

КОМПИЛЯТОРЫ

*принципы, технологии
и инструментов*

Второе
издание



Addison
Wesley

Альфред В. Ахо

Моника С. Лам

Рави Сети

Джеффри Д. Ульман

Компиляторы

принципы, технологии и инструментарий

Второе издание

Альфред В. Ахо

Колумбийский университет

Моника С. Лам

Станфордский университет

Рави Сети

Агауа

Джеффри Д. Ульман

Станфордский университет



Москва • Санкт-Петербург • Киев
2017

ББК 32.973.26-018.2.75

К63

УДК 681.3.07

Издательский дом "Вильямс"

Зав. редакцией *С.Н. Тригуб*

Перевод с английского и редакция канд. техн. наук *И.В. Красикова*

По общим вопросам обращайтесь в Издательский дом "Вильямс" по адресу:
info@williamspublishing.com, http://www.williamspublishing.com

Ахо, Альфред В., Лам, Моника С., Сети, Рави, Ульман, Джеффри Д.

К63 Компиляторы: принципы, технологии и инструментарий, 2-е изд. : Пер. с англ. — М. : ООО "И.Д. Вильямс", 2017. — 1184 с. : ил. — Парал. тит. англ.

ISBN 978-5-8459-1932-8 (рус.)

ББК 32.973.26-018.2.75

Все названия программных продуктов являются зарегистрированными торговыми марками соответствующих фирм.

Никакая часть настоящего издания ни в каких целях не может быть воспроизведена в какой бы то ни было форме и какими бы то ни было средствами, будь то электронные или механические, включая фотокопирование и запись на магнитный носитель, если на это нет письменного разрешения издательства Addison-Wesley Publishing Company, Inc.

Authorized translation from the English language edition published by Addison-Wesley Publishing Company, Inc, Copyright © 2007

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the publisher.

Russian language edition published by Williams Publishing House according to the Agreement with R&I Enterprises International, Copyright © 2017.

Научно-популярное издание

Альфред В. Ахо, Моника С. Лам, Рави Сети, Джеффри Д. Ульман

Компиляторы:

принципы, технологии и инструментарий

2-е издание

Литературный редактор *Л.Н. Красножон*

* Верстка *А.Н. Полинчик*

Художественный редактор *С.А. Чернокозинский*

Корректоры *Л.А. Гордиенко, Л.В. Чернокозинская*

Подписано в печать 31.01.2017. Формат 70x100/16

Гарнитура Times.

Усл. печ. л. 74,0. Уч.-изд. л. 53,0

Доп. тираж 200 экз. Заказ № 825.

Отпечатано в АО «Первая Образцовая типография»

Филиал «Чеховский Печатный Двор»

142300, Московская область, г. Чехов, ул. Полиграфистов, д.1

Сайт: www.chpd.ru, E-mail: sales@chpd.ru, тел. 8(499)270-73-59

ООО "И. Д. Вильямс", 127055, г. Москва, ул. Лесная, д. 43, стр. 1

ISBN 978-5-8459-1932-8 (рус.)

ISBN 0-321-48681-1 (англ.)

© Издательский дом "Вильямс", 2017

© Copyright Pearson Education, Inc., 2007

Оглавление

Предисловие	24
Глава 1. Введение в компиляцию	29
Глава 2. Простой синтаксически управляемый транслятор	75
Глава 3. Лексический анализ	155
Глава 4. Синтаксический анализ	251
Глава 5. Синтаксически управляемая трансляция	383
Глава 6. Генерация промежуточного кода	443
Глава 7. Среды времени выполнения	525
Глава 8. Генерация кода	617
Глава 9. Машинно-независимые оптимизации	705
Глава 10. Параллелизм на уровне команд	841
Глава 11. Оптимизация параллелизма и локальности	911
Глава 12. Межпроцедурный анализ	1061
Приложение А. Завершенный пример начальной стадии компилятора	1133
Приложение Б. Поиск линейно независимых решений	1163
Предметный указатель	1167

Содержание

Предисловие	24
Глава 1. Введение в компиляцию	29
1.1 Компиляторы	29
1.1.1 Упражнения к разделу 1.1	32
1.2 Структура компилятора	32
1.2.1 Лексический анализ	33
1.2.2 Синтаксический анализ	35
1.2.3 Семантический анализ	37
1.2.4 Генерация промежуточного кода	38
1.2.5 Оптимизация кода	39
1.2.6 Генерация кода	39
1.2.7 Управление таблицей символов	40
1.2.8 Объединение фаз в проходы	41
1.2.9 Инструментарий для создания компиляторов	41
1.3 Эволюция языков программирования	42
1.3.1 Переход к языкам высокого уровня	42
1.3.2 Влияние на компиляторы	44
1.3.3 Упражнения к разделу 1.3	45
1.4 “Компилятороведение”	45
1.4.1 Моделирование при проектировании и реализации компилятора	45
1.4.2 Изучение оптимизации кода	46
1.5 Применения технологий компиляторов	48
1.5.1 Реализация высокоуровневых языков программирования	48
1.5.2 Оптимизация для архитектуры компьютера	50
1.5.3 Разработка новых архитектур компьютеров	52
1.5.4 Трансляции программ	54
1.5.5 Инструментарий для повышения производительности программного обеспечения	55

1.6	Азы языков программирования	57
1.6.1	Понятия статического и динамического	58
1.6.2	Среды и состояния	58
1.6.3	Статическая область видимости и блочная структура	61
1.6.4	Явное управление доступом	65
1.6.5	Динамическая область видимости	65
1.6.6	Механизмы передачи параметров	68
1.6.7	Псевдонимы	70
1.6.8	Упражнения к разделу 1.6	70
1.7	Резюме к главе 1	72
1.8	Список литературы к главе 1	73
Глава 2.	Простой синтаксически управляемый транслятор	75
2.1	Введение	76
2.2	Определение синтаксиса	78
2.2.1	Определения грамматик	79
2.2.2	Выведение	81
2.2.3	Деревья разбора	82
2.2.4	Неоднозначности	84
2.2.5	Ассоциативность операторов	85
2.2.6	Приоритет операторов	86
2.2.7	Упражнения к разделу 2.2	89
2.3	Синтаксически управляемая трансляция	90
2.3.1	Постфиксная запись	91
2.3.2	Синтезированные атрибуты	92
2.3.3	Простые синтаксически управляемые определения	94
2.3.4	Обходы дерева	95
2.3.5	Схемы трансляции	97
2.3.6	Упражнения к разделу 2.3	99
2.4	Разбор	100
2.4.1	Нисходящий анализ	101
2.4.2	Предиктивный анализ	104
2.4.3	Использование пустых продукций	106
2.4.4	Разработка предиктивного анализатора	106
2.4.5	Левая рекурсия	107
2.4.6	Упражнения к разделу 2.4	109
2.5	Транслятор простых выражений	109
2.5.1	Абстрактный и конкретный синтаксис	110
2.5.2	Адаптация схемы трансляции	111
2.5.3	Процедуры для нетерминалов	113
2.5.4	Упрощение транслятора	114

2.5.5	Завершенная программа	115
2.6	Лексический анализ	118
2.6.1	Удаление пробельных символов и комментариев	119
2.6.2	Опережающее чтение	120
2.6.3	Константы	121
2.6.4	Распознавание ключевых слов и идентификаторов	121
2.6.5	Лексический анализатор	123
2.6.6	Упражнения к разделу 2.6	128
2.7	Таблицы символов	128
2.7.1	Таблица символов для области видимости	129
2.7.2	Использование таблиц символов	133
2.8	Генерация промежуточного кода	136
2.8.1	Два вида промежуточных представлений	136
2.8.2	Построение синтаксических деревьев	137
2.8.3	Статические проверки	142
2.8.4	Трехадресный код	144
2.8.5	Упражнения к разделу 2.8	151
2.9	Резюме к главе 2	152
Глава 3. Лексический анализ		155
3.1	Роль лексического анализатора	155
3.1.1	Лексический и синтаксический анализ	157
3.1.2	Токены, шаблоны и лексемы	157
3.1.3	Атрибуты токенов	159
3.1.4	Лексические ошибки	160
3.1.5	Упражнения к разделу 3.1	161
3.2	Буферизация ввода	162
3.2.1	Пары буферов	162
3.2.2	Ограничители	163
3.3	Спецификация токенов	165
3.3.1	Строки и языки	165
3.3.2	Операции над языками	166
3.3.3	Регулярные выражения	168
3.3.4	Регулярные определения	170
3.3.5	Расширения регулярных выражений	172
3.3.6	Упражнения к разделу 3.3	173
3.4	Распознавание токенов	177
3.4.1	Диаграммы переходов	179
3.4.2	Распознавание зарезервированных слов и идентификаторов	181
3.4.3	Завершение примера	183

3.4.4	Архитектура лексического анализатора на основе диаграммы переходов	184
3.4.5	Упражнения к разделу 3.4	187
3.5	Генератор лексических анализаторов Lex	191
3.5.1	Использование Lex	191
3.5.2	Структура программ Lex	192
3.5.3	Разрешение конфликтов в Lex	196
3.5.4	Прогностический оператор	197
3.5.5	Упражнения к разделу 3.5	198
3.6	Конечные автоматы	199
3.6.1	Недетерминированные конечные автоматы	200
3.6.2	Таблицы переходов	201
3.6.3	Принятие входной строки автоматом	201
3.6.4	Детерминированный конечный автомат	203
3.6.5	Упражнения к разделу 3.6	204
3.7	От регулярных выражений к автоматам	205
3.7.1	Преобразование НКА в ДКА	206
3.7.2	Моделирование НКА	210
3.7.3	Эффективность моделирования НКА	210
3.7.4	Построение НКА из регулярного выражения	213
3.7.5	Эффективность алгоритма обработки строк	218
3.7.6	Упражнения к разделу 3.7	221
3.8	Разработка генератора лексических анализаторов	221
3.8.1	Структура генерируемого анализатора	221
3.8.2	Распознавание шаблонов на основе НКА	223
3.8.3	ДКА для лексических анализаторов	225
3.8.4	Реализация прогностического оператора	226
3.8.5	Упражнения к разделу 3.8	228
3.9	Оптимизация распознавателей на основе ДКА	229
3.9.1	Важные состояния НКА	229
3.9.2	Функции, вычисляемые на синтаксическом дереве	231
3.9.3	Вычисление nullable, firstpos и lastpos	232
3.9.4	Вычисление followpos	234
3.9.5	Преобразование регулярного выражения непосредственно в ДКА	235
3.9.6	Минимизация количества состояний ДКА	237
3.9.7	Минимизация состояний в лексических анализаторах	242
3.9.8	Компромисс между скоростью и используемой памятью при моделировании ДКА	242
3.9.9	Упражнения к разделу 3.9	244

3.10	Резюме к главе 3	245
3.11	Список литературы к главе 3	247
Глава 4.	Синтаксический анализ	251
4.1	Введение	252
4.1.1	Роль синтаксического анализатора	252
4.1.2	Образцы грамматик	253
4.1.3	Обработка синтаксических ошибок	254
4.1.4	Стратегии восстановления после ошибок	256
4.2	Контекстно-свободные грамматики	258
4.2.1	Формальное определение контекстно-свободной грамматики	258
4.2.2	Соглашения об обозначениях	260
4.2.3	Порождения	261
4.2.4	Деревья разбора и порождения	263
4.2.5	Неоднозначность	265
4.2.6	Проверка языка, сгенерированного грамматикой	266
4.2.7	Контекстно-свободные грамматики и регулярные выражения	267
4.2.8	Упражнения к разделу 4.2	269
4.3	Разработка грамматики	272
4.3.1	Лексический и синтаксический анализ	273
4.3.2	Устранение неоднозначности	273
4.3.3	Устранение левой рекурсии	275
4.3.4	Левая факторизация	278
4.3.5	Не контекстно-свободные языковые конструкции	279
4.3.6	Упражнения к разделу 4.3	280
4.4	Нисходящий синтаксический анализ	281
4.4.1	Синтаксический анализ методом рекурсивного спуска	283
4.4.2	FIRST и FOLLOW	285
4.4.3	LL(1)-грамматики	288
4.4.4	Нерекурсивный предиктивный синтаксический анализ	292
4.4.5	Восстановление после ошибок в предиктивном синтаксическом анализе	295
4.4.6	Упражнения к разделу 4.4	298
4.5	Восходящий синтаксический анализ	301
4.5.1	Свертки	302
4.5.2	Обрезка основ	302
4.5.3	Синтаксический анализ “перенос/свертка”	304
4.5.4	Конфликты в процессе ПС-анализа	306
4.5.5	Упражнения к разделу 4.5	309

4.6	Введение в LR-анализ: простой LR	309
4.6.1	Обоснование использования LR-анализаторов	310
4.6.2	Пункты и LR(0)-автомат	311
4.6.3	Алгоритм LR-анализа	317
4.6.4	Построение таблиц SLR-анализа	322
4.6.5	Активные префиксы	326
4.6.6	Упражнения к разделу 4.6	328
4.7	Более мощные LR-анализаторы	330
4.7.1	Канонические LR(1)-пункты	331
4.7.2	Построение множеств LR(1)-пунктов	332
4.7.3	Канонические таблицы LR(1)-анализа	336
4.7.4	Построение LALR-таблиц синтаксического анализа	338
4.7.5	Эффективное построение таблиц LALR-анализа	344
4.7.6	Уплотнение таблиц LR-анализа	349
4.7.7	Упражнения к разделу 4.7	352
4.8	Использование неоднозначных грамматик	353
4.8.1	Использование приоритетов и ассоциативности для разрешения конфликтов	353
4.8.2	Неоднозначность "висящего else"	357
4.8.3	Восстановление после ошибок в LR-анализе	358
4.8.4	Упражнения к разделу 4.8	361
4.9	Генераторы синтаксических анализаторов	363
4.9.1	Генератор синтаксических анализаторов Yacc	364
4.9.2	Использование Yacc с неоднозначной грамматикой	368
4.9.3	Создание лексического анализатора в Yacc с помощью Lex	371
4.9.4	Восстановление после ошибок в Yacc	372
4.9.5	Упражнения к разделу 4.9	375
4.10	Резюме к главе 4	375
4.11	Список литературы к главе 4	378
Глава 5.	Синтаксически управляемая трансляция	383
5.1	Синтаксически управляемые определения	384
5.1.1	Наследуемые и синтезируемые атрибуты	384
5.1.2	Вычисление СУО в узлах дерева разбора	387
5.1.3	Упражнения к разделу 5.1	390
5.2	Порядок вычисления в СУО	391
5.2.1	Графы зависимостей	391
5.2.2	Упорядочение вычисления атрибутов	393
5.2.3	S-атрибутные определения	394
5.2.4	L-атрибутные определения	394

5.2.5	Семантические правила с контролируруемыми побочными действиями	396
5.2.6	Упражнения к разделу 5.2	398
5.3	Применения синтаксически управляемой трансляции	399
5.3.1	Построение синтаксических деревьев	400
5.3.2	Структура типа	404
5.3.3	Упражнения к разделу 5.3	406
5.4	Синтаксически управляемые схемы трансляции	406
5.4.1	Постфиксные схемы трансляции	407
5.4.2	Реализация постфиксной СУТ с использованием стека синтаксического анализатора	407
5.4.3	СУТ с действиями внутри продукции	410
5.4.4	Устранение левой рекурсии из СУТ	411
5.4.5	СУТ для L-атрибутных определений	414
5.4.6	Упражнения к разделу 5.4	421
5.5	Реализация L-атрибутных СУО	422
5.5.1	Трансляция в процессе синтаксического анализа методом рекурсивного спуска	423
5.5.2	Генерация кода “на лету”	425
5.5.3	L-атрибутные СУО и LL-синтаксический анализ	428
5.5.4	Восходящий синтаксический анализ L-атрибутных СУО	434
5.5.5	Упражнения к разделу 5.5	439
5.6	Резюме к главе 5	439
5.7	Список литературы к главе 5	441
Глава 6.	Генерация промежуточного кода	443
6.1	Варианты синтаксических деревьев	445
6.1.1	Ориентированные ациклические графы для выражений	445
6.1.2	Метод номера значения для построения ориентированных ациклических графов	447
6.1.3	Упражнения к разделу 6.1	450
6.2	Трехадресный код	450
6.2.1	Адреса и команды	450
6.2.2	Четверки	454
6.2.3	Тройки	455
6.2.4	Представление в виде статических единственных присваиваний	457
6.2.5	Упражнения к разделу 6.2	458
6.3	Типы и объявления	459
6.3.1	Выражения типов	459
6.3.2	Эквивалентность типов	461

6.3.3	Объявления	462
6.3.4	Размещение локальных имен в памяти	462
6.3.5	Последовательности объявлений	464
6.3.6	Поля в записях и классах	466
6.3.7	Упражнения к разделу 6.3	467
6.4	Трансляция выражений	468
6.4.1	Операции в выражениях	468
6.4.2	Инкрементная трансляция	470
6.4.3	Адресация элементов массива	471
6.4.4	Трансляция обращений к массиву	473
6.4.5	Упражнения к разделу 6.4	475
6.5	Проверка типов	477
6.5.1	Правила проверки типов	477
6.5.2	Преобразования типов	478
6.5.3	Перегрузка функций и операторов	481
6.5.4	Выведение типа и полиморфные функции	482
6.5.5	Алгоритм унификации	487
6.5.6	Упражнения к разделу 6.5	490
6.6	Поток управления	491
6.6.1	Булевы выражения	492
6.6.2	Код сокращенного вычисления	492
6.6.3	Инструкции потока управления	493
6.6.4	Трансляция логических выражений с помощью потока управления	496
6.6.5	Устранение излишних команд перехода	499
6.6.6	Булевы значения и код с переходами	501
6.6.7	Упражнения к разделу 6.6	502
6.7	Обратные поправки	504
6.7.1	Однопроходная генерация кода с использованием обратных поправок	504
6.7.2	Обратные поправки для булевых выражений	505
6.7.3	Инструкции потока управления	508
6.7.4	Инструкции break, continue и goto	511
6.7.5	Упражнения к разделу 6.7	512
6.8	Инструкции выбора	514
6.8.1	Трансляция инструкций выбора	514
6.8.2	Синтаксически управляемая трансляция инструкций выбора	515
6.8.3	Упражнения к разделу 6.8	517
6.9	Промежуточный код процедур	517

6.10	Резюме к главе 6	520
6.11	Список литературы к главе 6	521
Глава 7.	Среды времени выполнения	525
7.1	Организация памяти	525
7.1.1	Статическое и динамическое распределение памяти	527
7.2	Выделение памяти в стеке	528
7.2.1	Деревья активации	529
7.2.2	Записи активации	532
7.2.3	Последовательности вызовов	535
7.2.4	Данные переменной длины в стеке	539
7.2.5	Упражнения к разделу 7.2	540
7.3	Доступ к нелокальным данным в стеке	542
7.3.1	Доступ к данным при отсутствии вложенных процедур	542
7.3.2	Вложенные процедуры	543
7.3.3	Язык с вложенными объявлениями процедур	544
7.3.4	Глубина вложенности	545
7.3.5	Связи доступа	547
7.3.6	Работа со связями доступа	548
7.3.7	Связи доступа для процедур, являющихся параметрами	550
7.3.8	Дисплеи	551
7.3.9	Упражнения к разделу 7.3	554
7.4	Управление кучей	555
7.4.1	Диспетчер памяти	555
7.4.2	Иерархия памяти компьютера	557
7.4.3	Локальность в программах	559
7.4.4	Снижение фрагментации	561
7.4.5	Освобождение памяти вручную	565
7.4.6	Упражнения к разделу 7.4	568
7.5	Введение в сборку мусора	568
7.5.1	Цели проектирования сборщиков мусора	569
7.5.2	Достижимость	572
7.5.3	Сборщики мусора с подсчетом ссылок	574
7.5.4	Упражнения к разделу 7.5	576
7.6	Введение в сборку на основе отслеживания	576
7.6.1	Базовый сборщик мусора	577
7.6.2	Базовая абстракция	580
7.6.3	Оптимизация алгоритма “пометить и подмести”	582
7.6.4	Сборщики мусора “пометить и сжать”	583
7.6.5	Копирующие сборщики	587
7.6.6	Сравнение стоимости	589

7.6.7	Упражнения к разделу 7.6	590
7.7	Распределенная сборка мусора	591
7.7.1	Инкрементная сборка мусора	591
7.7.2	Инкрементный анализ достижимости	593
7.7.3	Основы частичной сборки	596
7.7.4	Сборка мусора по поколениям	597
7.7.5	Алгоритм поезда	599
7.7.6	Упражнения к разделу 7.7	603
7.8	Дополнительные вопросы сборки мусора	604
7.8.1	Параллельная сборка мусора	605
7.8.2	Частичное перемещение объектов	608
7.8.3	Консервативная сборка мусора для небезопасных языков программирования	608
7.8.4	Слабые ссылки	609
7.8.5	Упражнения к разделу 7.8	610
7.9	Резюме к главе 7	611
7.10	Список литературы к главе 7	614
Глава 8.	Генерация кода	617
8.1	Вопросы проектирования генератора кода	619
8.1.1	Вход генератора кода	619
8.1.2	Целевая программа	620
8.1.3	Выбор команд	621
8.1.4	Распределение регистров	623
8.1.5	Порядок вычислений	625
8.2	Целевой язык	625
8.2.1	Простая модель целевой машины	625
8.2.2	Стоимость программ и команд	629
8.2.3	Упражнения к разделу 8.2	630
8.3	Адреса в целевом коде	632
8.3.1	Статическое выделение памяти	632
8.3.2	Выделение памяти в стеке	635
8.3.3	Адреса имен времени выполнения	638
8.3.4	Упражнения к разделу 8.3	639
8.4	Базовые блоки и графы потоков	640
8.4.1	Базовые блоки	641
8.4.2	Информация о дальнейшем использовании	643
8.4.3	Графы потоков	644
8.4.4	Представление графов потоков	646
8.4.5	Циклы	646
8.4.6	Упражнения к разделу 8.4	647

8.5	Оптимизация базовых блоков	648
8.5.1	Представление базовых блоков с использованием ориентированных ациклических графов	648
8.5.2	Поиск локальных общих подвыражений	649
8.5.3	Устранение неиспользуемого кода	651
8.5.4	Применение алгебраических тождеств	652
8.5.5	Представление обращений к массивам	653
8.5.6	Присваивание указателей и вызовы процедур	655
8.5.7	Сборка базового блока из ориентированного ациклического графа	656
8.5.8	Упражнения к разделу 8.5	658
8.6	Простой генератор кода	660
8.6.1	Дескрипторы регистров и адресов	661
8.6.2	Алгоритм генерации кода	661
8.6.3	Разработка функции <i>getReg</i>	665
8.6.4	Упражнения к разделу 8.6	667
8.7	Локальная оптимизация	668
8.7.1	Устранение излишних загрузок и сохранений	668
8.7.2	Устранение недостижимого кода	669
8.7.3	Оптимизация потока управления	670
8.7.4	Алгебраические упрощения и снижение стоимости	671
8.7.5	Использование машинных идиом	671
8.7.6	Упражнения к разделу 8.7	671
8.8	Распределение и назначение регистров	672
8.8.1	Глобальное распределение регистров	672
8.8.2	Счетчики использований	673
8.8.3	Назначение регистров для внешних циклов	676
8.8.4	Распределение регистров путем раскраски графа	676
8.8.5	Упражнения к разделу 8.8	677
8.9	Выбор команд путем переписывания дерева	677
8.9.1	Схемы трансляции деревьев	678
8.9.2	Генерация кода путем замощения входного дерева	681
8.9.3	Поиск соответствий с использованием синтаксического анализа	684
8.9.4	Программы семантической проверки	685
8.9.5	Обобщенный поиск соответствий	686
8.9.6	Упражнения к разделу 8.9	688
8.10	Генерация оптимального кода для выражений	688
8.10.1	Числа Ершова	689
8.10.2	Генерация кода на основе помеченных деревьев выражений	690

8.10.3	Вычисление выражений при недостаточном количестве регистров	692
8.10.4	Упражнения к разделу 8.10	694
8.11	Генерация кода с использованием динамического программирования	695
8.11.1	Последовательные вычисления	696
8.11.2	Алгоритм динамического программирования	697
8.11.3	Упражнения к разделу 8.11	700
8.12	Резюме к главе 8	700
8.13	Список литературы к главе 8	702
Глава 9.	Машинно-независимые оптимизации	705
9.1	Основные источники оптимизации	706
9.1.1	Причины избыточности	706
9.1.2	Конкретный пример: быстрая сортировка	707
9.1.3	Трансформации, сохраняющие семантику	710
9.1.4	Глобальные общие подвыражения	710
9.1.5	Распространение копий	712
9.1.6	Удаление бесполезного кода	713
9.1.7	Перемещение кода	714
9.1.8	Переменные индукции и снижение стоимости	715
9.1.9	Упражнения к разделу 9.1	718
9.2	Введение в анализ потоков данных	719
9.2.1	Абстракция потока данных	720
9.2.2	Схема анализа потока данных	722
9.2.3	Схемы потоков данных в базовых блоках	723
9.2.4	Достигающие определения	725
9.2.5	Анализ активных переменных	733
9.2.6	Доступные выражения	735
9.2.7	Резюме	740
9.2.8	Упражнения к разделу 9.2	740
9.3	Основы анализа потока данных	743
9.3.1	Полурешетки	744
9.3.2	Передаточные функции	750
9.3.3	Итеративный алгоритм в обобщенной структуре	753
9.3.4	Смысл решения потока данных	755
9.3.5	Упражнения к разделу 9.3	759
9.4	Распространение констант	760
9.4.1	Значения потока данных для структуры распространения констант	761
9.4.2	Сбор в структуре распространения констант	762

9.4.3	Передаточные функции для структуры распространения констант	762
9.4.4	Монотонность структуры распространения констант	763
9.4.5	Недистрибутивность структуры распространения констант	764
9.4.6	Интерпретация результатов	765
9.4.7	Упражнения к разделу 9.4	767
9.5	Устранение частичной избыточности	768
9.5.1	Источники избыточности	769
9.5.2	Все ли избыточные вычисления могут быть устранены?	771
9.5.3	Отложенное перемещение кода	773
9.5.4	Ожидаемость выражений	774
9.5.5	Алгоритм отложенного перемещения кода	775
9.5.6	Упражнения к разделу 9.5	785
9.6	Циклы в графах потоков	787
9.6.1	Доминаторы	787
9.6.2	Упорядочение в глубину	790
9.6.3	Ребра в глубинном остоном дереве	792
9.6.4	Обратные ребра и приводимость	794
9.6.5	Глубина графа потока	795
9.6.6	Естественные циклы	796
9.6.7	Скорость сходимости итеративных алгоритмов потоков данных	798
9.6.8	Упражнения к разделу 9.6	801
9.7	Анализ на основе областей	803
9.7.1	Области	804
9.7.2	Иерархии областей для приводимых графов потоков	805
9.7.3	Обзор анализа на основании областей	808
9.7.4	Необходимые предположения о передачных функциях	810
9.7.5	Алгоритм анализа на основе областей	811
9.7.6	Обработка неприводимых графов потоков	816
9.7.7	Упражнения к разделу 9.7	818
9.8	Символический анализ	819
9.8.1	Аффинные выражения ссылочных переменных	819
9.8.2	Формулировка задачи потока данных	822
9.8.3	Символический анализ на основе областей	827
9.8.4	Упражнения к разделу 9.8	832
9.9	Резюме к главе 9	833
9.10	Список литературы к главе 9	838

Глава 10. Параллелизм на уровне команд	841
10.1 Архитектуры процессоров	842
10.1.1 Конвейерная обработка команд и задержки ветвления	842
10.1.2 Конвейерное выполнение	843
10.1.3 Многоадресные команды	844
10.2 Ограничения планирования кода	845
10.2.1 Зависимость через данные	846
10.2.2 Поиск зависимостей среди обращений к памяти	846
10.2.3 Компромиссы между использованием регистров и параллелизмом	848
10.2.4 Упорядочение фаз распределения регистров и планирования кода	851
10.2.5 Зависимость от управления	852
10.2.6 Поддержка опережающего выполнения	853
10.2.7 Базовая модель машины	855
10.2.8 Упражнения к разделу 10.2	856
10.3 Планирование базовых блоков	857
10.3.1 Графы зависимости данных	858
10.3.2 Планирование списков базовых блоков	859
10.3.3 Приоритетные топологические порядки	861
10.3.4 Упражнения к разделу 10.3	863
10.4 Глобальное планирование кода	864
10.4.1 Примитивное перемещение кода	865
10.4.2 Восходящее перемещение кода	867
10.4.3 Нисходящее перемещение кода	868
10.4.4 Обновление зависимостей данных	870
10.4.5 Алгоритм глобального планирования	870
10.4.6 Усовершенствованные методы перемещения кода	874
10.4.7 Взаимодействие с динамическими планировщиками	875
10.4.8 Упражнения к разделу 10.4	876
10.5 Программная конвейеризация	876
10.5.1 Введение	877
10.5.2 Программная конвейеризация циклов	879
10.5.3 Распределение регистров и генерация кода	882
10.5.4 Циклы с зависимыми итерациями	883
10.5.5 Цели и ограничения программной конвейеризации	884
10.5.6 Алгоритм программной конвейеризации	888
10.5.7 Планирование ациклических графов зависимости данных	889
10.5.8 Планирование графов с циклическими зависимостями	891
10.5.9 Усовершенствования алгоритма конвейеризации	899

10.5.10	Модульное расширение переменных	900
10.5.11	Условные инструкции	903
10.5.12	Аппаратная поддержка программной конвейеризации	904
10.5.13	Упражнения к разделу 10.5	904
10.6	Резюме к главе 10	907
10.7	Список литературы к главе 10	909
Глава 11.	Оптимизация параллелизма и локальности	911
11.1	Фундаментальные концепции	914
11.1.1	Многопроцессорность	914
11.1.2	Параллелизм в приложениях	917
11.1.3	Параллелизм на уровне циклов	918
11.1.4	Локальность данных	920
11.1.5	Введение в теорию аффинных преобразований	922
11.2	Пример посерьезнее: умножение матриц	927
11.2.1	Алгоритм умножения матриц	927
11.2.2	Оптимизации	930
11.2.3	Интерференция кэша	933
11.2.4	Упражнения к разделу 11.2	933
11.3	Пространства итераций	934
11.3.1	Построение пространств итераций вложенных циклов	934
11.3.2	Порядок выполнения вложенности циклов	936
11.3.3	Матричная запись неравенств	938
11.3.4	Добавление символьных констант	938
11.3.5	Управление порядком выполнения	939
11.3.6	Изменение осей	944
11.3.7	Упражнения к разделу 11.3	945
11.4	Аффинные индексы массивов	948
11.4.1	Аффинные обращения к данным	948
11.4.2	Аффинное и неаффинное обращения на практике	949
11.4.3	Упражнения к разделу 11.4	950
11.5	Повторное использование данных	951
11.5.1	Типы повторных использований	952
11.5.2	Собственные повторные использования	953
11.5.3	Собственное пространственное повторное использование	958
11.5.4	Групповое повторное использование	959
11.5.5	Упражнения к разделу 11.5	962
11.6	Анализ зависимости данных в массивах	964
11.6.1	Определение зависимостей данных доступов к массивам	965
11.6.2	Целочисленное линейное программирование	966

11.6.3	НОД	967
11.6.4	Эвристики для решения задачи целочисленного линейного программирования	969
11.6.5	Решение обобщенной задачи целочисленного линейного программирования	973
11.6.6	Резюме	975
11.6.7	Упражнения к разделу 11.6	976
11.7	Поиск параллельности, не требующей синхронизации	978
11.7.1	Вводный пример	978
11.7.2	Разбиения аффинного пространства	981
11.7.3	Ограничения разбиений пространства	982
11.7.4	Решение ограничений разбиений пространств	986
11.7.5	Простой алгоритм генерации кода	990
11.7.6	Устранение пустых итераций	993
11.7.7	Устранение проверок из внутреннего цикла	996
11.7.8	Преобразования исходного кода	998
11.7.9	Упражнения к разделу 11.7	1003
11.8	Синхронизация между параллельными циклами	1005
11.8.1	Постоянное количество синхронизаций	1006
11.8.2	Графы зависимостей программ	1007
11.8.3	Иерархическое время	1009
11.8.4	Алгоритм распараллеливания	1012
11.8.5	Упражнения к разделу 11.8	1013
11.9	Конвейеризация	1013
11.9.1	Что такое конвейеризация	1014
11.9.2	Последовательная сверхрелаксация: пример	1016
11.9.3	Полностью переставляемые циклы	1017
11.9.4	Конвейеризация полностью переставляемых циклов	1018
11.9.5	Общая теория	1021
11.9.6	Ограничения временного разбиения	1022
11.9.7	Решение временных ограничений с использованием леммы Фаркаша	1026
11.9.8	Преобразования кода	1029
11.9.9	Параллелизм с минимальной синхронизацией	1035
11.9.10	Упражнения к разделу 11.9	1037
11.10	Оптимизации локальности	1039
11.10.1	Временная локальность вычисляемых данных	1040
11.10.2	Сжатие массива	1041
11.10.3	Чередование частей	1044
11.10.4	Алгоритмы оптимизации локальности	1047
11.10.5	Упражнения к разделу 11.10	1050

11.11	Прочие применения аффинных преобразований	1050
11.11.1	Машины с распределенной памятью	1050
11.11.2	Процессоры с одновременным выполнением нескольких команд	1051
11.11.3	Векторные и SIMD-команды	1052
11.11.4	Предвыборка	1053
11.12	Резюме к главе 11	1054
11.13	Список литературы к главе 11	1057
Глава 12.	Межпроцедурный анализ	1061
12.1	Базовые концепции	1062
12.1.1	Графы вызовов	1062
12.1.2	Чувствительность к контексту	1064
12.1.3	Строки вызовов	1067
12.1.4	Контекстно-чувствительный анализ на основе клонирования	1069
12.1.5	Контекстно-чувствительный анализ на основе резюме	1070
12.1.6	Упражнения к разделу 12.1	1073
12.2	Необходимость межпроцедурного анализа	1075
12.2.1	Вызовы виртуальных методов	1076
12.2.2	Анализ псевдонимов указателей	1076
12.2.3	Параллельность	1077
12.2.4	Поиск программных ошибок и уязвимых мест	1077
12.2.5	SQL-ввод	1078
12.2.6	Переполнение буфера	1080
12.3	Логическое представление потока данных	1081
12.3.1	Введение в Datalog	1082
12.3.2	Правила Datalog	1083
12.3.3	Интенциональные и экстенциональные предикаты	1085
12.3.4	Выполнение программы Datalog	1088
12.3.5	Инкрементное вычисление программ Datalog	1090
12.3.6	Проблематичные правила Datalog	1091
12.3.7	Упражнения к разделу 12.3	1093
12.4	Простой алгоритм анализа указателей	1095
12.4.1	Сложность анализа указателей	1096
12.4.2	Модель указателей и ссылок	1097
12.4.3	Нечувствительность к потоку	1098
12.4.4	Формулировка с применением Datalog	1099
12.4.5	Использование информации о типе	1101
12.4.6	Упражнения к разделу 12.4	1102
12.5	Контекстно-нечувствительный межпроцедурный анализ	1105

12.5.1	Влияние вызовов методов	1105
12.5.2	Построение графа вызовов в Datalog	1107
12.5.3	Динамическая загрузка и отражение	1108
12.5.4	Упражнения к разделу 12.5	1109
12.6	Контекстно-чувствительный анализ указателей	1109
12.6.1	Контексты и строки вызовов	1110
12.6.2	Добавление контекста в правила Datalog	1113
12.6.3	Дополнительные наблюдения о чувствительности	1114
12.6.4	Упражнения к разделу 12.6	1115
12.7	Реализация Datalog с применением BDD	1115
12.7.1	Диаграммы бинарного выбора	1116
12.7.2	Преобразования диаграмм бинарного выбора	1118
12.7.3	Представление отношений при помощи BDD	1119
12.7.4	Операции отношений и BDD-операции	1120
12.7.5	Использование диаграмм бинарного выбора для анализа целей указателей	1123
12.7.6	Упражнения к разделу 12.7	1124
12.8	Резюме к главе 12	1124
12.9	Список литературы к главе 12	1128
Приложение А. Завершенный пример начальной стадии компилятора		1133
A.1	Исходный язык	1133
A.2	Main	1135
A.3	Лексический анализатор	1135
A.4	Таблицы символов и типы	1139
A.5	Промежуточный код выражений	1140
A.6	Переходы для булевых выражений	1144
A.7	Промежуточный код для инструкций	1149
A.8	Синтаксический анализатор	1153
A.9	Построение начальной стадии	1160
Приложение Б. Поиск линейно независимых решений		1163
Предметный указатель		1167