

Программные технологии для эффективного поиска логического вывода в исчислении позитивно-образованных формул

Диссертация на соискание учёной степени кандидата
технических наук

Специальность: 05.13.17 – теоретические основы
информатики

Александр Ларионов

ФГБОУ ВПО “Иркутский государственный университет”

Научный руководитель к.т.н. Е.А. Черкашин, ИДСТУ СО РАН

Иркутск - 2012

Актуальность проблемы/Предпосылки

1. Системы автоматического доказательства теорем (АДТ) успешно применяются для решения разного рода задач:

- верификация программных и аппаратных средств;
- математические теоремы, есть примеры решения открытых проблем (EQP: проблема Роббинса);
- планирование;
- исследование безопасности информационных потоков;
- логическое программирование;
- семантический веб и т.д.

Разработка систем АДТ ведётся с середины 20-ого века. Некоторые системы имеют более чем 20 летнюю историю. На сегодняшний день теоретической основой, как правило, является метод резолюции.

Актуальность проблемы/Предпосылки

2. Границы формальных систем:

2.1. Полуразрешимость исчислений первого порядка. Если формула является теоремой, то существует разрешимый алгоритм установки этого факта. Если формула не является теоремой, то нет.

2.2. Полнота исчислений первого порядка.

2.3. Теорема Гёделя о неполноте.

3. Обязательно найдутся задачи со сколь угодно большим минимальным выводом.

4. Планка решаемых задач повышается. Актуальна проблема расширяемости существующих систем АДТ. Гегемония метода резолюции.

Актуальность проблемы/Следствие

1. Любое отодвижение границ выводимости всегда актуально. Уменьшение шагов вывода, уменьшение скорости вывода. Разработка стратегий вывода для сужения пространства поиска в предположительно верном направлении.

2. Важен не только результат (доказано или не доказано), но и способ решения, для анализа задач. Иные способы решения задач необходимы.

3. Исследование исчислений не связанных с методом резолюции актуально.

Выбор: исчисление позитивно-образованных формул (ПО-формул) .

Современное состояние дел в АДТ

Свойства ПО-формул

Объект и предмет исследования

Объект исследования:

Исчисление позитивно-образованных формул.

Предмет исследования: свойства языка и исчисления П0- -формул, современных структур данных и алгоритмов поиска ЛВ в исчислениях первого порядка, разработка структур данных и алгоритмов, обеспечивающих создание новых программных систем и технологий эффективного поиска ЛВ в исчислениях П0- -формул. Эффективность поиска ЛВ задается следующими критериями: время решения задачи; количество шагов ЛВ; объём использованной оперативной памяти; ширина класса успешно решаемых задач.

Цель работы

Цель работы - - - разработка программной системы для эффективного поиска логического вывода в исчислении позитивно- -образованных формул.

Или тут свойства П0-формул

Основные задачи

1. Разработка эффективных структур данных представления П0--формул в памяти компьютера.
2. Создание оригинальных методик (стратегий) эффективного поиска ЛВ в исчислении П0--формул.
3. Адаптация существующих методик АДТ для исчисления П0--формул с целью повышения эффективности поиска ЛВ.
4. Разработка эффективной программной системы АДТ, создание инструментальных средств для программирования систем АДТ в исчислении П0--формул.
5. Апробация разработанных программных средств на решении тестовых и практических задач. Сравнение с другими системами АДТ.

Научная новизна

1. Изучены свойства исчисления П0--формул, влияющие на применимость известных методик повышения эффективности поиска ЛВ.
2. Впервые предложены и реализованы подходы обработки неограниченных переменных, стратегия k, m -ограничения, структуры данных экономного представления П0--формул с дизъюнктивным ветвлением, обработка предиката равенства без прямого представления аксиом равенства в виде П0--формулы.
3. Впервые, для исчисления П0--формул успешно применены широко используемые подходы: индексирование термов, параллельные схемы алгоритмов поиска ЛВ, разделение данных, кэширование результатов, системы переписывания термов.
4. Значительно расширен класс решаемых задач при помощи систем АДТ, базирующихся на исчислении П0--формул. Построены новые варианты решения задач АДТ по сравнению с другими методами АДТ.

Научная и практическая значимость

- 1.** Созданы система АДТ и инструментальная среда разработки систем АДТ, направленные на построение ЛВ формул исчисления П0--формул первого порядка.
- 2.** Выделены классы задач, для которых созданная система является более эффективной, чем современные высокопроизводительные системы АДТ и предложены специальные стратегии, повышающие эффективность поиска ЛВ.
- 3.** Реализована инфраструктура тестирования разработанных алгоритмов и программного обеспечения АДТ на тестовых задачах из библиотеки TRTP.

Научная и практическая значимость

1. Федеральная целевая программа «Научные и научно-педагогические кадры инновационной России» на 2009-2013 годы, госконтракт № П696.

2. Базовые проекты научно-исследовательской работы ИДСТУ СО РАН. Проект IV.31.2.4. “Методы и технологии разработки программного обеспечения для анализа, обработки, и хранения разноформатных междисциплинарных данных и знаний, основанные на применении декларативных спецификаций форматов представления информации и моделей программных систем”, № гос. регистрации: 01201001350. Программа IV.31.2. “Новые ГИС и веб-технологии, включая методы искусственного интеллекта, для поддержки междисциплинарных научных исследований сложных природных, технических и социальных систем с учетом их взаимодействия”. Проект IV.32.1.2. Информационно-вычислительные системы нового поколения для формаций автономных необитаемых аппаратов, № гос. Регистрации: 01201001346.

3. РФФИ 08-07-98005-р_сибирь_a “Программные технологии логико-математического моделирования динамики лесных ресурсов Байкальского региона”.

4. Программа “Университетский кластер”.

Результаты выносимые на защиту

1. Адаптированы методики других систем АДТ, повысившие эффективность поиска логического вывода: три типа параллельных стратегий (скорость), индексирование термов (скорость), один вариант разделения памяти (память).
2. Реализованы оригинальные стратегии, повышающие эффективность поиска ЛВ: дерево состояний вывода (память), k, m -ограничение (количество шагов), стратегии направленные на решение проблем неограниченных переменных (количество шагов, время), варианты разделения памяти (память, время).
3. Реализована подсистема управления ЛВ (количество шагов).
4. Разработана инфраструктура для взаимодействия с библиотекой задач TRTP. Решено множество сложных задач.

Публикации/В журналах рекомендованных ВАК РФ

1. Давыдов А.В., Ларионов А.А., Черкашин Е.А. Об исчислении позитивно-образованных формул для автоматического доказательства теорем // Моделирование и анализ информационных систем. 2010.Т. 17, N 4, с. 60–69.
2. Ларионов А.А., Черкашин Е.А., Терехин И.Н. Системные предикаты для управления логическим выводом в системе автоматического доказательства теорем для исчисления позитивно-образованных формул // Вестник Бурятского государственного университета, 2011, выпуск 9. Серия Математика. Информатика. с. 94-98.
3. Ларионов А.А., Черкашин Е.А. Параллельные схемы алгоритмов автоматического доказательства теорем в исчислении позитивно-образованных формул // Дистанционное и виртуальное обучение. Февраль 2012. No. 2, с. 93-100.
4. Davydov A.V., Larionov A.A., Cherkashin E.A. On the calculus of positively constructed formulas for automated theorem proving // Automatic Control and Computer Sciences (AC&CS). 2011. Volume 45, Issue 7, pp.402-407.

Публикации/Другие

5. Ларионов А.А., Терехин И.Н., Черкашин Е.А., Давыдов А.В. Программная система КВАНТ/4 для автоматического доказательства теорем // Труды ИМЭИ ИГУ. Математика и информатика : сб. научных трудов / под ред.: Ю. Д. Корольков [и др.]. - Иркутск : Изд-во ИГУ, 2011. - Выпуск 1 с. 77-85.

6. Ларионов А.А., Черкашин Е.А., Давыдов А.В. Программная система для автоматического доказательства теорем в исчислении позитивно-образованных формул. / Винеровские чтения / Труды IV Всероссийской конференции. Часть II. - Иркутск: ИрГТУ, 2011. - с.190-197.

7. А.В. Давыдов, А.А. Ларионов. Об исчислении позитивно-образованных формул для автоматического доказательства теорем. / Тр. 5-го международного симпозиума по компьютерным наукам в России. Семинар "Семантика, спецификация и верификация программ: теория и приложения". 14-15 июня 2010, Казань, с. 109-116.

8. A.A. Larionov, E.A. Cherkashin, A.V. Davydov Theorem Proving Software, Based on Method of Positively-Constructed Formulae. // MIPRO 2011. 34-th international convention on information and communication technology, electronics and microelectronics. Vol. III. May 23-27, 2011. Croatia, Opatija. //MIPRO: Croatia. - 2011., p.p. 365-368.

Представление работы

1. Международная конференция “Мальцевские чтения”, г.Новосибирск, 24-28 августа 2009 г.;
2. Семинар ИДСТУ СО РАН “Ляпуновские чтения”, ИДСТУ СО РАН, г. Иркутск, 21-23 декабря 2009 г.;
3. Всероссийская конференция молодых ученых “Математическое моделирование и информационные технологии”, г. Иркутск, 15-21 марта 2010 г.;
4. Международная конференция “Облачные вычисления. Образование. Исследования. Разработки”, г.Москва 15-16 апреля 2010 г.;
5. Международный симпозиум по компьютерным наукам в России. Семинар “Семантика, спецификация и верификация программ: теория и приложения”, г.Казань, 14-15 июня 2010 г.;
6. 4-ая Всероссийская конференция “Винеровские чтения”, г.Иркутск, 9-14 марта 2011г.
7. 34-ый международный симпозиум “MIPRO”, г.Опатия, Хорватия, 23-27 мая 2011г.
8. 4-ая Всероссийская мультikonференция по проблемам управления, с. Дивноморское, 3-8 октября 2011г.
9. 13-ая национальная конференция по искусственному интеллекту с международным участием (КИИ-2012), г. Белгород, 16-20 октября 2012 г.

Личный вклад автора

1. Стратегии поиска ЛВ, являющихся основой разработанных алгоритмов [1,2,3]
2. Адаптированы существующие алгоритмы поиска логического вывода в исчислении П0-формул [1,3,4,5,8];
3. Реализованы структуры данных для представления П0-формул [1,5,6,8];
4. Реализована программная система АДТ [2,5,8];
5. Тестирование системы АДТ на тестовых примерах АДТ [3,5,6].

Совместно с Черкашиным Е.А. проведена разработка структур данных для представления П0-формул с использованием разделения оперативной памяти. Совместно с Давыдовым А.В. созданы методики поиска ЛВ с использованием неограниченных переменных.

Соответствие паспорту специальности

Материал диссертации соответствует формуле специальности 05.13.17 -- Теоретические основы информатики. Диссертация посвящена исследованию процессов создания, накопления и обработки информации, представленной в виде фактов и знаний; созданию и исследованию новых моделей знаний, методов представления и обработки знаний; исследованию принципов создания и функционирования программных средств автоматизации представления обработки знаний.

В диссертации получены результаты по следующим пунктам ``Области исследований``: 2. Исследование информационных структур, разработка и анализ моделей информационных процессов и структур; 8. Исследование и когнитивное моделирование интеллекта, включая моделирование поведения, моделирование рассуждений различных типов, моделирование образного мышления. Результаты диссертации имеют научное и народнохозяйственное значение для решения проблем повышения производительности обработки знаний в ПО-исчислении, они позволяют развить научные основы современных информационных технологий обработки формализованных знаний средствами вычислительной техники и в ускорении на этой основе научно-технического прогресса.

Структура работы

- Введение
- Глава 1. Вспомогательные сведения
- Глава 2. Методики повышения производительности АДТ
- Глава 3. Реализация алгоритмов и программная система АДТ
- Глава 4. Применение программной системы
- Заключение
- Приложение

Всего 163 страницы.

Основного текста 131 страница.

Список литературы содержит 100 наименований.

Введение

Введение/ПО-формулы

$$\forall:T\{\exists:A(k),B(f(e))\}\begin{cases}\forall x:D(x)\{\exists:False\\ \forall x,y:P(x,y)\{\exists:False\\ \forall x,y:A(x),B(y)\begin{cases}\exists:D(y)\\ \exists z:P(x,z)\end{cases}\end{cases}\end{cases}$$

Формулы имеют древовидную структуру, поэтому можно пользоваться соответствующей терминологией.

- 1. Узел уровня 0 – **корень**.
- 2. Узлы уровня 1 – **базы**.
- 3. Узлы уровня 2 – **вопросы** к базе.
- 4. Более глубокие узлы – **консеквенты**.
- 5. Если у вопроса более одного дочерних узла, то вопрос обладает **дизъюнктивным ветвлением**.
- 6. Если глубина больше 3, то подформула называется **глубокой**.
- 7. Вопросы с False называются **целевыми**.
- 8. Подстановки для вопроса называются **ответными**.

Вопрос: $\forall \bar{y}:Q(\tilde{y})$

База: $\exists \bar{x}:B(\tilde{x})$

Ответ: $\theta:\bar{y}\rightarrow H^\infty$ если $Q(\tilde{y})\theta\subseteq B(\tilde{x})$

Правило вывода: ω

Введение/ПО-формулы/Особенности

1. Крупноблочная структура.
2. Единственное правило вывода.
3. Нет необходимости в сколемизации.
4. Процедура ЛВ фокусируется в ближайшей окрестности корня ПО–формулы.
5. Имеется естественный ИЛИ-параллелизм.
6. Вариабельность семантики.

Методики/Глава 2

Методики/Основные проблемы

Методики/Основные проблемы/Сложность представления

$$\forall:T\left\{\exists:A(k)\left\{\begin{array}{l}\forall x:P(k,x)\{\exists:C(x)\\ \forall x:A(x)\left\{\begin{array}{l}\exists:D(x)\\ \exists z:P(x,z)\end{array}\right.\end{array}\right.\right.\right.$$



$$\forall:T\left\{\begin{array}{l}\exists:A(k),D(k)\left\{\begin{array}{l}\forall x:P(k,x)\{\exists:C(x)\\ \forall x:A(x)\left\{\begin{array}{l}\exists:D(x)\\ \exists z:P(x,z)\end{array}\right.\end{array}\right.\\ \exists:A(k),P(k,z')\left\{\begin{array}{l}\forall x:P(k,x)\{\exists:C(x)\\ \forall x:A(x)\left\{\begin{array}{l}\exists:D(x)\\ \exists z:P(x,z)\end{array}\right.\end{array}\right.\end{array}\right.$$

- 1. Структура формул сложнее чем, например, в методе резолюции.
- 2. Используется два типа кванторов.



Необходимы специальные структуры данных для ПО-формул, совместимые с предлагаемыми методиками

Методики/Основные проблемы/Лимит памяти

1. Структура формулы усложняется вширь. Появляются новые ветви.
2. База фактов увеличивается.
3. Сложность термов увеличивается.
4. Формализация некоторых задач изначально велика. Например, 15M6.



Рано или поздно будет исчерпан лимит памяти.

Методики/Основные проблемы/Неограниченные переменные

$$\forall : T \left\{ \exists v : A(v) \left\{ \begin{array}{l} \forall x, y : C(x, y), Q(t(e, e)) \{ \exists : False \\ \forall x, \textcircled{y} : A(x) \{ \exists : C(x, y), Q(y) \end{array} \right. \right.$$

Эрбранов универс: $H^\infty = \{e, t(e, e), t(e, t(e, e)), t(t(e, e), e) \dots\}$

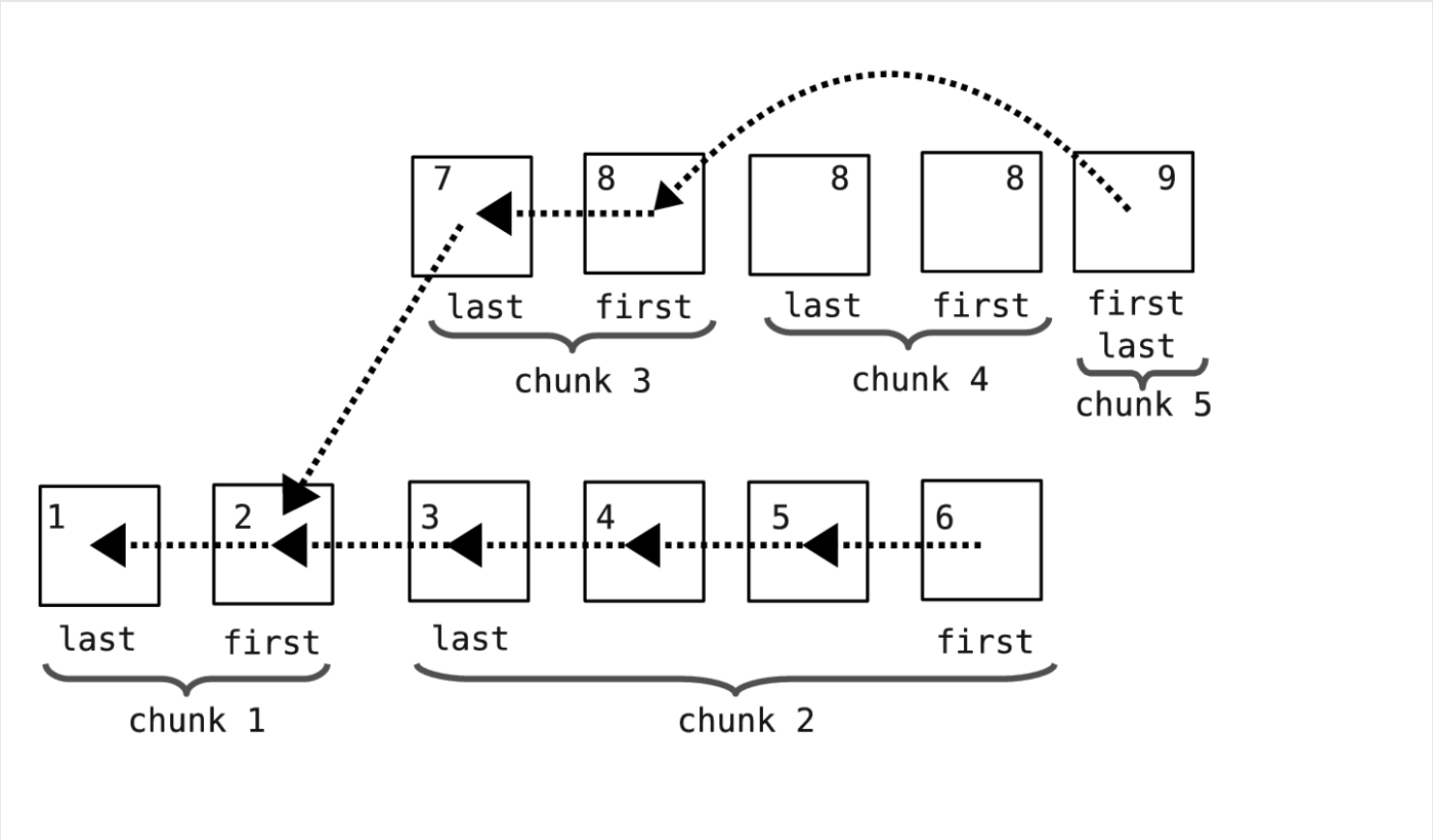
Ответ: $\theta : \{x \rightarrow v, y \rightarrow ?\}$

Подстановкой для неограниченной переменной может быть любой элемент, в общем случае бесконечного эрбранова универса. **Какой именно элемент выбрать?**

Методики/Дерево состояний вывода/Чанки

Чанк — связный список с выделенными первым и последним элементом.

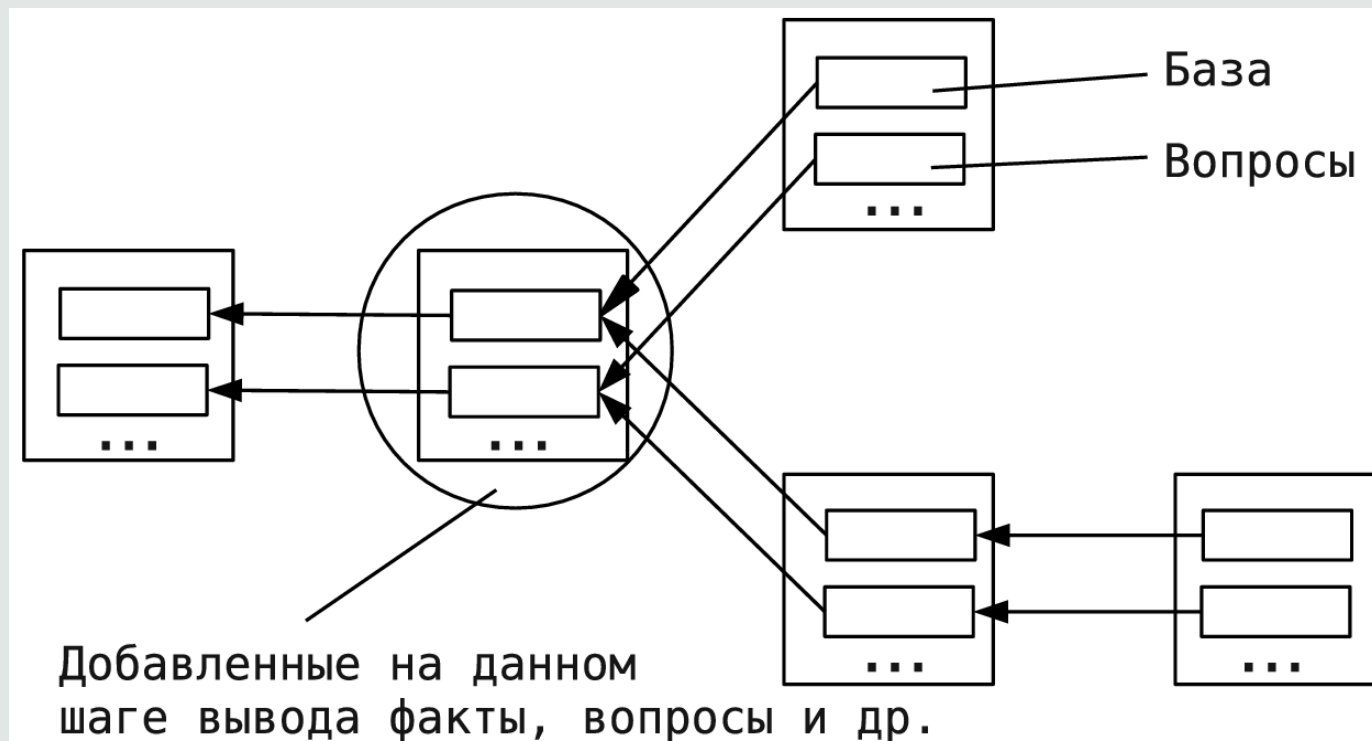
Задана операция связывания двух чанков.



Методики/Дерево состояний вывода

Связывая чанки можно построить дерево. От листов в корню. Каждый узел – чанк. Такое дерево используется для хранения состояний логического вывода, и называется **дерево состояний вывода** (ДСВ).

Каждый элемент чанка это: факт из базы, подформула-вопрос, ответы на вопросы и др.



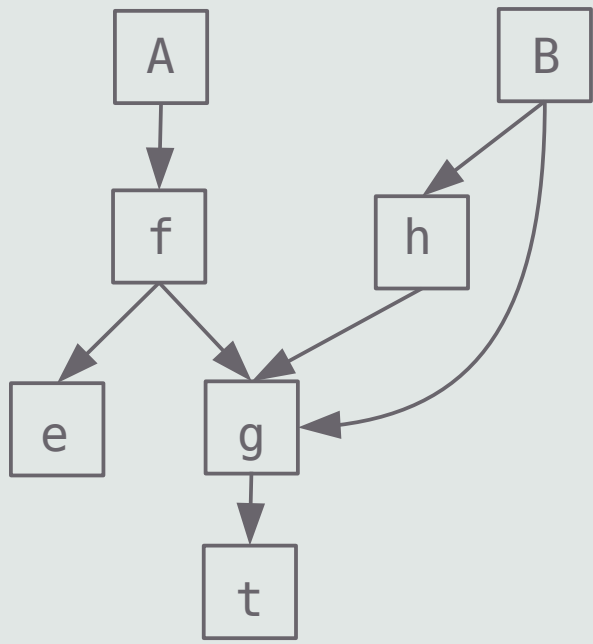
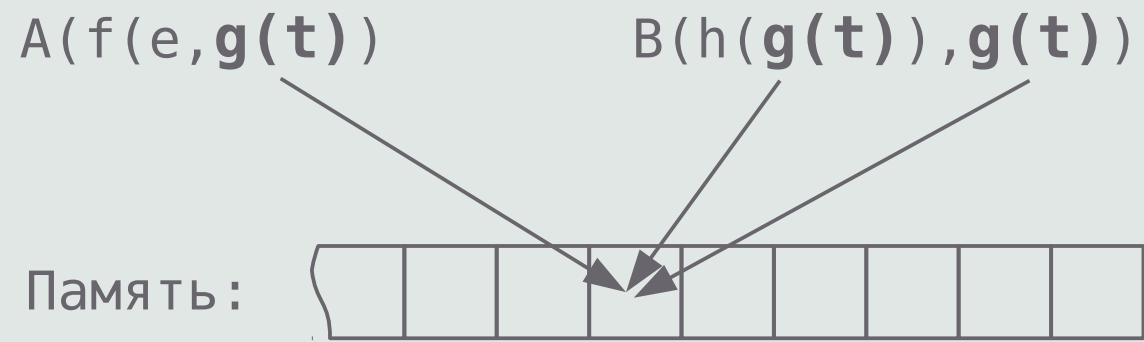
Методики/Дерево состояний вывода

1. Корень дерева хранит информацию об исходной базовой-подформуле.
2. Остальные узлы хранят подформулы-консеквенты, добавляемые после очередного шага вывода. Формула монотонно разрастается.
3. Базовая подформула на текущем шаге определяется совокупностью узлов ДСВ от листа до корня. Количество листов соответствует количеству базовых подформул.

Таким образом реализована возможность **отслеживать события** связанные с поиском логического вывода. Эти события обратимы, а значит реализуется **возврат (backtracking)**. Кроме того ДСВ позволяет **экономить память** (об этом ниже).

Методики/Экономия памяти/Разделение памяти/Термов

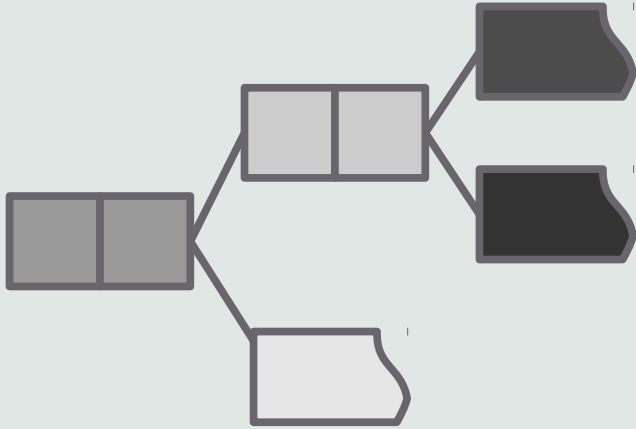
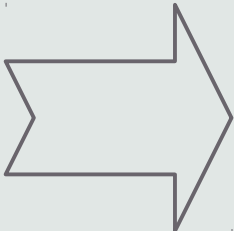
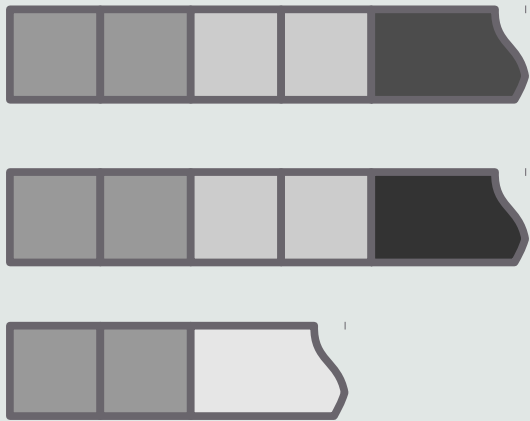
Разделение общих основных термов.



Методики/Экономия памяти/Разделение памяти/Базовых подформул/Схема

Разделение общих частей базовых подформул, полученных в результате расщепления формулы.

Базовые подформулы

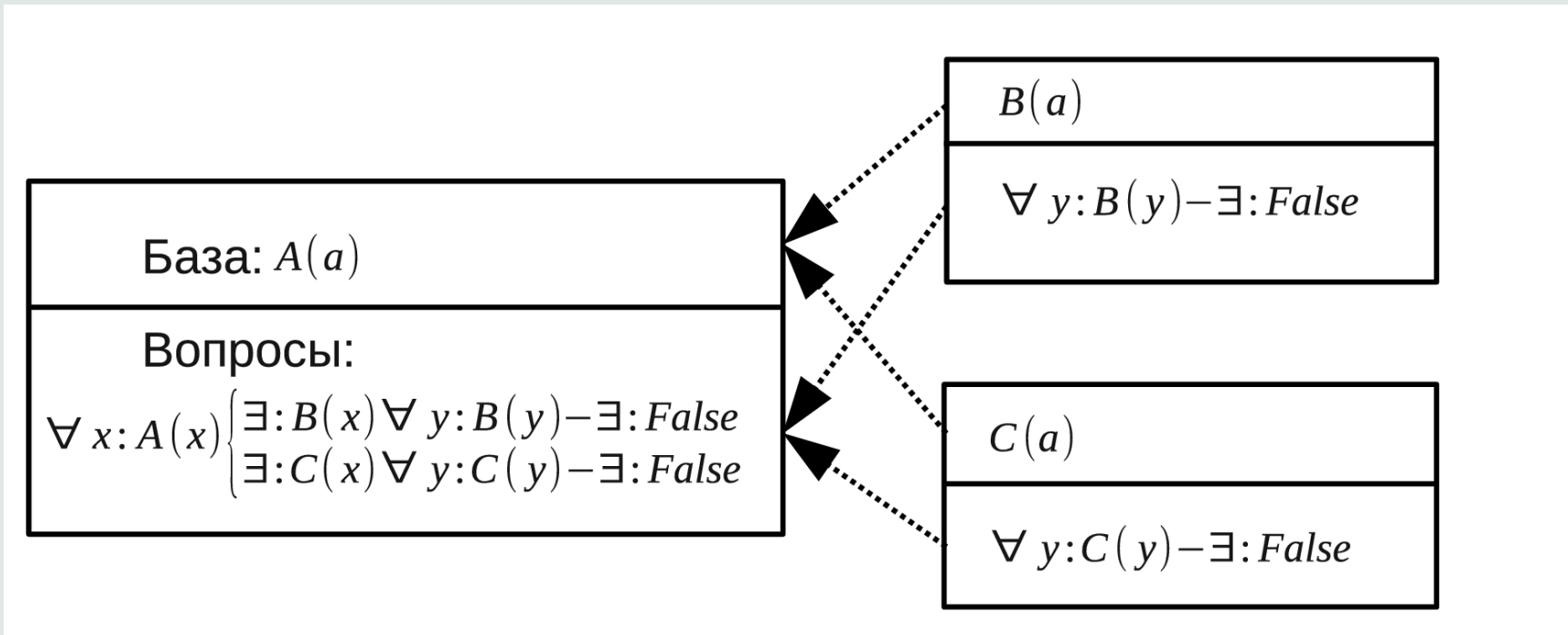


Методики/Экономия памяти/Разделение памяти/Базовых подформул/Пример

Базовая подформула:

$$\exists: A(a) T \left\{ \forall x: A(x) \left\{ \begin{array}{l} \exists: A(x) - \{ \forall y: A(y) - \{ \exists: False \\ \exists: B(y) - \{ \forall y: B(y) - \{ \exists: False \end{array} \right. \right.$$

Корень – формула до ответа на вопрос. Всё дерево – после.



Методики/Экономия памяти/Удаление излишков и веса выражений

1. Удаление неиспользуемых фактов. Пример:

$$\forall : T \{ \exists : A(n, m), B(f(e)) \left\{ \begin{array}{l} \forall x, y : A(x, y) \{ \exists : A(f(m), f(m)) \\ \forall x : A(x, x) \{ \exists : False \end{array} \right.$$

Очевидно, факт $B(f(e))$ никогда не используется.
Его можно удалить.

2. Выбор таких ответов, исполнение которых приводит к выражениям наименьшего веса. Под весом понимается количество узлов в древовидном представлении выражения.

Методики/Неограниченные переменные/Ленивая конкретизация

Идея: Вместо конкретного элемента эрбранова универса выбирается неопределённый эрбанов элемент (НЭЭ), который конкретизируется позднее по мере необходимости.

Пример:

$$\forall : True \{ \exists : True \left\{ \begin{array}{l} \forall x : True \{ \exists : A(x) \\ \forall x : A(f(x)) \{ \exists : B(f(x)) \\ \forall : B(f(a)) \{ \exists : False \end{array} \right. \}$$

$$H^{\infty} = \{a, f(a), f(f(a)) \dots\}$$

Ответ на первый вопрос: $\{x \rightarrow h_1\}$

Ответ на второй вопрос: $\{h_1 \rightarrow f(h_2)\}$

Ответ на третий вопрос: $\{h_2 \rightarrow a\}$

h_1 и h_2 являются неопределёнными эрбрановыми элементами (НЭЭ)

Методики/Неограниченные переменные/Ленивая конкретизация/Ограничения

Проблема: Конкретизация НЭЭ может производиться вхолостую (раньше нужного времени), и не приводить к успешному результату.

Ограничение 1: Ограничить количество одинаковых конкретизаций таких НЭЭ, которые изначально появились как результат подстановки для одной и той же переменной.

Ограничение 2: Сохранять конкретизируемый НЭЭ, при этом порождать новое выражение.

Методики/Неограниченные переменные/Фильтрация эрбранова универса

Идея: Фильтровать заведомо неподходящие элементы эрбранова универса.

Методики/ k, m -ограничение

Идея: Ответ на вопрос принимается, если за последующие k шагов вывода, заданное событие произойдёт m раз.

Два варианта спецификации:

1. k, m -опровержение. Обобщение k -опровержения, предложенного А.К. Жерловым. Если за k шагов будет опровергнуто по меньшей мере m базовых подформул.
2. k, m -конкретизация. Если за k шагов будет конкретизировано m НЭЭ.

Реализация использует ДСВ. В случае неудачи производится возврат, т.е. удаление узлов ДСВ от листа до точки возврата, с последовательной разконкретизацией НЭЭ.

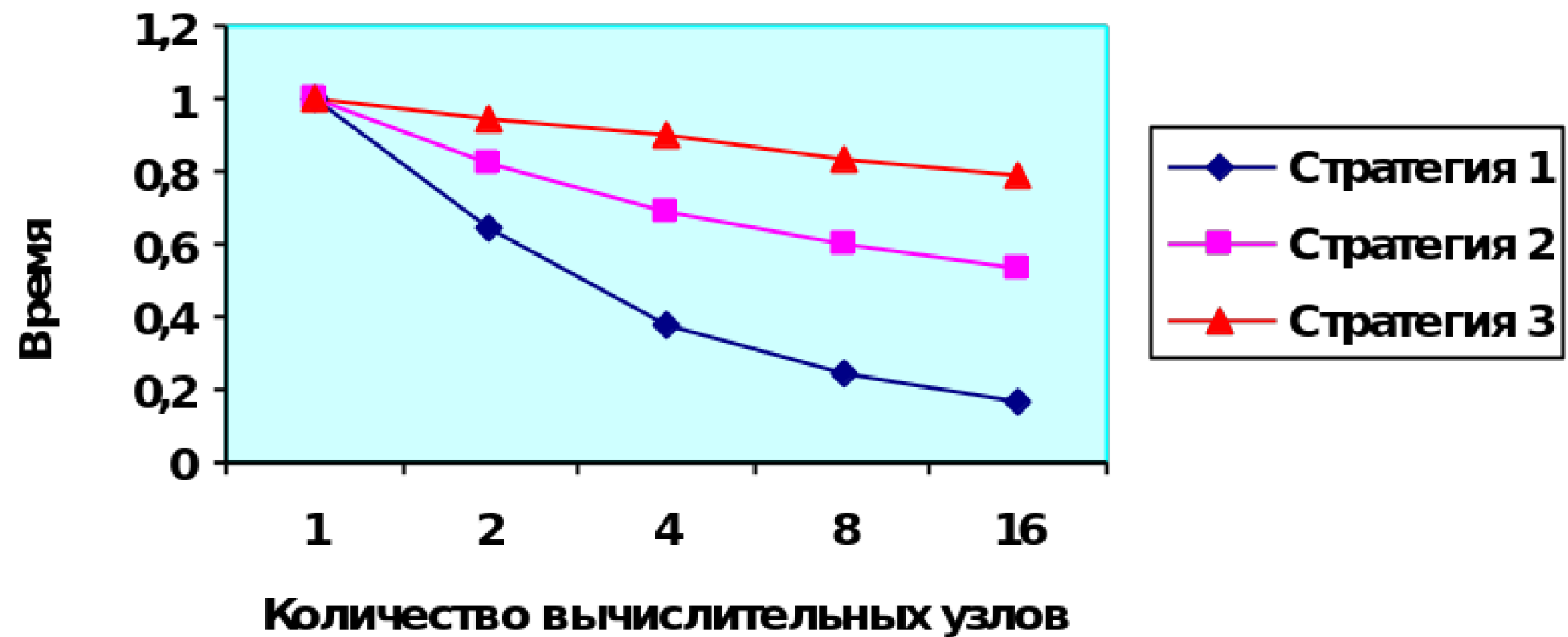
Методики/Параллельные стратегии

Идея использовать принципы Erlang: много **независимых** процессов.

Три вложенные стратегии:

- 1.** Независимые базы опровергать в отдельном процессе. Возможно на другом вычислительном узле. Особенно актуально для крупных формул с дизъюнктивными ветвлениями.
- 2.** Для каждого вопроса искать ответы в независимых процессах.
- 3.** Процедура поиска ответов использует процедуру поиска подстановок для каждого атома из конъюнкта вопроса, которые можно исполнять в независимых процессах. Накладные расходы на создание процессов могут перекрыть полезное время вычислений, поэтому использование стратегии возможно в особых случаях.

Методики/Параллельные стратегии/Тестирование



Методики/Индексирование

Проблема: Очень много термов. Поиск по ним нужно проводить быстро.

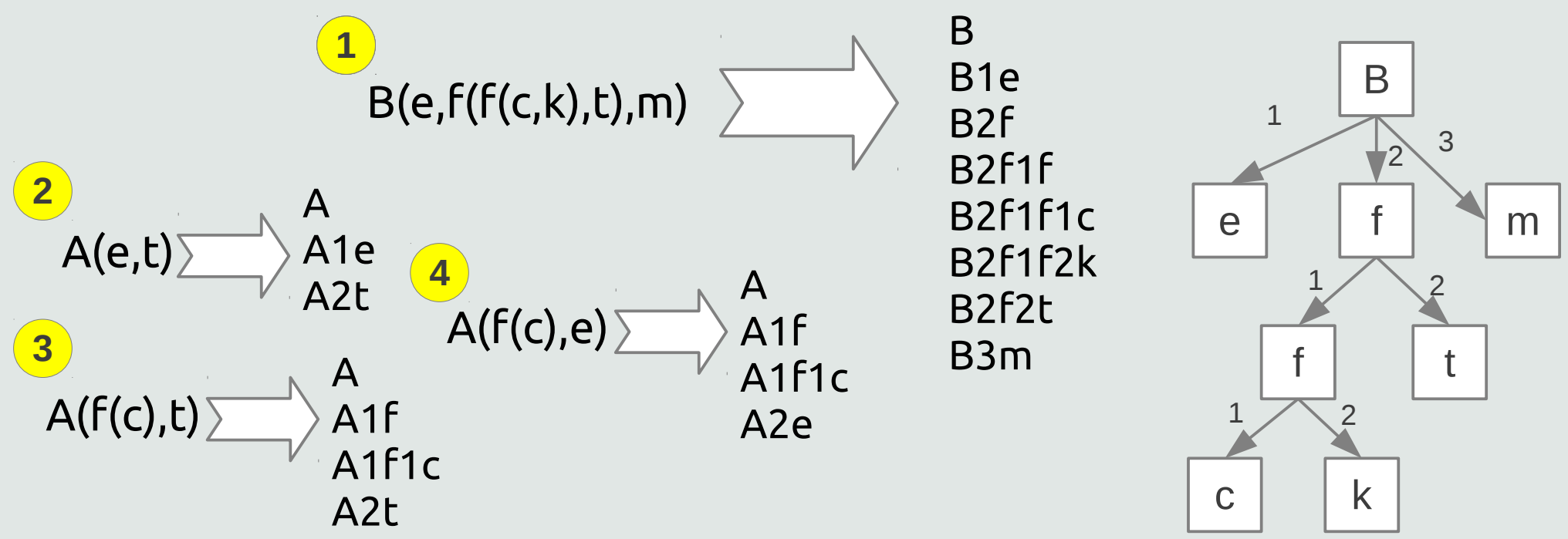
Одно из решений: Индексировать термы. Являются древовидными структурами, и требуют специальных методов индексирования.

Существующие методы индексирования: Индексирование путями (path indexing), индексирование деревом подстановок (substitution tree indexing), их модификации и др. Требуется адаптация для данной системы.

Поскольку, П0-формулы тоже обладают древовидной структурой, возможно обобщение методов индексирования термов до индексирования П0-формул.

Методики/Индексирование/Пример индексирования путями

Каждый терм представляется как список так называемых путей



Каждый путь указывает на множества соответствующих термов

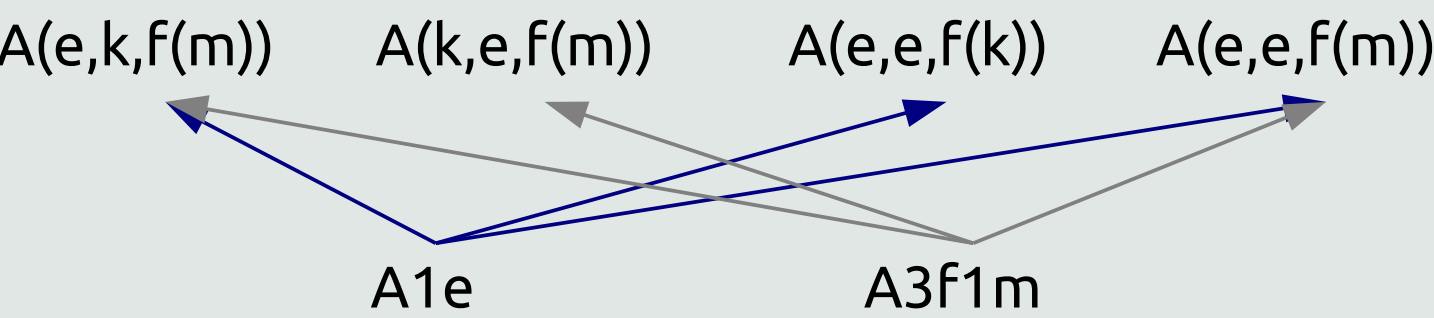
- | | | |
|-------------------|--------------------|-------------------|
| $A: \{2, 3, 4\};$ | $A1f1c: \{3, 4\};$ | $B2f1f2k: \{1\};$ |
| $A2t: \{2, 3\};$ | $A2t: \{2, 3\};$ | $B2f2t: \{1\};$ |
| $A1f: \{3, 4\};$ | $B3m: \{1\};$ | ... |

Методики/Индексирование/Пример индексирования путями

Вопрос:



База:



$$[A1e] \cap [A3f1m] = \{A(e,k,f(m)), A(e,e,f(m))\}$$

Данные два терма являются примерами $A(e,x,f(m))$

Методики/Индексирование/Литература

Индексирование термов является одной из основных тем исследования в области АДТ. Хотя, стали появляться комментарии вроде “Индексирование – необходимое зло”.

Литература:

1. Graf P. Substitution Tree Indexing // Proceedings of the 6th International Conference on Rewriting Techniques and Applications. 1995. P.117-131.
2. Graf P. Term Indexing (Lecture Notes in Computer Science / Lecture Notes in Artificial Intelligence). Springer; 1 edition. March 27, 1996.
3. McCune W. Experiments with Discrimination-Tree indexing and Path Indexing for Term Retrieval // Journal of Automated Reasoning. 1992. V.9(2). P.147-167.
4. Sekar, R., Ramakrishnan, I.V., Voronkov, A. Term Indexing. In: Robinson, A., Voronkov, A. (eds.) Handbook of Automated Reasoning, pp. 1853-1964. MIT Press, Cambridge. 2001.
5. Stickel M. The path-indexing method for indexing terms. / Stickel M. // Technical Note 473, Artificial Intelligence Center, SRI International, RAVENSWOOD AVE., MENLO PARK, CA 94025

Методики/Кэширование

Сохраняются некоторые повторяющиеся вычисления:

1. Процедуры поиска ответных подстановок.
2. Проверка свойств выражений.

Методики/Равенства

Проблема: Крайне неэффективно в явной форме использовать аксиомы равенства (рефлексивность, симметричность, транзитивность, подстановочность).

Ограничения: Необходимо сохранить особенности исчисления П0-формул.

Решение: Использовать системы переписывания термов (СПТ) для генерации эквивалентных фактов в базе.

Пример:

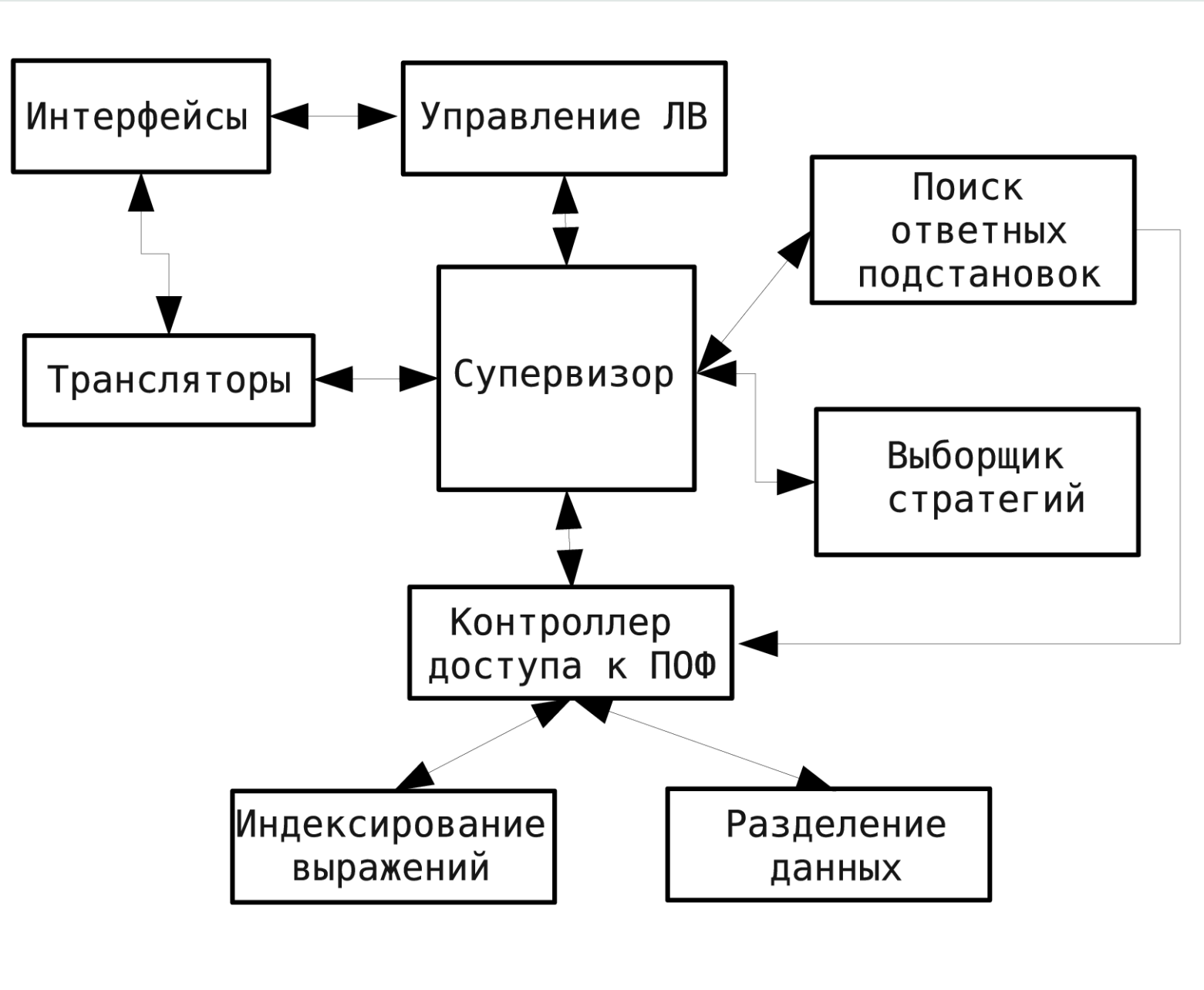
База: $\{A(e), = (e, f(e))\}$

Вопрос: $\forall x: A(f(f(x))) - \{\exists: False$

$e \rightarrow f(e)$ Получаем базу: $\{A(e), A(f(e)), A(f(f(e))) = (e, f(e))\}$

Программная система/Глава 3

Программная система/Архитектура



Программная система/Подсистемы

1. Контроллер доступа к формуле: анализ и модификация формулы.
2. Поиск ответных подстановок.
3. Выборщик стратегий.
4. Управление логическим выводом: стратегии, системные предикаты.
5. Трансляторы: ТРТР \rightarrow П0-формулы, П0-формулы \rightarrow Внутреннее представление
6. Супервизор: доступ ко всему ДСВ, совместимость стратегий, корень всех деревьев ДСВ.

Программная система/Процедуры

1. Процедуры мягкого и жесткого копирования выражений.
2. Процедура мэтчинга термов.
3. Композиция подстановок.
4. Шаг вывода (добавление новых узлов ДСВ).
5. Процедура возврата (backtracking).
6. Выборщик вопросов.
7. Предобработка формул.

Программная система/Транслятор

Приложения и тестирование/Глава 4

Приложения и тестирование/Библиотека TPTP

Библиотека TPTP (Thousands of Problems for Theorem Provers, www.tptp.org). Является де-факто стандартом для тестирования первопорядковых систем АДТ. Регулярно обновляется и растёт.

Тысячи задач, классифицированные по многим критериям: рейтинг (сколько state-of-the-art систем могут решить задачу), количественные характеристики формулы, тип формул (теорема, открытая проблема и т.п.), формат формул (FOF, CNF и др.), предметная область (геометрия, медицина, головоломки, алгебра, менеджмент, верификация, планирование и др.)

Рейтинг. В библиотеку включены лучшие системы АДТ мирового уровня. Если ни одна из систем не может решить задачу, значит рейтинг 1.0, если все решают, то 0.0. К классу `difficult` относятся задачи с рейтингом уже от 0.04.

Приложения и тестирование/Результаты

1. Решен ряд задач с рейтингом от 0.5 до 0.62.
2. Достаточно много задач рейтингом от 0.21 до 0.5
3. Очень много задач с рейтингом меньше 0.21.
4. Хорошие результаты в следующих предметных областях: геометрия, менеджмент, Syntactic, Common sense reasoning
5. Для решаемых ранее задач найден минимальный вывод.
6. Решаются некоторые очень крупные задачи: тысячи подформул-вопросов, тысячи атомов, большая глубина.
7. Некоторые satisfiable задачи распознаются как satisfiable.
8. Вкл/Выкл стратегий наглядно демонстрирует их пользу.
9. Выделены классы задач, наиболее подходящие для П0-формул.

Приложения и тестирование/Сравнение

1. Сильнейшие из сильнейших системы: EP, Vampire, iProver выигрывают в сложности решаемых задач, то есть могут решать некоторые задачи с рейтингом близким к 0.9. Как правило только эти системы и могут решить такие задачи, поэтому у них такой высокий рейтинг.
2. В простых и средних задачах, как правило разница незначительная с переменным успехом. Выигрыш по времени у просто сильнейших систем.
3. Выигрываем у многих других систем.
4. Возможны неожиданные результаты, когда не решается задача с рейтингом 0 или около. Это касается и остальных систем.

Приложения и тестирование/Статистика

Всего тестировалось задач: ~1200.

Решено: ~800.

Решено задач с difficult рейтингом: ~300.

Решено задач с рейтингом >0.5 : 12.

Нерешенные задачи: рейтинг >0.62 ; многие satisfiable задачи; рейтинг решенных задач с равенством низок.

Заключение/Результаты

Поставленная цель достигнута. Задачи решены.

1. Адаптированы некоторые общеупотребимые стратегии: три типа параллельных стратегий, индексирование термов, один вариант разделения памяти (data sharing).
2. Реализованы оригинальные стратегии предложенные именно для исчисления П0-формул: дерево состояний вывода, k,m-ограничение, стратегии направленные на решение проблем неограниченных переменных, ещё варианты разделения памяти.
3. Реализован транслятор формул из языка библиотеки TRTP в язык П0-формул.
4. Показана хорошая производительность системы на задачах из библиотеки TRTP.

Заключение/Дальнейшая работа

1. Дальнейшее улучшение производительности (всегда актуально). Упор на равенства.
2. Работать в направлении решения открытых проблем.
3. Регистрация системы в библиотеке TRTP.