

На правах рукописи

Иванова Галина Сергеевна

**МЕТОДОЛОГИЯ И СРЕДСТВА РАЗРАБОТКИ
АЛГОРИТМОВ РЕШЕНИЯ ЗАДАЧ АНАЛИЗА И СИНТЕЗА
СТРУКТУР ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ И
УСТРОЙСТВ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ**

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

**Автореферат
диссертации на соискание ученой степени
доктора технических наук**



Москва - 2007

Официальные оппоненты:

Доктор технических наук,
профессор
Жданов Владимир Сергеевич

Доктор технических наук,
профессор
Юркевич Евгений Владимирович

Доктор технических наук,
профессор
Солодовников Игорь Владимирович

Ведущая организация

~~Акционерное~~ акционерное общество

ий центр
ной техники»

Иванова Г.С.
Методология и средства
разработки алгоритмов
решения задач анализа и
синтеза структур программ...
2007

Защита диссертации

1 130 часов на ударст-

3a0
Be:

21795611

ла'

ВОЗВРАТИТЕ КНИГУ НЕ ПОЗЖЕ

ВЫСЫ-

обозначенного здесь срока

О госу-

да

ДИС

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность темы. К задачам структурного анализа и синтеза относят задачи определения и исследования свойств некоторого варианта структуры объекта, под которой понимают совокупность составляющих его элементов и связей между ними. К этому типу задач относятся многие задачи проектирования ЭВМ, телекоммуникационных систем и сетей, задачи оптимизации транспортных потоков и др. Среди наиболее известных задач данного класса следует назвать задачу назначения, позиционирования, разрезания (декомпозиции) функциональных схем, установления идентичности фрагментов схем и т.п.

Существенный интерес представляют также сравнительно новые задачи анализа и синтеза структур, возникающие при разработке программного обеспечения, например, задачи проектирования структуры адаптивных сайтов, распараллеливания алгоритмов, оптимизации программ при трансляции и компиляции, проектирования оптимальных структур данных и т.п. Причем количество подобных задач, возникающих в области вычислительной техники, увеличивается.

Большинство задач структурного анализа и синтеза относятся к классу комбинаторных. Существенной тенденцией является рост сложности проектируемых систем, который приводит к увеличению размера входа задач (при проектировании БИС сотни тысяч и миллионы компонентов, при разработке программного обеспечения – сотни и тысячи операторов). Анализ фундаментальных и прикладных работ в этой области показал, что программы решения комбинаторных задач структурного анализа и синтеза имеют:

- высокую временную сложность – точные алгоритмы решения многих задач данного класса являются NP -сложными, что и при сравнительно небольшой размерности входа (порядка 10^2) приводит к недопустимо большому времени выполнения соответствующих программ даже на современных ЭВМ;
- высокую емкостную сложность, как правило, полиномиально зависящую от размерности задачи.

Невозможность полного перебора при решении NP -сложных задач в сочетании со стремлением к приемлемой точности решения приводит к использованию приближенных методов, основанных на большом количестве эвристик.

Таким образом, разработка алгоритмов решения задач данного класса – многоэтапный, сложный и весьма трудоемкий процесс. Однако при большом разнообразии задач анализа и синтеза структур они имеют высокую степень общности, обусловленную использованием одних и тех же абстракций – графовых моделей. Это позволяет говорить об общих принципах разработки алгоритмов их решения, а следовательно, возможности создания средств автоматизации этого процесса.

НТБ МГТУ им. Н.Э. Баумана

МГТУ
им. Н.Э. БАУМАНА
БИБЛИОТЕКА



1795611
Иванова Г.С. Методология и

Для автоматизации достаточно трудоемких этапов перевода неформального описания алгоритма на уровне операций над моделями объектов в описание его на универсальном языке программирования, анализа вычислительной и емкостной сложности, а также выполнения оптимизирующих преобразований необходимо создание специализированного языка и средств анализа и оптимизации.

Исследованиями в области решения задач структурного анализа и синтеза занимались Р.П. Базилевич, А.М. Бершадский, А.Н. Мелихов, К.К. Морозов, М.И. Нечепуренко, Ф.А. Новиков, В.А. Овчинников, А.И. Петренко, А. Ахо, Т. Кормен, Ч. Лейзерсон, Р. Риверст, Р. Седжвик, С. Хидитниemi, Д. Хопкрофт и многие другие.

Однако в основном указанные ученые занимались постановкой задач, поиском и анализом методов и алгоритмов их решения, а не автоматизацией создания соответствующих программ.

К средствам автоматизации процесса разработки программ решения задач указанного класса можно отнести ограниченно-применимые язык обработки множеств *SETL* и систему «*Lambda-Upsilon-Omega*», в которой принята попытка автоматизации оценки вычислительной сложности, а также некоторые не универсальные библиотеки подпрограмм выполнения часто встречающихся операций над графами, предлагаемые рядом авторов: А. Ахо, Н. Виртом, Р. Седжвиком, коллективом под руководством М.И. Нечепуренко и др.

Систем, позволяющих автоматизировать процесс разработки алгоритмов решения задач структурного анализа и синтеза, на данный момент не существует. Создание такой системы позволит упростить и ускорить процесс разработки и оценки алгоритмов, обеспечивая пользователю возможность оперировать понятиями и терминами моделей предметной области.

Цель диссертационной работы. Создание методологии, а также комплекса математических, лингвистических и программных средств автоматизации разработки алгоритмов решения задач анализа и синтеза структур программного обеспечения и устройств вычислительной техники.

Задачи исследований. Для достижения указанной цели потребовалось:

1. На основе анализа объектов задач структурного анализа и синтеза, их свойств и характеристик определить совокупность математических абстракций, необходимых для представления указанных объектов.

2. Исследовать свойства и разработать модели неструктурных и структурных алгоритмов решения задач структурного анализа и синтеза, а также многоуровневых и комбинированных структур данных, представляющих графовые модели.

3. Создать аналитические средства оценки и преобразования конструкций алгоритмов, лингвистические средства описания их структуры и метод приведения алгоритмов к структурному виду.

4. Разработать синтаксис и семантику языков описания алгоритмов в операциях над множествами и в операциях над графами, а также трансляторы с них.

5. Разработать синтаксис и семантику языка описания графовых моделей, решить задачу выбора оптимального с точки зрения минимизации вычислительной сложности способа представления этих моделей, а также предложить способ синтеза оптимальных с той же точки зрения структур данных для реализации графовых моделей и макрогенерации их описаний.

6. Получить совокупность оптимизирующих преобразований алгоритмов на графах и сформулировать соответствующие синтаксические правила.

7. Создать методику и средства оценки и снижения вычислительной и емкостной сложности алгоритмов.

8. Создать методику и программную систему разработки и анализа эффективности алгоритмов решения задач структурного анализа и синтеза.

Методы исследований. Выполненная работа характеризуется комплексным подходом к решению поставленной проблемы. Результаты диссертационной работы получены на базе использования методов синтеза, анализа и оптимизации. Математическую основу составляет аппарат теории графов, теории множеств и математической логики, а также теория формальных языков и методы грамматического разбора.

Научная новизна работы. В работе предпринята попытка обобщить и углубить теоретические и прикладные исследования в области автоматизации процесса разработки алгоритмов решения задач на графах. В процессе исследования получены следующие новые научные результаты, представляемые на защиту:

1. Предложена многоуровневая схема описания, реализации и оптимизации алгоритмов решения задач анализа и синтеза структур как аппаратных, так и программных средств вычислительной техники.

2. Выполнена систематизация существующих и предложены новые структурные модели, позволяющие адекватно и более эффективно представлять объекты задач указанного класса.

3. Разработаны и формально описаны информационно-логическая модель структурного и неструктурного алгоритмов, а также модели комбинированных и иерархических структур данных.

4. Определены полные наборы инвариантов моделей структурных конструкций, сформулированы аксиоматика операции свертки уграфов структурных конструкций и теорема о необходимом и достаточном условии структурности алгоритма.

5. Формально поставлена задача приведения алгоритмов к структурному виду, разработан и реализован метод ее решения.

6. Обоснован синтаксис и определена семантика двух специализированных языков описания алгоритмов решения задач структурного анализа и

синтеза на уровне операций над графовыми моделями и операций над множествами.

7. Формально поставлена и решена задача выбора оптимального способа представления графовой модели множествами, намечены подходы к формальной постановке и решению задачи синтеза оптимальных структур данных для представления множеств и их отображений.

Практическая ценность работы. По результатам проведенных исследований создана программная система формального описания в терминах операций над множествами, реализации и анализа эффективности алгоритмов решения задач структурного анализа и синтеза, которая включает подсистему структуризации алгоритмов, транслятор со специализированного языка описания алгоритмов, анализатор вычислительной и емкостной сложности алгоритмов, библиотеки шаблонов описаний классов абстракций, макрогенератор описаний структур данных, подсистему оптимизации, а также ряд рабочих модулей, позволяющих выполнять ввод, модификацию текста алгоритма и структур данных. Система позволяет автоматизировать процесс получения программ алгоритмов решения задач структурного анализа и синтеза из описания их алгоритмов на специализированном языке, получать оценки потребления программы ресурсов вычислительной машины и выполнять оптимизирующие преобразования описания алгоритмов.

Апробация работы. Основные положения работы обсуждались на научно-технической конференции, посвященной 170 лет МГТУ им. Н.Э. Баумана (Москва, 2000), на межвузовской научно-технической конференции МГТУ им. Н.Э. Баумана (Москва, 2003), на Международной научно-технической конференции «Гражданская авиация на современном этапе развития науки, техники и общества» (Москва, 2003), на первой Международной научно-технической конференции, посвященной 90-летию со дня рождения академика В.Н. Челомея (Москва-Реутов, 2004).

Реализация и внедрения. Все основные теоретические и практические результаты работы в виде методик и программных средств внедрены в промышленность. Они использованы при выполнении опытно-конструкторских работ по темам И070406 и И070507 («Метеор-3М-ИКФС-2-МЭ») в НИИ ИСУ МГТУ им. Н.Э. Баумана, работ по договорам № 275/2004 («Разработка Информационной управляющей системы»), № 177/2006 («Разработка информационно-аналитической системы оценки технического состояния, эффективности защиты магистрального газопровода и прогнозирования коррозионного состояния») в ЗАО «РТСофт» и при создании системы управления контентом адаптивных сайтов RBC Contents 5.01 в РБК Софт (РосБизнесКонсалтинг).

Кроме этого, разработанные в диссертации методики, аналитические, лингвистические и программные средства используются при чтении лекций и проведении семинарских занятий по курсам «Алгоритмические языки и

программирование», «Системное программное обеспечение» и «Автоматизация конструкторского проектирования ЭВМ», а также выполнении курсовых и дипломных проектов.

Публикации. По результатам выполненных исследований опубликовано 15 статей, 3 учебника и 1 учебное пособие.

Объем и структура диссертации. Диссертационная работа включает введение, пять глав, заключение и список литературы, занимающих 411 страниц текста, в том числе 85 рисунков и 65 таблиц на 119 страницах, список использованной литературы из 115 наименований на 10 страницах.

СОДЕРЖАНИЕ РАБОТЫ

Во введении обоснована актуальность темы диссертации, рассмотрено современное состояние проблемы. Сформулированы цель и основные задачи исследования, приведен перечень основных научных результатов, выносимых на защиту, и изложено краткое содержание глав диссертации.

В первой главе «Определение предметной области и постановка задачи» выполнен анализ известных комбинаторных задач структурного анализа и синтеза, решаемых при проектировании технических и программных средств вычислительной техники.

В результате показано, что большое количество и разнообразие подобных задач, а также важность получения результатов в короткие сроки при условии высокой трудоемкости их решения делают актуальным создание средств автоматизации разработки алгоритмов решения задач рассматриваемого класса.

Содержательно определены основные задачи, составляющие проблему автоматизации разработки программ решения комбинаторных задач структурного анализа и синтеза.

Исследованы объекты задач структурного анализа и синтеза и их модели. В результате выявлена необходимость систематизации существующих графовых моделей, а также формулирования правил перехода от объектов к моделям, без которых автоматизация процесса разработки алгоритмов решения задач указанного класса невозможна.

Выполнен анализ процесса разработки алгоритмов решения задач рассматриваемого класса. По результатам анализа предложена четырехуровневая схема разработки алгоритмов решения задач данного класса (см. рис. 1), реализация которой существенно снижает трудоемкость создания средств автоматизации разработки алгоритмов и позволяет выполнять поуровневую оптимизацию получаемой программы.

Обоснована необходимость разработки двух специализированных языков описания алгоритмов решения задач структурного анализа и синтеза и оптимизирующих трансляторов с них, что существенно сокращает описание алгоритмов и позволяет выполнять его в терминах используемых моделей.

При этом выявлено, что язык описания алгоритмов на уровне операций над графами является надмножеством языка описания алгоритмов на уровне операций над множествами, поскольку каждая операция над графами реализуется определенной совокупностью операций над множествами. Кроме того, сформулировано основное требование к синтаксису разработанных языков – использование нотаций, близких к применяемым в теории графов, множеств и математической логики.

Постановка задачи разработки языка описания алгоритмов решения задач структурного анализа и синтеза на уровне графов позволила сформулировать задачу синтеза оптимальных с точки зрения минимизации вычислительной сложности алгоритма структур данных для представления графовых моделей, поскольку наличие такого языка предполагает рассмотрение графовых моделей в целом, а не представляющих их множеств.

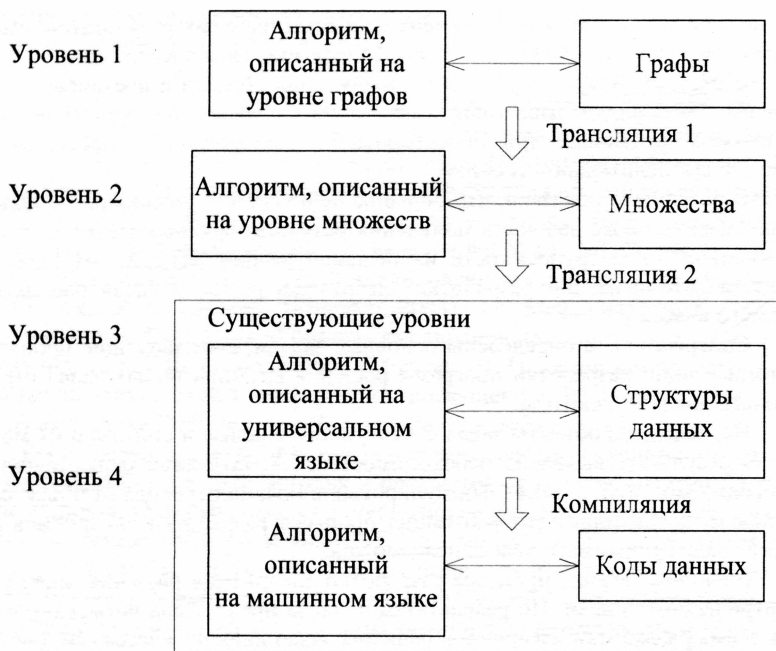


Рис. 1. Четырехуровневая схема формального описания алгоритмов и соответствующие представления данных

Показано, что реализация создаваемых алгоритмов в виде готовых программ возможна только при создании средств макрогенерации описаний структур данных и подпрограмм, выполняющих операции над ними. Использование при этом механизмов построения классов позволит оценивать

вычислительную и емкостную сложность сгенерированных конструкций, что создаст предпосылки автоматизации расчета вычислительной и емкостной сложности разрабатываемых алгоритмов.

Выявлена необходимость создания средств приведения описания алгоритмов к структурному виду, что позволяет не ограничивать разработчика алгоритмов использованием только структурных конструкций, обеспечивая возможность автоматического расчета вычислительной сложности алгоритма вне зависимости от используемых разработчиком алгоритмов конструкций.

Установлено, что для формальной постановки и решения задачи структуризации алгоритма, разработки универсальных средств оценки эффективности алгоритмов, определения оптимизирующих и структурирующих преобразований алгоритмов и определения подхода к постановке и решению задачи синтеза оптимальных структур данных необходимо создание моделей структурного и неструктурного алгоритма и данных.

Во второй главе «Анализ алгоритмов и данных и построение их моделей» выполнен анализ характеристик объектов, которые используются при решении задач структурного анализа и синтеза. Определено, что структура, на которой можно решать задачи перечисленных типов, полностью характеризуется следующими параметрами:

- функциональные свойства компонентов;
- связность компонентов с учетом направления связи и фактора неизвестности или известности порядка их соединения в пределах одной связи;
- топологические свойства компонентов, обуславливающие ограничения на построение соединений;
- метрические параметры компонентов и связей.

Показано, что обобщенной моделью, которая позволяет отображать всю перечисленную информацию, является ультраграф с кратными дугами и отношением порядка на вершинах ультрадуг.

Определен набор и выполнен анализ моделей, используемых для представления объектов, процесса решения и результатов задач структурного анализа и синтеза. Установлено соответствие между компонентами объекта и элементами модели при представлении этих моделей с одним и двумя универсумами, связи компонентов объекта сопоставлены отношениям элементов моделей. В табл. 1 показаны результаты такого сопоставления для связей n элементов с учетом и без учета направления сигналов и с учетом и без учета порядка подключения элементов.

Выполнен анализ элементов и свойств указанных моделей. Показано, что модели с одним универсумом имеют меньшую емкостную и вычислительную сложность, но являются ограниченно применимыми вследствие неполноты представляемой в них информации о связях компонентов.

Таблица 1

Представление связей n компонентов объекта в структурных моделях

Связь	Представление в модели с одним универсумом	Представление в модели с двумя универсумами
Связь n элементов без учета направления сигнала и порядка подключения (гиперребро)	Элемент n -арного симметричного отношения смежности $F \subset X^n$ – $X_j = \{x_i, x_k, \dots, x_r\} \in F,$ $ X_j = n$	Множество или мультимножество* из n элементов симметричного бинарного отношения инцидентности $\Gamma \subset X \times U = U \times X -$ $\{\{x_k, u_j\} \in \Gamma\}, x_k \in X_j, X_j = n,$ где X_j – множество вершин, соединенных ребром u_j
Связь n элементов, учитывающая порядок их подключения (гипердуга)	Элемент n -арного отношения смежности $\bar{F} \subset X^n$ – $X_j = (x_i, x_k, \dots, x_r) \in \bar{F},$ $ X_j = n$	Кортеж из n элементов симметричного бинарного отношения инцидентности $\Gamma \subset X \times U = U \times X -$ $(\{x_k, u_j\} \in \Gamma), x_k \in X_j, X_j = n,$ где X_j – множество вершин, соединенных ребром u_j
Связь n элементов без учета порядка подключения, но с учетом направления сигнала (ультрадуга)	Элемент бинарного отношения смежности на множестве подмножеств $\bar{F} \subset X^r \times X^s, r + s = n -$ $X_i, \bar{F} X_k, \text{ т. е. } (X_i, X_k) \in \bar{F},$ $ X_i + X_k = n, X_j = X_i, X_k,$ где X_i, X_k – множества вершин источников/приемников ребра u_j	Множество или мультимножество* из n элементов бинарных отношений инцидентности $\bar{\Gamma}_1 \subset X \times U, \bar{\Gamma}_2 \subset U \times X -$ $\{(x_i, u_j) \in \bar{\Gamma}_1\} \cdot \{(u_j, x_k) \in \bar{\Gamma}_2\},$ $x_i \in X_i, x_k \in X_k,$ $ X_i + X_k = n, X_j = X_i, X_k,$ где X_i, X_k – то же, что и слева
Связь n элементов с учетом порядка подключения и направления сигнала (ориентированная ультрадуга)	Элемент n -арного отношения смежности $\bar{F}_1 \subset X^n$ и элемент бинарного отношения смежности на множестве подмножеств $\bar{F}_2 \subset X^r \times X^s, r + s = n -$ $X_j = (x_i, x_k, \dots, x_r) \in \bar{F}_1, X_j = n,$ и $X_i, \bar{F}_2 X_k, \text{ т. е. } (X_i, X_k) \in \bar{F}_2,$ $ X_i + X_k = n, X_j = X_i, X_k,$ где X_i, X_k – то же, что и выше	Кортеж из n элементов бинарных отношений инцидентности $\bar{\Gamma}_1 \subset X \times U, \bar{\Gamma}_2 \subset U \times X -$ $(\{(x_i, u_j) \in \bar{\Gamma}_1\} \cdot \{(u_j, x_k) \in \bar{\Gamma}_2\}),$ $x_i \in X_i, x_k \in X_k,$ $ X_i + X_k = n,$ $X_j = X_i, X_k,$ где X_i, X_k – то же, что и выше

Кроме этого в работе исследован подкласс структурных иерархических моделей, необходимость применения которых при разработке программных и аппаратных средств вычислительной техники следует из применения блочно-иерархического подхода при проектировании указанных средств. Для них предложен способ представления:

$$G(X, XFX, XRX) \text{ или } G(X, U, XGU, UTX, XRX),$$

где $R = \{R^i / i = \overline{1, n}\}$ – множество отношений между элементами соседних уровней:

$$\{(x_j^{i-1}, x_k^i) / x_j^{i-1} \in X_k^{i-1}\} \subset R^i \subset X^{i-1} \times X^i,$$

x_k^i, x_j^{i-1} – вершины i -го и $i-1$ -го уровня соответственно;

X_k^{i-1} – подмножество вершин $i-1$ -го уровня, соответствующее x_k^i ;

X^i, X^{i-1} – множества вершин i -го и $i-1$ -го уровня соответственно;

R^i – отношение вхождения между вершинами $i-1$ -го и i -го уровней.

Получены формальные правила, определяющие отображение отношения вершин одного уровня в отношения вершин другого уровня.

Систематизированы матричные и аналитические способы задания графовых моделей. Полученные результаты позволяют сформулировать требования к языкам формального описания алгоритмов и определяют особенности выполнения операций над ними.

По результатам анализа структур алгоритмов определены их основные компоненты, а также отношения и свойства, что позволило построить базис алгоритма:

$$A(O, C) = \varphi(B),$$

где O и C – множества операторов и связей между ними;

$B = \{DB, FB, PB, T\}$ – базис алгоритма,

DB, FB, PB – символы переменных, функций и предикатов условий,

T – символы типов операторов – операторный базис алгоритма.

Предложена информационно-логическая модель алгоритма, позволяющая отобразить его компоненты, их отношения и свойства, которые должны быть учтены при решении задач структуризации и автоматического анализа вычислительной сложности разрабатываемых алгоритмов.

Модель, состоит из двух частей: управляющего графа алгоритма и двудольного графа «оператор–данные».

Так управляющим графом алгоритма (см. рис. 2, а) является взвешенный граф (см. рис. 2, б) вида:

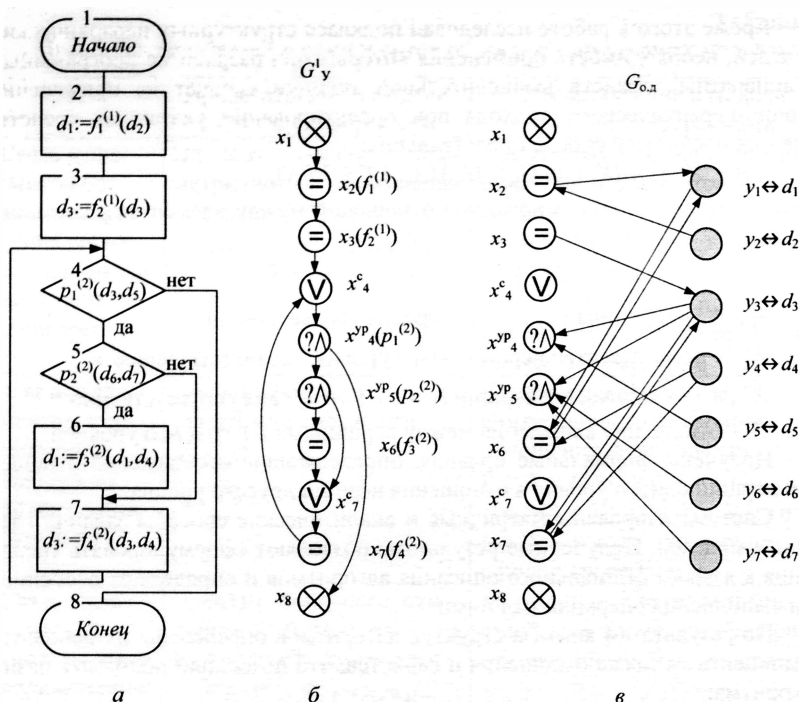
$$G_Y(<X, T, (F_B, P_B)>, \{\overline{U^1}, L, E>, \overline{U^2}\}),$$

где X, U – множества вершин и ребер графа;

$\overline{U^1}, \overline{U^2}$ – множества дуг условных и безусловных передач управления;

$L = \{true, false\}$;

E – множество вероятностей переходов.



Условные обозначения:

- $\otimes$ - терминальный блок; $\odot \wedge$ - объединенный блок вычисления
 $\ominus$ - обработка данных; условия и ветвления;
 $\vee$ - слияние потоков управления; $\circ$ - вершина данных

Рис. 2. Алгоритм (а), уграф (б) и двудольный граф «оператор-данные» (в)

Сформулированы правила перехода от объекта к модели:

- 1) $O \leftrightarrow X$;
- 2) $X R_1 T, X' R'_1 F_B, X'' R''_1 P_B, X \cup X''' = X$;
- 3) $C \leftrightarrow \bar{U}$;
- 4) $\bar{U}^1 Q_1 L, \bar{U}^2 Q_2 E$,

где R'_1, R''_1, Q_1, Q_2 – однозначные, возможно взаимнооднозначные отображения.

Двудольный граф «оператор-данные» (рис. 2, в):

$$G_{o,d}(Z, \vec{V}) = G_{o,d}(\{X, Y\}, \{\langle \bar{V}^1, B \rangle, \bar{V}^2\}),$$

где X, Y и $\bar{V}$ – подмножества вершин и множество ребер двудольного графа;
 $\bar{V}^1, \bar{V}^2$ – подмножества отношений данное-оператор и оператор-данное;
 $B = \{b_1, b_2\}$, b_1 – признак того, что входное данное – аргумент;
 b_2 – признак того, что входное данное аргументом не является.

Сформулированы правила перехода от объекта к модели:

1) $O \leftrightarrow X$; $D \leftrightarrow Y$, причем $X \cap Y = \emptyset$, $X \cup Y = Z$;

2) $v_k = \begin{cases} v_k(y_j, x_i), & \text{если } y_j \leftrightarrow d_j - \text{входное данное } o_i \leftrightarrow x_i, \\ v_k(x_i, y_j), & \text{если } y_j \leftrightarrow d_j - \text{выходное данное } o_i \leftrightarrow x_i. \end{cases}$

3) $\bar{V}^1 R_2 B$,

где D – множество символов переменных базиса DB ;

R_2 – однозначное отображение.

Интегральная модель алгоритма определена как

$$G(Z, U, V) = G_y \cup G_{o.d.}.$$

Выполненное формальное описание правил перехода от алгоритма к его модели позволяет осуществлять эту операцию автоматически.

Построены модели базовых структурных конструкций, определены их свойства и инварианты:

1. Модель оператора обработки данных представляет собой одновершинный кусок $G^0(X^0, U^0)$, такой что

$$X^0 = \{x_k\}; \quad t(x_k) = \text{«обработка данных»};$$

$$U^0 = U_{i,j}^0 = \{\bar{u}(\emptyset, x_k), \bar{u}(x_k, \emptyset)\}.$$

Формальные признаки данной конструкции:

$$|X^0| = 1; |U^0| = 2; \quad P(X^0) = S(X^0) = 1,$$

где $t(x_k)$ – тип k -ой вершины; $U_{i,j}^0$ – подмножество внешних ребер куска;

$P(X^0), S(X^0)$ – полустепени исхода и захода куска.

2. Модель конструкции следования (см. рис. 3, а) – простая цепь

$G^1(X^1, U^1)$, такая, что:

$$G^1(X^1, U^1) - U_{i,j}^1 = \{\bar{u}(\emptyset, x_k), \bar{u}(x_{k+n}, \emptyset)\}.$$

Формальные признаки данной конструкции:

$$P(X^1) = S(X^1) = 1,$$

$$(\forall x_i \in X^1) (p(x_i) = s(x_i) = 1 \ \& \ t(x_i) = \text{«обработка данных»}),$$

где $P(X^1), S(X^1)$ – полустепени исхода и захода куска $G^1(X^1, U^1)$,

$p(x_i), s(x_i)$ – полустепени исхода и захода вершины x_i ,

$t(x_k)$ – тип k -ой вершины.

3. Моделью конструкции ветвления (см. рис. 3, б) является кусок уграфа

$G^2(X^2, U^2)$, такой что $U_{i,j}^2 = \{\bar{u}(\emptyset, x_k), \bar{u}(x_{k+4}, \emptyset)\}$,

где x_k – вершина начала конструкции ветвления;

x_{k+4} – вершина конца тела цикла конструкции ветвления.

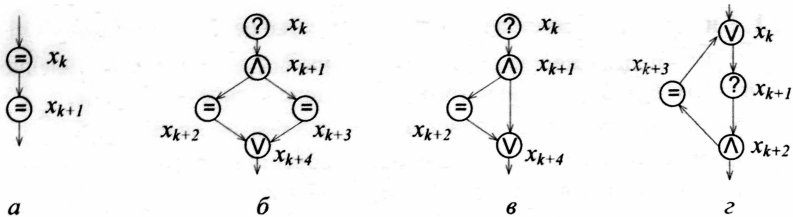


Рис. 3. Модели структурных конструкций:
а – следование; **б** – ветвление; **в** – цикл-пока

Формальные признаки данной конструкции:

$$\begin{aligned}
 P(X^2) = S(X^2) = 1, \quad t(x_k) = \text{«вычисление условия»}, \quad t(x_{k+4}) = \text{«слияние»}, \\
 (F^{+1}(x_k) = x_{k+1}) \& (t(x_{k+1}) = \text{«разветвление»}), \\
 (F^{+1}(x_{k+1}) = \{x_{k+2}, x_{k+3}\}) \& ((x_{k+2} = x_f \& x_{k+3} = x_h) \vee \\
 \vee (x_{k+2} = x_{k+4} \& x_{k+3} = x_h) \vee (x_{k+2} = x_f \& x_{k+3} = x_{k+4})) \& \\
 \& (t(x_h) = t(x_f) = \text{«обработка данных»}),
 \end{aligned}$$

где $P(X^2), S(X^2)$ – полустепени исхода и захода куска $G^2(X^2, U^2)$;

$t(x_k)$ – тип k -ой вершины.

4. Моделью конструкции цикла (см. рис. 3, в) является кусок управляющего графа $G^3(X^3, U^3)$, такой что $U_{i,j}^3 = \{\bar{u}(\emptyset, x_k), \bar{u}(x_{k+2}, \emptyset)\}$,

где x_k – вершина начала цикла, x_{k+2} – вершина конца цикла.

Формальные признаки данной конструкции:

$$\begin{aligned}
 P(X^3) = S(X^3) = 1, \quad (t(x_k) = \text{«слияние»}), \quad (t(x_{k+2}) = \text{«разветвление»}), \\
 (F^{+1}(x_{k+2}) \cap F^{-1}(x_k) = x_{k+3}) \& (t(x_{k+3}) = \text{«обработка данных»}), \\
 (F^{+1}(x_k) = x_{k+1}) \& (t(x_{k+1}) = \text{«вычисление условия»}),
 \end{aligned}$$

где $P(X^3), S(X^3)$ – полустепени исхода и захода куска $G^3(X^3, U^3)$;

$t(x_k)$ – тип k -ой вершины.

Для распознавания сложных структурных конструкций в работе предложено использовать свойство гомеоморфности уграфов этих конструкций уграфам базовых конструкций с точностью до уграфов вложенных конструкций.

Для доказательства гомеоморфизма конструкций была определена операция факторизации (свертки) вершин модели алгоритма, входящих в одну структурную конструкцию:

$$\text{factor}(G_j(X_j, U_j)) = \begin{cases} G_j(X_j, U_j), & \text{если } |X_j| = 1, \\ G_j^F(\{x_n\}, \{u(\emptyset, x_n), u(x_n, \emptyset)\}), & \\ t(x_n) = \text{«Изменение данных»}, & \text{если } |X_j| > 1. \end{cases}$$

Определена аксиоматика этой операции – правила свертки структурных конструкций:

$$\text{factor}(G^0) = G^{0F} = G^0,$$

$$\text{factor}(G_i \in \{G^1, G^{1F}, G^2, G^{2F}, G^3, G^{3F}\}) = G^{0F},$$

$$\text{factor}(\{G_k\} \subset G^{1C} \& G_k \in \{G^{2F}, G^{3F}\}) = G^{1F},$$

$$\text{factor}(G_{01}, G_{02} \subset G^{2C} \& G_{01}, G_{02} \in \{G^1, G^{1F}, G^2, G^{2F}, G^3, G^{3F}\}) = G^{2F},$$

$$\text{factor}(G_0 \subset G^{3C} \& G_0 \in \{G^1, G^{1F}, G^2, G^{2F}, G^3, G^{3F}\}) = G^{3F},$$

Сформулирована и доказана теорема о необходимых и достаточных условиях структурности алгоритма.

Теорема 1. Алгоритм является структурным тогда и только тогда, когда в результате последовательной факторизации кусков его уграфа, последний можно свести к уграфу

$$G(\{x_H, x^F, x_K\}, \{u(x_H, x^F), u(x^F, x_K)\}),$$

причем $(x_H), (x_K) = \langle \otimes \rangle$ – вершины начала и завершения алгоритма.

Получена модель структурного алгоритма:

$$G_y^C(X, U) = G_k \in \{G^{1C}, G^{2C}, G^{3C}, G^0\},$$

$$\text{где } G^{1C} = \begin{cases} G_k^C : \text{factor}(G_{k-1,1}, G_{k-1,2} \subset G_k^C) = G^{1F}, \\ G^1, \end{cases}$$

$$G^{2C} = \begin{cases} G_k^C : \text{factor}(G_{k-1,1}, G_{k-1,2} \subset G_k^C) = G^{2F}, \\ G^2, \end{cases}$$

$$G^{3C} = \begin{cases} G_k^C : \text{factor}(G_{k-1,1} \subset G_k^C) = G^{3F}, \\ G^3, \end{cases}$$

$$G_{k-1,1}, G_{k-1,2} \in \{G^{1C}, G^{2C}, G^{3C}, G^0\}.$$

Выполнен анализ базовых способов организации данных: векторной и списковой, определен набор компонентов структур данных и выявлены отношения между ними, что позволило построить модели векторов и списков.

Обобщенная модель векторной структуры (см. рис. 4) – смешанный граф вида:

$$G_V(\{Z^{AB}, <Z^{AE}, l^V, Q_o, >, Z^V, Z^{AV}\}, \{\bar{E}^{AB}, \bar{E}^{AE}, <\bar{E}^{AV}, l^A, >, E^V, \bar{E}^{IV}\}),$$

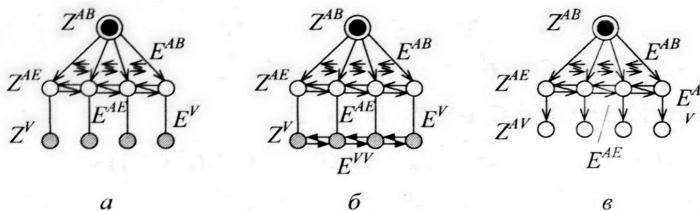
где $|Z^{AB}| = 1$;

для вектора несортированных данных $-Z^{AV} = \emptyset, \bar{E}^{AV} = \emptyset, \bar{E}^{IV} = \emptyset$, причём для характеристического вектора $-l^V = 1/8$ (байта);

для вектора сортированных данных $-Z^{AV} = \emptyset, \bar{E}^{AV} = \emptyset$;

для вектора адресов $-Z^V = \emptyset, E^V = \emptyset, \bar{E}^{IV} = \emptyset, l^V = 0, Q_o = \emptyset$;

для вектора адресов прямого доступа $-Z^V = \emptyset, E^V = \emptyset, l^V = 0, Q_o = \emptyset$.



Условные обозначения:

- | | |
|-----------------------------------|---|
| ● - вершина
начального адреса; | → - дуга, связывающая адреса; |
| ○ - вершина адреса
элемента; | → - дуга, связывающая данные; |
| ● - вершина данных; | — - ребро, связывающее адрес со
значением элемента |

Рис. 4. Модели вектора:

a – с несортированными элементами; *б* – с сортированными элементами;
в – содержащего адреса

Обобщенная модель списковой структуры (см. рис. 5):

$$G_s(\{Z^{AB}, \langle Z^{AE}, l^V, Q_0 \rangle, Z^V, Z^{AV}\}, \{\bar{E}^{AB}, l^A, \langle \bar{E}^{AE}, l^A \rangle, \bar{E}^{AV}, l^A, E^V, E^{VV}\}).$$

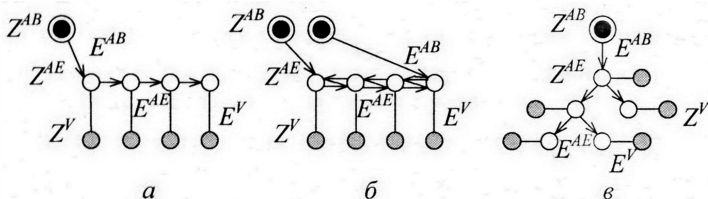


Рис. 5. Модели списковых структур данных:

a – линейный односвязный список; *б* – линейный двусвязный список;
в – древовидный список

$$G_v(\{Z^{AB}, \langle Z^{AE}, l^V, Q_0 \rangle, Z^V, Z^{AV}\}, \{\bar{E}^{AB}, \bar{E}^{AE}, \langle \bar{E}^{AV}, l^A \rangle, E^V, \bar{E}^{VV}\}),$$

где $Z^{AB}, Z^{AE}, Z^V, Z^{AV}$ – подмножества начальных вершин, вершин адресов элементов, элементов данных и адресов данных;

$\bar{E}^{AB}, \bar{E}^{AE}, \bar{E}^{AV}, E^V, \bar{E}^{VV}$ – подмножества ребер (дуг), связывающих вершины указанных выше подмножеств;

Q_0 – вычислительные сложности выполнения основных операций;

l^V, l^A – объем памяти, необходимый для хранения данных и адреса.

Определены правила перехода от объектов к их моделям, а также расчетные соотношения, позволяющие оценивать вычислительные сложности базовых операций и емкостную сложность исследуемой структуры, что создает предпосылки для автоматической генерации структур данных, представляющих графовые модели.

Модели иерархических и комбинированных структур данных предложено получать как объединение моделей базовых структур, использованных при построении сложной структуры. Получены правила перехода от объекта к модели для таких структур данных.

Определены расчетные соотношения для оценки вычислительной сложности выполнения основных операций и емкостной сложности таких структур по построенным моделям, что является основой для генерации оптимальных структур данных, включая комбинированные, а также автоматизации оценки их вычислительной и емкостной сложности.

В третьей главе «Исследование свойств алгоритмов и разработка метода их приведения к структурному виду» получена формальная постановка задачи структуризации, основанная на моделях структурных и неструктурных алгоритмов, что создает предпосылки для ее корректного решения:

выполнить преобразование

$$G^{yh}(<X_1, T>, <\vec{U}_1, P>) \xrightarrow{F_1} G^{yc}(<X_2, T>, <\vec{U}_2, P>) \text{ и} \\ G^{dh}(\{\langle X_1, T \rangle, \langle Y_1, C_1 \rangle\}, \{\langle \vec{V}_1^1, T_1^1 \rangle, \langle \vec{V}_1^2, T_1^2, D_1^3 \rangle\}) \xrightarrow{F_2} \\ \xrightarrow{F_3} G^{dc}(\{\langle X_2, T \rangle, \langle Y_2, C_2 \rangle\}, \{\langle \vec{V}_2^1, T_2^1 \rangle, \langle \vec{V}_2^2, T_2^2, D_2^3 \rangle\})$$

такое что $(\forall i \in I) \quad A^H(S_i) = A^C(S_i),$

где G^{yh} и G^{yc} – куски управляющих графов неструктурной и структурной программ без вершин, сопоставленных началу и концу программы;

$S = \{S_i / i \in I\}$ – множество всех допустимых наборов входных данных задачи, для решения которых предназначен данный алгоритм;

A^H – неструктурная программа, управляющим графом которой является граф G^{yh} ;

A^C – структурная программа, соответствующая преобразованным графам G^{yc} и G^{dc} ;

при выполнении следующих условий, вытекающих из свойств алгоритмов:

$$|X_1| \leq |X_2|, \quad X_1 \cap X_2 \neq \emptyset, \quad |\vec{U}_1| \leq |\vec{U}_2|, \quad \vec{U}_1 \cap \vec{U}_2 \neq \emptyset;$$

и существует разрезание B куска графа G^{yc} на совокупность кусков $B(G^{yc})$, удовлетворяющих условиям:

$$(\forall G_i^{yc} \in B(G^{yc})) \quad G_i^{yc} \neq \emptyset, i \in I,$$

$$(\forall G_i^{yc} \in B(G^{yc})) (G_i^{yc} \in G^B \vee G_i^{yc} \in G^{BF}),$$

$$\bigcup_{i \in I} G_i^{yc} = G^{yc},$$

$$(\forall G_i^{yc}, G_{i+1}^{yc} \in B(G^{yc})) (X_i^{yc} \cap X_{i+1}^{yc} = \emptyset \ \& \ U_i^{yc} \cap U_{i+1}^{yc} = U_{i,j+1} \ \& \ |U_{i,j+1}| = 1),$$

$$i, i+1 \in I,$$

$$(\forall G_i^{yc}, G_j^{yc} \in B(G^{yc})) (X_i^{yc} \cap X_j^{yc} = \emptyset \ \& \ U_i^{yc} \cap U_{i+1}^{yc} = \emptyset), \quad i, j \in I, \quad j \neq i \pm 1,$$

или другими словами

$$factor^k(G^{yc}) = G^{0F}.$$

Выполненный в работе анализ возможных вариантов нарушений структурности показал, что при неограниченном использовании оператора безусловной передачи управления возможно получение сложных нарушений структурности, не приводимых к структурному виду за один шаг. В связи с этим предложено осуществлять последовательную многошаговую структуризацию, ориентированную на замену неструктурных фрагментов на эквивалентные структурные и факторизацию структурных конструкций по мере их распознавания.

Выявлено, что решение данной задачи в исходном виде является *NP*-сложным, поскольку необходимо определять в уграфе куски, изоморфные структурным. Поэтому предложено от уграфа перейти к линейной цепочке символов, соответствующих его вершинам с точностью до определяющих конструкцию семантических свойств этих вершин. Такое преобразование позволило для распознавания структурных конструкций и нарушений структурности применить методы синтаксического анализа, используемые в теории формальных языков. При этом решение имеет полиномиальную сложность.

Определено, что для перехода к линейной цепочке символов принципиальное значение имеет порядок обхода уграфа алгоритма. Откуда следует необходимость выполнения нумерации вершин уграфа, основным требованием к которой является выявление структурных свойств уграфа.

Показано, что существующие алгоритмы нумерации вершин уграфа не удовлетворяют требованиям, предъявляемым в настоящей работе. Получены формальные правила, позволяющие выполнять разметку и специально разработанную *S*-нумерацию вершин уграфа алгоритма, проявляющие его структуру.

Правило 1. Вершины, кроме вершин слияния и вершин, следующих за вершиной ветвления, нумеруются последовательно:

$$S(x_j) = S(x_i) + 1: x_j \in F^{+1}(x_i) \ \& \ t_{xi} \in \{ \text{'начало'}, \text{'изменение данных'}, \text{'слияние'} \} \ \& \\ \& \ t_{xj} \in \{ \text{'изменение данных'}, \text{'ветвление'}, \text{'конец'} \},$$

где $F^{+1}(x_i)$ – множество вершин, в которые заходят дуги из x_i .

Следствие. Любой линейный участок простого пути (луч) уграфа – Ch_{x_q, x_r} , нумерованный по правилу 1, будет S -лучом:

$$\forall x_i, x_j \in Ch_{x_q, x_r} \& x_j \neq x_q \& x_i \neq x_r: x_j = F^{+1}(x_i) \Rightarrow S(x_j) = S(x_i) + 1.$$

Правило 2. Вершина слияния нумеруется последовательно, если она является первой вершиной цикла или завершает ветвление, обе ветви которого пронумерованы:

$$S(x_j) = S(x_i) + 1: x_j \in F^{+1}(x_i) \& t_{ij} = \text{'слияние'} \& \\ \& [\exists Cycle(x_j) (x_i \notin Cycle(x_j)) \vee (\forall x_k \in F^{-1}(x_j)) (\exists S(x_k))],$$

где $Cycle(x_j)$ – цикл, проходящий через вершину x_j .

Следствие 1. Для аранжируемых уграфов все простые пути от начальной вершины $x_0 - Ch(x_0, x_k)$ будут S -путями:

$$\forall x_j, x_i \in Ch(x_0, x_k) \& x_j \neq x_i \& x_i \neq x_k: x_j \in F^{+1}(x_i) \Rightarrow S(x_j) > S(x_i).$$

Следствие 2. Для неаранжируемых уграфов, если некоторая вершина принадлежит всем простым путям из начальной вершины в заданную, то путь из этой вершины в заданную вершину является S -путем:

$$x_j \in F^{+1}(x_i) \& x_i \neq x_0 \& x_i \neq x_j \& \forall Ch(x_0, x_j) \in CH(x_0, x_j) (x_i \in Ch(x_0, x_j)) \Rightarrow \\ \Rightarrow S(x_j) > S(x_i),$$

где $CH(x_0, x_j)$ – множество всех простых путей из начальной вершины в вершину x_j ;

$F^{+1}(x_i)$ – t -е прямое отображение вершины x_i во множество вершин уграфа.

Следствие 3. Все дуги, замыкающие цикл, будут S -обратными:

$$x_i \in F^{+1}(x_j) \& \exists Ch(x_j, x_i) (x_k \in Cycle(x_j) \& S(x_k) = \emptyset) \Rightarrow S(x_j) < S(x_i).$$

Правило 3. Последовательно с вершиной ветвления x_i нумеруется одна из вершин $x_j, x_k \in F^{+1}(x_i)$, если она принадлежит подмножеству простых циклов, включающих вершины x_i, x_j и начинающихся в вершине x_r , имеющей максимальный номер по сравнению с другими вершинами – входами циклов, а вторая вершина этому подмножеству не принадлежит, или, если такого подмножества не существует и обе вершины – потомки x_i еще не нумерованы:

$$S(x_j) = S(x_i) + 1: F^{+1}(x_i) = \{x_j, x_k\} \& t_{xi} = \text{'ветвление'} \& \\ \& [(\exists \{Circle_{x_r}(x_i)\} \subseteq \{Circle(x_i)\}) (x_r, x_j \in \{Circle_{x_r}\}, x_k \notin \{Circle_{x_r}\}) \vee \\ \vee (\exists \{Circle_{x_r}(x_i)\} \subseteq \{Circle(x_i)\}) (x_r, x_j \in \{Circle_{x_r}\}, \\ x_k \notin \{Circle_{x_r}\}) \& \exists (S(x_j), S(x_k))],$$

где $\{Circle(x_i)\}$ – множество простых циклов, включающих вершину x_i ;

$\{Circle_{x_r}(x_i)\}, \{Circle_{x_q}(x_i)\}$ – множества простых циклов, включающих вершину x_i и вершины x_r или x_q соответственно;

$x_r \in X_q \& S(x_r) = \max_{x_q \in X_q} \{S(x_q)\}$ – входная вершина внутреннего цикла;

$X_q = \{x_q \in X \text{ \& } l(x_q) = \text{'слияние'} \text{ \& } S(x_q) = S(x_m) + 1 \text{ \& } x_m \notin \{Circle_{xr}\}\}$ – множество входных вершин циклов.

Разработан алгоритм нумерации и определена его вычислительная сложность.

Определены синтаксис и семантика языка, который позволяет описывать как структурные, так и неструктурные конструкции в виде последовательности взвешенных терминальных символов (см. табл. 2).

Таблица 2

Терминальные символы языка описания конструкций алгоритма

№	Название символа	Символ	Семантика символа	Вес символа
1	Изменение данных	=	Последовательность вершин – операторов обработки данных	–
2	Начало ветвления	∧	Вершины ветвления, если при этом не фиксируется выход из цикла	Номер вершины – начала второй ветви
3	Конец ветвления	∨	Вершина слияния прямых потоков	–
4	Начало прямой дуги	°	Разрыв нумерации при завершении первой ветви ветвления	Номер вершины слияния, завершающей ветвление
5	Начало цикла	.∨	Вершина слияния прямого и обратного потоков	–
6	Выход из цикла	.∧	Вершины ветвления, обеспечивающие выход из цикла	Номер вершины, следующей за циклом
7	Конец цикла	•	Разрыв нумерации между последней вершиной цикла и вершиной, следующей за циклом	Номер вершины слияния, начинающей цикл

Предложения языка имеют формат:

$Sir = \langle D, W, S \rangle = (\langle d_1, w_{d1}, s_{d1} \rangle, \langle d_2, w_{d2}, s_{d2} \rangle \dots \langle d_M, w_{dM}, s_{dM} \rangle)$,

где $d_i \in \Sigma = \{\blacktriangleright, =, \wedge, \vee, \circ, .\vee, .\wedge, \bullet, \blacktriangleleft\}$ – терминальные символы, в множество которых добавлены символы начала и завершения алгоритма;

$w_{di} \in \mathbb{I}, |X|$ – их веса;

s_{di} – S-номера соответствующих вершин уграфа.

Разработан алгоритм формирования описания уграфа на предлагаемом языке, что позволяет осуществлять эту операцию автоматически.

Сформулированы синтаксические правила языка описания структурных алгоритмов:

<Структурный алгоритм> ::= ► <Следование> ◄.

<Следование> ::= <Следование> <Конструкция> | <Конструкция>.

<Конструкция> ::= = |

$\wedge$ <Следование> $\circ$ <Следование> $\vee$ |

$\wedge$ <Следование> $\vee$ |

$\cdot \vee \wedge$ <Следование> $\bullet$ |

$\vee$ <Следование> $\wedge \bullet$.

Определено, что грамматика языка описания структурных конструкций относится к классу контекстно-свободных (подклассу грамматик предшествования). Для этого языка построен распознающий автомат с магазинной памятью (см. табл. 3) и разработан алгоритм программы анализа, что позволило выявлять нарушения структурности алгоритма.

Все случаи, когда автомат выдает состояние ошибки, проанализированы и разделены на три группы: когда переход в состояние ошибки действительно вызван ошибкой описания алгоритма, недостижим ни при каких условиях и является результатом нарушения структурности.

Таблица 3

Переходы автомата, распознающего структурные алгоритмы

1. Основной распознаватель

Состояние	Входные символы							
	=	$\wedge$	$\cdot \wedge$	$\vee$	$\cdot \vee$	$\circ$	$\bullet$	◄
0	↓Y=1	↓Y=1	E ₁	E ₂	↓Y=1	E ₃	E ₄	E ₀
1	↓Y=1	↓Y=1	E ₁	E ₂	↓Y=1	E ₃	E ₄	K

2. Рекурсивный распознаватель «Конструкция»

Состояние	Входные символы							
	=	$\wedge$	$\cdot \wedge$	$\vee$	$\cdot \vee$	$\circ$	$\bullet$	◄
0	↓Y=1	↓Y=1	E ₁	E ₂	↓Y=1	E ₃	E ₄	E ₀
1	↓Y=1	↓Y=1	E ₁	E ₂	↓Y=1	E ₃	E ₄	K
X	↑Y, Reduce ²	2, Move	E ₁	E ₂	6, Move	E ₃	E ₄	E ₄₈
2	↓Y=3	↓Y=3	E ₅	E ₆	↓Y=3	E ₇	E ₈	E ₉
3	↓Y=3	↓Y=3	E ₁₀	↑Y, Reduce ² E ₄₆	↓Y=3	4, Move	E ₁₁	E ₁₂
4	↓Y=5	↓Y=5	E ₁₃	E ₁₄	↓Y=5	E ₁₅	E ₁₆	E ₁₇
5	↓Y=5	↓Y=5	E ₁₈	↑Y, Reduce ² E ₄₇	↓Y=5	E ₁₉	E ₂₀	E ₂₁

Таблица 3 (окончание)

Состояние	Входные символы							
	=	^	^	∨	∨	°	•	◀
6	↓Y=9	↓Y=9	7, Move	E ₂₂	↓Y=9	E ₂₃	E ₂₄	E ₂₅
7	↓Y=8	↓Y=8	E ₂₆	E ₂₇	↓Y=8	E ₂₈	E ₂₉	E ₃₀
8	↓Y=8	↓Y=8	E ₃₁	E ₃₂	↓Y=8	E ₃₃	↑Y, Reduce ² E ₄₈	E ₃₄
9	↓Y=9	↓Y=9	10, Move	E ₃₅	↓Y=9	E ₃₆	E ₃₇	E ₃₈
10	E ₃₉	E ₄₀	E ₄₁	E ₄₂	E ₄₃	E ₄₄	↑Y, Reduce ² E ₄₉	E ₄₅

В таблице использованы следующие условные обозначения:

K – конец разбора, E – состояние ошибки,

↓Y = <состояние> – рекурсивный вызов распознавателя, справа указан номер состояния после возврата из данного вызова;

↑Y – завершение рекурсивного вызова, следующее состояние определяется значением Y,

Reduce² – двукратная свертка в нетерминал «Следование»;

Move – операция переноса.

Выполнено исследование нарушений структурности алгоритма, в результате которого они разбиты на 9 групп, характеризующихся схожими свойствами:

- условие выхода из цикла в одной из ветвей ветвления, непосредственно следующее за началом ветвления;
- условие выхода из цикла в одной из ветвей ветвления, отделенное от начала ветвления вершиной изменения данных;
- пересечение ветвлений: вершина слияния непосредственно следует за вершиной ветвления;
- пересечение ветвлений: вершина ветвления отделена от вершины слияния;
- пересечение ветвления и цикла, являющееся следствием неверной последовательности вершин слияния;
- пересечение ветвления и цикла: наличие дополнительных входов в цикл;
- дополнительные выходы из цикла: два выхода из цикла подряд;
- дополнительные выходы из цикла: два выхода из цикла не подряд;
- условие выхода из цикла в середине тела цикла.

Для структуризации выявленных нарушений структурности разработаны 8 структурирующих преобразований, которые заключаются в изменении порядка следования нарушающих структурность вершин. Для каждого преобразования получены условия эквивалентности. Это позволяет утверждать, что исходный и полученный в результате корректного применения предложенных преобразований алгоритмы эквивалентны.

Так для преобразования «Вынесение из конструкции ветвления начала другого ветвления или условия выхода из цикла» получено, что исходный и преобразованный алгоритмы (см. рис. 6) эквивалентны, если:

$$\begin{cases} p'_n \leftrightarrow b'_{n2} = p_n \leftrightarrow b_{n2}, & \text{если } z \in Z(m_{01}) \\ p'_n \leftrightarrow b'_{n2} = \text{false}, & \text{если } z \in Z(m_{02}) \end{cases},$$

где $Z(m_{01})$ и $Z(m_{02})$ – множества комбинаций значений данных, при которых происходит передача управления на метки m_{01} или m_{02} , причем события передачи управления на эти метки не совместны;

p_n, p'_n – предикаты условий вершин ветвления исходного и результирующего уграфов;

$b_{n1}, b_{n2}, \dots \in \{true, false\}$ – логические переменные, совпадение значений которых с результатами вычисления предикатов условий приводит к осуществлению соответствующих переходов.

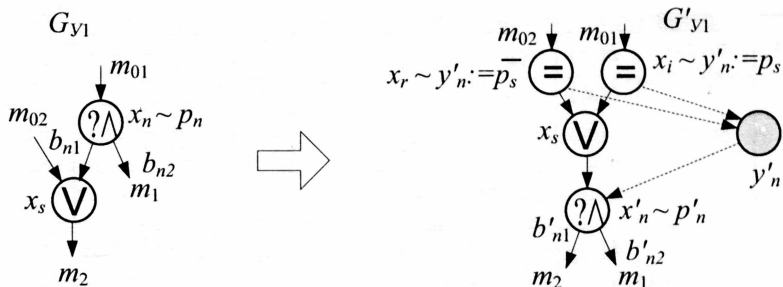


Рис. 6. Преобразование «Вынесение начала ветвления или условия выхода из цикла из конструкции ветвления»

Откуда определяется предикат ветвления преобразованного фрагмента уграфа:

$$p'_n = \begin{cases} p_n, & \text{если } (b'_{n2} = b_{n2}) \& [z \in Z(m_{01})], \\ \overline{p_n}, & \text{если } (b'_{n2} = \overline{b_{n2}}) \& [z \in Z(m_{01})], \\ true, & \text{если } (b'_{n2} = false) \& [z \in Z(m_{02})], \\ false, & \text{если } (b'_{n2} = true) \& [z \in Z(m_{02})]. \end{cases}$$

На базе эквивалентных преобразований алгоритма разработаны последовательности структуризации каждой группы нарушений структурности, что позволило построить алгоритм структуризации уграфов алгоритмов, имеющий полиномиальную сложность.

В четвертой главе «Математические основы и лингвистические средства разработки и анализа алгоритмов» выполнен анализ отношений компонентов графовых моделей и определена совокупность характеристик и элементарных операций над компонентами графовых моделей. В табл. 4 показаны характеристики, необходимые для полного описания структуры гиперграфов и ультраграфов.

Таблица 4
Характеристики элементов графовых моделей

№	Модель	Характеристики элементов при различных отношениях между ними						
		ΓX	ΓU	FU	$Q_n U$	$F_n X$	$R_n X_U$	$QX_U,$ $Q_n X_U$
1	Гипер- граф	$\rho(x_i)$	$v(u_j)$	$e(u_j)$	$q(u_j)$	$p_n(x_i)$	—	$q(x_i)$
2	Ориенти- рованный гиперграф	$\rho(x_i)$	$v(u_j)$	$e(u_j)$	$q(u_j)$	$p_n(x_i)$	$Number(x_i)$	$q(x_i)$
3	Ультра- граф	$\bar{b}(x_i)$ $\bar{b}(x_i)$	$\bar{v}(u_j)$ $\bar{v}(u_j)$	$\bar{e}(u_j)$ $\bar{e}(u_j)$ $\bar{e}(u_j)$ $\bar{e}(u_j)$	$q(u_j)$	$\bar{p}_n(x_i)$ $\bar{p}_n(x_i)$	—	$q(x_i)$
4	Ультра- граф с упорядо- ченными вершина- ми ребра	$\bar{b}(x_i)$ $\bar{b}(x_i)$	$\bar{v}(u_j)$ $\bar{v}(u_j)$	$\bar{e}(u_j)$ $\bar{e}(u_j)$ $\bar{e}(u_j)$ $\bar{e}(u_j)$	$q(u_j)$	$\bar{p}_n(x_i)$ $\bar{p}_n(x_i)$	$Number(x_i)$	$q(x_i)$

Построена классификация элементарных операций над графовыми моделями, что позволило определить множество операций (с учетом часто встречающихся в алгоритмах решения задач структурного анализа и синтеза), которые целесообразно реализовать в языке описания алгоритмов на уровне графов:

$$F_B^G = \{Op_{\text{мод}}^G, Op_{\text{пр}}^G, Op_{\text{хар}}^G\},$$

где $Op^G_{\text{мод}} = \{+, -, \cap, \cup, \setminus, \text{Fact}\}$ – символы операций преобразования (добавление и удаление вершин и ребер, пересечение, объединение и дополнение графов, подграфов и их кусков, а также свертка кусков);

$Op^G_{\text{пр.}} = \{\Gamma, \bar{\Gamma}, \bar{\Gamma}, F, \bar{F}, \bar{F}, F_n, \bar{R}_n, \bar{R}_n, Q_n, \bar{Q}_n, \bar{Q}_n\}$ – символы определения бинарных, n -арных, симметричных, несимметричных отношений инцидентности Γ , смежности F , порядка R и кратности Q ;

$Op^G_{\text{хар.}} = \{\rho, \bar{b}, \bar{b}, v, \bar{v}, \bar{v}, p, \bar{p}, \bar{p}, p_n, \bar{p}_n, \bar{p}_n, e, \bar{e}, \bar{e}, q, \bar{q}, \bar{q}, \text{Number}, \text{GammaBool}, \text{ComponentNumber}, \text{NumberOfComponents}\}$ – символы определения характеристик элементов по различным отношениям и отображениям.

Определена совокупность элементарных операций над данными множествами типов, выполнен их анализ, что позволило обоснованно сформировать множество операций над множествами, мультимножествами и кортежами, которые необходимо реализовать в языке описания алгоритмов на уровне множеств, представляющих графовую модель:

$$Op^M = \{Op^M_{\text{мод.}}, Op^M_{\text{пр.}}, Op^M_{\text{хар.}}\},$$

где $Op^M_{\text{мод.}} = \{., \text{Set}, \text{Set_weight}, \text{Insert}, \text{Delete}\}$ – символы операций модификации, возможно, взвешенных множеств, мультимножеств и кортежей;

$Op^M_{\text{пр.}} = \{ \in (\notin), \text{Duplicate}, \text{Next}, \text{Previous} \}$ – символы операций просмотра данных тех же типов;

$Op^M_{\text{хар.}} = \{\text{Card}, \text{Length}, n_Duplicate, \text{Number}, \text{Get}, \text{Get_weight}\}$ – символы операций просмотра характеристик.

Определено множество конструкций и предложен синтаксис языка описания алгоритмов на уровне операций над множествами.

Синтаксические правила основных конструкций языка:

<Список операторов> ::= <Помеченный оператор> |

<Список операторов>; <Помеченный оператор>

<Помеченный оператор> ::= <Метка>: <Оператор> | <Оператор>

<Оператор> ::= <Составной оператор> | <Присваивание> |

<Условный оператор> | <Цикл счетный> | <Выборка> |

<Безусловный переход> | <Вызов процедуры>

<Метка> ::= # <Целое без знака>

<Безусловный переход> ::= goto <Метка>

<Условный оператор> ::= <Логическое выражение> $\Rightarrow$ <Оператор>

<Выборка> ::= { <Элемент-переменная> $\in$ <Выражение множественное> / <Выражение логическое> } |

{ <Элемент-переменная> / <Выражение логическое> }

<Цикл счетный> ::= <Заголовок цикла>: <Оператор>

<Заголовок цикла> ::= <Заголовок счетного цикла> |

(<Заголовок счетного цикла>) & (<Выражение логическое>)

<Заголовок счетного цикла> ::=

$\forall$ <Идентификатор> = <Выражение арифметическое> ,

<Выражение арифметическое>

$\forall$ <Элемент-переменная> $\in$ <Выражение множественное>

<Элемент-переменная> ::= <Идентификатор> |

<Идентификатор> [<Идентификатор>] |

<Идентификатор> [<Идентификатор> , <Идентификатор>]

Определено множество дополнительных конструкций, которые необходимо включить в язык описания алгоритмов, чтобы обеспечить описание алгоритмов на уровне операций над графовыми моделями.

Сформулированы правила описания графовых моделей и их фрагментов: подграфов и кусков, а также наборы множеств для представления всех типов графовых моделей (см. табл. 5).

Таблица 5

Описание графовых моделей и набор подмножеств их отображений

№	Тип модели	Подтип	Подмножества отображений
1	Граф и мультиграф	а) Неориентированный <i>Undirected</i>	$AFX = \{ X\bar{F}X \}, UFU = \{ U\bar{F}U \},$ $XGU = \{ X\bar{G}U \}, UGX = \{ U\bar{G}X \}$
		б) Ориентированный <i>Directed</i>	$AFX = \{ X\bar{F}X, X\bar{F}X \},$ $UFU = \{ U\bar{F}U, U\bar{F}U, U\bar{F}U, U\bar{F}U \},$ $XGU = \{ X\bar{G}U, X\bar{G}U \},$ $UGX = \{ U\bar{G}X, U\bar{G}X \}$
2	Гиперграф	а) С неупорядоченными вершинами ребра <i>Unarranged</i>	$AFX = \{ X\bar{F}X \}, UFU = \{ U\bar{F}U \},$ $XGU = \{ X\bar{G}U \}, UGX = \{ U\bar{G}X \}$
		б) С упорядоченными вершинами в ребре <i>Arranged</i>	$AFX = \{ X\bar{F}X \}, UFU = \{ U\bar{F}U \},$ $XGU = \{ X\bar{G}U \}, UGX = \{ U\bar{G}^A X \}$
3	Ультраграф	а) С неупорядоченными вершинами ребра <i>Unarranged</i>	$AFX = \{ X\bar{F}X, X\bar{F}X, X\bar{F}X, X\bar{F}X \},$ $UFU = \{ U\bar{F}U, U\bar{F}U, U\bar{F}U, U\bar{F}U \},$ $XGU = \{ X\bar{G}U, X\bar{G}U \},$ $UGX = \{ U\bar{G}X, U\bar{G}X \}$
		б) С упорядоченными вершинами в ребре <i>Arranged</i>	$AFX = \{ X\bar{F}X, X\bar{F}X, X\bar{F}X, X\bar{F}X \},$ $UFU = \{ U\bar{F}U, U\bar{F}U, U\bar{F}U, U\bar{F}U \},$ $XGU = \{ X\bar{G}U, X\bar{G}U \},$ $UGX = \{ U\bar{G}X, U\bar{G}X, U\bar{G}^A X \}$

Получены необходимые расчетные соотношения и разработана методика оценки вычислительной и емкостной сложности алгоритмов решения задач структурного анализа и синтеза, описанных на уровне операций над графами и множествами, что позволило автоматизировать этот процесс.

Разработан алгоритм выбора оптимального способа представления графовых моделей множествами, базирующийся на анализе частоты выполнения операций и сложности их реализации при различных способах представления, что упрощает и ускоряет разработку алгоритма.

В общем случае задача выбора способа представления формулируется следующим образом.

Найти вариант представления графовой модели

$$l \in J: r_l = \min\{r_j / j \in J\}, \text{ при } L_l \leq L_{\text{доп}},$$

где J – количество комбинаций отображений в представлении графовой модели;

$L_j = \sum_{i=1}^K l_i$ – объем памяти для хранения отображений j -ой комбинации представления;

K – количество отображений в комбинации;

$r_j = \sum_{i=1}^K a_{i,j} f_i$ – количество операций, необходимых для получения недос-
тающих образов при j -м варианте представления;

f_i – кратность получения i -го отображения в алгоритме;

$a_{i,j}$ – количество операций, необходимых для получения i -го отображе-
ния:

$$A' = \{A_j \in A: \forall f_i \neq 0 \ \& \ a_{i,j} \neq 0 / i = \overline{1, n}\}.$$

Поскольку количество вариантов представления невелико ($K \leq 7$ – если рассматриваются табличные способы, $K \leq 15$ – если рассматриваются аналитические способы и $K \leq 22$ – если рассматриваются и те и другие способы одновременно), поставленная задача может быть решена методом перебора.

Выявлены основные факторы, влияющие на эффективность используемых для представления графовых моделей комбинированных структур данных, и предложен подход к формальной постановке задачи генерации оптимальных структур данных, что создает предпосылки для решения этой задачи.

Выполнен анализ существующих способов снижения вычислительной и емкостной сложности алгоритмов решения задач структурного анализа и синтеза и построена их классификация. Выявлены новые способы снижения вычислительной и емкостной сложности, определены факторы, которые необходимо учитывать при разработке решающих правил оптимизирующих преобразований. Разработана структура правила, описывающего оптимизирующее преобразование:

<Решающее правило> ::= <Условие существования> &
 <Условие целесообразности> &
 <Условие допустимости> &
 <Условие локальности> &
 <Заменяемая строка> → <Заменяющая строка>

Приведены примеры описания синтаксиса заменяемой и заменяющей строк, а также решающих правил оптимизирующих преобразований. Так правило исключения лишних вычислений при определении показателей связности, используемое для замены фрагментов вида

$$\forall i = 1, n-1: (\forall j = i+1, n: (s[i, j] := \text{Card}(\Gamma x[i] \cap \Gamma x[j])));$$

на фрагменты вида

$$\forall i=1, n: (\forall j=i+1, n: (s[i, j] := 0), \forall x[j] \in Fx[i]: (s[i, j] := \text{Card}(\Gamma x[i] \cap \Gamma x[j])));$$

при условии наличия в описании алгоритма отображения FX , где X – множество вершин гиперграфа, выглядит следующим образом

$$(\alpha \in A) \& (\gamma \in A) \rightarrow (\alpha ::= \beta),$$

где α, β – строки описания заменяемого и заменяющего фрагментов;

γ – строка, определяющая наличие отображения FX среди исходных данных алгоритма;

A – упорядоченное множество строк описания алгоритма.

Синтаксически заменяемый и заменяющий фрагменты при этом определяются следующим образом:

<Заменяемый фрагмент> ::=

$\forall$ <Индекс1> = 1, <Граница1> – 1:

($\forall$ <Индекс2> = <Индекс1> + 1, <Граница2>:

(<Матрица> [<Индекс1>, <Индекс2>] :=

$\text{Card}(\Gamma \langle \text{Множество1} \rangle [\langle \text{Индекс1} \rangle] \cap \Gamma \langle \text{Множество1} \rangle [\langle \text{Индекс2} \rangle]))$);

где <Индекс1>, <Индекс2> – идентификаторы целого типа;

<Граница1> – идентификатор целого типа – размерность множества вершин графа;

<Матрица> – идентификатор матрицы;

<Множество1> – идентификатор множества вершин графа.

<Заменяющий фрагмент> ::=

$\forall$ <Индекс1> = 1, <Граница1> – 1:

($\forall$ <Индекс2> = <Индекс1> + 1, <Граница2>:

(<Матрица> [<Индекс1>, <Индекс2>] := 0);

$\forall$ <Множество1> [<Индекс2>] $\in F$ <Множество1> [<Индекс1>]:

(<Матрица> [<Индекс1>, <Индекс2>] :=

$\text{Card}(\Gamma \langle \text{Множество1} \rangle [\langle \text{Индекс1} \rangle] \cap \Gamma \langle \text{Множество1} \rangle [\langle \text{Индекс2} \rangle]))$; .

Это преобразование допустимо, так как не требует дополнительной памяти, целесообразно, поскольку в любом случае снижает вычислительную сложность и локально, поскольку не требует изменений в других местах программы.

Полученные правила составили основу создания автоматического оптимизатора алгоритмов.

Разработаны полная и сокращенная методики описания, анализа и реализации алгоритмов решения задач структурного анализа и синтеза с использованием предлагаемых средств автоматизации, которые легли в основу созданного программного обеспечения.

В пятой главе «Реализация и экспериментальные исследования системы разработки алгоритмов» описаны назначение, состав (см. рис. 7) и функции основных частей системы описания, реализации и анализа алгоритмов решения задач структурного анализа и синтеза (см. рис. 8). Приведены результаты ее эксплуатации и экспериментальных исследований.



Рис. 7. Структура системы разработки алгоритмов и решения задач структурного анализа и синтеза

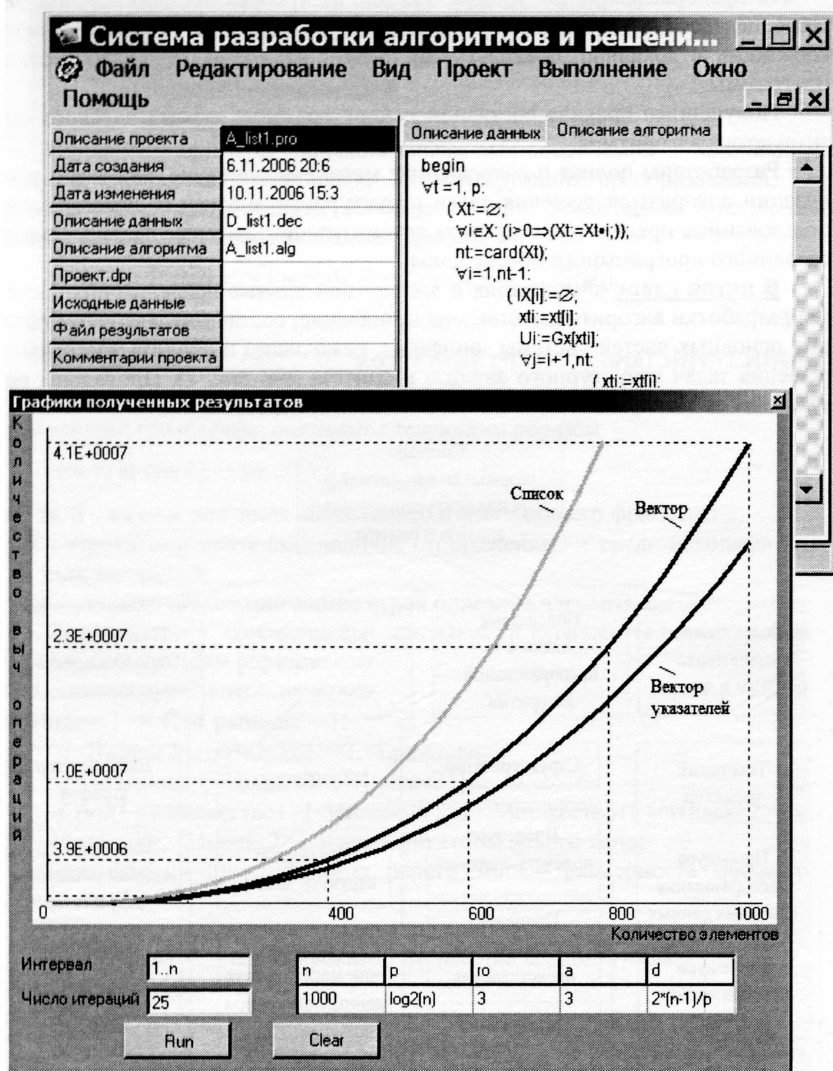


Рис. 8. Интерфейс системы

Программное обеспечение написано на *Delphi Pascal* и *Visual C++* и имеет объем ~ 22000 операторов.

Сформулированы цели экспериментальных исследований, даны характеристики объектов.

Экспериментальная проверка программы, реализующей метод структуризации, показала, что структурированная программа сортировки слияниями эквивалентна исходной с точки зрения результатов на всем диапазоне допустимых значений исходных данных. Время выполнения этой программы превышает время выполнения исходной при размере входа $N = 460000$ не более чем на 4 %, что согласуется с общепринятыми нормами.

Экспериментальные исследования различных программ решения задач структурного анализа и синтеза показали, что трудоемкость создания программ с использованием системы на порядок ниже трудоемкости создания тех же программ вручную.

Экспериментальная проверка результатов оценки системой функциональной вычислительной сложности показала, что реализованный в системе анализатор вычислительной сложности позволяет оперативно и с достаточной точностью оценивать вычислительную сложность разрабатываемого алгоритма и получать информацию, характеризующую вклад каждого фрагмента алгоритма в общую оценку, что делает возможным его использование для анализа алгоритма и оценки эффективности оптимизирующих преобразований.

Показано, что реализация разработанных в теоретической части методов и алгоритмов в виде единой программной системы обеспечивает автоматизацию разработки алгоритмов решения задач структурного анализа и синтеза, что позволяет существенно сократить трудоемкость подготовки таких задач к решению и обеспечить удовлетворяющие разработчика характеристики получаемого программного обеспечения (емкостную и вычислительную сложность). Таким образом, проведенные эксперименты и эксплуатация системы подтвердили результаты теоретических исследований, показали ее работоспособность и высокую эффективность применения.

ОСНОВНЫЕ РЕЗУЛЬТАТЫ РАБОТЫ

В результате проведенного с позиций системного подхода исследования проблемы разработки и анализа алгоритмов решения задач структурного анализа и синтеза предложена многоуровневая схема проектирования и реализации алгоритмов и сформулированы задачи, которые необходимо решить для разработки научных основ автоматизации указанного процесса.

1. Выполнена систематизация существующих и предложены новые структурные модели, позволяющие адекватно и более эффективно представлять объекты задач структурного анализа и синтеза программно-аппаратных средств вычислительной техники. Сформулированы правила соответствия компонентов объектов элементам графовых моделей.

2. Разработаны информационно-логическая модель структурного и неструктурного алгоритмов, а также модели комбинированных и иерархических структур данных. Определены формальные правила перехода от указанных объектов к их моделям. Полученные модели составили основу общего подхода к систематическому исследованию проблемы разработки и анализа алгоритмов.

3. Определены полные наборы инвариантов моделей структурных конструкций, сформулированы аксиоматика операции свертки уграфов структурных конструкций и теорема о необходимом и достаточном условии структурности алгоритма. В результате формально поставлена задача приведения алгоритмов к структурному виду, предложен и реализован метод ее решения, что позволило снять ограничение структурности используемых конструкций с предлагаемых языков описания алгоритмов решения задач структурного анализа и синтеза, а также обеспечило возможность автоматического анализа вычислительной сложности проектируемых алгоритмов.

4. Обоснован синтаксис и определена семантика двух специализированных языков описания алгоритмов решения задач структурного синтеза на уровне операций над графовыми моделями и операций над множествами. Применение указанных языков позволяет формулировать описание алгоритма с использованием абстракций предметной области и обеспечивает возможность автоматического выполнения частичной оптимизации алгоритма и синтеза оптимальных структур данных.

5. Поставлена и решена задача выбора оптимального в смысле минимизации вычислительной сложности способа представления графовой модели множествами. Намечены подходы к формальной постановке и решению задачи синтеза оптимальных с той же точки зрения структур данных для представления множеств и их отображений, выполнена классификация оптимизирующих преобразований. Определена структура формального правила оптимизирующего преобразования для поэтапной трансляции и сформулированы синтаксические правила двух наиболее употребимых оптимизирующих преобразований.

6. Теоретические результаты работы доведены до практической реализации в виде системы описания, реализации и анализа алгоритмов решения задач структурного анализа и синтеза. Экспериментальная проверка и эксплуатация системы подтвердили результаты теоретических исследований и основные научные положения, а также показали эффективность предложенных моделей, методов и лингвистических средств в обеспечении высокого качества проектируемых алгоритмов и сокращения сроков разработки.

Таким образом, на основании выполненных исследований сформулированы и обоснованы научные положения, совокупность решения которых можно квалифицировать как теоретическое обобщение и решение крупной научной проблемы создания методологии и комплекса математических,

лингвистических и программных средств автоматизации разработки алгоритмов решения задач анализа и синтеза структур программного обеспечения и устройств вычислительной техники.

Работы по теме диссертации

1. Иванова Г.С. Формальная постановка задачи структуризации алгоритмов // Вестник Московского государственного технического университета им. Н.Э. Баумана. Приборостроение. – 2005. – № 3(60). – С. 64-73.
2. Иванова Г.С. Нумерация вершин управляющего графа для решения задачи его структуризации // Информационные технологии. – 2005. – № 12. – С. 48-56.
3. Иванова Г.С. Математические модели структур данных // Информационные технологии. – 2006. – № 9. – С. 44-52.
4. Иванова Г.С. Модели объектов задач структурного синтеза // Наука и образование. Инженерное образование: Эл. науч. издание. – 2006. – № 12. (Номер гос. регистрации 0420600025\0053.)
5. Иванова Г.С. Способы представления структурных моделей // Наука и образование. Инженерное образование: Эл. науч. издание. – 2007. – № 1. (Номер гос. регистрации 0420700025\003.)
6. Иванова Г.С. Основы программирования: Учебник для вузов. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2004. – 416 с.
7. Иванова Г.С., Ничушкина Т.Н., Пугачев Е.К. Объектно-ориентированное программирование: Учебник для вузов / Под ред. Г.С. Ивановой. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2003. – 366 с.
8. Иванова Г.С. Технология программирования: Учебник для вузов. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2003. – 320 с.
9. Овчинников В.А., Иванова Г.С. Информационно-логическая модель алгоритма // Вестник Московского государственного технического университета им. Н.Э. Баумана. Приборостроение. – 2005. – № 2(59). – С. 109-121.
10. Овчинников В.А., Иванова Г.С. Проектно-исследовательская система решения задач структурного синтеза // Тез. докл. научно-технической конференции. – М., 2000. – С. 41.
11. Овчинников В.А., Иванова Г.С. Способы снижения вычислительной сложности реализации операций над множествами. // Научный вестник МГТУ ГА. Информатика. Прикладная математика. – 2002. – № 55. – С. 38-42.
12. Овчинников В.А., Иванова Г.С., Ничушкина Т.Н. Выбор структур данных для представления графов при решении комбинаторно-оптимизационных задач // Вестник Московского государственного технического университета им. Н.Э. Баумана. Приборостроение. – 2001. – № 2(43). – С. 39-51.

13. Иванова Г.С., Бакулина М.А. Автоматизация процесса разработки алгоритмов решения задач структурного синтеза на графах // Тез. Докл. Международной научно-технической конференции – М., 2003. – С. 166-167.
14. Иванова Г.С., Домосканов А.А. Оптимизация управления динамическим распределением оперативной памяти. // Современные информационные технологии: Сб. докл. и сообщ. Межвузовской юбилейной научно-технической конференции. – М., 2001. – С. 6-13.
15. Иванова Г.С., Ничушкина Т.Н. Проектирование синхронных интерфейсов интерактивных программных систем. // Научный вестник МГТУ ГА. Информатика. Прикладная математика. – 2002. – № 55. – С. 33-37.
16. Бакулина М.А., Иванова Г.С. Средства автоматизации процесса решения задач структурного синтеза // Аэрокосмические технологии: Материалы Первой международной научно-технической конференции, посвященной 90-летию со дня рождения академика В.Н. Челомея. – М., 2004. – С. 167-169.
17. Иванова Г.С., Пасечников К.А. Макрогенерация описаний структур данных для системы проектирования алгоритмов решения задач структурного синтеза // Современные информационные технологии: Сб. трудов каф., посвященный 175-летию МГТУ им. Н.Э. Баумана. – М.: Эликс+, 2004. – С. 69-73.
18. Иванова Г.С., Тихонов Ю.В. Введение в МПролог: Практическое пособие для программистов. – М.: Изд-во МГТУ, 1990. – 152 с.
19. Овчинников В.А., Иванова Г.С., Попов А.Ю. Применение метода ветвей и границ для решения задачи дихотомического разрезания гиперграфа // Вестник Московского государственного технического университета им. Н.Э. Баумана. Приборостроение. Информатика. – 1999. – № 2(34). – С. 78-91.

Подписано к печати 03.09.07. Заказ № 573

Объем 2,0 печ.л. Тираж 100 экз.

Типография МГТУ им. Н.Э. Баумана
105005, Москва, 2-я Бауманская ул., д.5
263-62-01