

А. В. ЗУБОВ, И. И. ЗУБОВА

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ЛИНГВИСТИКЕ

Рекомендовано

*Учебно-методическим объединением
по образованию в области лингвистики
в качестве учебного пособия для студентов
высших учебных заведений, обучающихся
по специальности 021800 —
Теоретическая и прикладная лингвистика*

УДК 800(075.8)
ББК 81.1я73
3-915

Рецензенты:

кафедра теоретической и прикладной лингвистики
лингвистического факультета Московского государственного
областного университета (зав. кафедрой —
доктор филологических наук, профессор *Ю. Н. Марчук*);
кандидат филологических наук,
старший научный сотрудник ИНИОН РАН *С. Ю. Семенова*

Зубов А. В.

3-915 Информационные технологии в лингвистике: Учеб. пособие для студ. лингв. фак-тов высш. учеб. заведений / А. В. Зубов, И. И. Зубова — М.: Издательский центр «Академия», 2004. — 208 с.

ISBN 5-7695-1531-7

Пособие содержит как общие, так и профессиональные знания, необходимые будущему специалисту-лингвисту. В нем рассмотрены основные понятия современных информационных технологий, показаны возможности их использования в проведении лингвистических исследований, практической обработке текстов и обучении иностранным языкам на базе персональных компьютеров.

Для студентов лингвистических факультетов высших учебных заведений. Может быть полезно аспирантам и преподавателям информатики.

УДК 800(075.8)
ББК 81.1я73

© Зубов А. В., Зубова И. И., 2004
© Образовательно-издательский центр «Академия», 2004
© Оформление. Издательский центр «Академия», 2004

ISBN 5-7695-1531-7

*Любимому сыну Антону, будущему
пользователю суперсовременных
информационных технологий,
посвящается*

ПРЕДИСЛОВИЕ

Становление современного информационного общества приводит к коренным изменениям во всех сферах жизни и деятельности человека. В сознании людей все больше утверждается мысль, что будущий стратегический потенциал общества будут составлять не вещество и энергия, а информация и научные знания. В недалеком будущем реально защищенным в социальном плане может быть лишь только широкообразованный человек, способный гибко перестраивать направление и содержание своей деятельности в связи со сменой технологий или требований рынка [98, 5—7]¹.

В наши дни владение информационными технологиями ставится в один ряд с такими качествами, как умение читать и писать [193, 10]. Сегодня специалист с высшим образованием должен свободно ориентироваться в мировом информационном пространстве, иметь необходимые знания и навыки поиска, обработки и хранения информации с использованием современных информационных технологий, компьютерных систем и сетей [111, 21].

Вполне естественно, что приобретаемые в процессе изучения информационных технологий знания, умения и навыки зависят как от уровня обучения, так и от основной специальности обучаемого. Для студентов, обучающихся по специальности «Теоретическая и прикладная лингвистика», необходимы более углубленные знания по проблемам алгоритмизации, моделированию лингвистических задач, современным языкам программирования. Эти знания позволят им получить четкое представление о том, как ставится и решается лингвистическая задача с помощью компьютера: от ее словесной формулировки к алгоритму и компьютерной программе.

Все это позволит студентам в будущем эффективно использовать информационные технологии для автоматического распознавания и обработки текста и речи, статистического анализа тек-

¹ В квадратных скобках дана ссылка на список литературы в конце книги. Первая цифра обозначает порядковый номер книги в списке, последующие цифры — номера страниц. Ссылка на другие книги списка литературы указана после точки с запятой.

стов, моделирования в филологических исследованиях, обучения языкам [139]. Все сказанное выше возможно без знания высшей математики и построения сложных математических моделей. Более того, как отметил известный специалист по автоматической обработке текстов Терри Виноград: «ЭВМ — это языковые машины: основа их могущества заключается в способности манипулировать лингвистическими знаками — символами, которым приписывается некоторый смысл» [37, 90]. Таким образом, естественный язык занимает в информатике фактически центральное место.

Основное отличие данной книги от других книг по использованию информационных технологий в гуманитарных науках заключается в том, что в ней по отношению к письменному тексту не только ставятся задачи и предлагаются методы их решения, но и приводятся детальные процедуры их выполнения. Такой подход позволит студенту быть готовым к практическому созданию описанных в книге систем и разработке процедур решения аналогичных задач.

ЛИНГВИСТИКА И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

ЛИНГВИСТИКА: РАЗДЕЛЫ И НАПРАВЛЕНИЯ

Существует достаточно большое число определений понятия **лингвистика**. В «Лингвистическом энциклопедическом словаре» (1990) лингвистика (языкознание, языковедение) определяется как «наука о естественном человеческом языке вообще и о всех языках мира как индивидуальных его представителях» [115, 618]. Несколько конкретизируя приведенное выше определение, Ю. С. Маслов пишет, что она исследует сущность и природу языка, проблему его происхождения и общие законы его развития и функционирования [127, 4]. Более детально задачи лингвистики рассмотрены в упомянутом энциклопедическом словаре [115, 618—622], а также в работе В. В. Звегинцева [60, 9—22].

Существуют многочисленные попытки выделения внутри лингвистики отдельных ветвей или направлений [60, 3—36; 155, 5—14; 90, 261—262; 22; 181, 5—12; 45; 189, 4—9; 49, 5—31; 48, 4—28; 143, 3—6; 124, 3—23]. В работах [60, 3—36; 189, 4—9] четко отделяются друг от друга теоретическая и прикладная лингвистика. Общим для перечисленных и некоторых других работ этого периода (конец 60-х — начало 70-х годов XX века) является твердое убеждение, что эти два направления лингвистики взаимосвязаны и дополняют друг друга, что успешное функционирование прикладной лингвистики возможно лишь на базе лингвистических теорий, разработанных в рамках теоретической лингвистики.

Некоторые из существовавших в то время и новых зарождающихся подходов к анализу языковых явлений (машинный перевод, автоматическая обработка речевой информации, порождающая грамматика, дескриптивная лингвистика, математическая лингвистика и др.) почти единогласно относились авторами к проблемам прикладной лингвистики.

Несколько позже Р. Г. Пиотровский, четко разделяя теоретическую и прикладную лингвистику (но не признавая последнюю как самостоятельный раздел языкознания), выделяет структурное и математическое языкознание, а также «новую» лингвистику, к которой относит инженерную лингвистику [155, 5—15]. В последней он выделяет далее вычислительную лингвистику,

экспериментальную фонетику, лингвистическое обеспечение систем научно-технической информации [155, 11].

Группа ученых Ленинградского университета [22, 13], опираясь на три основных аспекта любой достаточно развитой области знания — теорию, эксперимент, практику, — и учитывая их диалектическую взаимосвязь в процессе познания, выделила в языкознании три взаимосвязанных направления: теоретическую лингвистику, экспериментальную лингвистику и прикладную лингвистику. При этом структурную лингвистику авторы считают симбиозом таких наук, как теоретическое языкознание, психология, логика, семиотика, математика, математическая лингвистика. С их точки зрения, это совокупность методов теоретической лингвистики и математики [22, 4]. В последние годы в рамках прикладной лингвистики выделяют также компьютерную лингвистику [49, 9—11; 124, 23; 17].

Как видно из вышесказанного, «прикладная лингвистика» — понятие, до сих пор не имеющее четкого определения и конкретного конечного перечня решаемых ею задач. Тем не менее попытки разобраться с этими проблемами были и есть.

Одно из первых определений понятия **прикладная лингвистика** принадлежит В. В. Звегинцеву. Он определил ее как новую область лингвистики, «которая осуществляет реализацию лингвистических знаний с целью решения всякого рода практических задач» [60, 23]. Далее, несколько детализируя сказанное, автор отмечает: «Прикладная лингвистика представляет новый взгляд на задачи изучения языка. Исходя из этого нового взгляда, она производит переоценку достигнутого в науке о языке, направляет по определенному руслу лингвистические исследования и, конечно, комплектуется собственной тематикой» [60, 24].

В «Лингвистическом энциклопедическом словаре» прикладная лингвистика определяется как «направление в языкознании, занимающееся разработкой методов решения практических задач, связанных с использованием языка» [115, 397]. При этом задачи прикладной лингвистики делятся на традиционные, или «вечные», и «новые» [45, 126, 127]. К числу первых относят создание и совершенствование письменности, разработку систем транскрипции устной речи, систем транслитерации иноязычных слов, унификацию и стандартизацию научно-технической терминологии, создание словарей различных типов, перевод с языка на язык, обучение языку и т. д. [115, 397; 45, 126]. К новым задачам прикладной лингвистики относятся те, появление которых обусловлено современной научно-технической революцией, характеризующейся укреплением взаимосвязи общественных, естественных и технических наук. Перечень таких задач достаточно широк [115, 397; 45, 127—132; 181]. Наиболее удачным представляется определение прикладной лингвистики и перечня решаемых ею задач,

сделанное А. Е. Кибриком. По мнению автора, «прикладная лингвистика — раздел языкознания, в котором разрабатываются методы решения практических задач, связанных с оптимизацией использования языка как важнейшего средства человеческой коммуникации» [97, 261]. Множество задач, решаемых прикладной лингвистикой, А. Е. Кибрик выделяет с опорой на те функции языка, которые оптимизируются задачами прикладной лингвистики.

В частности, он выделяет следующие четыре функции [97, 262]:

- 1) оптимизация способов фиксации и хранения речевой информации;
- 2) оптимизация способов передачи информации;
- 3) оптимизация интеллектуальных способностей человека, связанных с использованием языка;
- 4) оптимизация использования языка как средства массовой коммуникации.

В рамках каждого из этих разделов автор выделяет конкретные задачи. Их также можно условно разделить на «вечные» и «новые» (ср. с. 6). К числу задач первого типа, отмечаемых А. Е. Кибриком, можно отнести создание алфавитов и письменностей, создание систем транскрипции и транслитерации, задачи сурдопедагогики и задачи использования языка в медицине, языковое планирование, языковое строительство, нормализацию языка и т. д.

«Новые» задачи — это такие проблемы, решение которых возможно с использованием современных информационных технологий.

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И ПРИЧИНЫ, СПОСОБСТВУЮЩИЕ ИХ ПОЯВЛЕНИЮ

До последнего времени мощь любого государства определялась уровнем развития промышленности, новизной и эффективностью ее технической базы. Именно с опорой на новейшее техническое оборудование совершенствовалась технология материального производства. Под словом **технология** (от греч. *téchnē* — ‘искусство, мастерство, умение’ и *lógos* — ‘слово, учение’) понимается «совокупность методов обработки, изготовления, изменения состояния, свойства, формы сырья, материала или полуфабриката, используемых в процессе производства продукции» [205, 1338]. Такая технология была основой индустриального общества.

Семидесятые годы XX века стали периодом создания персональных компьютеров и началом развития информационного общества. Основное отличие этого общества от индустриального заключается в том, что наряду с высокоэффективной технологией материального производства все бóльшую роль начинают играть

информационные технологии как совокупность законов, методов и средств получения, хранения, передачи, распространения, преобразования информации с помощью компьютеров.

Наряду с приведенным выше существуют и иные определения понятия «информационные технологии» [86, 23; 235].

Появлению и бурному развитию информационных технологий способствовали следующие достижения научно-технического прогресса:

1) создание персональных компьютеров с большой памятью и большой скоростью выполнения операций;

2) разработка звуковых плат, дающих возможность воспроизводить и записывать речь, звуки и музыку в большом диапазоне частот;

3) изобретение видеоплат, позволяющих выводить на экран компьютеров изображение с телеэкранов и видеомagneтофонов;

4) разработка специальных (мультимедийных) компьютеров, позволяющих воспроизводить на экране дисплеев цвет, звук, музыку, движение;

5) создание специальных процессоров и устройств, способных передавать информацию в сети от одного компьютера к другому;

6) разработка устройств электронной связи (модемов), позволяющих передавать информацию на далекие расстояния (по телефонным линиям, кабелям, радиоканалам и т.п.);

7) создание электронной оргтехники, связанной с персональными компьютерами и позволяющей осуществлять высокоскоростную печать документов, их копирование и размножение.

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ЛИНГВИСТИКЕ

Конкретизируя определение понятия «информационные технологии» по отношению к лингвистике, можно сказать, что **информационные технологии в лингвистике** — это совокупность законов, методов и средств получения, хранения, передачи, распространения, преобразования информации о языке и законах его функционирования с помощью компьютеров. Если соотнести это определение с теми задачами, которые решает современная прикладная лингвистика [45, 127—132; 97, 262; 181; 115, 397], то можно отметить, что понятие «информационные технологии» в лингвистике относится в основном к задачам прикладной лингвистики. К их числу можно отнести:

1) создание систем искусственного интеллекта;

2) создание систем автоматического перевода;

3) создание систем автоматического аннотирования и реферирования текстов;

4) создание систем порождения текстов;

- 5) создание систем обучения языку;
- 6) создание систем понимания устной речи;
- 7) создание систем генерации речи;
- 8) создание автоматизированных информационно-поисковых систем;
- 9) создание систем атрибуции и дешифровки анонимных и псевдоанонимных текстов;
- 10) разработка различных баз данных (словарей, карточек, каталогов, реестров и т. п.) для гуманитарных наук;
- 11) разработка различного типа автоматических словарей;
- 12) разработка систем передачи информации в сети Интернет и т. д.

Эти комплексные задачи включают целый ряд более мелких проблем. К их числу относится автоматизация следующих процессов:

- 1) построение словарей текстов;
- 2) морфологический анализ слова;
- 3) определение значения многозначного слова;
- 4) синтаксический анализ предложения;
- 5) поиск слова в словаре;
- 6) порождение предложения и т. д.

Некоторые из приведенных выше задач детально рассматриваются в данной книге.

БУДУЩЕЕ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

Философы, психологи и другие специалисты отмечают, что в будущем социально защищенным может считаться лишь тот человек, который способен гибко перестраивать направление и содержание своей деятельности в связи со сменой технологий или требований рынка [98, 5; 239]. Чтобы подготовить такого человека, необходимо заменить традиционную технологию получения новых знаний более эффективной организацией познавательной деятельности обучаемых в ходе учебного процесса. Это можно сделать с использованием современных информационных технологий. Именно они могут продемонстрировать обучаемому тот факт, что любой информационный ресурс¹ представляет реальную ценность лишь в том случае, когда к нему организован соответствующий доступ. С их помощью будущих специалистов можно научить правильной организации хранения информации и выбору адекватных форм ее представления [216, 34]. «Интеллектуальная собственность, представленная в цифровом формате, станет главной “валютой” XXI века» [19, 40].

¹ Об информационных ресурсах см. с. 156—175.

Технология получения и распространения новых знаний уже сейчас неотделима от Internet. Формируется инфраструктура, объединяющая в единое целое глобальные и местные телекоммуникационные каналы, радио, телевидение, телефонные линии связи (одна планета — одна сеть). Все это не просто создает сверхинтеллект, а формирует новые измерения сознания, феномен сверхпсихологических изменений в личности человека [19, 39; 98, 6]. Широкое распространение в недалеком будущем получат видеоконференции и дистанционное обучение¹ [19, 40; 117, 93, 94; 54; 85; 178, 103—261, 522—555].

К концу XXI века ученые изучат принципы работы мозга на уровне отдельных нейронов и научатся обращаться с ним, как со сложным электронным объектом [19, 41]. Это даст новый толчок к созданию систем искусственного интеллекта, систем автоматического порождения текстов, их перевода, реферирования и т. д. В ближайшие годы должна найти решение проблема распознавания и синтеза устной речи, широкое распространение получат электронная коммерция (e-commerce) мобильная цифровая телефонная связь, геновая инженерия [19, 39, 40; 43; 117, 92, 93].

¹ О дистанционном обучении, сети Интернет и видеоконференциях см. с. 142—145, 178—189.

ОСНОВНЫЕ СОСТАВЛЯЮЩИЕ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

СТРУКТУРА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

В составе современных информационных технологий можно выделить следующие составляющие:

- 1) теоретические основы информационных технологий;
- 2) методы решения задач информационными технологиями;
- 3) средства решения задач, используемые в информационных технологиях:

- а) аппаратные средства;
- б) программные средства.

Рассмотрим подробнее эти составляющие.

ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

Теоретическую основу информационных технологий составляют важнейшие понятия и законы информатики. В свою очередь понятие «информатика» тесно связано с понятием «информация».

Слово **информация** (от лат. *informatio* — ‘разъяснение, изложение’) в обычном житейском понимании обозначает некоторые сведения о внешнем и внутреннем мире, которые мы используем для регулирования своего поведения. Более строго это понятие раскрывается разными способами [160, 7; 202, 4, 5; 226, 13—16]. Выберем один из них, который более всего подходит к рассматриваемым в пособии лингвистическим задачам, и определим информацию как определенным образом связанные сведения, данные, понятия, отраженные в нашем сознании и изменяющие наши представления о реальном мире [226, 16].

Информация обладает разными свойствами [152, 45—48; 226, 18—20]. Наиболее важными из них являются: ценность, достоверность, полнота, актуальность, логичность, компактность. **Ценность** информации определяется тем, насколько она важна для достижения цели, стоящей перед ее получателем. **Полнота** информации связана с тем, насколько много в ней сведений,

позволяющих получателю информации достичь своей цели. **Актуальность** информации определяется необходимостью ее немедленного использования для достижения какой-либо цели. **Компактность** информации — способность представить ее в наиболее сжатом виде. Понятия **достоверность** и **логичность** информации не требуют особых пояснений.

Выделяют различные виды информации. При этом для ее классификации по видам разработано много подходов, использующих разнообразные признаки и особенности информации. Так, в зависимости от того, какими органами чувств воспринимается информация, ее делят на визуальную, аудиальную (звуковую, фонетическую), аудиовизуальную, тактильную. По направленности информации всем членам общества или каким-то его группам различают информацию массовую, предназначенную для всех членов общества, и специальную — для специалистов в различных областях науки, техники, культуры, производства. Специальную информацию подразделяют на научную, техническую, производственную, эстетическую и т. п.

В каждом виде специальной информации выделяют подвиды. Например, в зависимости от области науки и научной информации выделяют информацию физическую, математическую, биологическую, лингвистическую и т. д. Так, **лингвистической информацией** называют множество определенным образом связанных сведений, данных, понятий о языке и правилах его функционирования, отраженных в нашем сознании и влияющих на наше речевое поведение.

Слово «информатика» также не имеет единого определения. С современной точки зрения **информатика** — это наука о законах и методах получения, хранения, передачи, распространения, преобразования и использования информации в естественных и искусственных системах с применением компьютера.

В зависимости от вида информации выделяют различные типы информатики. Так, различают информатику социальную, экономическую, научную, научно-техническую, статистическую, биологическую, медицинскую и т. п.

Наука, изучающая законы и методы организации и переработки с помощью компьютера лингвистической информации, называется **лингвистической информатикой**. Вспомнив, что понимается под лингвистической информацией, можно сказать, что **объектом исследования** лингвистической информатики будет структура слов, словосочетаний, предложений, текстов.

Ее интересуют правила, объединяющие нижестоящие языковые единицы в вышестоящие, правила перевода предложений и текстов, способы построения рефератов и аннотаций, пути обучения языкам и целый ряд других вопросов, связанных с языком и речью.

МЕТОДЫ РЕШЕНИЯ ЗАДАЧ С ИСПОЛЬЗОВАНИЕМ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

Основным методом решения различных задач информационными технологиями является **метод моделирования**. Суть его заключается в том, что для решения какой-либо задачи строится модель некоторого объекта, явления или процесса. Этот метод используется человеком очень давно. Существуют различные определения понятий «модель» и «моделирование» [16, 13; 17, 22; 235, 45—53]. За основу примем следующее определение понятия «модель» [103, 46]: **модель** — это формализованное описание объекта, системы нескольких объектов, процесса или явления, выраженное конечным набором предложений какого-либо языка, математическими формулами, таблицами, графиками, специальными знаками или какими-нибудь схемами.

Описание считается формализованным, если оно понятно не только человеку, но и некоторому устройству, например компьютеру. Предположим, архитектор, разрабатывая план какого-либо города или поселка, строит его модель в виде таких специальных знаков, как квадрат, прямоугольник, круг, которые обозначают целые дома, заводы, улицы и т. п.

Модель взаимосвязи в треугольнике его сторон (катетов a и b и гипотенузы c) выражается формулой $c^2 = a^2 + b^2$.

Модель распределения словоформ какого-либо текста по частоте употребления может быть представлена в виде таблицы 1.

По отношению к моделируемому объекту, процессу или явлению модель должна удовлетворять целому ряду свойств. Важнейшими из них являются следующие [163, 23—28].

1. Модель выступает в качестве упрощенного аналога изучаемого объекта (процесса, явления).
2. Модель не должна быть сложнее самого оригинала.
3. Метод изучения объекта (процесса, явления) путем его моделирования должен быть более экономным по сравнению с другими возможными методами изучения того же объекта.
4. Построенная модель должна быть предельно простой и логически корректной, не содержащей противоречий.
5. Модель должна по возможности иметь общий (универсальный) характер, позволяющий использовать ее для изучения дру-

Т а б л и ц а 1

Словоформа	Частота
Информация	73
Компьютер	46
Технология	27

гих подобных объектов (процессов, явлений). Например, построив на материале английского текста модель его реферирования, опирающуюся на ключевые слова текста (см. с. 56—75), необходимо, чтобы эта модель работала и для текстов других языков.

6. Модель должна отражать наиболее существенные черты реального объекта, процесса или явления, которые важны для проводимого в данный момент процесса моделирования.

Существуют различные виды моделей. При использовании информационных технологий в лингвистике выделяют следующие типы моделей [17, 26, 27; 18; 165, 23, 24; 235, 46—53].

1. **Структурные модели** служат для изучения и описания внутреннего строения некоторого объекта. Например, такая модель строится, если необходимо изучить систему согласных какого-либо языка или устройство речевого аппарата человека.

2. **Функциональные модели** позволяют изучать поведение некоторого объекта, течение некоторого процесса или же этапы реализации некоторого явления. Например, функциональная модель строится, если необходимо смоделировать процесс создания некоторого текста человеком. Такая же модель создается для объяснения процесса перевода текста с одного языка на другой.

3. **Динамические модели** создаются при необходимости найти объяснение некоторых процессов или явлений в их временном развитии. Так, если требуется узнать, как со временем менялось произношение некоторого слова, строят динамическую модель такого процесса.

Особая роль в лингвистике отводится функциональным моделям, позволяющим раскрыть суть функционирования языка, механизма производства и восприятия речи и текста. Нельзя заглянуть в мозг человека и посмотреть, как в нем осуществляются операции с буквами, звуками, словами, предложениями при всевозможном использовании языка. Поэтому для решения таких задач в рамках функциональных моделей выделяют **воспроизводящие инженерно-лингвистические модели** (ВИЛМ). Они представляют собой компьютерные системы, поведение которых, с одной стороны, имитирует поведение реальных лингвистических объектов, а с другой стороны, позволяет хотя бы частично воспроизвести эти реальные объекты [164, 25, 26].

Многочисленные примеры используемых в лингвистике моделей можно найти в работах [64; 83; 84; 96; 103; 123; 156; 164; 191; 215].

Как отмечалось выше, существуют разные способы формализованного описания объекта, процесса или явления: формулы, таблицы, графики, схемы, наборы предложений естественного языка и т. д. Все эти способы составляют основу алгоритмического решения задач с помощью ПК.

Общие понятия об алгоритме

С точки зрения современной психологии **задача** в самом общем понимании — это некоторая цель, поставленная в конкретных условиях и требующая исполнения, решения [170, 39]. Примерами интеллектуальных задач являются следующие: 1) решить полное квадратное уравнение $ax^2 + bx + c = 0$; 2) составить таблицу значений x^2 , x^3 и $1/x$ величины x , меняющейся с некоторым шагом k от некоторого начального значения n до некоторого конечного значения m ; 3) найти среди группы русских глаголов те, которые употреблены в инфинитиве; 4) составить реферат научного текста; 5) перевести текст с английского языка на русский и т. д.

Чтобы решить задачу, необходимо знать ее **начальные условия**, а также **метод** или **способ** ее решения. Так, чтобы решить полное квадратное уравнение, необходимо знать конкретные значения коэффициентов a , b и c (начальные условия). В качестве метода решения этого уравнения надо использовать правило вычисления значений x_1 и x_2 :

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}.$$

Для выделения из группы русских глаголов инфинитивных форм необходимо, чтобы среди анализируемых глаголов были эти инфинитивные формы (начальные условия). А способ решения сводится к следующей проверке: оканчивается ли соответствующий глагол на *-ть*, *-чь*, *-ти*. Чтобы провести такую проверку, надо выполнить определенные действия: выделить у глагола две последние буквы, сравнить их с окончаниями *-ть*, *-ти*, *-чь* и т. д. Чтобы перевести текст на русский язык, необходимо иметь, как минимум, англо-русский словарь и знать английскую и русскую грамматики, лексикологию и еще многое другое. Все это начальные условия. В качестве метода решения этой задачи выступают те правила перевода текстов, которым обучают в вузе.

Таким образом, метод или способ решения некоторой задачи сводится к поиску определенных правил. Согласно «Словарю русского языка» С. И. Ожегова **правило** — это предписание, устанавливающее порядок чего-нибудь [149, 529]. Точное предписание о выполнении в определенном порядке некоторой последовательности действий (физических или умственных), приводящее к решению некоторой типовой задачи, называют **алгоритмом** (ср. [237, 23, 24]). Например, при необходимости сварить кофе последовательность физических действий будет такой: вскипятить нужное

количество воды, засыпать кофе в горячую воду (одну-две чайные ложки на стакан воды), нагреть воду до кипения (но не кипятить) и т.д. Определенные последовательности физических действий выполняются человеком и при решении таких задач, как «добраться из дома в университет», «найти в большом городе нужный дом», «изготовить на станке какую-то деталь» и т.п. Примеры задач, для решения которых необходимо выполнить определенную последовательность умственных действий, приведены выше.

Слово **алгоритм** происходит от слова *algorithmi* — латинской формы написания имени великого математика IX века аль-Хорезми. Он впервые четко сформулировал правила выполнения арифметических действий. Сейчас это понятие используется для обозначения последовательности любых действий (арифметических, логических, взятия логарифмов, вычисления синуса и т.п.).

Алгоритмы обладают следующими основными свойствами: дискретностью, результативностью, массовостью, детерминированностью и формализованностью [237, 25, 26].

Дискретность алгоритма заключается в том, что он разбивается на конечное число действий-шагов (предписаний, команд), которые могут быть пронумерованы. Причем только после выполнения одного предписания можно перейти к выполнению другого.

Результативность алгоритма означает, что при всех начальных условиях число шагов алгоритма конечно, и он приводит к решению задачи.

Массовость алгоритма предполагает, что по данному алгоритму может быть решен целый ряд типовых задач (они отличаются лишь различными начальными условиями).

Детерминированность алгоритма заключается в том, что при многократном решении одной и той же задачи с одинаковыми начальными условиями всегда получается один и тот же результат.

Формализованность алгоритма состоит в том, что тот, кто его выполняет (человек, машина), может не вникать в смысл того, что он делает согласно предписаниям алгоритма, и все равно придет к верному результату.

Между задачей и ее алгоритмом соответствие неоднозначное. Очень мало задач имеют только один алгоритм решения. Например, задача «позвонить по междугороднему телефону» для данного типа телефонного автомата имеет единственный алгоритм, представленный в виде правила пользования этим телефонным аппаратом. Большинство задач могут иметь несколько алгоритмов решения. Так, есть несколько правил приготовления кофе, можно различными путями добраться из дома в университет, несколькими способами составить по тексту его реферат и т.д. В то же время есть задачи, алгоритм решения которых до сих пор неизве-

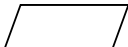
стен. Так, неясно, как человек пишет стихи, повесть или научную статью. Нет точных предписаний, как переводить текст с одного языка на другой и т.д.

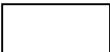
Способы записи алгоритмов

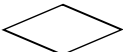
Существует несколько способов записи алгоритмов решения задач. Наиболее известны следующие: словесный, графический и табличный.


Словесное представление алгоритма решения задачи сводится к тому, что составляющие алгоритм шаги (предписания) записываются в виде слов и предложений естественного языка.


При **графическом представлении** алгоритма его шаги изображаются разными геометрическими фигурами (блоками), образующими блок-схему алгоритма. Связи между блоками обозначены стрелками, соединяющими соответствующие фигуры. Чаще всего в качестве таких геометрических фигур используются следующие:

1. Параллелограмм  используется для обозначения действий ввода информации в компьютер и вывода информации из него.

2. Прямоугольник  используется для записи вычислительных и некоторых других действий.

3. Ромб  используется для проверки различных условий.

4. Овал  используется для обозначения начала и конца алгоритма.

5. Круг  служит для указания тех блоков алгоритма, на которые передается управление от блоков первых трех типов.

При **табличном представлении** алгоритма его шаги записываются в графах специальных таблиц. Чаще всего такой способ записи алгоритма используется для выполнения различных вычислений по формулам.

Продemonстрируем три способа записи алгоритма на примере следующей задачи: «Решить полное квадратное уравнение $ax^2 + bx + c = 0$ (1)» (слово «решить» в данном случае означает, что надо найти то значение x , при котором левая часть равенства

обращается в нуль). Как было отмечено выше, способ решения

этой задачи определяется равенством $x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$ (2),

где величина $b^2 - 4ac$ называется дискриминантом. Тогда формула (2) может быть вычислена по следующему **словесному алгоритму**.

1. Присвоить коэффициентам a , b и c конкретные начальные числовые значения.

2. Вычислить значение $b^2 - 4ac$.

3. Если $b^2 - 4ac \geq 0$, то выполнить шаг 6. Если $b^2 - 4ac < 0$, то выполнить шаг 4.

4. Сделать вывод: «Уравнение корней не имеет».

5. Перейти к шагу 8.

6. Вычислить $x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$.

7. Вычислить $x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$.

8. Закончить работу.

Запись процесса решения этой же задачи в виде **блок-схемы** (в графическом виде) может быть представлена следующим образом (схема 1).

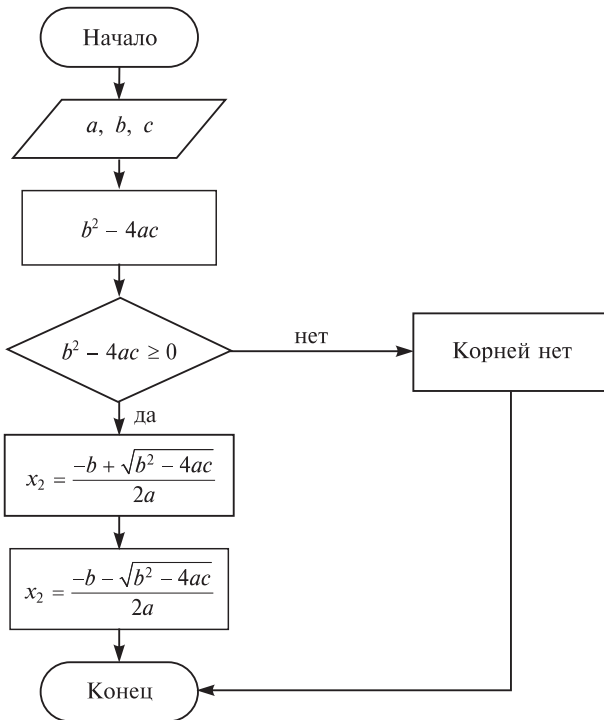
Наконец, алгоритм решения того же уравнения в **табличной форме** может быть задан следующим образом (например, при значениях коэффициентов: $a = 1$, $b = -7$, $c = 12$, т.е. для уравнения $x^2 - 7x + 12 = 0$) (табл. 2).

Т а б л и ц а 2

a	b	c	b^2	ac	$4ac$	$b^2 - 4ac$	$\sqrt{b^2 - 4ac}$	$2a$
1	-7	12	49	12	48	1	± 1	2
...

Окончание табл. 2

$-b + \sqrt{b^2 - 4ac}$	$-b - \sqrt{b^2 - 4ac}$	$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$	$x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$
$7 + 1 = 8$	$7 - 1 = 6$	$8/2 = 4$	$6/2 = 3$
...



СРЕДСТВА РЕШЕНИЯ ЗАДАЧ, ИСПОЛЬЗУЕМЫЕ В ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЯХ

Аппаратное и программное обеспечение информационных технологий

Как уже отмечалось, к используемым в информационных технологиях средствам решения задач относятся:

- 1) аппаратное обеспечение информационных технологий (*hardware*);
- 2) программное обеспечение информационных технологий (*software*).

К **средствам аппаратного обеспечения** информационных технологий относятся компьютер и периферийные устройства, т.е. различные устройства хранения, ввода и вывода данных. Их разновидности и условия функционирования достаточно детально опи-

саны в большом числе специальных изданий и учебников по информатике [235; 87; 186; 110; 188; 259].

Как отмечалось выше, **программное обеспечение** (ПО) компьютера является вторым важным средством решения задач, используемых в информационных технологиях. Программное обеспечение современных персональных компьютеров условно делится на три группы: 1) системное ПО; 2) прикладное ПО; 3) прикладные инструментальные средства.

В системном программном обеспечении (первая группа ПО), в свою очередь, выделяют две группы программ: 1) операционные системы и 2) дополнительные системные программы (программы-утилиты и драйверы).

Операционная система (ОС) — это главная программа, загружаемая в оперативную память компьютера после его включения. Основные функции ОС сводятся к следующему:

- 1) управление работой персонального компьютера (управление внутренними функциями ПК, осуществление контроля за выполнением операций, распределение памяти и т. п.);
- 2) запуск на выполнение прикладных программ;
- 3) обеспечение пользователю удобного способа общения с компьютером.

Современные операционные системы разрабатываются по следующим двум направлениям [133]:

- 1) создание принципиально новых операционных систем, связанных с широким использованием цифровых способов обработки информации;
- 2) использование в будущих версиях операционных систем лингвистических технологий: технологии речевого и рукописного ввода информации, автоматического самообучения компьютерных систем и т. д.

Результатом работ первого направления является новая операционная система BeOS фирмы «Be». Это мультимедийная ОС, рассчитанная на многопроцессорные компьютеры. Обработка на таких ПК мультимедийных данных проходит значительно быстрее, чем на компьютерах с ОС Windows NT или UNIX. Решаемая под управлением ОС BeOS задача разбивается на множество небольших задач, которые выполняются одновременно. В зависимости от сложности основной задачи таких небольших задач может насчитываться до тысячи. Данная операционная система тоже поддерживает принцип многозадачности, т. е. с помощью ОС BeOS можно одновременно решать несколько сложных задач. Основное назначение такой системы — удовлетворение потребностей разработчиков различных приложений для обработки мультимедийных данных (видео, звука, графики, трехмерной анимации) [214].

Второе направление развития операционных систем сформулировано в ряде публичных выступлений руководителя фирмы

«Microsoft» Билла Гейтса. Эта крупнейшая фирма продолжает разрабатывать серию ОС Windows. Примерами являются ОС Windows XP — следующая после ОС Windows 2000 сетевая офисная операционная система и ОС Millenium — операционная система для домашних мультимедийных ПК, имеющая еще более упрощенный пользовательский интерфейс и гораздо больше возможностей по обработке мультимедийной информации, в том числе и огромных потоков данных, поступающих из сети Интернет.

Как отмечалось выше, к системному ПО относятся и программы-утилиты. **Утилита** — это программа, расширяющая возможности операционной системы, помогающая работать с компьютерной системой и повышающая ее эффективность. К числу таких программ можно отнести:

1) программы-архиваторы — служат для упаковки больших объемов информации с высоким коэффициентом сжатия;

2) программы для создания резервных копий обрабатываемой информации — позволяют быстро копировать данные на объемные носители информации;

3) антивирусные программы — предотвращают заражение компьютера «вирусом» (специальной программой, уничтожающей или искажающей информацию в памяти компьютера) и ликвидирующие последствия такого заражения;

4) программы для диагностики компьютера — проверяют работоспособность всех устройств компьютера и т. д.

Важным классом системных программ являются **программы-драйверы**, необходимые для управления устройствами компьютера (чаще всего устройствами ввода-вывода). Наиболее распространены драйверы клавиатуры, мыши, принтера, сканера и т. д.

Вторую большую часть программного обеспечения компьютера составляют **программы пользователя**, или **прикладные программы**. С их помощью можно решать некоторые профессиональные задачи, связанные с обработкой информации в науке, промышленности, сельском хозяйстве, образовании, медицине, культуре и т. д. Невозможно перечислить все программы, составляющие эту часть ПО. В качестве отдельных примеров назовем следующие.

1. Текстовые процессоры служат для подготовки и печати различных документов. Такие программы позволяют проводить редактирование и форматирование текста, вставлять в текст графические изображения, таблицы, формулы, диаграммы и т. д. Для оформления документа они предлагают пользователю сотни различных шрифтов. К их числу относятся, например, такие программы, как MS Word, Corel WordPerfect, Lotus WordPro и т. д.

2. Программы автоматического преобразования графической информации в текстовую используются для преобразования полученного с помощью сканера графического изображения текста в текстовый файл с целью его дальнейшей обработки соответ-

ствующими программами (например, текстовыми процессорами, программами-переводчиками и т. п.). Наиболее известны российские программы CuneiForm (фирма «Cognitive Technologies») и FineReader (фирма «ABBYY Software House»).

3. Системы машинного перевода (МП) позволяют переводить текст с одного языка на другой без вмешательства человека. К наиболее известным системам МП, поддерживающим перевод с иностранных языков на русский и обратно, относятся системы PROMT (российская фирма «PROMT») и Socrat (российская фирма «Арсенал»).

4. Системы автоматического аннотирования и реферирования текста позволяют создать аннотацию и реферат научно-технического текста без участия человека (например, программа Аннотатор российской фирмы «MediaLingua»).

5. Настольно-издательские системы используются для создания с помощью компьютера различных видов полиграфической продукции. К их числу относятся, например, такие программы, как PageMaker (американская фирма «Adobe»), Ventura (американская фирма «Corel») и т. д.

6. Обучающие программы, например программы обучения с помощью компьютера английскому языку English Gold и English Platinum¹ и т. д.

7. Экспертные системы широко используются как в промышленности, например для диагностики неисправности отдельных приборов, так и в медицине, например при определении болезни и метода ее лечения.

8. Программы создания и обработки электронных таблиц (табличные процессоры) позволяют обрабатывать различные экономические и статистические данные, представленные в виде сложных таблиц (ведомостей, статистических отчетов и т. п.). В настоящее время наиболее известны такие программы, как MS Excel, Lotus 1-2-3 и др.

9. Системы управления базами данных (СУБД) позволяют осуществлять создание и модификацию больших совокупностей структурированных определенным образом данных (баз данных), а также поиск в них информации. Практически все современные СУБД содержат средства для создания баз данных.

В состав **прикладных инструментальных средств** (третья группа ПО) входят различные средства разработки программного обеспечения. К ним прежде всего относятся языки и системы программирования.

Язык программирования, или **алгоритмический язык**, — это искусственный язык, используемый для представления алгоритма решения задачи в виде, понятном компьютеру. Существуют

¹ Подробнее см. с. 139—142.

различные подходы к классификации языков программирования. Согласно одному из современных подходов языки программирования делятся на следующие четыре типа: 1) языки ассемблера; 2) языки системного уровня; 3) языки описания сценариев; 4) языки промежуточного типа.

Языки ассемблера представляют записанные в алгоритме действия в виде машинных кодов (например, 01 — сложить, 02 — вычесть и т. д.).

В языках системного уровня действия алгоритма записываются в виде отдельных английских слов или их частей. Существуют определенные правила (синтаксис) следования таких слов друг за другом. Подобные языки могут быть простыми и достаточно сложными. К числу простых можно отнести практически неиспользуемые сейчас языки программирования Algol, Fortran, Cobol, используемые — PL/1, Quick BASIC и др. Более сложными современными языками этого типа являются C, C++, Pascal, Turbo Pascal, Java и т. д.

Языки описания сценариев служат для связывания готовых программ в новые более сложные программы. В настоящее время к ним относятся такие языки программирования, как Perl, TCL, Visual BASIC, JavaScript, языки оболочек ОС UNIX и т. д.

К языкам промежуточного типа относится, например, язык Lisp. Он обладает свойствами и языков программирования системного уровня, и языков описания сценариев.

Система программирования (система разработки приложений) — это интегрированный набор средств разработки программ, обычно включающий язык программирования, средства компоновки и отладки программы, а также обширную библиотеку готовых к использованию программных модулей.

Элементы алгоритмического языка QBASIC

Язык BASIC был создан в 1965 году в США. Слово BASIC является аббревиатурой английского словосочетания Beginner's All-purpose Symbolic Instruction Code и переводится как «многоцелевой язык программирования для начинающих». Позже появились более совершенные версии этого языка. К ним относятся GW-BASIC, Quick BASIC, QBASIC (усеченный вариант языка Quick BASIC), Turbo BASIC, Visual BASIC и др. Каждая из версий имеет свои особенности и специфичное применение [42; 61; 95].

Для того чтобы в полной мере проследить переход от словесной формулировки лингвистической задачи к ее компьютерной реализации, рассмотрим основные составляющие языка QBASIC [61, 59—128]. К ним относятся:

- 1) алфавит языка QBASIC;

- 2) типы исходных данных;
- 3) операторы обработки исходных данных.

В алфавит языка QBASIC входят следующие знаки:

- 1) все прописные и строчные буквы латинского алфавита;
- 2) десятичные цифры от 0 до 9;
- 3) знаки арифметических действий (см. табл. 3);
- 4) знаки логических действий (см. табл. 4);
- 5) знаки-разделители: ":" (*точка*), ";" (*запятая*), "." (*точка с запятой*), ":" (*двоеточие*), "'" (*апостроф*), "(" (*открывающая скобка*), ")" (*закрывающая скобка*), "_" (*знак подчеркивания*);
- 6) специальные знаки: "#" (*знак номера*), "\$" (*знак доллара*), "&" (*коммерческое "И"*), "!" (*восклицательный знак*), "%" (*знак процента*).

С помощью этих знаков на языке QBASIC описываются действия по обработке текстовых и числовых данных, т. е. создаются программы.

Чтобы познакомиться с тем, каким образом компьютеру передается обрабатываемая информация (каким образом она «описывается»), необходимо знать, какие же данные машина принимает для обработки. В основном компьютер, как и мы, люди, имеет дело с двумя видами информации:

1) информацией, представленной буквами (буквосочетаниями, словами, предложениями, текстами). Ее называют *символьной* или *строковой*;

2) информацией, представленной числами. Ее называют *вещественной*¹.

Каждый тип информации может быть представлен в виде констант (постоянных величин) и переменных, которые в процессе решения задачи могут изменяться.

Строковые константы — это обычные буквы, слова, предложения, тексты любого естественного языка, последовательности цифр, заключаемые в кавычки. Например: «ВЕСНА», «СЕГОДНЯ 29 МАРТА 2001 ГОДА», «ПЕТРОВ И.В.», «ТЕХТ», «FILE» и т. п.

Каждая **строковая переменная** имеет имя (точно так же, как каждый человек имеет свое имя, фамилию, город — название и т. д.). Этим именем называется некоторая область компьютерной памяти, где будут размещаться соответствующие данные (суффикс, слово, предложение, текст, текст с числами и т. д.). В разных алгоритмических языках существуют свои правила записи таких имен. В языке QBASIC для записи имени строковой переменной используется любая латинская буква, входящая в его алфавит. Другим знаком может быть одна из десятичных цифр от 0 до 9 и знак \$. Знак \$ в имени строковой переменной должен всегда стоять на последнем месте. Например: X\$, K2\$, S9\$, BUK\$, STROKA\$ и т. д.

¹ Для упрощения дальнейшего изложения в понятие «вещественная» включаются как дробные, так и целые числа.

Вещественные и целые константы — это привычные для человека дробные и целые числа: 5; 2077; 0,5; 29,0995 и т. п.

Вещественная переменная, как и строковая, должна иметь свое имя. Оно записывается точно так же, как и имя строковой переменной, только в конце имени не ставится знак \$. Так, следующие имена *X*, *K2*, *S9*, *BUK*, *STROKA* служат для размещения в них целых и дробных чисел, а не элементов текста.

Еще одной составляющей языка QBASIC являются операторы обработки исходных данных. **Оператор** — это условная запись действия, выполняемого компьютером над некоторой информацией (данными). В самом общем виде оператор языка QBASIC записывается так:

НС *Имя оператора/содержание оператора*
где НС — номер строки программы или номер оператора в программе обработки данных, может принимать значение от 0 до 65535, не является обязательным;

имя оператора — это одно из слов или частей слов английского языка, обозначающее то действие, которое этот оператор выполняет;

содержание оператора — это какая-то константа или переменная, какое-то арифметическое или логическое выражение¹ или же номер какого-либо другого оператора той же программы. Например:

```
20      LET X=5 ИЛИ LET X=5
120     LET Y=X+Z ИЛИ LET Y=X+Z
1745    IF B$="ТЬ" THEN 70 ИЛИ IF B$="ТЬ" THEN 70
2014    GOTO 1775 ИЛИ GOTO 1775
225     PRINT W$ ИЛИ PRINT W$
```

Можно выделить пять основных групп операторов: 1) арифметических действий; 2) логических действий; 3) управления программой; 4) ввода и вывода информации; 5) специальных.

Простейшим из операторов является **оператор присваивания**, который в общем виде записывается так:

```
НС I=K
```

где НС — номер строки программы; *I* — имя любой переменной, которой присваивается значение *K* (вместо *K* может быть число, слово, предложение, какая-то другая переменная или арифметическое выражение).

Например:

```
20      I=5
        B$="ВЕCHA"
        Z=Y+X
```

¹ Арифметическое или логическое выражение — это несколько переменных, соединенных соответственно знаками арифметических или логических действий (см. табл. 3 и 4).

В первом случае переменной I присписывается значение, равное 5, во втором — в область памяти с именем $B\$$ передается слово ВЕСНА. В третьем случае переменной Z присваивается значение, получаемое в результате сложения значений переменных Y и X .

Операторы арифметических действий выполняют различные арифметические операции над переменными (табл. 3).

Т а б л и ц а 3

Т а б л и ц а 4

Операторы арифметических действий

Операторы логических действий

Тип арифметического действия	Знак действия
Сложение	+
Вычитание	-
Умножение	*
Деление	/
Возведение в степень	^

Тип арифметического действия	Знак действия
Равно	=
Меньше	<
Больше	>
Меньше или равно	<=
Больше или равно	>=

Приведенные в таблице 3 знаки арифметических действий используются при вычислении различных арифметических выражений. Например, пусть необходимо подсчитать следующее значение Y :

$$Y = X^2 - 5X + 3$$

для $X=5$. Тогда программу решения данной задачи¹ на языке QBASIC можно записать так (см. табл. 3):

```
X=5
Y=X^2-5*X+3
```

Сначала переменной X присваивается значение, равное 5, а затем компьютер вычисляет значение Y , подставляя везде вместо X значение 5.

В операторах логических действий выполняются следующие логические операции (табл. 4). Примеры, иллюстрирующие применение этих операторов, будут приведены после рассмотрения третьей группы операторов.

Операторы управления программой. Чаще всего используются следующие операторы управления программой.

1. Оператор безусловного перехода `GO TO` («перейти к»):

```
HC GO TO HC1
```

¹ П р о г р а м м а — это последовательность операторов, которая указывает компьютеру, какие действия и в какой последовательности должны выполняться.

Он означает следующее: «перейти к выполнению оператора, который имеет номер HC1». Например, оператор GO TO 1200 позволяет компьютеру перейти к выполнению оператора под номером 1200.

2. Оператор условного перехода IF THEN («если то»):

```
HC IF условие THEN HC1
```

Этот оператор работает следующим образом: если (IF) выполнено некоторое условие, то (THEN) машина должна перейти к выполнению оператора под номером HC1. Если же это условие не выполнено, то компьютер должен выполнить тот оператор, который стоит ниже. Например, если часть программы записана так:

```
40      IF B$="ТЬ" THEN 70
        IF B$="ТИ" THEN 85
```

то оператор 40 выполняется компьютером следующим образом: если то, что находится в строковой переменной с именем B\$ совпадает (равно) с буквосочетанием *ть*, то компьютер должен выполнить оператор под номером 70. Если же в B\$ находятся любые другие буквы, то компьютер будет выполнять следующий оператор. То же самое действие будет выполнено и в случае, если оператор IF будет записан в виде:

```
HC IF условие GOTO HC1
```

Этот же оператор может быть записан и в следующей форме:

```
HC IF условие THEN оператор
```

В этом случае при выполнении условия будет выполнен некоторый оператор, стоящий за словом THEN. Если же условие не выполнено, то будет выполняться следующий в программе оператор. Например:

```
IF X>0 THEN Y=X^2+5
```

3. Оператор конца программы END («конец»):

```
HC END
```

Он пишется в конце всей программы и позволяет компьютеру закончить решение задачи. Например:

```
X=5
Y=X^2+5*X+3
END
```

Операторы ввода и вывода информации. Для решения любой задачи необходима исходная информация, которая должна быть введена в память компьютера с устройства ввода. Полученные в результате решения задачи данные должны быть выведены из компьютера на устройство вывода. Простейший способ ввода

информации в память машины — это использование оператора присваивания (см. об этом операторе на с. 25). Обычно он используется в тех случаях, когда число переменных, которым в начале работы присваиваются (вводятся в переменные) определенные начальные числовые или строковые значения, невелико.

Рассмотрим, как исходная информация может быть введена в компьютер с клавиатуры дисплея. Для ввода информации любого типа (строковой или числовой) используется оператор ввода INPUT («ввести»), который в полном виде записывается так:

```
HC INPUT P1, P2, ...
```

где HC, как всегда, номер оператора программы, а P1, P2, ... — имена переменных, которым с клавиатуры передаются исходные данные. Если вводится одно данное (одно слово, одно число), то используется только одна переменная P1. Например, в компьютер необходимо ввести для последующего анализа русский глагол *ходить*. Это действие можно записать следующим образом:

```
INPUT W$
```

Как только компьютер встретит данный оператор, он высветит на экране дисплея знак вопроса (?). Человек должен набрать на клавиатуре слово *ходить* (при этом данное слово появляется и на экране дисплея) и нажать на клавиатуре клавишу ENTER. Слово *ходить* попадет в область оперативной памяти с именем W\$.

В случаях, когда в машинную память вводится одно или несколько предложений (текст), целесообразнее использовать оператор ввода LINPUT, который записывается точно так же, как и оператор INPUT:

```
HC LINPUT P1, P2, ...
```

Выполняется он аналогично оператору INPUT, но не «отсекает» при этом пробел, который следует за последним знаком предложения или текста (что делает оператор INPUT). Такой пробел необходим для опознавания конца предложения или текста.

После того как слово *ходить* проанализировано и компьютер, например, принял решение, что это русский глагол в неопределенной форме, результат надо напечатать на принтере. Для вывода информации на принтер используется оператор PRINT, который записывается так:

```
HC PRINT D1, D2, ...
```

где HC — номер оператора программы, а D1, D2, ... — выводимые на печать данные. Вместо D1, D2, ... могут стоять либо имена переменных (как в операторах ввода LINPUT и INPUT), либо строко-

вые или числовые константы. Такой строковой константой может быть, например, фраза «ГЛАГОЛ УПОТРЕБЛЕН В НЕОПРЕДЕЛЕННОЙ ФОРМЕ». Итак, если необходимо вывести на принтер проанализированный глагол *ходить*, который находится, например, в области памяти с именем *W\$*, и результат проведенного анализа в виде упомянутой выше фразы, то для этого можно использовать следующую команду:

```
PRINT W$; " — ГЛАГОЛ УПОТРЕБЛЕН В НЕОПРЕДЕЛЕННОЙ ФОРМЕ"
```

При выполнении этого оператора на экране дисплея (на бумаге принтера) появится следующая информация:

```
ХОДИТЬ — ГЛАГОЛ УПОТРЕБЛЕН В НЕОПРЕДЕЛЕННОЙ ФОРМЕ
```

Если сравнить данный оператор с общим видом оператора PRINT, то можно заметить, что вместо *D1* в операторе указана переменная *W\$*, а вместо *D2* — приведенная выше фраза. Как видно из результата выполнения оператора вывода, между словом *ходить* и фразой на экране дисплея поставлен один знак пробела. Если необходимо, чтобы между выводимыми данными было более одного пробела, то между *D1*, *D2*, ... в операторе PRINT надо ставить не точку с запятой, а запятую. Эти знаки называются разделителями информации.

Рассмотрим далее некоторые из **специальных операторов** языка QBASIC.

1. Оператор LEN (часть английского слова *length* — ‘длина’). Он служит для определения числа знаков (букв и т.д.) в строковых данных. В общем виде этот оператор записывается так:

```
HC I=LEN (K)
```

где HC — номер оператора программы; *I* — имя переменной, в которую будет передан результат работы оператора LEN, т.е. число знаков (букв, цифр и т.п.), находящихся в обрабатываемом данном *K*. Вместо *K* может стоять имя переменной или любая строковая константа. Последняя заключается в кавычки или апострофы. Например, если необходимо узнать, сколько букв в слове *ходить*, достаточно написать следующий оператор:

```
I=LEN ("ХОДИТЬ")
```

После выполнения этого оператора в переменной *I* будет число 6 (6 букв в слове *ходить*). Вместо *K* в операторе LEN может стоять переменная, в которой уже находится какое-то слово. Так, если после ввода слово *ходить* оказалось в области памяти с именем *W\$*, то в этом случае для подсчета числа букв надо написать оператор

```
I=LEN (W$)
```

Итог будет тот же: $I=6$.

2. Оператор `INSTR` дает возможность узнать, в каком месте слова, словосочетания, предложения или текста находится некоторая его часть (буква, суффикс, приставка, основа слова, слово и т. п.). Общий вид этого оператора таков:

$$\text{HC } I = \text{INSTR } (n, W\$, B\$)$$

где `HC` — номер оператора программы; `I` — имя переменной, в которую передается результат работы этого оператора, т. е. номер первого найденного символа (буквы, цифры, знака) строки символов, находящейся в переменной с именем `W$`; `n` — номер позиции начала поиска в строковой переменной `W$` (`n` принимает значения 1, 2, 3, 4 и т. д.); `B$` — имя переменной, содержащей искомую часть заданной строки символов (буква, суффикс и т. д.). Например, если в переменной `W$` находится слово *ходить* (`W$="ХОДИТЬ"`) и необходимо узнать, в каком месте этого слова находится окончание *-ть* (`B$="ТЬ"`), то это действие на языке `QBASIC` записывается следующим образом:

$$I = \text{INSTR } (W\$, B\$)$$

Поскольку в операторе отсутствует параметр `n`, то компьютер будет проводить поиск с начала слова *ходить* и в итоге в переменной `I` окажется число 5 (так как начало окончания *-ть* начинается с 5-й позиции).

3. Оператор `LEFT$` позволяет выделить в строковом данном (слове, словосочетании, предложении и т. п.) несколько левых символов (букв, цифр, знаков). Например, результатом действия

$$B\$ = \text{LEFT\$ } (W\$, n)$$

при условии, что `W$="ПРИЕХАТЬ"`, а `n=3`, будет помещено в переменную `B$` приставки *при-* (`B$="ПРИ"`).

4. Оператор `RIGHT$`, общий вид которого выглядит следующим образом:

$$\text{HC } B\$ = \text{RIGHT\$ } (W\$, n)$$

позволяет выделить в строке символов, содержащейся в переменной с именем `W$`, определенное число (`n`) правых символов и направить их в переменную с именем `B$`. Например, если в слове *ходить*, расположенном в переменной `W$` (`W$="ХОДИТЬ"`), необходимо выделить две последние (правые) буквы и поместить их в область памяти с именем `B$`, то такое действие будет записано на языке `QBASIC` в виде команды

$$B\$ = \text{RIGHT\$ } (W\$, 2)$$

5. Оператор `MID$` (часть английского слова *middle* — ‘середина’) записывается в следующем виде:

$$\text{HC } B\$ = \text{MID\$ } (W\$, n, k)$$

Он позволяет выделить в строковом данном, находящемся в переменной с именем $W\$$, начиная с символа под номером n , K последующих символов. Результат такого выделения записывается в область памяти с именем $B\$$. Например, в переменной $W\$$ находится слово *приходила* ($W\$="ПРИХОДИЛА"$). Необходимо выделить его часть — *-ход-*. В данном случае оператор $MID\$$ будет иметь вид

$$B\$=MID\$(W\$, 4, 3)$$

т. е. компьютер поместит в переменную $B\$$ строку из трех последовательных символов, выделение которых из слова *приходила* начнется с 4-й позиции ($B\$="ХОД"$).

ОБЩИЕ ПРИНЦИПЫ РЕШЕНИЯ ЛИНГВИСТИЧЕСКИХ ЗАДАЧ МЕТОДОМ МОДЕЛИРОВАНИЯ

ОСНОВНЫЕ ЭТАПЫ РЕШЕНИЯ ЗАДАЧИ

Как было отмечено выше, основным методом решения лингвистических задач является метод **моделирования**. Моделирование — процесс творческий. И модель в любом случае отражает наиболее важные и существенные особенности моделируемого объекта, процесса или явления (оригинала модели).

В общем, процесс моделирования на компьютере включает следующие этапы [165, 24—34]:

- 1) постановка задачи;
- 2) разработка модели;
- 3) проведение компьютерного эксперимента;
- 4) анализ результатов работы компьютерной модели.

В свою очередь, каждый этап моделирования может быть представлен в виде некоторой последовательности конкретных действий.

Говоря о *постановке задачи*, выделяют следующие действия:

- 1) описание решаемой задачи;
- 2) формулирование цели моделирования;
- 3) анализ оригинала модели.

Описание решаемой задачи может быть представлено словесно, в виде формулы или нескольких формул, в виде таблиц, графиков и т. п.

Цели моделирования могут быть различными [165, 23]: объяснение сути некоторого явления или процесса, создание объектов с заранее заданными свойствами, определение последствий воздействия на некоторый объект, процесс или явление, принятие правильного решения и т. п.

При анализе оригинала модели (некоторого объекта, процесса или явления) в нем прежде всего выделяются наиболее важные, существенные черты и свойства. Если оригинал представляет собой совокупность более мелких составляющих, то на этапе анализа его расчлениают, разделяют на ряд более мелких объектов и выявляют отношения между этими составляющими. При компьютерном моделировании стараются выделять такие черты и свойства оригинала, которые мог бы легко опознать компьютер. Эти черты и свойства называют **формальными**.

На этапе *непосредственной разработки модели*, опираясь на результаты детального анализа ее оригинала, создается алгоритм решения задачи.

Проведение компьютерного эксперимента связано с созданием на основе алгоритма компьютерной программы на каком-либо алгоритмическом языке и отладкой этой программы (устранением ошибок программирования).

Наконец, в процессе *анализа результатов работы* компьютерной модели выявляются логические ошибки в самой компьютерной программе и алгоритме (формуле, графике и т. п.), который послужил основой компьютерной модели. Такие ошибки исправляются внесением в алгоритм и программу соответствующих изменений.

Рассмотрим подробнее эти этапы и соответствующие им действия в процессе создания конкретных воспроизводящих инженерно-лингвистических моделей (ВИЛМ) [162, 25—40]. Поскольку решение лингвистических задач на ПК имеет свою специфику, введем некоторые основные понятия, связанные с этой процедурой. Все модели, создаваемые ниже, будут понятны, если принять единые определения основных лингвистических единиц.

Информация поступает в компьютерную память в виде цепочки символов, каждый из которых занимает один байт памяти. Цепочка буквенных символов, находящаяся в тексте между двумя знаками пробела, называется *словоупотреблением*. Следовательно, компьютер читает не слово, а словоупотребление. Словоупотребление, находящееся вне предложения или текста, будем называть *словоформой*. Несколько словоформ, имеющих одно и то же лексическое значение, образуют *слово*, или *лексему*. Например, в тексте¹

СКОРО_ПРИДЕТ_ВЕСНА_. _ВЕСНОЙ_ЛЕГЧЕ_ДЫШИТСЯ_.
ПРИХОДИ_, ВЕСНА_!

8 словоупотреблений. Если бы из единиц этого текста составлялся алфавитно-частотный словарь, то в нем оказалось бы 7 словоформ. В таком словаре словоформы располагаются по алфавиту и при каждой словоформе указывается абсолютная частота (F) употребления словоформы (как сумма зафиксированных в тексте соответствующих словоупотреблений):

Словоформа	F
1. ВЕСНА	2
2. ВЕСНОЙ	1
3. ДЫШИТСЯ	1

¹ При вводе текстов в память компьютера все знаки препинания отделяются от слов знаком пробела (_).

4. ЛЕГЧЕ	1
5. ПРИДЕТ	1
6. ПРИХОДИ	1
7. СКОРО	1

Как видно, два словоупотребления *весна* исходного текста (предложения) преобразуются вне его в одну словоформу *весна*.

Если составить из этого же текста алфавитно-частотный словарь слов, то он будет включать 5 слов:

Слово	<i>F</i>
1. ВЕСНА	3
2. ДЫШАТЬСЯ	1
3. ЛЕГКО	1
4. ПРИХОДИТЬ	2
5. СКОРО	1

Здесь словоформы *весна* и *весной* относятся к одному словарному слову *весна*, а глаголы *придет* и *приходи* являются словоформами лексемы *приходить*. Таким образом, если, например, в лингвистической задаче сказано, что в предложении необходимо найти слово *машина*, это означает, что искать в нем надо все словоформы, относящиеся к этому слову: *машина*, *машины*, *машине*, *машину*, *машиной*, *машин*, *машинам*, *машинах*¹ (или, если это возможно, общую часть таких слов. В данном примере это возможно, и потому для компьютерного поиска задается слово *машин*).

Предложением с компьютерной точки зрения называется цепочка словоупотреблений между двумя знаками конца предложения (точкой, вопросительным знаком, восклицательным знаком, многоточием). Пример такого предложения приведен выше. **Текст** в компьютере — это линейная последовательность подобных предложений.

Проиллюстрируем этапы построения инженерно-лингвистической модели на одном простейшем примере.

Постановка задачи.

Описание задачи: задана группа из 10 русских глаголов, среди которых есть глаголы в инфинитиве без частицы *-ся* или *-сь*. Необходимо найти в этой группе глаголы в инфинитиве и напечатать их.

Формулирование цели процесса моделирования: создать такую модель опознавания инфинитивной формы русского глагола, ко-

¹ Вместо 12 словоформ (6 форм в единственном числе и 6 форм во множественном) здесь приведены 9 форм, так как форма *машины* совпадает в род. падеже ед. числа, им. падеже мн. числа и вин. падеже мн. числа. Форма *машине* одинакова для дат. падежа ед. числа и предл. падежа ед. числа.

торая объясняла бы процесс выделения подобных форм и давала бы верные результаты для любого исходного количества глаголов, удовлетворяющих заданному описанию.

Решению каждой лингвистической задачи должен предшествовать тщательный анализ соответствующего лингвистического материала: конкретных букв (звуков), слов, словосочетаний, предложений, абзацев, текстов. Для поиска наиболее важных, существенных признаков, положенных в основу алгоритма решения сформулированной выше задачи, необходимо изучить большое число русских глаголов в самых разных грамматических формах и попытаться найти какие-то черты, которые отличают глагол в неопределенной форме от глаголов в других грамматических формах. Причем эти признаки должны быть формальными, т. е. любой человек, даже не знающий русского языка, ориентируясь на них, сможет решить поставленную задачу.

Анализируя, например, слова *неть, беречь, косила, переделал, работаю, идти, пошел, толочь, бегать, пройду, читать, нести, прочитаю, ходить, сделаю, хранить, уехать, стеречь, подарю, берегу, играть* и т. п., можно сделать вывод, что русский глагол в инфинитиве заканчивается буквосочетаниями *-ть, -чь, -ти*¹. Это и есть те формальные признаки, на которые будет опираться модель.

Разработка модели.

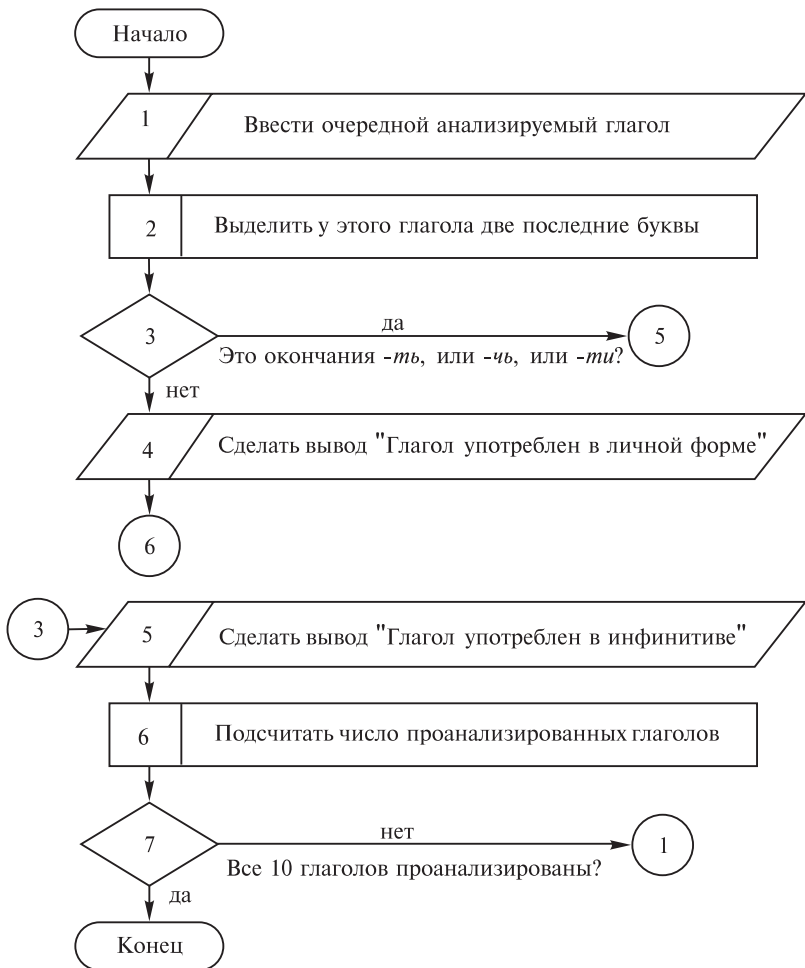
Как было сказано выше, любая воспроизводящая инженерно-лингвистическая модель представляется в виде словесно-графического алгоритма, опирающегося на выделенные в процессе анализа оригинала модели признаки. Как же отыскать в группе из 10 русских глаголов глаголы в неопределенной форме? Очевидно, необходимо выполнить следующую цепочку «умственных» действий:

1. Взять очередной анализируемый глагол.
2. Выделить у этого глагола две последние буквы.
3. Посмотреть, совпадают ли эти буквы с буквосочетаниями *-ть, -чь, -ти*. Если да, то перейти к действию 4; если нет, то выполнить действие 6.
4. Сделать вывод: «Данный глагол употреблен в инфинитиве».
5. Перейти к действию 7.
6. Сделать вывод: «Данный глагол употреблен в личной форме».
7. Подсчитать число проанализированных глаголов.
8. Проверить, все ли 10 глаголов проанализированы. Если нет, то перейти к выполнению действия 1; если да, то перейти к действию 9.
9. Закончить работу.

¹ К такому же выводу можно прийти, если посмотреть соответствующие разделы русской грамматики [194, 108, 109].

Так построен словесный алгоритм задачи. В словесно-графической форме он представляется следующей блок-схемой (схема 2).

С х е м а 2



При таком изображении алгоритма решения задачи все его шаги (действия) представляются в виде пронумерованных блоков. В кружках указаны номера блоков, к которым передается управление (см. блоки 1, 5, 6), или же номера блоков, от которых пришло указание на выполнение действия (см. блок 3).

Проведение компьютерного эксперимента.

На данном этапе на конкретном алгоритмическом языке должна быть написана программа решения задачи. Этот этап сопровождается выполнением ряда следующих действий.

1. Путем анализа построенного алгоритма выделяют основные переменные (области памяти компьютера, ср. с. 24, 25). В данном случае ими будут:

- а) место в памяти компьютера для размещения каждого анализируемого глагола;
- б) место в памяти компьютера для размещения двух последних букв глагола;
- в) место в памяти компьютера (счетчик) для размещения числа проанализированных глаголов.

2. В зависимости от конкретного алгоритмического языка этим переменным присваивают определенные имена. Если использовать для программирования алгоритмический язык QBASIC, то выделенным выше переменным можно дать следующие имена:

W\$ — строковая переменная для размещения анализируемого глагола;

B\$ — строковая переменная для размещения двух последних букв глагола;

I — числовая переменная (счетчик) для размещения числа проанализированных глаголов.

3. Следующий далее процесс написания компьютерной программы заключается в замене блоков алгоритма (см. схему 2) одним или несколькими операторами какого-либо алгоритмического языка. Заменяя блоки операторами языка QBASIC, получаем следующую программу, моделирующую действия человека при определении им инфинитивной формы русского глагола (табл. 5).

Т а б л и ц а 5

Номер блока алгоритма	Номер оператора программы	Оператор языка QBASIC	Пояснения
1	10	INPUT "ВВЕДИТЕ ОЧЕРЕДНОЙ РУССКИЙ ГЛАГОЛ", W\$	см. с. 28
2		B\$=RIGHT\$(W\$, 2)	см. с. 30, 31
3		IF B\$="ТЬ" OR B\$="ЧЬ" OR B\$="ТИ"	см. с. 26, 27
		GOTO 50	
4		PRINT W\$; "— ГЛАГОЛ УПОТРЕБЛЕН В ЛИЧНОЙ ФОРМЕ"	см. с. 28, 29
		GOTO 60	см. с. 26, 27
5	50	PRINT W\$; "— ГЛАГОЛ УПОТРЕБЛЕН В ИНФИНИТИВЕ"	см. с. 28, 29
6	60	I=I+1	см. с. 25, 26
7		IF I<=10 THEN 10	см. с. 26, 27
		END	см. с. 27

4. Отладку программы, т.е. устранение в ней различных ошибок программирования, выполняет программист, хорошо знающий соответствующий алгоритмический язык.

Анализ результатов работы компьютерной программы.

Для проверки универсальности созданной ВИЛМ на вход компьютерной программы подают разное количество различных русских глаголов. Во всех случаях программа должна давать правильный результат. Если относительно какого-то глагола компьютер дал неверный ответ, то это может быть следствием следующих причин:

- 1) неверно построен алгоритм;
- 2) сделана ошибка при программировании;
- 3) учтены не все формальные признаки, определяющие русский глагол в инфинитиве.

В 1-м и 3-м случае вводятся изменения и дополнения в алгоритм, во 2-м случае исправляются ошибки программирования.

Моделирование процесса определения лексико-грамматического значения слова *zu* в немецком предложении

Задачи подобного рода часто встречаются в процессах автоматического анализа текстов и при их переводе с одного языка на другой. В обоих случаях среди нескольких лексических или грамматических значений одного и того же слова приходится выбирать одно единственное, специфичное для данного текста. Как правило, это делается методом *контекстологического анализа* омонимичного или полисемичного (многозначного) слова. Этот метод хорошо описан в работах [64; 124, 138—160; 167], и суть его будет ясна в ходе решения данной задачи.

Строго не придерживаясь изложенной выше процедуры решения лингвистических задач, но сохраняя ее суть, представим процесс решения этой задачи следующей последовательностью действий.

Постановка задачи.

Описание задачи: в память компьютера с клавиатуры поочередно вводятся любые немецкие предложения. Необходимо составить алгоритм и программу, позволяющие определить, в какой функции (и с каким лексическим значением) в каждом немецком предложении употреблено слово *zu* (если, конечно, оно есть в предложении).

Анализ оригинала модели: изучение употреблений слова *zu* в немецких предложениях [60, 350—354], а также данные немецких грамматик (например, [146, 357—378]) — свидетельствует о том,

что слово *zu* употребляется в следующих пяти функциях: 1) предлог; 2) отделяемая приставка; 3) частица при инфинитиве; 4) в конструкции с причастием I; 5) усилительная частица. Более детальный анализ немецких предложений позволяет построить следующую таблицу диагностирующих признаков (табл. 6), однозначно определяющих функцию слова *zu* и его возможный перевод на русский язык.

Т а б л и ц а 6

Таблица диагностирующих признаков для распознавания функции слова *zu*

Функция в предложении	Формальные диагностирующие признаки				
	Левое окружение слова		Перевод слова <i>zu</i>	Правое окружение слова	
	Слова	Признаки		Слова	Признаки
Предлог	—	—	<i>к, в</i>	<i>dem, den, der, diesem, dieser, dessen, einem, einer, jeder, jeden, deren, denen</i>	Признак имени существительного в немецком языке — две одинаковые первые буквы
	<i>bis</i>		<i>вплоть до</i>	—	
Отделяемая приставка	—	—	—	<i>und, oder, als</i>	Знаки препинания ";", ". ", ",", "
Частица при инфинитиве	<i>um</i>	—	<i>чтобы</i>	<i>sein, seinem, tun</i>	Окончания инфинитива: <i>-en, -rn, -ln</i>
	<i>ohne</i>	—	<i>без того, чтобы</i>	—	
В конструкции с причастием I	—	—	<i>который, следует</i>	—	Суффиксы причастия I и окончания имен существительных: <i>-ende, -nden, -lnde, -nder, -ndes, -rnde</i>
Усилительная частица	Формальные выделители не обнаружены		—	Формальные выделители не обнаружены	