

Посвящаем эту книгу:

- *нашим родителям, которые привили нам жажду знаний и были нам опорой, когда мы приобретали навыки, необходимые для ее удовлетворения;*
- *нашим детям, которые еще раз показали нам, каким прекрасным может быть процесс обучения и взросления;*
- *нашим половинкам, без которых эта книга никогда не была бы написана.*

Содержание

От издательства	15
Об авторах	16
Предисловие	17
 Глава 1. Обзор процесса компиляции	 22
1.1. Введение	22
1.2. Структура компилятора	28
1.3. Краткий обзор трансляции	32
1.3.1. Фаза анализа	32
1.3.2. Оптимизатор	37
1.3.3. Фаза синтеза	39
1.4. Техническая разработка	44
1.5. Итоги и перспективы	45
Замечания к этой главе	46
Упражнения	47
 Глава 2. Сканеры	 49
2.1. Введение	49
2.2. Распознавание слов	52
2.2.1. Формализм для описания распознавателей	54
2.2.2. Распознавание более сложных слов	56
2.3. Регулярные выражения	58
2.3.1. Формализация нотации	60
2.3.2. Примеры регулярных выражений	61
2.3.3. Свойства замкнутости РВ	64
2.4. От регулярного выражения к сканеру	67
2.4.1. Недетерминированные конечные автоматы	67
2.4.2. РВ в НКА: построение Томпсона	70
2.4.3. НКА в ДКА: построение подмножеств	71
2.4.4. ДКА в минимальный ДКА	76
2.4.5. Использование ДКА в качестве сканера	81
2.5. Реализация сканеров	84
2.5.1. Таблично управляемые сканеры	85
2.5.2. Непосредственно кодируемые сканеры	88
2.5.3. Сканеры, написанные вручную	90
2.5.4. Практические вопросы реализации	91
2.6. Дополнительные темы	95
2.6.1. Преобразование ДКА в регулярное выражение	96

2.6.2. Регулярные выражения, свободные от замыканий	97
2.6.3. Альтернатива алгоритму минимизации ДКА	98
2.7. Итоги и перспективы	101
Замечания к этой главе	101
Упражнения	102

Глава 3. Парсеры..... 105

3.1. Введение	105
3.2. Выражение синтаксиса..... 107	107
3.2.1. Почему бы не использовать регулярные выражения?	108
3.2.2. Контекстно-свободные грамматики..... 109	109
3.2.3. Более сложные примеры	112
3.2.4. Кодирование семантики в структуре..... 115	115
3.2.5. Нахождение вывода для заданной строки..... 118	118
3.3. Нисходящий разбор	119
3.3.1. Преобразование грамматики..... 121	121
3.3.2. Нисходящие парсеры рекурсивного спуска..... 132	132
3.3.3. Таблично управляемые LL(1)-парсеры..... 133	133
3.4. Восходящий разбор..... 138	138
3.4.1. Алгоритм LR(1)-разбора	140
3.4.2. Построение таблиц LR(1)-разбора..... 147	147
3.4.3. Ошибки при построении таблиц..... 157	157
3.5. Практические вопросы..... 160	160
3.5.1. Восстановление после ошибки	161
3.5.2. Унарные операторы..... 162	162
3.5.3. Обработка контекстно-зависимой неоднозначности..... 163	163
3.6. Дополнительные темы	165
3.6.1. Оптимизация грамматики	165
3.6.2. Уменьшение размера таблиц LR(1)-разбора	167
3.7. Итоги и перспективы	172
Замечания к этой главе	172
Упражнения	173

Глава 4. Промежуточные представления 176

4.1. Введение	176
4.2. Таксономия	179
4.3. Графовые ПП..... 183	183
4.3.1. Деревья, связанные с синтаксисом	183
4.3.2. Графы..... 187	187
4.4. Линейные ПП..... 192	192
4.4.1. Код для стековой машины..... 194	194
4.4.2. Трехадресный код..... 195	195
4.4.3. Представление линейных кодов..... 196	196
4.4.4. Построение ГПУ из линейного кода..... 198	198
4.5. Таблицы символов	201
4.5.1. Разрешение имен..... 202	202

4.5.2. Реализация таблиц	205
4.6. Пространства имен	207
4.6.1. Пространства имен в ПП.....	209
4.6.2. Форма с единственным статическим присваиванием	212
4.7. Размещение значений в памяти	215
4.7.1. Модели памяти.....	215
4.7.2. Хранение значений в регистрах	218
4.7.3. Отнесение значений к областям данных	219
4.8. Итоги и перспективы.....	222
Замечания к этой главе	222
Упражнения	223

Глава 5. Синтаксически управляемая трансляция

5.1. Введение.....	227
5.2. Предварительные сведения.....	230
5.3. Синтаксически управляемая трансляция	232
5.3.1. Первый пример	232
5.3.2. Трансляция выражений	235
5.3.3. Трансляция управляющих конструкций.....	242
5.4. Моделирование окружения именованного	245
5.4.1. Лексические иерархии	246
5.4.2. Иерархии наследования	252
5.4.3. Видимость.....	256
5.4.4. Разрешение имен на этапе компиляции	257
5.5. Информация о типе	258
5.5.1. Применения типов при трансляции	259
5.5.2. Компоненты системы типов	261
5.5.3. Вывод типа выражения.....	266
5.6. Размещение в памяти	270
5.6.1. Классы памяти и области данных	270
5.6.2. Размещение в виртуальном адресном пространстве	273
5.6.3. Назначение памяти	275
5.6.4. Место назначения памяти в процессе трансляции	280
5.6.5. Ограничения выравнивания и дополнение	280
5.7. Дополнительные темы.....	282
5.7.1. Структура грамматики и ассоциативность	282
5.7.2. Более трудные проблемы вывода типов.....	285
5.7.3. Относительные смещения и производительность кеша	287
5.8. Итоги и перспективы.....	289
Замечания к этой главе	290
Упражнения	291

Глава 6. Реализация процедур.....

6.1. Введение.....	294
6.2. Предварительные сведения.....	298
6.3. Поддержка именованного на этапе выполнения	301

6.3.1. Поддержка времени выполнения в Algol-подобных языках	302
6.3.2. Поддержка объектно-ориентированных языков на этапе выполнения	308
6.4. Передача значений между процедурами	313
6.4.1. Передача параметров	313
6.4.2. Возврат значений	316
6.4.3. Обеспечение адресуемости нелокальных переменных	318
6.5. Стандартные соглашения о связях	324
6.6. Дополнительные темы	329
6.6.1. Явное управление кучей	330
6.6.2. Неявное освобождение	333
6.7. Итоги и перспективы	339
Замечания к этой главе	340
Упражнения	341

Глава 7. Форма кода

7.1. Введение	346
7.2. Арифметические операторы	349
7.2.1. Вызовы функций в выражении	351
7.2.2. Выражения со смешанными типами	352
7.2.3. Уменьшение спроса на регистры	354
7.3. Методы доступа к значениям	356
7.3.1. Методы доступа к скалярным переменным	356
7.3.2. Методы доступа к агрегатам	359
7.3.3. Проверка диапазона	366
7.4. Булевы и реляционные операторы	368
7.4.1. Аппаратная поддержка реляционных выражений	369
7.4.2. Варианты аппаратной поддержки	371
7.5. Управляющие конструкции	374
7.5.1. Условное выполнение	375
7.5.2. Циклы и итерация	377
7.5.3. Предложения выбора	381
7.6. Операции со строками	385
7.6.1. Длина строки	385
7.6.2. Присваивание строке	385
7.6.3. Конкатенация строк	387
7.6.4. Оптимизация операций со строками	387
7.7. Вызовы процедур	388
7.7.1. Вычисление фактических параметров	389
7.7.2. Сохранение и восстановление регистров	390
7.8. Итоги и перспективы	392
Замечания к этой главе	393
Упражнения	394

Глава 8. Введение в оптимизацию

8.1. Введение	398
---------------------	-----

8.2. Предварительные сведения.....	400
8.2.1. Примеры.....	402
8.2.2. Соображения по поводу оптимизации.....	406
8.2.3. Возможности для оптимизации	409
8.3. Область оптимизации	410
8.4. Локальная оптимизация	414
8.4.1. Локальная нумерация значений	414
8.4.2. Балансировка дерева по высоте	421
8.5. Региональная оптимизация	430
8.5.1. Суперлокальная нумерация значений	431
8.5.2. Разворачивание цикла.....	434
8.6. Глобальная оптимизация.....	437
8.6.1. Нахождение неинициализированных переменных с помощью анализа живучести	438
8.6.2. Глобальное размещение кода	443
8.7. Межпроцедурная оптимизация	450
8.7.1. Встраивание	450
8.7.2. Размещение процедуры в памяти.....	454
8.7.3. Прагматические аспекты межпроцедурной оптимизации	458
8.8. Итоги и перспективы.....	461
Замечания к этой главе	462
Упражнения	464
Глава 9. Анализ потока данных	468
9.1. Введение	468
9.2. Итеративный анализ потока данных.....	471
9.2.1. Доминирование	471
9.2.2. Анализ живучести переменных	476
9.2.3. Ограничения на анализ потока данных.....	480
9.2.4. Другие задачи о потоке данных	483
9.3. Форма с единственным статическим присваиванием.....	488
9.3.1. Наивный метод построения SSA-формы	490
9.3.2. Границы доминирования	492
9.3.3. Размещение ϕ -функций.....	495
9.3.4. Переименование	499
9.3.5. Трансляция программы в форме SSA	505
9.3.6. Использование формы SSA	513
9.4. Межпроцедурный анализ	517
9.4.1. Построение графа вызовов	518
9.4.2. Межпроцедурное распространение констант.....	521
9.5. Дополнительные темы	525
9.5.1. Структурный анализ потока данных и приводимость	525
9.5.2. Ускорение итеративного каркаса доминирования	529
9.6. Итоги и перспективы.....	531
Замечания к этой главе	532
Упражнения	533

Глава 10. Скалярная оптимизация	537
10.1. Введение	537
10.2. Устранение мертвого кода	542
10.2.1. Устранение бесполезного кода	543
10.2.2. Устранение бесполезного потока управления	545
10.2.3. Устранение недостижимого кода	549
10.3. Перемещение кода	550
10.3.1. Отложенное перемещение кода	550
10.3.2. Поднятие кода	558
10.4. Специализация	560
10.4.1. Оптимизация хвостового вызова	560
10.4.2. Оптимизация листового вызова	562
10.4.3. Перенос параметров	563
10.5. Устранение избыточности	564
10.5.1. Идентичность значений и идентичность имен	565
10.5.2. Нумерация значений на основе доминирования	566
10.6. Обеспечение других преобразований	569
10.6.1. Клонирование суперблока	570
10.6.2. Клонирование процедуры	572
10.6.3. Расчленение цикла	572
10.6.4. Переименование	573
10.7. Дополнительные темы	575
10.7.1. Комбинирование оптимизаций	575
10.7.2. Снижение стоимости	580
10.7.3. Выбор последовательности оптимизаций	590
10.8. Итоги и перспективы	592
Замечания к этой главе	592
Упражнения	593
Глава 11. Выбор команд	596
11.1. Введение	596
11.2. Предварительные сведения	600
11.2.1. Влияние дизайна ISA на выбор команд	601
11.2.2. Поясняющий пример	605
11.2.3. Ситуативное сопоставление	607
11.3. Выбор команд с помощью оконной оптимизации	608
11.3.1. Оконная оптимизация	608
11.3.2. Упроститель	610
11.3.3. Сопоставитель	614
11.4. Выбор с помощью сопоставления с деревом-образцом	616
11.4.1. Представление деревьев	617
11.4.2. Правила переписывания	618
11.4.3. Вычисление покрытий	622
11.4.4. Инструменты	630
11.5. Дополнительные темы	631
11.5.1. Обучение оконным образцам	631

11.5.2. Генерирование последовательностей команд	633
11.6. Итоги и перспективы.....	634
Замечания к этой главе	635
Упражнения	636

Глава 12. Планирование команд..... 638

12.1. Введение	638
12.2. Предварительные сведения.....	641
12.2.1. Архитектурные средства, влияющие на производительность	642
12.2.2. Задача планирования команд.....	645
12.3. Локальное планирование	649
12.3.1. Алгоритм	650
12.3.2. Переименование	651
12.3.3. Построение графа зависимостей	653
12.3.4. Вычисление приоритетов.....	655
12.3.5. Планирование списка	656
12.3.6. Прямое и обратное планирование списка.....	660
12.4. Региональное планирование.....	663
12.4.1. Суперлокальное планирование	663
12.4.2. Планирование трасс	665
12.4.3. Клонирование с учетом контекста.....	668
12.5. Дополнительные темы	669
12.5.1. Стратегия организации программного конвейера	670
12.5.2. Алгоритм программного конвейера.....	673
12.5.3. Последний пример	677
12.6. Итоги и перспективы.....	679
Замечания к этой главе	679
Упражнения	681

Глава 13. Распределение регистров

13.1. Введение	684
13.2. Предварительные сведения.....	687
13.2.1. Пространство имен для распределения: интервалы жизни переменных	687
13.2.2. Интерференция.....	690
13.2.3. Код выгрузки	692
13.2.4. Классы регистров	693
13.3. Локальное распределение регистров.....	696
13.3.1. Переименование в локальном распределителе	698
13.3.2. Распределение и назначение	700
13.4. Глобальное распределение с помощью раскрашивания.....	704
13.4.1. Нахождение глобальных интервалов жизни	706
13.4.2. Построение графа интерференции.....	708
13.4.3. Консолидация операций копирования	710
13.4.4. Оценка глобальной стоимости выгрузки.....	712
13.4.5. Раскрашивание графа	714

13.4.6. Вставка кода выгрузки и восстановления	717
13.4.7. Обработка совмещенных классов регистров	718
13.5. Дополнительные темы	723
13.5.1. Вариации на тему консолидации	723
13.5.2. Вариации на тему выгрузки	725
13.5.3. Другие формы интервалов жизни	727
13.6. Итоги и перспективы	732
Замечания к этой главе	732
Упражнения	733

Глава 14. Оптимизация на этапе выполнения

14.1. Введение	737
14.2. Предварительные сведения	741
14.2.1. Модель выполнения	743
14.2.2. Триггеры компиляции	745
14.2.3. Гранулярность оптимизации	747
14.2.4. Источники улучшения	749
14.2.5. Построение оптимизатора этапа выполнения	752
14.3. Оптимизация горячих трасс	753
14.3.1. Поток выполнения	754
14.3.2. Связывание трасс	759
14.4. Оптимизация горячих методов	762
14.4.1. Горячие методы в смешанной среде	762
14.4.2. Горячие методы в платформенной среде	769
14.5. Дополнительные темы	772
14.5.1. Уровни оптимизации	773
14.5.2. Замещение в стеке	773
14.5.3. Управление кешем методов	775
14.5.4. Управление изменениями исходного кода	777
14.6. Итоги и перспективы	779
Замечания к этой главе	779
Упражнения	780

Приложение А. ILOC

А.1. Введение	783
А.2. Соглашения об именовании	785
А.3. Вычислительные операции	785
А.4. Операции перемещения данных	786
А.5. Операции потока управления	788
А.6. Таблицы кодов операций	791

Приложение В. Структуры данных

В.1. Введение	794
В.2. Представление множеств	795
В.2.1. Представление множеств упорядоченными списками	796
В.2.2. Представление множеств битовыми векторами	798

В.2.3. Представление разреженных множеств	799
В.2.4. Роль хеш-таблиц	800
В.3. Реализация ПП	801
В.3.1. Непосредственные графовые представления.....	801
В.3.2. Линейные промежуточные формы.....	806
В.4. Реализация хеш-таблиц	809
В.4.1. Выбор хеш-функции	809
В.4.2. Метод цепочек	810
В.4.3. Метод открытой адресации	812
В.4.4. Хранение записей символов.....	813
В.5. Гибкий дизайн таблицы символов	814
Замечания	816
Литература.....	818
Предметный указатель.....	845

От издательства

Отзывы и пожелания

Мы всегда рады отзывам наших читателей. Расскажите нам, что вы думаете об этой книге – что понравилось или, может быть, не понравилось. Отзывы важны для нас, чтобы выпускать книги, которые будут для вас максимально полезны.

Вы можете написать отзыв на нашем сайте www.dmkpress.com, зайдя на страницу книги и оставив комментарий в разделе «Отзывы и рецензии». Также можно послать письмо главному редактору по адресу dmkpress@gmail.com; при этом укажите название книги в теме письма.

Если вы являетесь экспертом в какой-либо области и заинтересованы в написании новой книги, заполните форму на нашем сайте по адресу http://dmkpress.com/authors/publish_book/ или напишите в издательство по адресу dmkpress@gmail.com.

Список опечаток

Хотя мы приняли все возможные меры для того, чтобы обеспечить высокое качество наших текстов, ошибки все равно случаются. Если вы найдете ошибку в одной из наших книг, мы будем очень благодарны, если вы сообщите о ней главному редактору по адресу dmkpress@gmail.com. Сделав это, вы избавите других читателей от недопонимания и поможете нам улучшить последующие издания этой книги.

Нарушение авторских прав

Пиратство в интернете по-прежнему остается насущной проблемой. Издательство «ДМК Пресс» очень серьезно относится к вопросам защиты авторских прав и лицензирования. Если вы столкнетесь в интернете с незаконной публикацией какой-либо из наших книг, пожалуйста, пришлите нам ссылку на интернет-ресурс, чтобы мы могли применить санкции.

Ссылку на подозрительные материалы можно прислать по адресу электронной почты dmkpress@gmail.com.

Мы высоко ценим любую помощь по защите наших авторов, благодаря которой мы можем предоставлять вам качественные материалы.

Об авторах

Кит Д. Купер – профессор вычислительной техники в университете Райса. Занимался широким кругом проблем в области оптимизации откомпилированного кода, включая анализ межпроцедурного потока данных и его приложения, нумерацию значений, алгебраическую реассоциацию выражений, распределение регистров и планирование выполнения команд. Он читал самые разные курсы, в т. ч. введение в программирование, алгоритмическое мышление, организацию компьютеров, коллективное программирование, несколько вариантов курса по конструированию компиляторов и курс по оптимизации кода для аспирантов. Исполнял должности заведующего кафедрой информатики в университете Райса, заведующего кафедрой вычислительной и прикладной математики там же и заместителя декана по исследованиям в инженерной школе при университете Райса. Является членом Ассоциации вычислительной техники (ACM).

Линда Торкзон в течение долгого времени работала старшим научным сотрудником кафедры информатики в университете Райса. Занималась широким кругом задач в области компиляции, включая анализ межпроцедурного потока данных, подстановку встраиваемых функций, распределение регистров, адаптивную оптимизацию и проектирование сред программирования. Несколько раз читала курс по компиляторам для студентов младших курсов в университете Райса. Д-р Торкзон занимала должность исполнительного директора Центра исследований по параллельным вычислениям при Национальном центре фундаментальных наук и технологий. Она также работала исполнительным директором компании HiPerSoft, Лос-Аламосского института информатики и проекта по разработке приложений виртуальных сетей. Была главным исследователем в проекте разработки адаптируемых сред компиляции, спонсируемым агентством DARPA, целью которого являлось проектирование оптимизирующего компилятора, который мог бы автоматически подстраивать свои алгоритмы и стратегии оптимизации под новые платформы. Также занимала должность директора программы присвоения степени мастера-профессионала по информатике в университете Райса.

Предисловие

ИЗМЕНЕНИЯ В ТРЕТЬЕМ ИЗДАНИИ

В третье издание книги «Разработка компилятора» (ЕаСЗе) изменения вносились по двум главным причинам: новые способы использования технологии трансляции языков программирования и изменившаяся техническая подготовка наших студентов. Эти две движущие силы побуждали нас постоянно пересматривать способы преподавания методов конструирования компиляторов. В третье издание включены эти изменения, протестированные на аудитории.

В издании ЕаСЗе материал организован по-новому: некоторые темы, присутствовавшие в предыдущих изданиях, исключены, а вместо них добавлены новые – по крайней мере в книге. В издание ЕаС2е уже были включены важные изменения материалов по оптимизации (главы 8, 9 и 10), и эти главы перешли в третье издание почти без изменений. Для ЕаСЗе изменения были внесены в разные места текста, но основной упор был сделан на реорганизацию и переработку средней части книги: главы с 4 по 7. Широкое использование JIT-компиляторов побудило нас добавить главу, посвященную введению в эти методы. Конкретно:

- материал о промежуточных представлениях перемещен в главу 4, перед синтаксически управляемой трансляцией. Студенты должны ознакомиться с этим материалом до изложения синтаксически управляемой трансляции;
- материал об атрибутивных грамматиках, который в двух предыдущих изданиях находился в главе 4, исключен. Атрибутивные грамматики – по-прежнему интересная тема, но на данный момент преобладающей парадигмой в фазе анализа компилятора является ситуативная трансляция;
- в главе 5 теперь более глубоко излагается механизм синтаксически управляемой трансляции и его применение. В эту главу перенесено несколько относящихся к трансляции тем, которые раньше находились в главах 4, 6 и 7;
- глава 7 получила новую организацию и структуру, основанную на интенсивных экспериментах в учебной аудитории;
- в главе 13 основное внимание уделяется двум распределителям: локальному, основанному на алгоритме Бэста, и глобальному, основанному на работе Хайтина. В разделе «Дополнительные темы» главы 13 изучаются модификации базовой схемы Хайтина–Бриггса, порождающие такие методы, как распределители на основе линейного сканирования, распределители на основе SSA и итеративные объединяющие распределители;

В предыдущих изданиях алгоритм Бэста назывался «локальным восходящим» алгоритмом.

- в главе 14 теперь представлен обзор оптимизации на этапе выполнения, или JIT-компиляции. JIT-компиляторы теперь можно встретить всюду. Студенты должны понимать, как работают JIT-компиляторы и как они связаны с классической компиляцией перед выполнением.

Нашей целью по-прежнему является учебник и курс, на котором студенты знакомятся с важнейшими проблемами современных компиляторов и получают базовые знания, позволяющие справляться с этими проблемами.

ОРГАНИЗАЦИЯ

В третьем издании материал разбит на четыре примерно одинаковых по размеру части.

- Первая часть, главы 2 и 3, посвящена проектированию фазы анализа компилятора (front end) и алгоритмам, на которых основаны инструментальные средства, автоматизирующие конструирование фазы анализа. В процессе описания алгоритмов, которые выводят сканеры из регулярных выражений, а парсеры – из контекстно-свободных грамматик, вводятся некоторые ключевые идеи, в т. ч. понятие алгоритма с неподвижной точкой.
- Во второй части (главы 4–7) изучается преобразование исходного кода в промежуточную форму. Здесь рассматриваются виды кода, которые фаза анализа может генерировать для оптимизатора и фазы синтеза (back end).
- В третьей части, в которую входят главы 8, 9, 10 и 14, приводится обзор методов оптимизации кода. В главе 8 представлен общий взгляд на виды оптимизации, выполняемые компилятором. В главах 9 и 10 более глубоко изучаются анализ потока данных и скалярная оптимизация. Глава 14 тематически примыкает к этой части, но в ней предполагается знание материала, составляющего четвертую часть.
- Четвертая часть, главы 11–13, посвящена основным алгоритмам, встречающимся в фазе синтеза: выбор команд, планирование выполнения команд и распределение регистров. В третьем издании мы переработали материал по распределению регистров, так что теперь количество идей уменьшилось, но те, что остались, рассматриваются глубже. Благодаря новой главе студенты получают основательные знания, позволяющие понять большинство современных алгоритмов распределения.

Обычно из программы для студентов младших курсов мы опускаем главы 9 и 10. Наш опыт показывает, что глава 14 вызывает у студентов больший интерес.

Программа для студентов младших курсов представляет собой линейную выжимку из этого материала. За недостатком времени мы часто опускаем главы 9 и 10. Материал главы 14 был подготовлен в ответ на вопросы слушателей курса.

Подход

Конструирование компилятора – упражнение из области инженерного проектирования. Автор компилятора должен выбрать путь в пространстве проектирования, изобилующем различными альтернативами, каждая из которых характеризуется стоимостью, набором преимуществ и сложностью. Любое решение оказывает влияние на результирующий компилятор. Качество конечного продукта зависит от решений, принятых на каждом шаге.

Поэтому часто не существует единственно верного ответа на вопрос о том, какое из множества решений выбрать при проектировании компилятора. Даже для «хорошо проработанных» и «решенных» проблем есть нюансы дизайна и реализации, которые влияют как на поведение компилятора, так и на качество порождаемого им кода. При принятии решения учитываются различные соображения. Например, выбор промежуточного представления оказывает значительное влияние на остальные части компилятора – от требований к памяти и времени работы до легкости применения различных алгоритмов. Однако на принятие решения часто не хватает времени. В четвертой части рассматривается вопрос о промежуточных представлениях и некоторые проблемы, которые следует учитывать при выборе представления. Этот вопрос поднимается еще в нескольких местах книги – как прямо в тексте, так и опосредованно в упражнениях.

В третьем издании исследуется все пространство проектирования компилятора и уделяется внимание как глубине проблем, так и широте охвата возможных решений. Описываются некоторые способы разрешения проблем компиляции, а также ограничения, из-за которых эти решения стали привлекательными. Авторы компиляторов должны понимать параметры проблем и их решения. Они должны сознавать, что решение одной проблемы может повлиять на спектр возможностей и ограничений в других частях компилятора. И только в этом случае они смогут сделать обоснованный и разумный выбор.

Философия

Этот учебник является воплощением нашей философии построения компиляторов, которая сложилась в результате сорока с лишним лет исследований, преподавания и практики. Например, промежуточные представления должны раскрывать те детали, которые имеют значение в конечном коде; это убеждение побуждает нас отдавать предпочтение низкоуровневым представлениям. Значения должны храниться в регистрах до тех пор, пока распределитель не обнаружит, что регистров не хватает; такая практика приводит к компиляторам, оперирующим в терминах виртуальных регистров, которые хранят значения в памяти, только если этого невозможно избежать. Данный подход также повышает важность эффективных алгоритмов на фазе синтеза. Каждый компилятор должен включать этап оптимизации, это упро-

щает остальные части компилятора. При отборе материала и способа его изложения мы руководствовались своим многолетним опытом.

Несколько слов об упражнениях

Курс по конструированию компиляторов дает возможность исследовать сложные проблемы в контексте конкретного приложения, основные функции которого понятны любому студенту, обладающему достаточной подготовкой для посещения курса по компиляторам. В большинстве вариантов такого курса упражнения на программирование играют важную роль.

Мы читали этот курс в вариантах, когда студенты создают простой компилятор от начала до конца – начиная со сгенерированных сканера и парсера и заканчивая кодогенератором для упрощенного набора RISC-команд. Мы также читали этот курс в вариантах, когда студенты пишут программы для решения четко поставленных узких задач, таких как распределение регистров или планирование команд. Выбор упражнений сильно зависит от роли, которую данный курс играет в общем учебном плане.

В некоторых учебных заведениях курс по компиляторам является кульминацией для студентов старших курсов, в нем идеи, с которыми слушатели познакомились на других курсах, соединяются в крупном практическом проекте, подразумевающем как проектирование, так и реализацию. Слушатели такого курса могут написать полный компилятор какого-нибудь простого языка или модифицировать компилятор с открытым исходным кодом, добавив в него какое-то языковое средство или новую архитектурную возможность. В таком варианте курса материал может излагаться линейно, близко к организации текста в учебнике.

В других вузах кульминация может достигаться в иных курсах или иными способами. В таком случае преподаватель может уделить больше внимания упражнениям, относящимся к конкретным алгоритмам или их реализациям; например, на семинарах можно разработать распределитель регистров или проход перебалансировки дерева с выравниванием высоты. В этом варианте курса можно опускать некоторые представленные в учебнике темы и изменять порядок изложения, подстраивая его под нужды семинаров. Мы обнаружили, что студенты, посещающие этот курс, умеют программировать на языке ассемблера, поэтому не затрудняются с пониманием того, как должен работать планировщик команд или распределитель регистров.

При любом сценарии курс должен иметь связи с другими курсами в учебном плане для младших курсов. Очевидно, он связан с курсами по организации компьютера, программированию на языке ассемблера, операционным системам, компьютерной архитектуре, алгоритмам и формальным языкам. Но немало и менее очевидных связей. Материал главы 7 о копировании литер поднимает вопросы производительности, которые критически важны для реализации стека протоколов. Материал главы 2 имеет разнообразные применения: от фильтрации URL-адресов до задания правил для брандмау-

Конец ознакомительного фрагмента.

Приобрести книгу можно

в интернет-магазине

«Электронный универс»

e-Univers.ru