На правах рукописи

ОТПУЩЕННИКОВ ИЛЬЯ ВЛАДИМИРОВИЧ

МЕТОДЫ И СРЕДСТВА ПРЕОБРАЗОВАНИЯ ПРОЦЕДУРНЫХ ОПИСАНИЙ ДИСКРЕТНЫХ ФУНКЦИЙ В БУЛЕВЫ УРАВНЕНИЯ

05.13.11 – Математическое и программное обеспечение
вычислительных машин, комплексов и компьютерных сетей

АВТОРЕФЕРАТ
диссертации на соискание ученой степени
кандидата технических наук

Иркутск – 2011

Работа выполнена в Учреждении Российской академии наук Институте динамики систем и теории управления Сибирского отделения РАН (ИДСТУ СО РАН).

Научный руководитель:	кандидат технических наук, доцент **Семёнов Александр Анатольевич**
Официальные оппоненты:	доктор физико-математичеких наук, профессор **Корольков Юрий Дмитриевич**
	кандидат технических наук, доцент **Тренькаев Вадим Николаевич**
Ведущая организация:	**Институт систем информатики им. А. П. Ершова СО РАН** (г. Новосибирск)

Защита состоится 14 июня 2011 г. в 15:00 ч. на заседании диссертационного совета Д 003.021.01 в ИДСТУ СО РАН по адресу: 664033, г. Иркутск, ул. Лермонтова, 134.

С диссертацией можно ознакомиться в библиотеке и на официальном сайте www.idstu.irk.ru ИДСТУ СО РАН.

Автореферат разослан 13 мая 2011 г.

Ученый секретарь
диссертационного совета,
д.ф.-м.н. А. А. Щеглова

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Диссертация посвящена разработке метода преобразования алгоритмических описаний дискретных функций в булевы уравнения и реализации этого метода в форме программного комплекса.

Актуальность темы исследования. Интенсивное развитие вычислительных технологий, наблюдающееся в последние годы, сделало актуальным проблематику разработки новых алгоритмов решения комбинаторных задач больших размерностей. Следствием этого стало появление новых направлений в вычислительных разделах дискретной математики и математической логики. Как известно, подавляющее число прикладных комбинаторных задач являются NP-трудными в общей постановке. Тем не менее во многих случаях тщательный анализ проблемы и правильный выбор алгоритмов и технологий позволяют найти решение за приемлемое время.

Одним из достижений в исследовании комбинаторных проблем является прогресс в решении систем логических (булевых) уравнений большой размерности. На сегодняшний день существуют алгоритмы, которые позволяют решать системы, содержащие сотни тысяч булевых переменных и уравнений (булевых ограничений). К булевым уравнениям эффективно сводятся многочисленные комбинаторные задачи. Процедура перехода от исходной постановки к системе булевых уравнений называется пропозициональным кодированием. Задачи, сводящиеся к булевым уравнениям, возникают в таких областях, как синтез и верификация схем в микроэлектронике, исследование безопасности коммуникационных протоколов, обоснование корректности программ, криптоанализ, исследование свойств динамических дискретно-автоматных моделей. Отметим, что работ, посвященных пропозициональному кодированию перечисленных задач, немного в сравнении с потоком статей по алгоритмам решения булевых уравнений. Результаты по пропозициональному кодированию представлены в работах С. Прествича, Дж. Маркеса-Сильвы, Ф. Массаччи, Л. Марраро, И. Линс, И. Гента, Г. А. Опарина, В. Г. Богдановой, А. П. Новопашина, Дж. Гу, Н. Эйена, Н. Сорренсона и др.

Многие комбинаторные задачи естественным образом связаны с общей проблемой обращения полиномиально вычислимых дискретных функций, когда по известному алгоритму вычисления функции и слову из области ее значений требуется найти прообраз этого слова. Для решения данной проблемы можно использовать пропозициональный подход, который предполагает пропозициональное кодирование алгоритмов вычисления рассматриваемых функций и применение булевых решателей для решения получающихся систем уравнений. Таким образом, интерес представляет задача разработки и программной реализации аппарата, позволяющего преобразовывать алгорит-

мические описания дискретных функций в булевы уравнения. Для осуществления таких преобразований можно применять различные методы. Наиболее естественным представляется подход, при котором для описания алгоритмов вычисления функций используется некоторый процедурный язык, позволяющий автоматически строить системы булевых уравнений в процессе интерпретации программ, написанных на данном языке. При этом необходимо предусмотреть возможность генерации пропозиционального кода преобразуемого алгоритма в различных формах. На сегодняшний день наиболее успешно решаются уравнения вида КНФ=1 (КНФ — конъюнктивная нормальная форма). Задачи поиска решений уравнений данного типа называются SAT-задачами, а специальные программные средства, предназначенные для их решения — SAT-решателями. В зависимости от выбранного метода решения для итоговых представлений преобразуемых алгоритмов могут использоваться и другие формы: уравнения вида ДНФ=0 (ДНФ — дизъюнктивная нормальная форма), системы алгебраических уравнений над GF(2), «И-НЕ» графы (And-Inverter Graphs) и т.д.

Резюмируя все сказанное, отметим актуальность и практическую значимость задачи разработки и программной реализации метода преобразования алгоритмических описаний дискретных функций, выполняемых на специальном процедурном языке, в булевы уравнения различных типов.

Цель и задачи исследования. Целью диссертации является разработка общего подхода к проблеме преобразования алгоритмов вычисления всюду определенных дискретных функций в булевы уравнения, а также создание на этой основе инструментального средства, предназначенного для исследования широкого класса функций, возникающих в приложениях.

Для достижения указанной цели были поставлены следующие задачи.

1. Разработать общую методологию преобразования высокоуровневых описаний алгоритмов вычисления дискретных функций в булевы уравнения; сформулировать общие принципы и описать базовые механизмы таких преобразований.

2. Разработать проблемно-ориентированный язык процедурного типа, предназначенный для описания алгоритмов вычисления всюду определенных дискретных функций и обладающий семантикой, обеспечивающей корректное построение булевых уравнений.

3. Разработать и реализовать структуры данных и алгоритмы, позволяющие автоматизировать преобразования в булевы уравнения процедур вычисления дискретных функций, описанных на специально созданном языке.

4. Создать программный комплекс, интегрирующий все разработанные алгоритмы и структуры данных, который осуществляет преобразование про-

грамм вычисления дискретных функций, оптимизирует получаемые булевы структуры, а также выполняет проверку их корректности.

5. С использованием этого программного комплекса создать набор шаблонов, хранящих пропозициональные коды ряда функциональных примитивов, широко используемых при описании поведения дискретных систем.

Методы и инструменты исследования. Теоретическая часть исследований базируется на теории множеств, методах дискретной математики, теории булевых функций, теории вычислительной сложности, криптографии, теории процедурных языков программирования. Экспериментальная часть использует современные средства разработки программного обеспечения (язык программирования C++, среда разработки и сборки приложений Visual Studio 2008 Express Edition).

Научная новизна. В диссертации предложен новый подход к преобразованию процедурных описаний дискретных функций в булевы уравнения. Для описания алгоритмов вычисления дискретных функций разработан новый проблемно-ориентированный язык TA, обладающий C-подобным синтаксисом и оригинальной семантикой, которая обеспечивает построение систем булевых уравнений в процессе интерпретации TA-программ. Для преобразования TA-программ в булевы уравнения разработаны новые алгоритмы и структуры данных. Все разработанные алгоритмы интегрированы в программный комплекс Transalg, с применением которого были построены новые пропозициональные коды ряда криптографических преобразований и оптимизационных задач с псевдобулевыми ограничениями.

Достоверность результатов. Достоверность полученных в работе результатов обеспечивается использованием апробированых методов и средств, строгостью математических выкладок и подтверждается результатами вычислительных экспериментов.

Теоретическая и практическая значимость работы заключается в предложенной общей методологии преобразования процедурных описаний дискретных функций в булевы уравнения, а также в возможности практического использования данной методологии, реализованной в виде программного комплекса, для исследования различных свойств дискретных систем.

Соответствие диссертации паспорту научной специальности. В диссертации представлен метод для осуществления преобразований алгоритмов, вычисляющих дискретные функции, в булевы уравнения, разработан проблемно-ориентированный язык программирования (язык TA), предназначенный для описания алгоритмов вычисления дискретных функций. Семантика языка TA обеспечивает эффективное построение булевых уравнений в процессе интерпретации TA-программ. Разработанные алгоритмы и струк-

туры данных реализованы в виде программного комплекса. Таким образом, тематика диссертации соответствует пунктам 1, 2 области исследований специальности 05.13.11.

Апробация работы. Результаты диссертации докладывались и обсуждались на 5-й Международной научной конференции «Параллельные вычислительные технологии» (Москва, 2011 г.), на IX Всероссийской школе-семинаре с международным участием Sibecrypt-10 (Тюмень, 2010 г.), на XIV Байкальской Международной школе-семинаре «Методы оптимизации и их приложения» (Северобайкальск, 2008 г.), на IX и X Всероссийских конференциях молодых ученых «Математическое моделирование и информационные технологии» (Иркутск, 2007 г., 2009 г.), на ежегодных конференциях «Ляпуновские чтения и презентация информационных технологий» (Иркутск, 2007–2010 гг.), а также на семинарах Института динамики систем и теории управления СО РАН, Института цитологии и генетики СО РАН, Института математики им. С. Л. Соболева СО РАН, Института систем информатики им. А. П. Ершова СО РАН.

Результаты диссертации были получены в рамках следующих проектов:

– проект СО РАН «Интеллектные методы и инструментальные средства создания и анализа интегрированных распределенных информационно-аналитических и вычислительных систем для междисциплинарных исследований с применением ГИС, GRID и Веб-технологий» (№ гос. регистрации 01.2.00708582), 2007–2009 гг.;

– проекта СО РАН «Интеллектные методы автоматизации решения задач в параллельных и распределенных вычислительных средах» (№ гос. регистрации 01.2.01001348), 2010–2011 гг.;

– грант РФФИ № 07-01-00400-а «Характеризация сложности обращения дискретных функций в задачах криптографии и интервального анализа»;

– грант РФФИ № 11-07-00377-а «Разработка параллельных алгоритмов решения булевых уравнений и их реализация в GRID-системах»;

– грант Президента РФ НШ-1676.2008.1.

Публикации и личный вклад автора. Наиболее значимые результаты диссертации представлены в работах [1–9], в число которых входят 4 статьи в журналах, рекомендованных ВАК РФ, 1 статья в тематическом сборнике, 2 полных текста докладов в материалах международных конференций, а также свидетельство о регистрации программы для ЭВМ.

Результаты главы 1 опубликованы в работах [4, 5]; результаты главы 2 опубликованы в [1–3]; результаты главы 3 опубликованы в [1–3, 6–8].

Все результаты, выносимые на защиту, получены автором лично. В основных публикациях по теме диссертации научному руководителю принадле-

жат постановки задач и некоторые теоретические результаты. В работах [2], [6] и [8] Заикину О. С. принадлежат результаты по распараллеливанию схем решения SAT-задач. Все результаты по разработке и реализации алгоритмов сведения комбинаторных задач к булевым уравнениям принадлежат автору.

Структура работы. Диссертация состоит из введения, трех глав, заключения, списка литературы из 121 наименования и четырех приложений. Объем диссертации — 128 страниц. Диссертация содержит 19 рисунков и 7 таблиц.

ОСНОВНОЕ СОДЕРЖАНИЕ РАБОТЫ

В первой главе приводятся необходимые сведения из теории дискретных функций, описываются основные принципы преобразования программ для формальных вычислительных моделей в булевы уравнения.

Рассматриваются алгоритмически вычислимые за полиномиальное от n время семейства функций вида

$$f_n : \{0,1\}^n \to \{0,1\}^*, dom\, f_n = \{0,1\}^n, n \in \mathbb{N}.$$

Пропозициональное кодирование алгоритма, вычисляющего произвольную функцию f_n, состоит в построении символьного описания всех возможных эволюций соответствующего вычисления на произвольных входах из $\{0,1\}^n$. Схематично данный процесс можно представить в следующем виде:

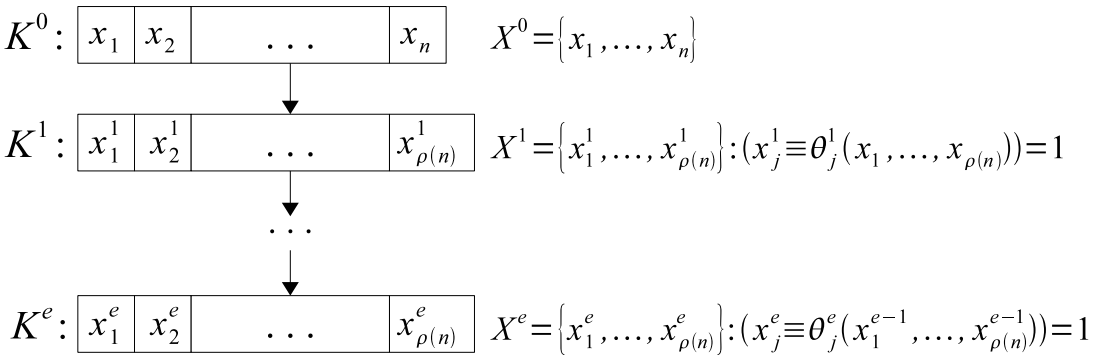

Рис. 1. Последовательность конфигураций памяти вычислительного устройства

Здесь $K^0, K^1, \ldots, K^e$ — последовательность конфигураций памяти вычислительного устройства (K^0, K^e — начальная и заключительная конфигурации). Каждой конфигурации K^i ставится в соответствие множество булевых переменных X^i, $i = 0, \ldots, e$. Переменные этих множеств связаны между собой системами булевых уравнений[1].

[1] Семенов А. А. Трансляция алгоритмов вычисления дискретных функций в выражения пропозициональной логики // Прикладные алгоритмы в дискретном анализе. Серия: дискретный анализ и информатика, вып. 2. 2008. Иркутск: Изд-во ИГУ. С. 70–98.

Здесь же приведен краткий обзор основных методов решения булевых уравнений. В заключительной части главы дано обоснование актуальности разработки программного комплекса, предназначенного для преобразования в булевы уравнения описаний алгоритмов вычисления дискретных функций, выполненных на специализированном процедурном языке.

Во второй главе описаны все базовые алгоритмы и структуры данных, используемые в процессе преобразований процедурных описаний дискретных функций в булевы уравнения; приведена архитектура программного комплекса Transalg и дано описание основных его функциональных возможностей.

В разделе 2.1 представлен проблемно-ориентированный язык (язык ТА), предназначенный для описания алгоритмов, вычисляющих дискретные функции, с целью их преобразования в булевы уравнения. Язык ТА является процедурным языком с блочной структурой и С-подобным синтаксисом. Каждый блок — это список инструкций ТА-программы. Программа на языке ТА представляет собой набор определений функций, а также объявлений и определений глобальных переменных и констант. В языке ТА реализованы все основные примитивные конструкции, характерные для процедурных языков программирования (объявление/определение переменной или массива переменных; определение именованных констант; определение функции; оператор присваивания; составной оператор; оператор условного перехода; оператор цикла; оператор вызова функции; оператор возврата из функции).

Язык ТА относится к языкам программирования с операционной семантикой[2]. Основная его особенность заключается в том, что процесс интерпретации произвольной ТА-программы является по своей сути ее символьным исполнением[3]. Символьное исполнение предполагает расширенную семантику, в соответствии с которой выполняемая программа по множеству символов, кодирующих вход рассматриваемой функции f_n, $f_n : \{0,1\}^n \to \{0,1\}^*$, $dom\, f_n = \{0,1\}^n$, строит систему булевых уравнений $S(f_n)$. Подстановка в $S(f_n)$ произвольного $y \in range\, f_n$ дает совместную систему $S_y(f_n)$, из произвольного решения которой можно эффективно выделить такое значение $x \in \{0,1\}^n$, что $f_n(x) = y$. Для нахождения решений системы $S_y(f_n)$ можно использовать современные булевы решатели.

Раздел 2.2 посвящен описанию основных механизмов интерпретации ТА-программ. Операционная (интерпретационная) семантика языка ТА описывается через ввод в рассмотрение абстрактной машины языка ТА (ТА-машины). Произвольная конфигурация ТА-машины фиксирует момент интерпретации

[2] Касьянов В. Н., Поттосин И. В. Методы построения трансляторов. — Новосибирск: Наука, 1986. 344 С.

[3] King J. C. Symbolic execution and program testing // Communications of the ACM. 1976. Vol. 19, №7. Pp. 385–394.

некоторой инструкции ТА-программы. При этом в памяти ТА-машины отображается информация о связи переменных, фигурирующих в тексте ТА-программы, с переменными, присутствующие в системе булевых уравнений, и с логическими выражениями (термами).

Следует особо подчеркнуть, что переменные, встречающиеся в тексте ТА-программы (далее «переменные программы»), являются идентификаторами областей памяти ТА-машины. Переменные, встречающиеся в системе булевых уравнений, — это булевы переменные, называемые далее переменными кода. Переменные программы и переменные кода связываются в памяти ТА-машины при помощи специальной структуры данных `var_object`.

Основным типом данных, используемым в зыке ТА, является тип `bit`. Как правило, переменная, имеющая этот тип, в каждой конфигурации ТА-машины связана с некоторой переменной кода и некоторым термом. Более точно, значениями переменной x, имеющей тип `bit`, в памяти ТА-машины являются слова вида

$$x|T(u_1, \ldots, u_k).$$

Здесь x — переменная кода, $T(u_1, \ldots, u_k)$ — терм исчисления высказываний над переменными кода (булевыми переменными) $u_1, \ldots, u_k$ («|» — разделяющий символ). Допустимыми операциями над переменными, имеющими тип `bit`, являются произвольные операции исчисления высказываний.

Процессу символьного исполнения ТА-программы соответствует последовательность конфигураций ТА-машины: $K^0, \ldots, K^e$ (K^0 — начальная, K^e — заключительная конфигурации). Для каждого $i \in \{0, \ldots, e\}$ обозначим через X^i множество переменных кода, связанных с переменными программы в конфигурации K^i. Таким образом,

$$X = \bigcup_{i=0}^{e} X^i$$

— это множество всех переменных кода, введенных в процессе символьного исполнения рассматриваемой ТА-программы. Множество X^0 образовано булевыми переменными, которые кодируют входные данные дискретной функции. Множество переменных кода, которые кодируют выход этой функции, будем обозначать через Y ($Y \subset X$).

Переменные программы, связанные в конфигурации K^0 с переменными кода из множества X^0, объявляются в тексте ТА-программы с атрибутом `_in`. Переменные программы, которые в процессе символьного исполнения связываются с переменными кода из множества Y, объявляются с атрибутом `_out`. Переменные программы с атрибутами `_in` и `_out` — это переменные с глобальной областью видимости.

Кроме типа `bit`, в языке ТА используются вспомогательные типы: целочисленный тип `int`, представляющий четырехбайтовое знаковое целое число, а также тип `bool` — целое число из множества $\{0, 1\}$.

Далее на примере простейших криптографических примитивов (регистров сдвига с линейной обратной связью, РСЛОС) демонстрируется необходимость разработки техники сокращения избыточности кода преобразуемого алгоритма. Для этой цели вводится специальный словарь термов. Данный словарь, который обозначается через Σ, содержит термы над переменными кода преобразуемой программы. Словарь Σ является динамически расширяемым. В начальном состоянии в Σ находятся только переменные множества X^0 (то есть переменные, кодирующие входную информацию). В дальнейшем каждый новый терм, попадающий в словарь, является результатом преобразования некоторой операции присваивания следующего вида:

$$\mathrm{x} = \Phi(\mathrm{x}_{j_1}, \ldots, \mathrm{x}_{j_r}).$$

Здесь x — переменная программы, которая через структуру `var_object` связана с некоторой переменной кода, а $\Phi(\mathrm{x}_{j_1}, \ldots, \mathrm{x}_{j_r})$ — выражение над переменными программы. Результатом интерпретации выражения $\Phi(\mathrm{x}_{j_1}, \ldots, \mathrm{x}_{j_r})$ является некоторый терм $\varphi(x_{j_1}, \ldots, x_{j_k})$ над переменными кода. Затем осуществляется проверка словаря Σ на предмет наличия в нем терма φ. Если $\varphi \notin \Sigma$, то данный терм связывается с новой переменной кода $\tilde{x}$ и добавляется в словарь Σ. При этом в систему булевых уравнений добавляется уравнение вида

$$(\tilde{x} \equiv \varphi(x_{j_1}, \ldots, x_{j_k})) = 1.$$

Если же $\varphi \in \Sigma$, это означает, что представленная данным термом информация уже учтена в пропозициональном коде программы и ей соответствует отдельная переменная кода x'. В этом случае переменная программы x связывается с переменной кода x' через структуру `var_object`. Данный прием позволяет избегать ввода булевых переменных, кодирующих одну и ту же информацию.

Подраздел 2.2.2 посвящен описанию процесса интерпретации основных операторов языка ТА. Наиболее трудным этапом этого процесса является интерпретация операторов условного перехода.

Оператор условного перехода — это оператор вида (в псевдокоде)

if $\Phi(\mathrm{x}_{j_1}, \ldots, \mathrm{x}_{j_r})$ then Оператор_1;
else Оператор_2;

здесь $\Phi(\mathrm{x}_{j_1}, \ldots, \mathrm{x}_{j_r})$ — выражение над переменными программы, а Оператор_1 и Оператор_2 — произвольные операторы языка ТА (оператор присваивания,

оператор условного перехода, оператор цикла, оператор вызова функции или составной оператор). Интерпретация оператора условного перехода начинается с построения по выражению Φ терма φ над переменными кода. После этого выполняется интерпретация операторов Оператор_1 и Оператор_2, в процессе которой строятся два альтернативных множества конфигураций ТА-машины Δ_1 и Δ_2. Данные множества образованы конфигурациями, в которых зафиксированы изменения в памяти ТА-машины, являющиеся результатами выполнения операторов Оператор_1 и Оператор_2, соответственно. Затем интерпретатор определяет множество переменных программы, которые изменили свое значение хотя бы в одной из конфигураций, представленных в Δ_1 и Δ_2. Пусть x — произвольная такая переменная, изменяющая свое значение по крайней мере в одной из конфигураций K_1 и K_2 ($K_1 \in \Delta_1$, $K_2 \in \Delta_2$). Пусть значениями x в K_1 и K_2 являются слова

$$\tilde{x}|\delta_1,\ \tilde{x}|\delta_2,$$

соответственно. Здесь $\tilde{x}$ — новая переменная кода, δ_1 и δ_2 — некоторые термы над переменными кода. В результате интерпретации оператора условного перехода с переменной программы x связывается переменная кода $\tilde{x}$, в словарь термов Σ добавляется терм

$$\varphi \cdot \delta_1 \vee \overline{\varphi} \cdot \delta_2,$$

а в систему булевых уравнений добавляется уравнение

$$(\tilde{x} \equiv \varphi \cdot \delta_1 \vee \overline{\varphi} \cdot \delta_2) = 1.$$

Далее рассматривается конструкция из нескольких вложенных операторов условного перехода (в псевдокоде).

```
if Φ1(...) then Оператор_1;
else if Φ2(...) then Оператор_2;
    ...
  else if Φn(...) then Оператор_n;
  else Оператор_n+1;
```

В соответствии со сказанным выше каждое выражение Φ_i, $i = 1, \ldots, n$, преобразуется в терм φ_i над множеством переменных кода, строится $n+1$ альтернативное множество $\Delta_1, \ldots, \Delta_{n+1}$ конфигураций ТА-машины и определяется множество переменных программы, каждая из которых изменила свое значение хотя бы в одной из конфигураций, представленных в $\Delta_1, \ldots, \Delta_{n+1}$. Пусть

x — такая переменная и ее значениями в соответствующих $n+1$ альтернативных конфигурациях ТА-машины являются слова

$$\tilde{x}|\delta_1, \ldots, \tilde{x}|\delta_{n+1}.$$

В результате интерпретации оператора условного перехода с переменной программы x связывается переменная кода $\tilde{x}$, в словарь термов Σ добавляется терм

$$\varphi_1 \cdot \delta_1 \vee \overline{\varphi}_1 \cdot \varphi_2 \cdot \delta_2 \vee \ldots \vee \overline{\varphi}_1 \cdot \overline{\varphi}_2 \cdot \ldots \cdot \overline{\varphi}_{n-1} \cdot \varphi_n \cdot \delta_n \vee \overline{\varphi}_1 \cdot \ldots \cdot \overline{\varphi}_n \cdot \delta_{n+1},$$

а в систему булевых уравнений добавляется уравнение

$$(\tilde{x} \equiv \varphi_1 \cdot \delta_1 \vee \overline{\varphi}_1 \cdot \varphi_2 \cdot \delta_2 \vee \ldots \vee \overline{\varphi}_1 \cdot \overline{\varphi}_2 \cdot \ldots \cdot \overline{\varphi}_{n-1} \cdot \varphi_n \cdot \delta_n \vee \overline{\varphi}_1 \cdot \ldots \cdot \overline{\varphi}_n \cdot \delta_{n+1}) = 1.$$

Также в этом подразделе кратко описан процесс интерпретации остальных операторов языка ТА (оператора цикла, оператора вызова функции, оператора возврата из функции).

В подразделе 2.2.4 приведены основные механизмы, используемые при преобразовании в булевы уравнения целочисленных операций. Необходимость таких преобразований возникает при работе с комбинаторными задачами, в которых фигурируют функции над целыми числами (например, в некоторых дискретных моделях генных сетей). В процессе преобразования целые числа представляются булевыми массивами, при этом в тексте ТА-программы присутствуют операции над буквенными обозначениями чисел. Например, результатом преобразования ТА-программы

```
__in bit a[n];
__in bit b[n];
__out bit c[n+1];
void main(){
    c = a + b;
}
```

будет система булевых уравнений следующего вида:

$$\begin{cases} (c_0 \equiv a_0 \oplus b_0) = 1 \\ (p_0 \equiv a_0 \cdot b_0) = 1 \\ (p_j \equiv \mathrm{maj}(a_j, b_j, p_{j-1})) = 1,\ j = \overline{1, n-1} \\ (c_i \equiv a_i \oplus b_i \oplus p_{i-1}) = 1,\ i = \overline{1, n-1} \\ (c_n \equiv p_{n-1}) = 1. \end{cases}$$

Запись «$\mathrm{maj}(x, y, z)$» обозначает терм $x \cdot y \vee x \cdot z \vee y \cdot z$; для кодирования битов переноса вводятся новые переменные кода p_i, $i = \overline{0, n-1}$. Аналогичным образом выглядят механизмы преобразования операций умножения, вычитания

и сравнения пар целых чисел. Помимо перечисленных реализованы процедуры преобразования разнообразных сдвиговых операций, используемых в криптографических функциях.

В разделе 2.3 приведено описание программного комплекса Transalg, в котором были реализованы все представленные во второй главе алгоритмы и структуры данных. Здесь же описаны различные форматы выходных данных, генерируемых комплексом Transalg.

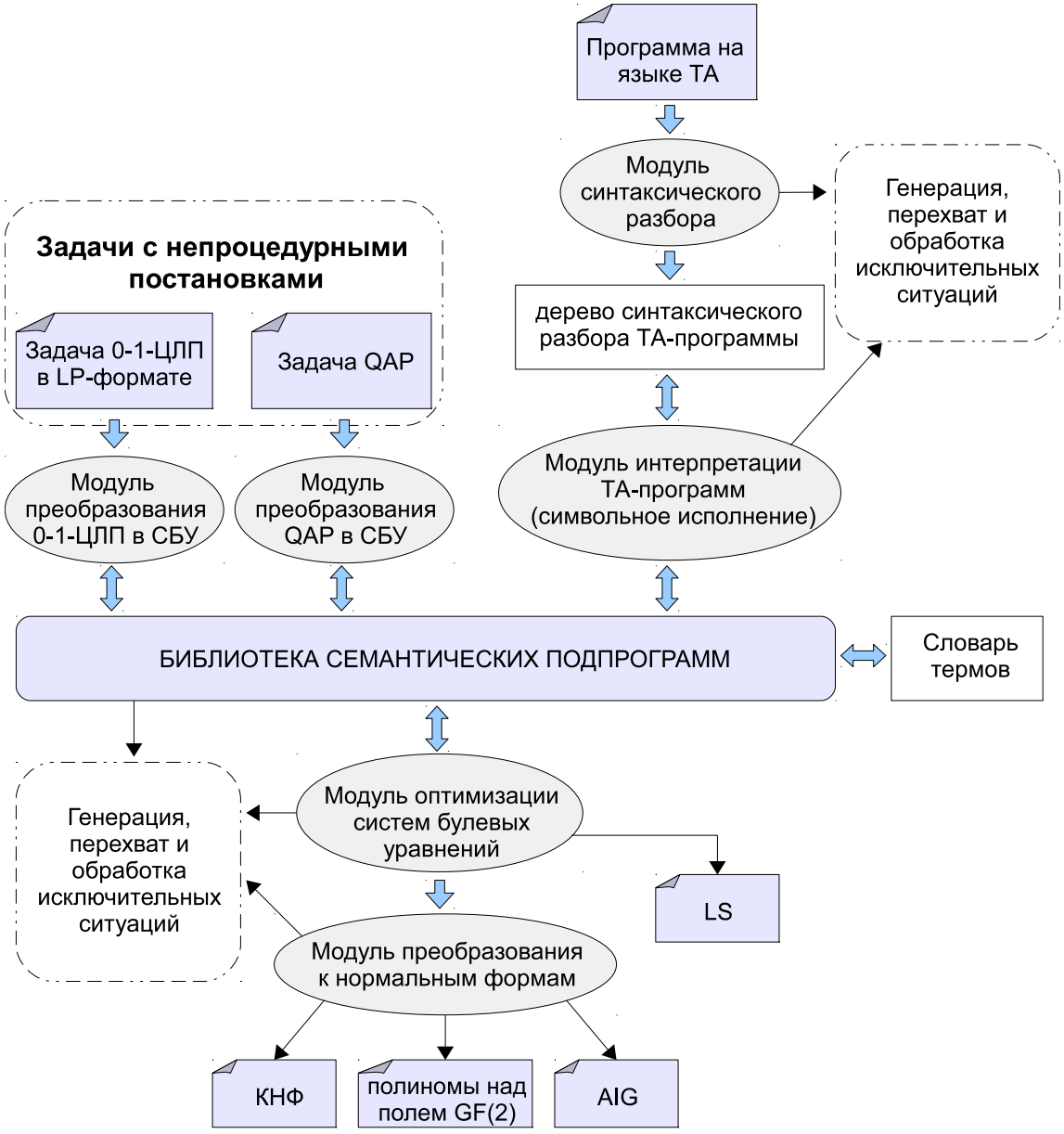

Рис. 2. Архитектура программного комплекса Transalg

Основное назначение комплекса Transalg состоит в преобразовании алгоритмов вычисления дискретных функций в булевы уравнения. Процедурные описания рассматриваемых функций в виде программ на языке ТА поступа-

ют на вход модулю синтаксического разбора. При синтаксическом разборе ТА-программ используется стандартная для теории компиляции техника.

Разработанные в диссертации алгоритмы и структуры данных, обеспечивающие символьное исполнение ТА-программ, реализованы в виде библиотеки семантических подпрограмм. В данной библиотеке присутствуют подпрограммы для интерпретации каждого оператора и операции языка ТА. Символьное исполнение ТА-программы включает в себя обход дерева синтаксического разбора, идентификацию конструкций языка ТА и вызовы соответствующих семантических подпрограмм, которые при помощи структур `var_object` используют информацию о переменных ТА-программы и взаимодействуют со словарем термов. Результатом работы семантических подпрограмм является некоторый фрагмент пропозиционального кода рассматриваемого алгоритма, то есть система булевых уравнений.

Помимо процедурных описаний функций на языке ТА комплекс Transalg позволяет работать с непроцедурными постановками задач 0-1-ЦЛП (0-1-целочисленное линейное программирование) и QAP (квадратичная задача о назначениях). Подробное описание механизмов преобразования данных задач в булевы уравнения приведено в третьей главе диссертации (раздел 3.3).

В комплексе Transalg переход от систем булевых уравнений к уравнениям вида КНФ=1 осуществляется при помощи преобразований Цейтина[4], дополненных процедурами минимизации булевых функций в классе КНФ[5]. Кроме этого, предусмотрен вывод пропозиционального кода алгоритма в форме системы полиномиальных уравнений над GF(2), а также возможность построения И-НЕ-графа, представляющего алгоритм вычисления рассматриваемой функции в форме схемы из функциональных элементов над базисом $\{\&, \neg\}$.

В третьей главе диссертации представлены результаты применения программного комплекса Transalg к решению разнообразных комбинаторных задач.

В разделе 3.1 приведены результаты преобразования в булевы уравнения (и в конечном счете в SAT-задачи) ряда криптографических функций. Рассмотрены генераторы ключевого потока Брюера, Рюппеля и Гиффорда, а также генератор, используемый в системе шифрования A5/1. Полученный комплексом Transalg пропозициональный код генератора A5/1 позволил осуществить его успешный логический криптоанализ в GRID-системе[6].

---

[4] Цейтин Г. С. О сложности вывода в исчислении высказываний // Записки научных семинаров ЛОМИ АН СССР. 1968. Т.8. С. 234–259.

[5] Минимизация булевых функций в комплексе Transalg осуществляется при помощи свободно распространяемой утилиты Espresso (http://embedded.eecs.berkeley.edu/pubs/downloads/espresso).

[6] Посыпкин М. А., Заикин О. С., Беспалов Д. В., Семенов А. А. Решение задач криптоанализа поточных шифров в распределенных вычислительных средах // Труды ИСА РАН. 2009. Т. 46. С. 119–137.

В подразделе 3.1.4 при помощи комплекса Transalg решается задача преобразования в SAT алгоритма блочного шифрования DES. Особо подчеркнем, что данная задача положила начало целому направлению, известному сегодня как «логический криптоанализ». Одним из базовых примитивов шифра DES являются перестановки. Перестановки в DES задаются таблицами натуральных чисел, которые не являются секретными. Произвольная перестановка применяется к некоторому множеству битов обрабатываемого слова. При пропозициональном кодировании операции перестановки Transalg не создает новых переменных кода, поскольку ему достаточно обновить связи переменных программы с уже существующими элементами словаря термов Σ. Данный факт означает, что операции перестановки не вносят в пропозициональный код алгоритма новой информации, никак не усложняя задачу логического криптоанализа. В результате применения комплекса Transalg к кодированию алгоритма DES был получен (см. таблицу 1) пропозициональный код (в КНФ), существенно более экономный, чем код, приведенный в работе Ф. Массаччи и Л. Марраро[7].

Таблица 1. Кодирование процесса шифрования алгоритмом DES одного блока открытого текста длиной 64 бита

Программный комплекс Transalg				F. Massacci, L. Marraro	
Без минимизации		Минимизация			
Переменные	Дизъюнкты	Переменные	Дизъюнкты	Переменные	Дизъюнкты
1912	37888	1912	26400	10336	61935

В разделе 3.2 перечислены результаты применения комплекса Transalg к исследованию поведения дискретных моделей генных сетей. Рассматривались задачи поиска циклов для одного класса дискретных автоматов, моделирующих поведение генных сетей[8]. Комплекс Transalg генерировал SAT-задачи по TA-программам, описывающим динамику функционирования генной сети с дополнительными условиями на возникновение цикла определенной длины. Получаемые SAT-задачи решались известным SAT-решателем[9] MiniSat 2.0. Для всех тестов либо были найдены циклы (длины до 10), либо было установлено, что таких циклов нет.

Заключительный раздел 3.3 третьей главы посвящен преобразованию в SAT оптимизационных задач с псевдобулевыми ограничениями. Данная

[7] Massacci F., Marraro L. Logical Cryptanalysis as a SAT Problem // J. of Automated Reasoning. 2000. Vol. 24, no. 1–2. Pp. 165–203.

[8] Евдокимов А. А., Кочемазов С. Е., Семенов А. А. Применение символьных вычислений к исследованию дискретных моделей некоторых классов генных сетей // Вычислительные технологии. 2011. Т. 16, №1. С. 30–47.

[9] http://minisat.se/MiniSat.html

тематика является сравнительно новой и интенсивно исследуемой. Преобразования осуществлялись при помощи дополнительного модуля комплекса Transalg, созданного специально для рассматриваемого класса задач. Основным авторским компонентом примененных преобразований являются процедуры декомпозиции преобразуемых выражений с последующим использованием минимизации булевых функций. Получаемые пропозициональные коды задач из семейства 0-1-ЦЛП (целочисленное линейное программирование) сравнивались с кодами, полученными известной программой[10] MiniSat+. Следует отметить, что коды, построенные комплексом Transalg, содержали меньшее число переменных, но большее число дизъюнктов, чем коды, построенные программой MiniSat+. По времени решения соответствующие SAT-задачи оказались сопоставимы.

В заключительной части раздела описано преобразование в булевы уравнения известной комбинаторной проблемы — квадратичной задачи о назначениях (QAP). Данная задача известна как «одна из наиболее труднорешаемых комбинаторных проблем»[11]. Следует отметить, что эффективность «традиционных» (комбинирующих принципы ветвей, границ и отсечений) методов решения задач дискретной оптимизации на QAP невысока. Это требует разработки в применении к QAP новых подходов и алгоритмов. С использованием комплекса Transalg в диссертации были осуществлены сведения задач из семейства QAP к итеративному поиску решений SAT-задач.

В заключении сформулированы основные результаты диссертационной работы.

В приложениях к диссертации содержится грамматика языка TA в нотации Бэкуса-Наура, тексты некоторых TA-программ (программа шифра DES, программа генератора A5/1 и программа, описывающая функционирование генной сети).

РЕЗУЛЬТАТЫ, ВЫНОСИМЫЕ НА ЗАЩИТУ

1. Разработан метод преобразования алгоритмов вычисления дискретных функций в булевы уравнения, который может использоваться для анализа различных свойств алгоритмически вычислимых функций за счет перехода к булевым уравнениям и последующего применения булевых решателей.

2. Создан проблемно-ориентированный язык описания алгоритмов вычисления дискретных функций (язык TA), который обладает семантикой,

[10] Een N., Sorensson N. Translating Pseudo-Boolean Constraints into SAT // J. on Satisfiability, Boolean Modeling and Computation. 2006. Vol. 2. Pp. 1–25.

[11] Cela E. The Quadratic Assignment Problem. Theory and Algorithms. — Kluwer Academic Publishers. 1998. 287 p.

обеспечивающей эффективное построение булевых уравнений в процессе интерпретации ТА-программ.

3. Разработаны алгоритмы и структуры данных, используемые при преобразовании в булевы уравнения инструкций ТА-программ, вычисляющих дискретные функции.

4. Создан программный комплекс Transalg, основное назначение которого состоит в генерации булевых уравнений по процедурным описаниям дискретных функций, выполненным на языке ТА.

СПИСОК ОСНОВНЫХ ПУБЛИКАЦИЙ ПО ТЕМЕ ДИССЕРТАЦИИ

1. Отпущенников И. В., Семенов А. А. Технология трансляции комбинаторных проблем в булевы уравнения // Прикладная дискретная математика. 2011. № 1. С. 96–115.

2. Заикин О. С., Отпущенников И. В., Семенов А. А. Параллельные алгоритмы решения проблемы выполнимости в применении к оптимизационным задачам с булевыми ограничениями // Вычислительные методы и программирование: Новые вычислительные технологии (Электронный научный журнал). 2011. Т. 12. С. 205–212.

3. Отпущенников И. В., Семенов А. А. Преобразования алгоритмов вычисления дискретных функций в булевы уравнения // Известия ИГУ. Серия: математика. 2011. Том 4. № 1. С. 83–96.

4. Семенов А. А., Отпущенников И. В., Кочемазов С. Е. Пропозициональный подход в задачах тестирования дискретных автоматов // Современные технологии. Системный анализ. Моделирование. 2009. № 4. С. 48–56.

5. Семенов А. А., Отпущенников И. В. Об алгоритмах обращения дискретных функций из одного класса // Прикладные алгоритмы в дискретном анализе. Серия: дискретный анализ и информатика, вып. 2. Изд-во ИГУ. 2008. С. 127–156.

6. Семенов А. А., Заикин О. С., Отпущенников И. В., Буров П. С. О некоторых особенностях задач обращения дискретных функций // Труды XIV Байкальской Международной школы-семинара «Методы оптимизации и их приложения». — Иркутск: ИСЭМ СО РАН. 2008. Т. 1. С. 498–505.

7. Отпущенников И. В., Семенов А. А. Программная трансляция алгоритмов в пропозициональную логику применительно к комбинаторным задачам // Прикладная дискретная математика. Приложение. 2010. № 3. С. 81–82.

8. Заикин О. С., Отпущенников И. В., Семенов А. А. Параллельные алгоритмы решения SAT в применении к оптимизационным задачам с булевыми ограничениями // Труды V Международной конференции «Параллельные вычислительные технологии» (ПАВТ 2011). Москва, МГУ. 2011. С. 501–508.

9. Отпущенников И. В., Семенов А. А. Инструментальное средство трансляции алгоритмов вычисления дискретных функций в выражения исчисления высказываний Transalg 1.0. Свидетельство о государственной регистрации программы для ЭВМ № 2011611151 (03.02.2011).

Редакционно-издательский отдел
Института динамики систем и теории управления СО РАН
664033, Иркутск, ул. Лермонтова, д. 134
Подписано к печати 11.05.2011
Формат бумаги 60 × 84 1/16, объем 1,2 п. л.
Заказ 10. Тираж 100 экз.

Отпечатано в ИДСТУ СО РАН