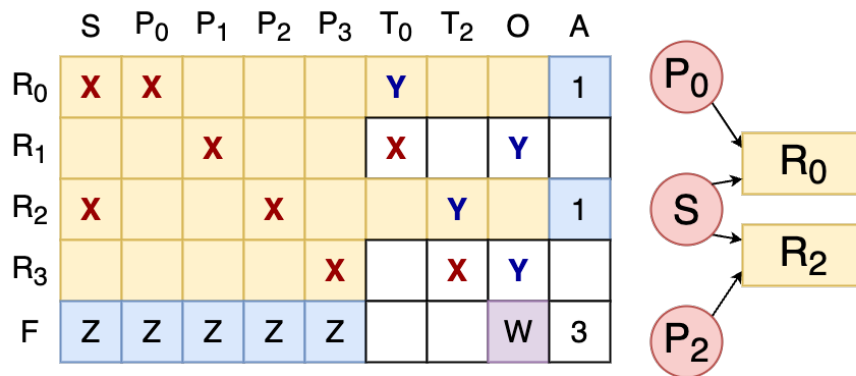


Рисунок Б.4 — Миварная таблица и логический вывод по миварной сети на этапе № 3

В результате активации будут вычислены значения параметров T_0 и T_2 (рисунок Б.5). Множество известных параметров queue после завершения этапа № 4 примет вид $(P_0, P_1, P_2, P_3, T_0, T_2)$.

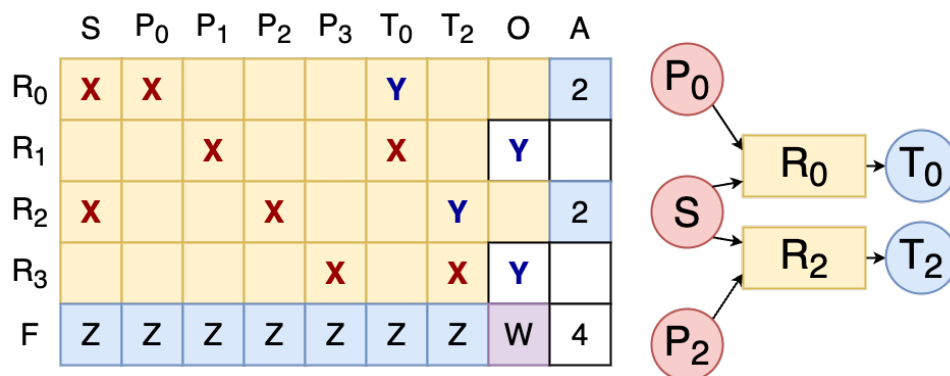


Рисунок Б.5 — Миварная таблица и логический вывод по миварной сети на этапе № 4

Далее из множества queue извлекается параметр P_0 , рассмотрение которого не приводит к выявлению новых правил, готовых к активации. После этого из множества извлекается параметр P_1 . Для данного параметра обнаруживается правило R_1 , готовое к активации (рисунок Б.6).

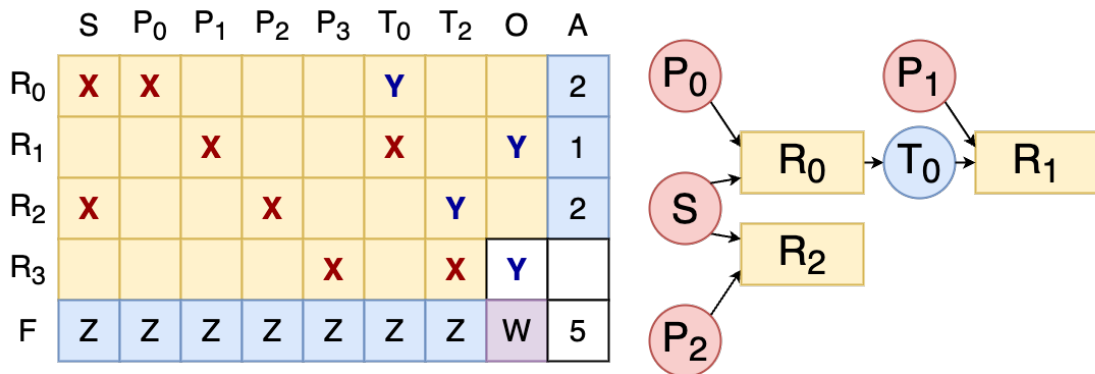


Рисунок Б.6 — Миварная таблица и логический вывод по миварной сети на этапе № 5, полученные в результате работы машины логического вывода программного комплекса КЭСМИ Wi!Mi «Разуматор»

На этапе № 5 других правил, готовых к активации, обнаружено не было, поэтому активируется только правило R_1 . В результате его активации вычисляется значение параметра O (рисунок Б.7). Поскольку вычисление значения параметра O является целью задачи построения логического вывода, МЛВ завершает обработку миварной сети.

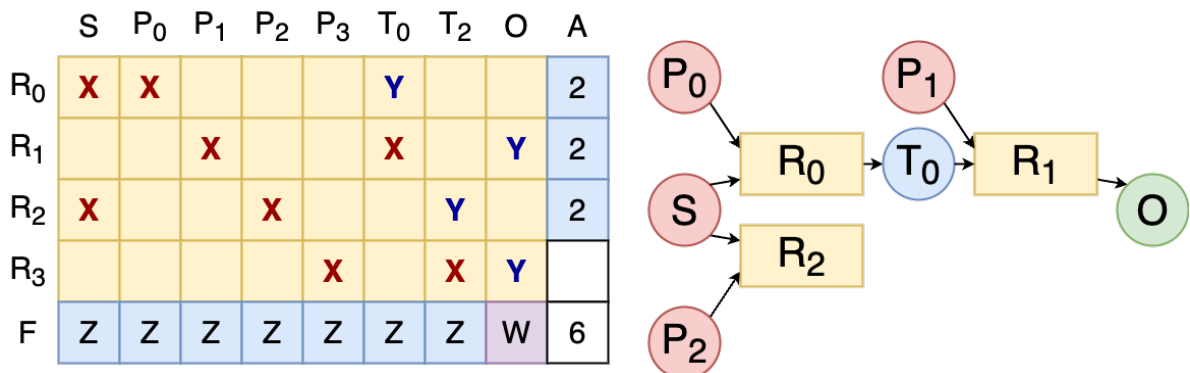


Рисунок Б.7 — Миварная таблица и логический вывод по миварной сети на этапе № 6, полученные в результате работы машины логического вывода программного комплекса КЭСМИ Wi!Mi «Разуматор»

Как отмечалось ранее в параграфе 4.2.2, в МЛВ программного комплекса КЭСМИ Wi!Mi «Разуматор» версии 2.1 параллельная активация правил R_1 и R_3 невозможна в силу особенностей обработки миварной сети (алгоритм 4.3).

Предположим, что отсутствует требование об обязательном вычислении всех выходных параметров при активации правила. В этом случае, если при активации правила R_1 значение параметра O не было вычислено, потребовались бы дополнительные этапы, на которых предпринималась бы попытка его вычисления с использованием правила R_3 . Это, в свою очередь, приводило бы к увеличению числа этапов обработки и, как следствие, к дополнительным временным затратам.

Рассмотрим процесс решения задачи построения логического вывода при использовании МЛВ программного комплекса “Balabza.Razumator”.

Множество известных параметров на этапе № 2 имеет вид (S, P_0, P_1, P_2, P_3) . В соответствии с алгоритмом 4.4 данное множество обрабатывается полностью. В результате обработки выявляются правила R_0 и R_2 , для которых на текущем этапе известны все входные параметры, что позволяет выполнить их активацию после завершения этапа № 3 (рисунок Б.4).

Пусть в результате активации правил R_0 и R_2 вычисляются значения параметров T_0 и T_2 (рисунок Б.5). Значения данных параметров обнаруживаются впервые, в связи с чем они используются для поиска следующих правил, готовых к активации. В результате такого поиска выявляются правила R_1 и R_3 . Их активация будет выполнена после завершения этапа № 5 (рисунок Б.8).

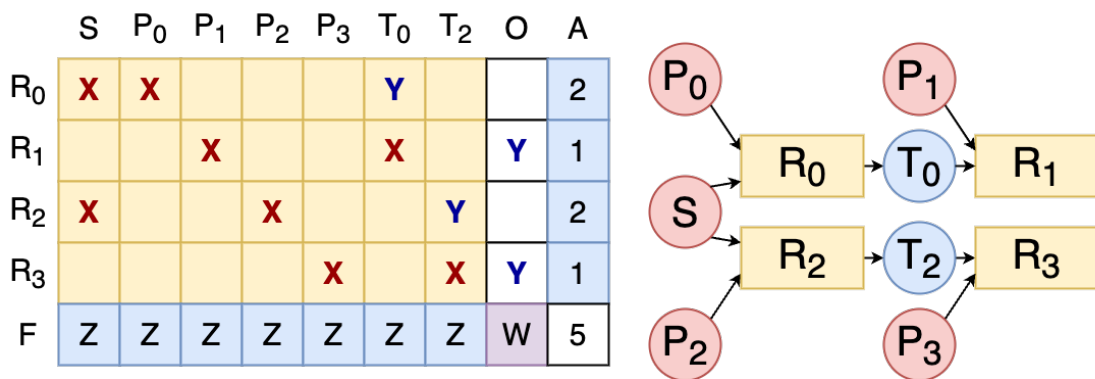


Рисунок Б.8 — Миварная таблица и логический вывод по миварной сети на этапе № 5, полученные в результате работы машины логического вывода программного комплекса “Balabza.Razumator”

На этапе № 6 значение параметра O может быть зафиксировано в результате активации правила R_1 или R_3 (рисунок Б.9). В случае возникновения конфликта при фиксации значения параметра O применяется алгоритм разрешения конфликтов, описанный в параграфе 4.2.4. После фиксации значения параметра O задача построения логического вывода считается решённой, и МЛВ завершает обработку миварной сети.

Рассмотренный пример решения задачи построения логического вывода по миварной сети наглядно демонстрирует различия между алгоритмами 4.3 и 4.4. Алгоритм 4.4, реализованный в программном комплексе “Balabza.Razumator”, обеспечивает более высокую производительность в рамках одного этапа обработки миварной сети за счёт возможности параллельной активации правил, готовых к выполнению при текущем наборе известных параметров.

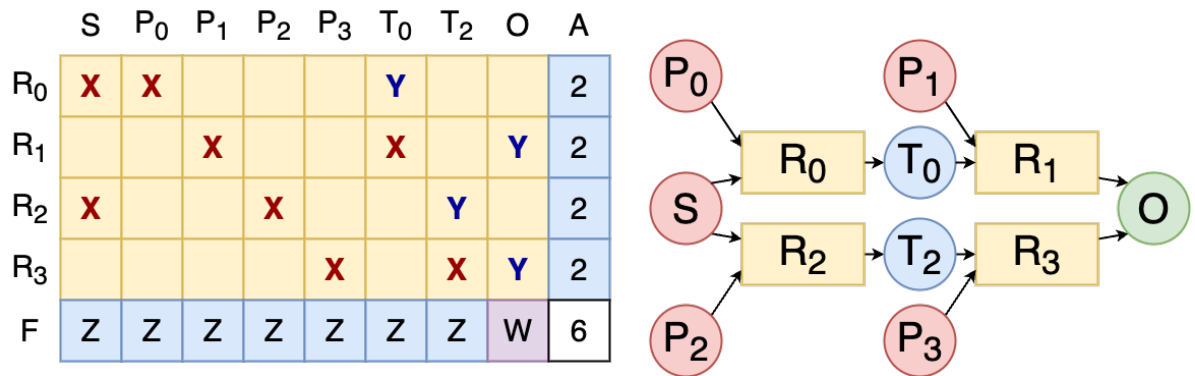


Рисунок Б.9 — Миварная таблица и логический вывод по миварной сети на этапе № 6, полученные в результате работы машины логического вывода программного комплекса “Balabza.Razumator”