Ахо А., Ульман Дж.

Теория синтаксического анализа, перевода и компиляции. Синтаксический анализ

Том 1

Москва
«Книга по Требованию»

УДК 53
ББК 22.3
А95

Ахо А.

А95 Теория синтаксического анализа, перевода и компиляции. Синтаксический анализ: Том 1 / Ахо А., Ульман Дж. – М.: Книга по Требованию, 2012. – 613 с.

ISBN 978-5-458-27408-1

Особенность книги состоит в том, что она трактует теоретические вопросы в связи с потребностями реализации языков программирования, и этим она отличается от книг по системному программированию.

ISBN 978-5-458-27408-1

Эта книга является репринтом оригинала, который мы создали специально для Вас, используя запатентованные технологии производства репринтных книг и печати по требованию.

Сначала мы отсканировали каждую страницу оригинала этой редкой книги на профессиональном оборудовании. Затем с помощью специально разработанных программ мы произвели очистку изображения от пятен, клякс, перегибов и попытались отбелить и выровнять каждую страницу книги. К сожалению, некоторые страницы нельзя вернуть в изначальное состояние, и если их было трудно читать в оригинале, то даже при цифровой реставрации их невозможно улучшить.

Разумеется, автоматизированная программная обработка репринтных книг – не самое лучшее решение для восстановления текста в его первозданном виде, однако, наша цель – вернуть читателю точную копию книги, которой может быть несколько веков.

Поэтому мы предупреждаем о возможных погрешностях восстановленного репринтного издания. В издании могут отсутствовать одна или несколько страниц текста, могут встретиться невыводимые пятна и кляксы, надписи на полях или подчеркивания в тексте, нечитаемые фрагменты текста или загибы страниц. Покупать или не покупать подобные издания – решать Вам, мы же делаем все возможное, чтобы редкие и ценные книги, еще недавно утраченные и несправедливо забытые, вновь стали доступными для всех читателей.

Серия Книжный Ренессанс
www.samizday.ru/reprint

ОТ РЕДАКТОРА ПЕРЕВОДА

Перед читателями — перевод фундаментального труда американских ученых А. Ахо и Дж. Ульмана, трактующего широкий круг проблем, связанных с языками программирования и методами их реализации на вычислительных машинах. Основой для него до известной степени послужила большая серия статей по теории языков и перевода, опубликованная авторами в различных научных журналах в течение 4—5 предшествующих лет.

Первый том посвящен теории языков, общим вопросам теории перевода и (главным образом) методам синтаксического анализа контекстно-свободных языков. Во втором томе рассматриваются вопросы, более тесно связанные с реализацией языков, в частности вопросы оптимизации анализаторов и объектного кода. На русском языке уже имеется ряд изданий, касающихся этих тем или даже целиком посвященных им, но книга А. Ахо и Дж. Ульмана сравнительно мало пересекается с ними. Причин этому две. Во-первых, здесь собран и систематизирован очень большой по объему материал — по широте охвата вопросов синтаксического анализа контекстно-свободных языков книга, по-видимому, не имеет себе равных ни в нашей, ни в зарубежной литературе. Во-вторых, авторы сосредоточили основное внимание лишь на тех вопросах, которые уже нашли или могут найти приложения на практике при создании компиляторов и других связанных с языками частей математического обеспечения.

Приятен стиль авторов, значительно облегчающий чтение книги. Введение новых более или менее сложных понятий и концепций начинается, как правило, с неформального их изложения и примеров. При этом делаются намеки, которые позволяют лучше понять следующее затем строгое, аккуратное и формальное построение. Описание многочисленных алгоритмов построено по аналогичной схеме: сначала дается неформальное разъяснение сути их работы, затем четкая формулировка алгоритма, иллюстрируемая часто на примере, и, наконец, доказательство правильности алгоритма. Каждый раздел заканчивается солидным списком задач различной трудности — от чисто технических и учебных упражнений до нерешенных научных проблем. Благодаря такому характеру изложения эту книгу, содержащую большой и трудный материал, легко читать на любом уровне: поверхностного чтения (только по неформальным и общим описаниям), знакомства с фактическим материалом (определения, теоремы, алгоритмы), глубокого изучения (разбор или воспроизведение доказательств, решение трудных задач). Для свободного владения излагаемым материалом полезен разбор примеров и решение большинства приводимых задач.

В целом книга будет интересной для широкого круга читателей: от студентов, изучающих основы теории языков программирования, до специалистов — разработчиков систем математического обеспечения и научных работников. Вместе с авторами можно надеяться, „что алгоритмы и понятия, изложенные в этой книге, переживут следующее поколение вычислительных машин и языков программирования и что хотя бы некоторые из них найдут применение не только при построении компиляторов, но и в других областях“.

В. М. Курочкин

Посвящается Адриенне и Холли

ПРЕДИСЛОВИЕ

Эта книга задумана как пособие для одно- или двухсеместрового курса лекций по теории компиляции для студентов старших курсов. В ней дается теоретическая трактовка предмета, имеющего практическое значение. При ее написании мы исходили из следующих трех соображений.

(1) В преподавании такой быстро развивающейся области, какой является наука о вычислительных процессах, правильный педагогический принцип состоит в том, чтобы больше внимания уделять идеям, а не техническим подробностям реализации. Мы надеемся, что алгоритмы и понятия, изложенные в этой книге, переживут следующее поколение вычислительных машин и языков программирования и что хотя бы некоторые из них найдут применение не только при построении компиляторов, но и в других областях.

(2) Прогресс в построении компиляторов достиг того этапа, когда компилятор можно расчленить на много составных частей и каждую часть подвергнуть запланированной оптимизации. Поэтому важно снабдить людей, предпринимающих попытки такой оптимизации, подходящими математическими средствами.

(3) Для полного понимания некоторых из самых полезных и самых эффективных алгоритмов компиляции (например, алгоритма LR (k)-анализа) требуется основательная математическая подготовка. Мы полагаем, что хорошая теоретическая подготовка становится все более существенной для тех, кто строит компиляторы.

Включая в книгу трудные теоремы, имеющие отношение к компиляции, мы старались сделать изложение как можно более доступным. Для этого приводится много примеров, причем в каждом из них используется какая-нибудь малая грамматика, а не те большие грамматики, что встречаются на практике. Мы надеемся, что в тех случаях, когда трудно следить за теоретическими построениями, для иллюстрации основных идей этих примеров будет достаточно.

О пользовании книгой

Книга возникла из записей лекций, прочитанных на старших курсах Принстонского университета и Стивенсовского технологического института. По ним читались как односеместровый курс, так и двухсеместровый. В первом случае курсу по теории компиляции предшествовал курс теории конечных автоматов и контекстно-свободных языков, поэтому не было необходимости излагать материал гл. 0, 2 и 8. Остальные же главы излагались подробно.

В случае двухсеместрового курса бóльшая часть материала первого тома излагалась в первом семестре, а бóльшая часть второго, исключая гл. 8,— во втором. При этом доказательствам и технике доказательств уделялось больше внимания, чем в односеместровом курсе.

Ясно, что одни разделы книги более важны, чем другие. Поэтому нам хочется кратко пояснить читателю, как мы оцениваем относительную важность различных частей первого тома. Общее замечание состоит в том, что большинство доказательств, по-видимому, можно пропустить. Мы включили доказательства всех главных результатов потому, что считаем их необходимыми для глубокого понимания предмета. Однако в курсах, посвященных компиляции, обычно предпочитают не особенно углубляться во многие вопросы, причем разумный уровень понимания достигается при довольно поверхностном знакомстве с доказательствами.

В гл. 0 (математические основы) и 1 (обзор компиляции) почти весь материал существен, за исключением, быть может, разд. 1.3, в котором рассматриваются приложения синтаксического анализа, не связанные с компиляцией.

Мы считаем, что каждое понятие и теорема, введенные в гл. 2 (теория языков), найдут применение где-нибудь в остальных девяти главах. Однако в курсе лекций по компиляторам некоторый материал следует опустить. Подходящим кандидатом для этого служит довольно трудный материал об уравнениях с регулярными коэффициентами из разд. 2.2.1. Придется опустить тогда часть материала из разд. 2.2.2, касающегося праволинейных грамматик (а результат об эквивалентности между ними и конечными автоматами вывести другим способом), и материал из разд. 2.4.5 о преобразовании грамматики в грамматнку в нормальной форме Грейбах методом Розенкранца.

Понятия, излагаемые в гл. 3 (перевод), очень важны для остальной части книги. Однако разд. 3.2.3 об иерархии синтаксически управляемых переводов довольно труден и его можно опустить.

Мы думаем, что разд. 4.1 о методах разбора с возвратами менее важен, чем разд. 4.2, в котором рассматриваются табличные методы.

Глава 5 (однопроходный синтаксический анализ) большей частью очень важна. Максимальное предпочтение мы предлагаем отдать LL-грамматикам (разд. 5.1), LR-грамматикам (разд. 5.2), грамматикам предшествования (разд. 5.3.2 и 5.3.4) и грамматикам операторного предшествования (разд. 5.4.3). Другие разделы при необходимости можно опустить.

Глава 6 (алгоритмы с возвратами) менее важна, чем бо́льшая часть гл. 5 или разд. 4.2. Если надо выбирать, то мы предпочли бы изложить разд. 6.1, а не 6.2.

Организация книги

Вся книга состоит из двух томов:

I. Синтаксический анализ (гл. 0—6) и

II. Компиляция (гл. 7—11). (Во втором томе рассматриваются оптимизация анализаторов, теория детерминированного разбора, перевод, работа с таблицами и оптимизация кода.)

В конце каждого раздела (с номером $i.\ j$) приводятся упражнения, проблемы и замечания по литературе. Проблемы делятся на открытые и предлагаемые для дальнейшего исследования, а в упражнениях звездочками указывается степень трудности. Для решения упражнения, помеченного одной звездочкой, требуется одна существенная догадка, а для упражнения с двумя звездочками — более чем одна.

Чтение курса по этой книге рекомендуется сопровождать лабораторными работами по программированию, в ходе которых должны быть спроектированы и реализованы какие-то части компилятора. В конце некоторых разделов книги приведены упражнения на программирование, которые можно использовать в этих лабораторных работах.

Благодарности

Многие люди внимательно прочли различные части рукописи и серьезно помогли нам при ее подготовке к печати. Мы особенно хотим поблагодарить Джона Бруно, Стефена Чена, Джеймса Гимпеля, Жана Ихбиа, Брайана Кернигана, Дугласа Мак-Илроя, Роберта Мартина и Роберта Морриса, а также рецензентов Томаса Читэма, Майкла Фишера и Уильяма Мак-Кимана. Важные замечания сделали многие студенты, пользовавшиеся нашими записями лекций, среди них Алан Демерс, Нахед Эль Джабри, Мэтью Хехт, Петер Хендерсон, Петер Майка, Томас Петерсон, Рави Сети, Кеннет Силлз и Стивен Сквайрз.

Мы благодарны также Ханне Крессе и Дороти Лючиани за то, что они великолепно напечатали рукопись. Кроме того, мы выражаем признательность лабораториям компании „Белл телефон“ за содействие при подготовке рукописи. Она была ускорена с помощью UNIX, операционной системы для вычислительной машины PDP-11, разработанной Деннисом Ричи и Кеннетом Томпсоном.

Альфред В. Ахо
Джефри Д. Ульман

0 ПРЕДВАРИТЕЛЬНЫЕ МАТЕМАТИЧЕСКИЕ СВЕДЕНИЯ

Чтобы говорить ясно и точно, нам нужен точный и правильный язык. В этой главе описывается язык, которым мы будем пользоваться, обсуждая вопросы синтаксического анализа, трансляции и другие предметы, содержащиеся в нашей книге. Этот язык является главным образом языком элементарной теории множеств, к которому добавлены некоторые первоначальные понятия теории графов и математической логики. Читатели, знакомые с основами этих областей математики, могут только бегло просмотреть главу и использовать ее как справочник обозначений и определений.

0.1. ОСНОВНЫЕ ПОНЯТИЯ ТЕОРИИ МНОЖЕСТВ

В этом разделе будет сделан краткий обзор некоторых из самых основных понятий теории множеств, таких, как отношения, функции, упорядочения, а также обычные операции над множествами.

0.1.1. Множества

В дальнейшем мы будем предполагать, что существуют объекты, называемые *атомами*. Этим словом обозначается первоначальное понятие,— иначе говоря, термин „атом" остается не определенным. Что называть атомом, зависит от рассматриваемой области. Часто бывает удобно считать атомами целые числа или буквы некоторого алфавита.

Мы будем также постулировать абстрактное понятие *принадлежности*. Если a принадлежит A, то пишут $a \in A$. Отрицание этого утверждения записывается так: $a \notin A$. Предполагается, что если a — атом, то ему ничто не принадлежит, т. е. $x \notin a$ для всех x из рассматриваемой области.

Будут также использоваться некоторые примитивные объекты, называемые *множествами*, которые не являются атомами. Если

A—множество, то его *элементы*—это те объекты a (не обязательно атомы), для которых $a \in A$. Каждый элемент множества представляет собой либо атом, либо другое множество. Предполагается, что каждый элемент множества появляется в нем точно один раз. Если A содержит конечное число элементов, то A называется *конечным множеством*, и часто пишут $A = \{a_1, a_2, \ldots, a_n\}$, если $a_1, \ldots, a_n$—все элементы множества A и $a_i \neq a_j$ для $i \neq j$. Заметим, что порядок элементов не играет роли. Можно было бы, например, написать $A = \{a_n, \ldots, a_1\}$. Мы резервируем символ $\varnothing$ для обозначения *пустого множества*, т. е. множества, в котором нет элементов. Заметим, что атом тоже не имеет элементов, но пустое множество не атом и атом не является пустым множеством.

Утверждение $\# A = n$ означает, что множество A имеет n элементов.

Пример 0.1. Пусть атомами будут неотрицательные целые числа. Тогда $A = \{1, \{2, 3\}, 4\}$—множество. Элементами A служат 1, $\{2, 3\}$ и 4. Элемент $\{2, 3\}$ множества A сам является множеством, состоящим из атомов 2 и 3. Однако атомы 2 и 3 не принадлежат множеству A. Можно писать $A = \{4, 1, \{3, 2\}\}$. Заметим, что $\# A = 3$. □

Один из полезных способов определения множества—определение с помощью *предиката*, т. е. утверждения, содержащего одно или несколько неизвестных и принимающего в зависимости от значений неизвестных одно из двух значений—*истина* или *ложь*. Множество, определяемое с помощью предиката, состоит в точности из тех элементов, для которых предикат истинен. Однако надо быть осторожным при выборе предиката для определения множества, иначе может оказаться, что мы пытаемся определить множество, которое, возможно, и не существует.

Пример 0.2. Только что отмеченное явление называется *парадоксом Рассела*. Пусть $P(X)$—предикат „X не является элементом самого себя", т. е. $X \notin X$. Тогда мы могли бы подумать, что можно определить множество Y всех тех X, для которых $P(X)$ истинно, т. е. Y состоит в точности из тех множеств, которые не являются элементами самих себя. Так как большинство обычных множеств не являются элементами самих себя, возникает искушение допустить, что множество Y существует.

Но если Y существует, мы должны суметь ответить на вопрос: „Является ли Y элементом самого себя?" А это приводит к невозможной ситуации. Если $Y \in Y$, то $P(Y)$ ложно, и Y не является элементом самого себя по определению Y. Отсюда невозможно, чтобы $Y \in Y$. Допустим наоборот, что $Y \notin Y$. Тогда по определению Y снова $Y \in Y$. Мы видим, что $Y \notin Y$ влечет $Y \in Y$, а $Y \in Y$

влечет $Y \notin Y$. Так как либо $Y \in Y$, либо $Y \notin Y$ истинно, то оба эти утверждения истинны — ситуация, которую мы считаем невозможной. Единственный выход из положения состоит в том, чтобы предположить, что Y не существует. $\square$

Обычный способ избежать парадокса Рассела заключается в том, чтобы определять множества только с помощью предикатов $P(X)$ вида „X принадлежит A и $P_1(X)$", где A — известное множество, а P_1 — произвольный предикат. Если множество A подразумевается, то мы будем вместо „X принадлежит A и $P_1(X)$" писать просто $P_1(X)$.

Если $P(X)$ — предикат, будем обозначать множество объектов X, для которых $P(X)$ истинно, через $\{X \mid P(X)\}$.

Пример 0.3. Пусть $P(X)$ — предикат „X — неотрицательное четное число", т. е. $P(X)$ имеет вид „X принадлежит множеству неотрицательных целых чисел и $P_1(X)$", где $P_1(X)$ — предикат „X четно". Тогда $A = \{X \mid P(X)\}$ будет множеством, которое часто записывают так: $\{0, 2, 4, \ldots, 2n, \ldots\}$. Если по ходу дела ясно, что речь идет о множестве неотрицательных целых чисел, то можно писать $A = \{X \mid X \text{ четно}\}$. $\square$

Мы не останавливаемся здесь подробно на аксиоматической теории множеств. Интересующемуся читателю рекомендуем книги Халмоша [1960] и Суппеса [1960] (см. список литературы).

Определение. Говорят, что множество A *содержится* в множестве B, и пишут $A \subseteq B$, если каждый элемент из A является элементом из B. Иногда в этом случае говорят, что B *содержит* (или *включает*) A, и пишут $B \supseteq A$. Говорят также, что A — *подмножество* B, а B — *надмножество* A.

Если B содержит[1]) элемент, не принадлежащий A, и $A \subseteq B$, то говорят, что A *собственно содержится* в B, и пишут $A \subsetneqq B$ (или что B *собственно включает* A, и пишут $B \supsetneqq A$). Можно также сказать, что A — собственное подмножество B или что B — собственное надмножество A.

Два множества A и B называются *равными*, если $A \subseteq B$ и $B \subseteq A$.

Для того чтобы графически изобразить включение множеств, часто пользуются так называемыми *диаграммами Венна*. На рис. 0.1 показана диаграмма Венна для отношения $A \subseteq B$.

---

[1]) Русский термин „содержит" (и его производные) обозначает в силу традиции два разных понятия: множество *B содержит множество A*, т. е. $B \supseteq A$, или $A \subseteq B$, и множество *B содержит элемент b*, т. е. $b \in B$. Из контекста каждый раз ясно, о чем идет речь, и можно надеяться, что у читателя трудностей по этой причине не возникнет. — *Прим. ред.*

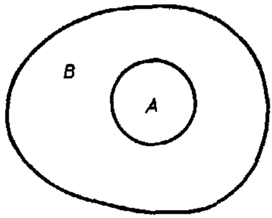

Рис. 0.1. Диаграмма Венна для включения множеств: $A \subseteq B$.

0.1.2. Операции над множествами

Существует несколько основных операций над множествами, с помощью которых можно строить новые множества.

Определение. Пусть A и B — множества. *Объединением* множеств A и B (записывается $A \cup B$) называется множество, содержащее все элементы из A вместе со всеми элементами из B. Формально $A \cup B = \{x \mid x \in A$ или $x \in B\}$ [1].

Пересечением множеств A и B (записывается $A \cap B$) называется множество, состоящее из всех тех элементов, которые принадлежат обоим множествам A и B. Формально $A \cap B = \{x \mid x \in A$ и $x \in B\}$.

Разностью множеств A и B (записывается $A - B$) называется множество тех элементов из A, которые не принадлежат B.

Если A — множество всех элементов, рассматриваемых в данной ситуации (иногда его называют *универсальным* и обозначают через U), то разность $U - B$ часто обозначается $\overline{B}$ и называется *дополнением* множества B.

Заметим, что мы говорим об универсальном множестве как о множестве всех элементов, „рассматриваемых в данной ситуации". При этом мы должны быть уверены в том, что U существует. Например, если взять в качестве U „множество всех множеств", то снова получится парадокс Рассела. Заметим также, что $\overline{B}$ не определено, если не ясно, по отношению к какому универсальному множеству рассматривается операция дополнения.

---

[1]) Заметим, что существование множества $A \cup B$ не гарантировано, так что возможность определения с помощью предиката вызывает сомнение. В аксиоматической теории множеств существование множества $A \cup B$ принимается за аксиому.