Содержание

Предисловие

Эта книга – рассказ об информатике словами тех, кто ее создавал. Появилась она по двум причинам. Во-первых, чтобы избавить читателей, живущих в XXI веке, от иллюзии, будто сложившиеся в этой области соглашения были дарованы современной культуре в готовом виде. Информатика имеет богатую семейную историю, которую следует знать студентам и всем пашущим на этой ниве. А во-вторых, чтобы помочь читателям разглядеть, как появлялись на свет важнейшие идеи, как робкие, неуклюжие шажки постепенно превращались в твердую и уверенную поступь – а иногда в течение многих лет никуда не приводили, чтобы возобновить движение после длительного перерыва. Информатика – все еще юная и динамичная наука; всякий осматривающийся в ней сегодня может разглядеть какое-то новшество, которое завтра станет каноном, но пока находится в зачаточном состоянии и едва различимо.

Чтобы рассказать вам эту историю, я отобрал, трепеща в душе, 46 статей, начиная с античных времен и заканчивая 1980 годом, и предпослал каждой краткий очерк с описанием контекста. Каждая из этих статей внесла запоминающийся вклад в свою область. Можно было бы отобрать много других работ в дополнение к этим или вместо них, да и дата отсечения, 1980 год, выбрана достаточно произвольно, хотя она все же представляет тот момент, после которого информатика стала настолько разветвленной, что попытка составить такой небольшой сборник заведомо потерпела бы неудачу.

Эта книга написана только в образовательных целях, чтобы документировать истоки научной области; она не является ни критическим переизданием статей, ни историей вопроса. Во введении статьи перечисляются в историческом контексте, но для тех читателей, кто жаждет более подробного освещения истории событий со всеми нюансами, рекомендую работу Priestley (2011), автору которой удалось удачно избежать излишнего доверия к своекорыстным воспоминаниям действующих лиц и заблуждений типа «после того значит вследствие того». За сведениями о ранней истории вычислительных машин отсылаю читателя к работам Pratt (1987) и Jones (2016).

Эту книгу можно использовать в качестве основы односеместрового курса для студентов старших курсов и аспирантов, и в таком качестве она действительно использовалась в Гарварде и МТИ. Она также может послужить путеводителем для любопытствующих профессионалов. При отборе и редактировании статей я руководствовался, в частности, следующими принципами.

1. Многие статьи значительно сокращены, чтобы привлечь внимание к основным идеям, опуская технические детали, которые сегодня уже не представляют большого интереса. Например, я опустил далекие от элегантности детали кода универсальной машины Тьюринга. Опущенный текст обозначается многоточием <...>. Нет нужды говорить, что читатели, жадные до деталей, могут обратиться к полному тексту статей. Каждая глава начинается библиографической отсылкой к статье, хотя указанный в ней год может не совпадать с годом, указанным в заголовке самой статьи, потому что некоторые работы сначала были представлены устно или были переработаны после первой публикации. Кроме того, изобретение или открытие, описываемое в статье, могло иметь место раньше даты публикации.
2. Я отдавал предпочтение коротким и удобным для восприятия статьям перед длинными и трудными, пусть даже более важными.
3. Я включал только статьи, но не отрывки из книг (за исключением «Законов мышления» Буля и очерка, давшего название сборнику Брукса «Мифический человеко-месяц»).
4. Я не старался дать выжимку из технических отчетов, в которых определены такие важнейшие языки программирования, как FORTRAN, COBOL или ALGOL, как бы важны они ни были для развития отрасли.
5. Также опущены ранние усилия по систематизации материала, в частности доклад *Curriculum 68* (Atchison et al., 1968) и история языков программирования в работе Jean Sammet (1972).
6. Количество страниц, включенных в книгу, не может считаться мерой значимости автора. При таком критерии Дональд Кнут был бы недооценен, а Эдсгер Дейкстра – переоценен. С другой стороны, такие великие имена, как Бэкус, Хомский, Чёрч, Флойд, Грэй, Клини, Ньюэлл, Лэмпорт, Лэмпсон, Рабин, Скотт и Тарьян, вообще не упомянуты. Приношу извинения тем, кто не нашел здесь своей любимой статьи или автора.

И еще несколько чисто редакторских замечаний.

- Все ссылки собраны в общей библиографии в конце книги, при их оформлении используется Чикагское руководство по стилю. Документы, которые цитируются в оригинальных статьях, но не в той части текста, которая включена в книгу, в библиографию не включены.
- Опечатки, встречающиеся в оригинальных статьях, исправлены, пунктуация приведена к единому стандарту.
- Нумерация разделов и рисунков единая во всей книге: § 33.7 – это раздел 7 (или, возможно, седьмой ненумерованный раздел) оригинальной статьи, которая здесь напечатана в главе 33.
- Оригинальная нумерация теорем и лемм сохранена.
- Сноски опущены или включены прямо в текст в скобках.
- В нескольких местах я добавил примечания редактора, оформленные в виде «[Примечание редактора: комментарий]».

Я выражаю благодарность слушателям курса CS191, прочитанного в 2019 году в Гарвардском университете, и курса 6.S897, прочитанного в 2020 году

в Массачусетском технологическом институте (МТИ), за тщательную корректуру. Особенно остроглазыми проявили себя Брайан Сапожников и Адхем Мегид. Спасибо также Питеру Дэннингу, Биллу Газарху, Уоррену Гольдфарбу, Мэттью Лена, Марианте Маллиарис, Таше Шенстейн, Ллойду Стриклэнду, Шерри Тэркл и Джоэлю Вахману за полезные замечания и поправки. Разумеется, за все оставшиеся ошибки ответственность несу только я.

Гарри Льюис
Июль 2020

Введение: корни и рост информатики

Все начиналось с чисел; инструментами первых бухгалтеров были кучки *calculi* (камушков). Более поздние нотационные и механические изобретения были придуманы астрономами, а также военными инженерами и мореплавателями. Нужда в вычислениях возрастала по мере того, как люди учились понимать физический мир и управлять им, особенно в эпоху Просвещения и во время последующих войн.

Но интеллектуальные истоки информатики не сводятся только к бухгалтерии, астрономии и баллистике. Информатика – дитя логики, математики и человеческого воображения. В силу столь разнородной интеллектуальной родословной и лишь косвенной связи с миром природы эта область знаний с трудом отвоевывала свое право на существование на протяжении большей части XX столетия. К тому времени компьютерные вычисления проникли во все отрасли науки, техники и экономики, а равно в математику, и стихли семантические войны на тему того, правомерно ли называть наукой «компьютерную науку» или какую-то другую «науку об искусственном» (Simon, 1996). То, что эта область знаний была признана наукой, – заслуга первопроходцев XX века, разработавших первые курсы и учебные программы для колледжей и университетов, часто несмотря на сопротивление со стороны математических и инженерных факультетов, от которых отпочковались. Мы не можем рассказать здесь их историю, но будет справедливо отметить, что это изложение становления информатики могло бы выглядеть совершенно иначе, если бы основоположники системы образования организовали предмет по-другому.

Логика таилась в ветвях генеалогического древа еще со времен античности, вступая лишь в неловкие эпизодические контакты с вычислением. Так продолжалось до тех пор, пока в середине XIX века она не стала инструментом метаматематики. Алгоритмы превратились из абстрактного в конкретное в первые десятилетия XX века, когда были включены в программу метаматематики, имеющую целью установить, какая математика может быть признана истинной. А научные и коммерческие расчеты, долгое время бывшие движущей силой ряда последовательных улучшений механического вычислительного устройства (калькулятора), получили мощный толчок со стороны физики и техники во время Второй мировой войны.

На протяжении всего пути в серьезной работе нет-нет да и мелькала нескромная идея: а не может ли человек вдохнуть жизнь в создаваемые им машины? По мере становления информатики это мифическое видение закручивалось тугим водоворотом вокруг математики вычислений, предвещая безудержность технологической сингулярности. Качество машинного зрения, синтетической речи и ловкости роботов постоянно улучшается, порождая дебаты о том, какие последствия это сулит отдельным людям и обществу в целом.

Честно говоря, дата рождения информатики выбрана произвольно. Мы относим к доисторическим временам такие работы, как «Арифметика» Диофанта, написанная в III веке (Diophantus, 1910) и «Алгебра» аль-Хорезми, датируемая IX веком (al Khwarizmi, 1915), какими бы выдающимися интеллектуальными качествами они ни обладали. (Диофант мелькнет в главе 5 этой книги, поскольку изучал задачи теории чисел, которые после обобщения оказываются рекурсивно неразрешимыми.)

Но начать нужно с Аристотеля, заложившего понятие о высказываниях, которые могут включать переменные, представляющие свойства (глава 1). Такое высказывание может быть истинным при любых значениях переменных, может быть истинным только иногда или вообще никогда. Этот логический анализ имеет прямое отношение к информатике. Что делает цифровой компьютер «универсальным»? Тот факт, что одни и те же двоичные логические элементы могут означать разные вещи в разные моменты времени. Регистр памяти может содержать время суток, когда используется в одной программе, или адрес при использовании в другой программе. Аристотелю мы обязаны пониманием того, что одни и те же правила логики могут применяться к разным явлениям, – того, что логика дает общую основу для рассуждений.

В начале XVII века Кеплер сформулировал математические законы движения планет, Декарт свел геометрию к алгебре, а Паскаль математически охарактеризовал жидкости. Поскольку наблюдаемые физические явления описывались математическими формулами, известные с античных времен задачи вычисления площадей и объемов криволинейных фигур и тел приобрели важное практическое значение – тем более важное, что с развитием оптики измерения небесных тел становились все более точными. Развитие математики непрерывных величин заложило основу для открытия исчисления бесконечно малых, совершенного почти одновременного Исааком Ньютоном в Англии и Готфридом Лейбницем на Континенте.

Лейбниц был опытным вычислителем. Паскаль изобрел механическую суммирующую машину, наблюдая за работой отца – сборщика налогов. Лейбниц усовершенствовал это устройство, наделив его возможностью выполнять умножение и деление, и тем самым построил одну из первых вычислительных машин с вложенными циклами (см., однако, стр. 95). Лейбниц осознал преимущества двоичной системы счисления и написал об этом задолго до того, как ее полезность оценили другие; даже пионер компьютеростроения Говард Хатауэй Эйкен, который жил и работал на 250 лет позже, с большой неохотой отказался от десятичной арифметики. Но самое главное, что помимо открытия анализа бесконечно малых Лейбниц придумал исчисление идей. В главе 2 описывается лишь одно из многих утопических предложений по

рационализации дел человеческих, и Лейбниц недалеко продвинулся в осуществлении своего плана. Зная, что в XIII веке Раймонд Луллий построил машину для выполнения некоторых силлогистических выводов, Лейбниц замечает, что когда его исчисление примет законченную форму, человечество «будет иметь инструмент, который сослужит возвышенному разуму службу не хуже той, что Телескоп сослужил совершенному зрению». Логические атомы, которыми оперирует его исчисление, вновь возникают двумя столетиями позже в работе Джорджа Буля (см. ниже) и становятся базовыми фактами в таких языках логического программирования, как Prolog и Datalog.

Конструкция Аналитической машины Чарльза Бэббиджа (глава 3) могла бы ознаменовать наступление века компьютеров, поскольку устройство должно было быть программируемым и адаптируемым. Но Бэббидж не смог ее построить. У него постоянно не хватало денег, несмотря на заявления о том, что машина почти готова, – в 1835 году Бэббидж (Babbage 1989, стр. 245) писал, что «самые серьезные трудности уже преодолены и планы будут исполнены в течение нескольких месяцев», – и ее важность для национальной обороны: «управление с помощью Аналитической машины всеми астрономическими вопросами, от которых так сильно зависит флот, едва ли оставит Ее Величество равнодушной к предмету», писал он принцу Альберту (Babbage, 1843). Тем не менее ученица Бэббиджа Ада Лавлейс, рассуждая об этой так и не заработавшей машине, сумела предвосхитить самые разные концепции современного программирования.

У Джорджа Буля (George Boole, 1854, здесь глава 4) была иная повестка. Он был на 25 лет младше Бэббиджа и не получил университетского образования, но их пути, по крайней мере однажды, пересеклись. Однако у Буля была своя идея: актуализировать Аристотеля, выразить правила человеческого мышления в математической форме. Его работа по логике была слишком новаторской и выбивалась из общего русла. Ее влияние на вычисления было ограниченным – до 1930-х годов, когда Клод Шеннон положил ее в основу проектирования цифровых схем.

К началу XX века логики использовали методы самой математики для математизации идеи доказательства и стремились довести до логического завершения план Лейбница. Величайшая проблема Гильберта, проблема разрешения (Entscheidungsproblem) – определить, возможно ли доказать формализованные математические утверждения, – была строго поставлена лишь спустя несколько лет после его знаменитого обращения к Международному конгрессу математиков 1900 года (мы включили отрывки из него в главу 5). Но «Математические проблемы», с одной стороны, предвосхищают механизированную логику в своем обращении к финитным методам, а с другой – разделяют оптимизм Лейбница в том смысле, что если математики всего мира будут трудиться достаточно усердно, то они приведут свой дом в порядок.

Этого не случилось, по крайней мере не так, как представлял себе Гильберт. Сначала Гёдель, потом Чёрч, а потом еще и Тьюринг (Turing 1936, здесь глава 6) открыли мир метаматематики, который даже вообразить себе было невозможно до XX века. И хотя каждый из них оказал значительное влияние на информатику, вклад Тьюринга оказался наиболее важен, потому что (а) была

убедительно формализована идея вычислительной машины, а стало быть, и идея о том, что можно доказывать утверждения о ней; (б) был убедительно сформулирован принцип вычислительной универсальности и показано, что его можно воплотить в жизнь с помощью устройств, состоящих из простых частей; (в) сочетание (а) и (б) с неопровержимой логикой доказывало, что у вычислимости есть пределы.

Отрицательное решение гильбертовой Entscheidungsproblem стало кульминацией его статьи. В середине доказательства был использован важный технический прием – устройство хранения данных, в котором автомат мог хранить программы для других автоматов. Для завершения доказательства Тьюринг прибег к диагональному методу, предложенному в 1891 году Кантором – для совершенно другой цели, доказательства несчетности множества действительных чисел (Cantor 1996). Спустя десять лет Эккерт и Мочли воспользовались аналогичной идеей хранимой программы при модификации своего компьютера ENIAC, хотя сделали это из чисто практических соображений, а не под влиянием теоретической работы Тьюринга – хранение программы в памяти на электровакуумных лампах ускоряло работу машины и упрощало изменение программы (Priestley 2011, стр. 125). Когда Джон фон Нейман в 1945 году вошел вместе с Эккертом и Мочли в группу по проектированию машины EDVAC, пришедшей на смену ENIAC, он взял на себя труд по составлению пояснительной записки (глава 10), и с тех пор конструкция машины с хранимой программой известна (не вполне заслуженно) под названием «архитектуры фон Неймана». Та же идея примерно в то же время была использована в Манчестерской малой экспериментальной машине (Baby), а чуть позже в проекте ACE Тьюринга. В своем предложении Тьюринг (Turing 1945, стр. 3) цитирует отчет об EDVAC, но идея хранимой программы, по-видимому, из тех, что почти одновременно рождаются в головах разных людей, когда для этого приспело время.

Но вернемся в 1930-е годы. Когда Тьюринг переезжал в Принстон, чтобы продолжить свои исследования по логике, прикладной математик Говард Хатауэй Эйкен (Aiken et al., 1964, здесь глава 7) в Гарварде работал над старой проблемой численных расчетов, в которой не было ощутимого продвижения со времен Лейбница. Эйкен спроектировал громоздкий электромеханический компьютер Mark I для печати таблиц математических функций – некоторые из них, например функции Бесселя, тянулись из математического анализа в традициях XVIII века, тогда как другие, например баллистические траектории, требовались для нужд современного вооружения. Его проект закончился ссорой с компанией IBM, которая финансировала создание машины; вопрос стоял почти так же, как в случае с Бэббиджем: «Принадлежит ли честь и слава конструктору компьютера или людям, которые его построили и запустили»?

Эйкен был не единственным, кто размышлял об автоматических вычислениях в конце 1930-х годов. Конрад Цузе работал над собственными электромеханическими калькуляторами в Берлине, а Джон Винсент Атанасов разрабатывал электронную машину в Айове. Все они трудились независимо, хотя Эккерт и Мочли, работавшие по контракту с армией США в Пенсильванском университете, конечно же, знали о работе Атанасова, что впоследствии привело к ожесточенному патентному спору.

В то время как компьютеры еще только вылуплялись из яйца в нескольких разных местах, работы по телефонии повсюду шли полным ходом. Существовало несколько способов соединить коммутаторы проводами, получив один и тот же функциональный результат, и изворотливые инженеры освоили искусство минимизации необходимого оборудования. Начав работать над этими проблемами в бытность свою аспирантом МТИ, Клод Шеннон (Claude Shannon 1938, здесь глава 8) понял, что законы мышления Буля, о которых он узнал на курсе философии, являются также законами построения электрических схем. Если перевести схему на язык булевой логики, то полученную логическую формулу можно будет упростить, а затем вернуться к языку схем, получив более экономичную конструкцию. Эти методы оказались необычайно важны для проектирования цифровых компьютеров, они используются и по сей день.

У двоичной системы счисления много преимуществ с точки зрения электротехники. Мало того что булеву логику можно использовать для упрощения сложных выражений, так она еще упрощает восстановление истинного значения ослабленного сигнала в случае, когда возможных значений всего два: один уровень напряжения представляет 0, а другой 1. Когда в 1940-х годах двоичная система прочно заняла свое место, механизация логики ускорилась. Группа Эккерта, Мочли и фон Неймана воспользовалась двоичным представлением при проектировании компьютера EDVAC (von Neumann 1993, здесь глава 10), именно ей принадлежит честь раннего анализа алгоритмов двоичной арифметики. Шеннон опубликовал вторую основополагающую работу, увязав технику связи с двоичным представлением данных и попутно определив «бит» (Shannon 1948, здесь глава 12). А Хэмминг (Hamming 1950, здесь глава 13) предложил общий метод включения дополнительных битов в двоичные данные, так чтобы ошибки в процессе передачи данных можно было обнаруживать и при определенных условиях исправлять.

Все эти несомненно важные разработки происходили постепенно. У архитектуры с хранимой программой было много прародителей, и Шеннон отдавал должное Найквисту и другим инженерам-связистам своего времени. Работа нейрофизиолога Уоррен Мак-Каллока и логика-самоучки Уолтера Питтса (McCulloch and Pitts 1943, здесь глава 9) совершенно иного рода. Ничего подобного ранее не публиковалось. Литература Возрождения уподобляла человеческое тело набору рычагов, а в XIX веке эту аналогию расширили, связав потребление телом энергии с паровой машиной. Идея о том, что мышление есть форма вычисления, появилась еще до Лейбница и восходит по крайней мере к Томасу Гоббсу (Hobbes 1655, стр. 2): «под “рассуждением” я понимаю вычисление» (*per ratiocinationem autem intelligo computationem*). Но объяснение самого мозга как особого вида механизма было чем-то совершенно новым, а еще более смело выглядела попытка связать возбуждение нейронов по принципу «все или ничего» с коммутацией электрических схем и тем самым развить логическое исчисление Уайтхеда и Рассела (Whitehead and Russell 1910). Мак-Каллок и Питтс не считали эту аналогию просто игрой ума, они заявили на весь мир, что отныне тайны человеческого разума разрешены – осталось лишь несколько деталей, которые будут уточнены позже. Их работа стала единственной, цитируемой в первой редакции отчета фон Неймана по

конструированию компьютера (глава 10). Эволюционируя, нервные сети превратились в важную модель вычислений, хотя большая часть деталей работы Мак-Каллока и Питтса заменена другими идеями. Принципиально важным переосмыслением нейронной модели с приближением ее к реальности стала статья Фрэнка Розенблатта (Rosenblatt 1958a, здесь глава 18) о перцептронах, хотя нервные сети и по сей день занимают свое место в теоретической информатике, далеко уйдя от своих нейроанатомических истоков.

Двойственная идея о том, что машина может действовать как человек, имеет глубокие корни в мифологии. Гомер описывал механических слуг божественного кузнеца Гефеста:

> Сильно хромая, шатаясь на слабых ногах. Отодвинул
> В сторону, прочь от горнила, меха ...
> ... и с толстою палкой, хромая,
> Двинулся к двери. Навстречу ему золотые служанки
> Вмиг подбежали, подобные девам живым, у которых
> Разум в груди заключен, и голос, и сила, – которых
> Самым различным трудам обучили бессмертные боги.
> Под руки взяли владыку служанки...
>
> *Гомер*, Илиада, песнь 18

Уже в VIII веке до н. э. в этом отрывке описано несколько разделов искусственного интеллекта: общий интеллект, обучение, речь, подвижность. Есть мнение, что Буль обсуждал «думающую машину» Бэббиджа в ходе их единственной известной встречи в 1862 году, хотя леди Лавлейс предостерегала его от надежд на то, что Аналитическая машина будет способна на оригинальные мысли. Но кошку уже выпустили из мешка; вскоре после того Сэмюель Батлер в статье «Дарвин среди машин» предвидел, что машины могут эволюционировать и стать разумными (Butler 1863, оригинальная публикация анонимна).

К концу 1940-х годов Тьюринг уже смог объединить все, что ему было известно о компьютерах (не только теоретически – он конструировал и строил вычислительные машины), с дошедшими из глубины веков рассуждениями о думающих машинах и современной ему британской аналитической философией. В результате на свет появилась статья «Вычислительные машины и разум» (Turing, 1950, здесь глава 14), в которой Тьюринг представил себе – в качестве умозрительной альтернативы метафизическому думающему компьютеру – машину, которая сможет успешно обмануть исследователя, заставив его поверить, что он ведет диалог не с машиной, а с человеком. Бесстрастный разбор контраргументов (в том числе высказанных леди Лавлейс) дал начало оживленным спорам и ответным статьям.

В 1964 году Джозеф Вейценбаум (Weizenbaum 1966, здесь глава 27) написал простенькую программу, которая поддерживала бессодержательный диалог с человеком, просто подставляя употребленные человеком слова в ответы компьютера на синтаксически правильные позиции. Вейценбаум рассматривал эту программу, которую назвал Элиза, как техническую демонстрацию обработки естественного языка, пусть и ограниченную. Она изначально не

была предназначена для ведения осмысленного диалога, но неуемное желание людей общаться с ней так, как они общались бы с людьми, породило важные этические вопросы о взаимодействии человека и машины. Норберт Винер (Wiener 1960, здесь глава 19) уже призывал профессиональных компьютерщиков задуматься об этических последствиях своей работы, когда размышлял о последствиях автоматизации и умениях игровых компьютеров – иначе говоря, о последствиях бездумного препоручения компьютерам решений, которые следовало бы оставить за человеком.

1950-е годы стали чем-то вроде кембрийской эпохи в конструировании компьютеров. Подробная конструкторская документация по EDVAC открыла дорогу всевозможным вариациям и улучшениям, а также экспериментам с новыми элементами памяти и переключателями – как в стенах академических учреждений, так и в коммерческом секторе. Изобретение микрокода Морисом Уилксом в 1952 году (Wilkes 1981, здесь глава 15) включено в качестве примера появления в простой форме идеи, важной для решения конкретной насущной задачи: он просто устал перепаивать компьютерные схемы всякий раз, как в систему команд вносились улучшения. Фантастическое увеличение количества логических компонентов, ставшее возможным в результате перехода на транзисторы, а позже на интегральные схемы, привело в действие закон безумного экспоненциального роста, названный в честь Гордона Мура (Moore 1965, здесь глава 25) и опубликованный в журнальной статье с мультяшным изображением домашнего компьютера – задолго до того, как кто-то мог представить себе, для чего такое устройство могло бы понадобиться.

Грейс Хоппер осознала – еще до того, как это стало очевидно прочим, – что оборудование скоро станет самой дешевой частью вычислительной системы, потому что компьютер покупается один раз, а инвестиции в новое математическое обеспечение могут продолжаться бесконечно (Hopper, 1952, здесь глава 16). Кроме того, она поняла, что языки высокого уровня – не просто удобство, а необходимость, т. к. никакой владелец кодовой базы не может позволить себе переписывать ее с нуля всякий раз при покупке нового компьютера.

Языки Fortran, Algol и Cobol появились в 1950-е годы благодаря Хоппер, которой была ясна важность применения компьютеров в коммерческом секторе. Мы не можем здесь воздать должное ни одному из этих языков программирования и их создателям, за исключением Джона Маккарти, который рискнул адаптировать лямбда-исчисление Алонзо Чёрча, разработанное с целью отправить на покой Entscheidungsproblem, и превратил его в язык функционального программирования (McCarthy 1960, здесь глава 21). По ходу дела он связал логическую традицию непосредственно с искусством программирования: если трактовка вычисления как манипулирования символами – ключ к пониманию пределов вычислимости, то почему бы не сделать манипулирование символами примитивом практического языка программирования? Маккарти также уверенно поставил во главу компьютерного программированиия рекурсивный стиль определения функции, который прочно занял место в метаматематике начала XX века и был систематизирован Розой Петер (Péter 1951).

Влияние Алонзо Чёрча невозможно переоценить. К числу его докторантов в Принстонском университете относятся Тьюринг, Майкл Рабин и Дана Скотт. А Маккарти, который также писал докторскую диссертацию в Принстоне во времена Чёрча, заложил основы искусственного интеллекта, первой системы с разделением времени (Corbató et al. 1962, здесь глава 23), а также символического и функционального программирования.

Конструирование компьютерных систем не только для вычислений, но и в качестве устройств, помогающих человеку в процессах мышления и запоминания, можно возвести к публикации в популярном журнале статьи Ванневара Буша (Bush 1945a, здесь глава 11) «Как мы можем мыслить». Он пытался вообразить, как человеческому мышлению могла бы в будущем помочь технология, но не имел ясного представления о том, что помочь может именно компьютер. В следующем десятилетии начали появляться огромные, громоздкие компьютеры, и провидцы всех мастей стали придумывать, как люди когда-нибудь смогут работать с ними более удобно. Дж. К. Р. Ликлайдер (Licklider 1960, здесь глава 20) представлял себе кооперацию людей и компьютеров, а Дуг Энгельбарт (Engelbart 1962, здесь глава 22) работал над тем, чтобы воплотить эту идею в жизнь. Айвен Сазерленд (Sutherland 1963, здесь глава 24) открыл область машинной графики в своей докторской диссертации в МТИ, в стенах которого Буш грезил о мыслительном помощнике почти двадцатью годами ранее.

Изучение формально-математических абстрактных методов описания и анализа вычислений развивалось во многих направлениях. На протяжении столетий было создано много ручных алгоритмов – одни хуже, другие лучше, – а описания механических программируемых калькуляторов Лавлейс и Эйкена предусматривают некоторые варианты программирования для повышения производительности или точности. Алгоритмы на графах вошли в состав исследования операций в годы Второй мировой войны и выделились в важный раздел информатики. Крускал (Kruskal 1956, здесь глава 17) предложил алгоритм построения минимального остовного дерева, который теперь изучают практически все студенты компьютерных специальностей, а Эдмондс (Edmonds 1965) явно говорил об эффективности алгоритмов при обсуждении максимального паросочетания.

Область изучения алгоритмов и их эффективности естественным образом расширялась, поскольку компьютеры нуждались в точной постановке задачи, а программирование ставило новые вопросы. Если говорить о положительной стороне, то Фолькер Штрассен открыл совершенно новые и неожиданные алгоритмы решения старых задач умножения и обращения матриц (Strassen 1969, здесь глава 30), а Эдсгер Дейкстра (Dijkstra 1965, здесь глава 26) математически строго решил сложную задачу управления конкурентностью. С другой стороны, Стивен Кук (Cook 1971b, здесь глава 34) модифицировал данное Тьюрингом доказательство возможности описать программу логическими формулами и применил его к конкретному случаю классов $\mathcal{NP}$ и логики высказываний, поставив тем самым до сих пор не разрешенную задачу: верно ли, что $\mathcal{P} = \mathcal{NP}$? А Ричард Карп (Karp 1972, здесь глава 36) показал, что самые разные известные задачи, характеризуемые комбинаторным взрывом, являются по существу вариациями одной и той же проблемы. Кнут (Knuth 1976,

здесь глава 43) предложил нотацию, которая теперь применяется практически повсеместно для сравнения вычислительной сложности алгоритмов, а также – в не менее восхитительной статье (Knuth 1974b) – предложил терминологию $\mathcal{P}$ и $\mathcal{NP}$, с которой научное сообщество согласилось.

Вместе с абстрагированием алгоритмов росло стремление рассматривать сами программы более формально и абстрактно, даже если это вело к отказу от части выразительной способности языков программирования. Так, Дейкстра (Dijkstra 1968a, здесь глава 29) предложил полностью избавиться от переходов, но это радикальное предложение не было принято; Хоар (Hoare 1969, здесь глава 31) предложил рассматривать программы как логические формулы, которые можно было подвергнуть формальной верификации; а ДеМилло с соавторами (DeMillo et al. 1979, здесь глава 44) резко возражал против плана верификации. Лисков и Зиллес (Liskov and Zilles 1974, здесь глава 39) предложили применять те же стандарты абстрагирования к данным, и это предложение было воспринято положительно, что впоследствии привело к объектно-ориентированному программированию.

Линия развития ведет от самой первой операционной системы с разделением времени, спроектированной Корбато с сотрудниками (Corbató et al. 1962, здесь глава 23), к системе мультипрограммирования «THE», разработанной Дейкстрой (Dijkstra 1968b, здесь глава 28), и системе Unix (Ritchie and Thompson 1974, здесь глава 37), которая ныне существует во многих вариантах.

Такие большие программные системы становилось все труднее писать и поддерживать в работоспособном состоянии. В 1960-е и 1970-е годы исправление ошибок в нескольких многомиллионных проектах сразу после их запуска обошлось в миллионы долларов, а некоторые пришлось вообще отправить в мусорную корзину. Программная инженерия возникла как искусство практиков, этой теме были посвящены две классические статьи: Ройса (Royce 1970, здесь глава 33) и Брукса (Brooks 1995, впервые опубликована в 1975 году, здесь глава 40). Каждый программист должен прочесть обе.

Для обработки больших наборов данных также понадобились новые методы. Кодд (Codd 1970, здесь глава 32) определил реляционную модель, концептуально элегантную, но потребовавшую немалых усилий для практической реализации. Ныне она является основой индустрии управления данными. Поиск в больших текстовых базах данных теперь считается само собой разумеющимся аспектом поиска в вебе, но на самом деле это старая задача информационного поиска. Карен Спарк Джонс (Jones 1972, здесь глава 35) открыла полезные принципы релевантности терминов, сопоставляя частоту встречаемости в документе (которая предполагает релевантность) с частотой встречаемости во всем корпусе документов (которая предполагает, что слово недостаточно специфично и потому не может являться полезным ключом индексирования).

На 1970-е годы пришелся взрывной рост сетей. Серф и Кан (Cerf and Kahn 1974, здесь глава 38) заложили основы протоколов интернета, которые мало изменились по сравнению с первоначальным описанием, а Меткалф и Боггс (Metcalfe and Boggs 1976, здесь глава 41) описали протоколы локальных сетей

Ethernet. Принятие обоих протоколов в качестве некоммерческих стандартов открыло возможность для взаимодействия любых компьютеров между собой.

В условиях вездесущей связанности потребовалось улучшить защищенность информации. Шифрование издавна применялось для секретного обмена сообщениями, но традиционные методы в интернете имели ограниченную пригодность, потому что ключ шифрования-дешифрирования приходилось передавать по тому же самому незащищенному каналу, что и сообщение. Диффи и Хеллман (Diffie and Hellman 1976a, здесь глава 42) ошеломили сообщество, предложив решение (которое, как оказалось, частично предвидел Ральф Меркл и сотрудники Центра правительственной связи, британской секретной службы). Ривест с сотрудниками (Rivest et al. 1978, здесь глава 45) разработали необходимый математический аппарат; их алгоритм широко используется и сегодня, хотя его безопасность основывается на недоказанных предположениях. Современный всплеск интереса к квантовым вычислениям в немалой степени обусловлен тем, что квантовые компьютеры можно будет использовать для вскрытия кодов RSA (Shor, 1999). Блестящая статья Шамира (Shamir 1979, здесь глава 46), последняя в этой книге, посвящена разделению секретов таким образом, что для восстановления требуется определенный уровень кооперации; эту задачу можно поставить без всякой связи с компьютерами, и для ее решения достаточно школьной математики, но смысл она имеет только в век компьютеров.

1 Первая аналитика (~350 год до н. э.)[1]

Аристотель

Некоторые идеи, встречающиеся в информатике, настолько хорошо знакомы, что даже трудно представить, что когда-то они были новыми. Такова идея логики. Методы счета, дошедшие до нас из тьмы веков, слабо повлияли на конструкцию современных компьютеров, но принципы двузначной логики лежали в основе цифровых вычислительных машин.

Компьютеры универсальны. Большинство компьютеров для игры в шахматы или для бухгалтерских расчетов – это самые обычные компьютеры, которые можно использовать как для игр, так и для деловых целей. Бит, который сегодня означает, что конь на поле f3 является белым или черным, завтра может означать, что книга имеется в наличии. Логические правила, встроенные в оборудование компьютера, можно использовать для манипулирования как той, так и другой информацией. Но абстрактные идеи вещей и свойств и логические правила рассуждения о них существовали не всегда. Это идеи Аристотеля, они стали первыми и непременными шагами в направлении вычислений, касающихся вещей и их свойств.

Аристотель (384–322 до н. э.) был великим систематизатором. Во многих работах, большая часть которых утрачена, он подвергал анализу и классификации все, что можно себе вообразить. В книге «Первая аналитика» миру была явлена первая логическая система. Ее цель – делать выводы из посылок способом, зависящим только от формы аргументации, но не от убедительности оратора или от чего-то, не упомянутого в посылках. Данное Аристотелем объяснение логической дедукции лежит в основе всей современной логики.

---

[1] Текст печатается по изданию: *Аристотель*. Аналитики / пер. Б. А. Фохта. Гос. изд-во полит. литературы, 1952.

Технический лексикон Аристотеля сильно затрудняет чтение. По счастью, архаические детали нам не важны. Он говорит об идее предиката – свойства, которым вещь может обладать или не обладать (в терминологии Аристотеля «приписывается» или «не приписывается»). Это подводит нас к современной идее о вещи как элементе множества всех вещей, обладающих некоторым свойством. Аристотелеву идею «принадлежности» можно интерпретировать как отношение между множеством и подмножеством. Например, сказать, что свойство A не принадлежит ни одной из вещей B, – все равно, что сказать, что ни один элемент множества B не является элементом A, т. е. что B не пересекается с A (или что A и B дизъюнктны). Таким образом, пример «если A приписывается всем B, а B – всем C, то по необходимости A приписывается всем C» – это утверждение о транзитивности отношения надмножества: если $A \supseteq B$ и $B \supseteq C$, то $A \supseteq C$. В распоряжении Аристотеля не было такой нотации, но люди, формализовавшие логику и теорию множеств, стояли на его плечах.

Убедительные математические рассуждения существовали и до Аристотеля, но именно Аристотель первым отделил форму таких рассуждений от их содержания. При этом он показал, как следует рассуждать механически – сопоставляя высказывания с общими шаблонами и делая выводы, которые следовали с железной неотвратимостью. Аристотель не проектировал и не строил логических вычислительных машин, но применяемый им лексикон наводит на мысль, что он описывает процесс вычисления. Слово, которое ниже переведено как «силлогизм», – в оригинале συλλογισμός и означает «подсчет» или «вычисление».

Аристотель также описывает метод, позволяющий показать, что претендующие на истину формы логического вывода не являются универсально верными. Для этого он использует контрпримеры. Он приглашает читателя сделать общий вывод из посылок $A \supseteq B$ и $B \cap C = \emptyset$. На самом деле из этих двух посылок нельзя вывести никакого заключения о соотношении между A и C. Ибо если считать, что A = животные, B = лошади и C = люди, то все посылки удовлетворены (все лошади – животные, но лошадь – не человек) и $A \supseteq C$ (все люди – животные). Однако если A = животные, B = люди и C = камни, то посылки тоже удовлетворены (все люди – животные, но человек – не камень), но A и C не пересекаются (ни один камень не является животным). (В современном рассуждении мы должны были бы добавить, что C не может одновременно быть подмножеством A и не пересекаться с A при условии, что C не пусто.) Этот метод и по сей день используется для опровержения предположений и доказательства независимости гипотез.

Прежде всего следует сказать о предмете исследования и о том, кем оно должно быть выполнено, именно: что исследовать должно доказательство и что это – дело доказывающей науки. Далее необходимо определить, что такое посылка, термин, силлогизм, а также какой силлогизм совершенный и какой – несовершенный; затем – что значит «это целиком содержится или не содержится в этом» и что значит «что-либо приписывается всем или ни одному».

Посылка есть высказывание, утверждающее или отрицающее что-нибудь о чем-нибудь. Высказывание же это бывает или общим, или частным, или неопределенным. Общим я называю суждение, когда (А), например, присуще всем или не присуще ни одному (Б), частным – когда (А) присуще или не присуще некоторым или присуще не всем (Б), неопределенным – когда нечто одно присуще или не присуще другому, без указания на то, присуще ли оно всему или не всему другому (как, например, суждение «противоположности изучаются одной и той же наукой» или «удовольствие не есть благо»); отличается же доказывающее суждение от диалектического, ведь доказывающее суждение есть принятие одного из членов противоречия (ибо тот, кто доказывает, не спрашивает, а утверждает), а диалектическое суждение есть вопрос относительно членов противоречия. При образовании силлогизмов это различие не имеет никакого значения как в том, так и в другом случае. Ибо как тот, кто доказывает, так и тот, кто спрашивает, одинаково строят силлогизм из положений о том, что нечто присуще или не присуще (чему-нибудь другому), так что силлогистическое суждение есть вообще утверждение или отрицание чего-нибудь о чем-нибудь по указанному выше способу; при этом доказывающим оно будет в том случае, если оно истинно и взято из предположений, выдвинутых с самого начала; диалектическим же оно является для вопрошающего как вопрос относительно членов противоречия, а для строящего умозаключения – как принятие того, что кажется, и того, что вероятно, как об этом сказано в *Топике*.

В дальнейшем изложении будет точно сказано о том, что такое суждение и чем отличаются друг от друга суждения силлогистическое, аподиктическое и диалектическое; а пока достаточно и того, что определено сейчас.

Термином я называю то, на что разлагается суждение, то, что приписывается, и то, чему приписывается, независимо от того, присоединяется или отнимается то, что выражается посредством глаголов «быть» и «не быть»; *силлогизм* же есть высказывание, в котором при утверждении чего-либо из него необходимо вытекает нечто отличное от утвержденного и именно в силу того, что это есть. Под словами же «в силу того, что это есть», я разумею, что это отличное вытекает благодаря этому, а под словами «вытекает благодаря этому» – что оно не нуждается ни в каком постороннем термине, чтобы следовать с необходимостью. Совершенным силлогизмом я называю такой, который для выявления необходимости заключения не нуждается ни в чем другом, кроме того что принято. Несовершенным я называю такой, который хотя и является необходимым благодаря положенным в основание данного силлогизма терминам, но нуждается в одном или нескольких суждениях, которых нет в посылках.

Выражения «нечто одно целиком содержится в другом» и «нечто одно приписывается всему другому» означают одно и то же. Говорим же мы «нечто одно приписывается всему другому», когда не может быть указана ни одна часть подлежащего, о которой нечто другое не высказывалось бы. И точно так же, когда мы говорим, что «ничего ничему другому не приписывается».

Всякое суждение есть или суждение о том, что присуще, или о том, что необходимо присуще, или о том, что возможно присуще; и из этих суждений, в зависимости от того, приписывается ли что-либо в них или не приписы-

вается, одни бывают утвердительными, другие – отрицательными; и далее, одни утвердительные и отрицательные бывают общими, другие – частными, третьи – неопределенными.

Суждение о присущем, если оно общеотрицательное, необходимо допускает обращение в отношении своих терминов, например: если никакое удовольствие не есть благо, то и никакое благо не есть удовольствие. Общеутвердительное же суждение тоже необходимо допускает обращение, однако не в общее, а в частное, например: если всякое удовольствие есть благо, то какое-нибудь благо есть удовольствие; из частных суждений утвердительное необходимо допускает обращение его в частное же (ибо если какое-нибудь удовольствие есть благо, то и какое-нибудь благо будет удовольствием); обращение же частноотрицательного суждения не необходимо, ибо если некоторым живым существам не присуще быть людьми, то отсюда не следует, что некоторым людям не присуще быть живыми существами.

Возьмем сперва в качестве общеотрицательного суждения АБ. Если А не присуще ни одному Б, то и Б не будет присуще ни одному А. Ибо если бы оно было присуще чему-нибудь, например В, то было бы неправильно заключить, что А не присуще ни одному Б, так как В есть также часть Б. Если же А присуще всему Б, то Б будет присуще некоторым А, ибо если бы Б не было присуще ни одному А, то и А не было бы присуще ни одному Б; но ведь было предположено, что А присуще всем Б. Точно так же обстоит дело с обращением и в том случае, если суждение частное. Ибо если А присуще некоторым Б, то и Б необходимо будет присуще некоторым А. Если же Б не было бы присуще ни одному А, то и А не было бы присуще ни одному Б. Но если А некоторым Б не присуще, то не необходимо, чтобы и Б не было присуще некоторым А, как, например, в том случае, если Б есть живое существо, а А – человек; ибо не всем живым существам присуще быть людьми, однако всем людям присуще быть живыми существами. <…>

После того как мы дали эти определения, мы укажем теперь, посредством чего, когда и каким образом строится всякий силлогизм; затем придется говорить о доказательстве. О силлогизме мы должны говорить раньше, чем о доказательстве, потому что силлогизм есть нечто более общее: ведь (всякое) доказательство есть некоторого рода силлогизм, но не всякий силлогизм – доказательство.

Итак, если три термина так относятся между собой, что последний целиком содержится в среднем, а средний целиком содержится или не содержится в первом, то необходимо, чтобы для двух крайних терминов образовался совершенный силлогизм. *Средним* термином я называю тот, который сам содержится в одном, в то время как в нем самом содержится другой, и по положению он является средним; *крайними* же я называю и тот, который содержится в другом, и тот, в котором содержится другой. В самом деле, если А приписывается всем Б, а Б – всем В, то А необходимо приписывается всем В. А как следует понимать выражение «приписывается всем», об этом было сказано выше. Точно так же если А не приписывается ни одному Б, а Б приписывается всем В, то А не будет присуще ни одному В. Если же первый термин присущ всему среднему, а средний не присущ ни одному последнему, то для крайних терминов нельзя будет построить никакой силлогизм, ведь из того,

что здесь имеется, ничего не следует с необходимостью, ибо в таком случае первый термин возможно присущ и всем и ни одному последнему, так что ни частное, ни общее заключение не вытекает здесь с необходимостью. Но так как здесь ничего с необходимостью не вытекает, то нельзя построить силлогизм. Пусть терминами для случая, когда первый термин присущ всему последнему, будут: живое существо – человек – лошадь; для случая, когда он ему вовсе не присущ: живое существо – человек – камень. Не получится также никакого силлогизма и тогда, когда ни первый не присущ среднему, ни средний не присущ ни одному последнему. Для случая, когда первый термин присущ всему последнему, терминами пусть будут: наука – линия – врачебное искусство; для случая, когда он ему вовсе не присущ: наука – линия – единица.

Итак, если термины взяты в общих посылках, то ясно, когда по этой фигуре может быть построен силлогизм и когда нет; и если силлогизм получается, то термины необходимо должны находиться друг к другу в указанном выше отношении, и если термины находятся друг к другу в таком отношении, то силлогизм получится.

Если же один из терминов взят в общей посылке, а другой – не в общей и первый является большим крайним, в утвердительной или в отрицательной посылке, второй же – в утвердительной посылке – меньшим крайним, то с необходимостью получится совершенный силлогизм. Если же термин, взятый в общей посылке, является меньшим крайним или оба термина находятся в каком-либо другом отношении, силлогизм невозможен. Большим крайним термином я называю тот, в котором содержится средний термин, меньшим же – тот, который подчинен среднему. Пусть А будет присуще всем Б, а Б – некоторым В, в таком случае если выражение «быть приписываемым всем» понимать в указанном выше смысле, то А будет необходимо присуще некоторым В. Если же А не присуще ни одному Б, а Б присуще некоторым В, то А необходимо не присуще некоторым В, ибо что значит «не быть приписываемым ни одному», – это также было определено; так что (и здесь) получится совершенный силлогизм. Точно так же обстоит дело, когда суждение БВ является неопределенным и утвердительным. Ведь силлогизм здесь будет таким же – берется ли (Б В) как неопределенное или как частное…

2 Истинный метод (1677)

Готфрид Вильгельм Лейбниц

Готфрид Вильгельм Лейбниц (1646–1716) был человеком универсальных знаний – философ с широчайшим кругом интересов, юрист и политический мыслитель, а также выдающийся и плодовитый математик. Он мог бы участвовать в конкурсе за звание первого специалиста по информатике. Паскаль участвовал бы в этом конкурсе благодаря изобретению суммирующей машины. Были и другие – как до, так и после Лейбница, но именно Лейбниц построил калькулятор с вложенными циклами, который умел умножать и делить (см. стр. 95). Еще важнее то, что он изобрел двоичную арифметику (рис. 3.1) и сконструировал двоичный калькулятор (который так никогда и не был построен).

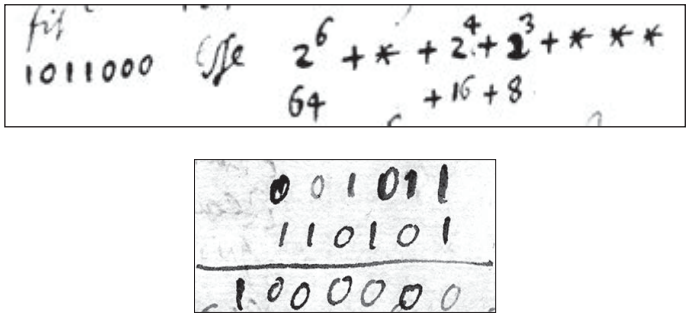

Рис. 2.1. Преобразование из двоичной системы
в десятичную и двоичное суммирование.
Из работы Лейбница «De progressione dyadica» (1679)

Лейбниц делит с Исааком Ньютоном честь открытия того, что впоследствии было названо исчислением бесконечно малых. В наши дни это исчисление так прочно ассоциируется с математикой, что мы уже забыли о вычис-

лениях, ставших побудительным мотивом для его открытия, например о том, как найти площадь фигуры посредством суммирования площадей тонких полосок. Предложенное Лейбницем обозначение *dx* для бесконечно малого приращения *x* дожило до наших дней, поскольку оно чрезвычайно упрощает запись правил, почти не выразимых в точечной нотации Ньютона. Например, тождество $\frac{dy}{dx} = \frac{dy}{du} \cdot \frac{du}{dx}$ очень неудобно записывать с помощью обозначения $\dot{y}$ для производной *y*, которое было предложено Ньютоном.

Лейбниц понимал, как важна хорошая система обозначений для ясного мышления. Он познакомился с сочинениями Аристотеля в возрасте 14 лет и в процессе получения образования написал диссертацию на тему использования систематической логики в юридических рассуждениях (Leibniz 1666). Он разработал формальную систему обозначений для логических рассуждений, раннюю форму математической логики (Struik, 1969, стр. 123). Его старания оформить рассуждения в виде логической системы с формальными правилами вылились в грандиозный план унификации всех человеческих знаний в систему, которая устранила бы все споры; факты, будучи один раз установлены, давали бы затем неопровержимые ответы. Тогда все рассуждения свелись бы к тривиальной подстановке, и в результате мир стал был бесспорно лучше. Знаменитый оптимизм Лейбница – его уверенность в том, что мы живем в лучшем из миров, – смешивался, таким образом, с этим ранним техноутопизмом.

И хотя математические работы принесли ему признание, его оптимизм подвергался осмеянию. В 1759 году Вольтер выставил его в карикатурном виде под личиной доктора Панглоса в «Кандиде». Сегодня его мечта о создании совершенного мира посредством логики и механизированных рассуждений кажется наивной и плохо согласующейся с теологической картиной мира, на который он хотел ее распространить. И тем не менее его идея сведения всего и вся к логике возникает вновь и вновь; наиболее известные примеры дают работы Мак-Каллока и Питтса (McCulloch and Pitts 1943, здесь стр. 120) и Буша (Bush 1945a, здесь стр. 156), а также все автоматизированные системы поддержки принятия решений.

Так как счастье заключается в удовлетворении желаний и так как непреходящая удовлетворенность зависит от уверенности в будущем – уверенности, которая основана на наших знаниях о природе Бога и души, – то отсюда вытекает, что для истинного счастья необходимо знание.

Но знание опирается на доказательство и на придумывание доказательств *некоторым методом*, который не всем известен. Ибо хотя любой человек способен судить о доказательстве (так как оно не заслуживало бы такого названия, если бы не каждый из тех, кто внимательно рассмотрел его, был им убежден и доволен), все же не каждый способен ни придумать доказательство самостоятельно, ни ясно изложить его, когда оно найдено, в силу отсутствия досуга или метода.

Истинный метод во всей своей широте представляется мне вещью, до настоящего времени неизвестной и нигде, кроме математики, не практи-

ковавшейся. Даже в самой математике он все еще далек от совершенства, что мне повезло доказать некоторым (по общему мнению, входящим ныне в число первейших математиков этого столетия) посредством вызывающих удивление доказательств. И я рассчитываю привести несколько примеров этого, которые, быть может, будут небезынтересны будущим поколениям.

И тем не менее, если метода, применяемого математиками, оказалось недостаточно для открытия всего того, что можно было ожидать от них, он, по крайней мере, смог предохранить их от ошибок, и если они не сказали всего того, что должны были бы сказать, то не сказали и ничего такого, чего говорить были не должны.

Если бы все те, кто занимается другими науками, подражали математикам хотя бы в этом пункте, то мы были бы очень счастливы и давно имели бы незыблемую метафизику, а также добрые нравы, которые зависят от нее, потому что метафизика включает в себя знание о Боге и душе, знание, которое должно направлять нашу жизнь.

Более того, мы имели бы науку о движении, которая является ключом к физике, а стало быть, и к медицине. Я верю, что мы в настоящее время сейчас пребываем в состоянии устремления к ней, и некоторые мои первые мысли, в силу их удивительной простоты, были приняты с такими рукоплесканиями наиболее образованными людьми нашего времени, что я полагаю, нам надлежит лишь провести некоторые опыты, правильно спланированные и продуманные (а не случайные, методом проб и ошибок, как часто бывает), чтобы на их основе возвести твердыню неоспоримой и доказательной физики.

Ныне же причина того, что искусство доказательства до сих пор практикуется только в математике, еще не осознана всеми, ибо если бы причина затруднений была известна, то уже давно было бы придумано лекарство. Причина же в том, что математика несет внутри себя средства собственной поверки. Ибо когда мне предъявляют ложную теорему, мне не нужно ни исследовать ее, ни даже знакомиться с доказательством, потому что я обнаружу ее ложность апостериори с помощью простого опыта, не требующего ничего, кроме бумаги и чернил, а именно вычисления, которое вскроет ошибку, сколь бы незначительной она ни была. Если бы в других предметах было так же просто рассуждать на основе опытов, то и не было бы столь различных мнений. Беда, однако, в том, что в физике опыты трудны и стоят дорого, а в метафизике вообще невозможны, если только Бог, ради нашего блага, не совершит чуда и не сделает тайные, невещественные вещи доступными для нашего познания.

Эта трудность не является непреодолимой, хотя на первый взгляд кажется нам именно таковой. Но те, кто готов склонить слух к тому, что я собираюсь сказать об этом, вскоре переменят свое мнение. Должно отметить тогда, что испытания или опыты, производимые в математике, дабы защититься от ложных рассуждений (как, например, вычеркивание девяток, вычисление Лудольфа ван Цейлена длины окружности, таблицы синусов и прочее), применяются не к самой вещи, а к символам, подставленным нами вместо вещи. [Примечание редактора: «вычеркивание девяток» – это способ проверки делимости числа на 9 путем складывания всех его цифр. Аристотель аппроксимировал число π, вычислив площадь правильного 96-угольника; Лудольф ван Цейлен (1540–1610) использовал с этой целью правильный многоугольник

с 2^{62} сторонами и вычислил π с 35 десятичными знаками, что заняло у него несколько лет (Ludolph van Ceulen 1596).] Ибо произвести вычисление над числами, например удостовериться в том, что произведение 1677 на 365 равно 612 105, было бы невозможно, если бы пришлось для этого сложить 365 кучек по 1677 камушков в каждой, а затем пересчитать их все, чтобы проверить, получилось ли указанное число. Вот почему мы с готовностью делаем это со знаками на бумаге, применяя проверку девяток или еще какой-нибудь способ. Аналогично, когда кто-то предлагает предположительно точную квадратуру круга, нам не нужно изготавливать материальный круг и обвязывать его веревкой, чтобы проверить, действительно ли длина этой веревки или отношение длины окружности к диаметру равно указанной величине; это было бы трудно, потому что если ошибка составляет тысячную (или того меньшую) часть диаметра, потребовалось бы построить большую окружность и с высокой точностью. И все же мы опровергаем неверную квадратуру кругу с помощью опыта и путем вычисления или проверки чисел. Но эта проверка производится только на бумаге и, следовательно, над знаками, представляющими вещь, а не над самой вещью.

Это соображение фундаментально в данном вопросе, и хотя многие очень умные люди, особенно в нашем столетии, заявляли, что предоставят нам доказательства в физике, метафизике, этике и даже в политике, юриспруденции и медицине, тем не менее они либо ошибались (поскольку все шаги скользкие и трудно не упасть, если не руководствоваться какими-то указаниями), либо даже если им удавалось найти доказательство, они не могли сделать свои рассуждения убедительными для всех (поскольку не существовало еще способа проверить аргументы с помощью каких-то простых испытаний, доступных каждому).

Отсюда ясно, что если бы мы могли найти знаки или символы, пригодные для выражения всех наших мыслей столь же ясно и точно, как арифметика выражает числа, а аналитическая геометрия выражает линии, то мы могли бы достигнуть во всех предметах, *поддающихся рассуждениям*, всего того, что можно сделать в арифметике и геометрии.

Ибо все вопросы, опирающиеся на рассуждение, можно было бы разрешить путем перегруппировки этих символов и с помощью некоторого вычисления, которое сделало бы изобретение красивых вещей очень простым делом. Ибо нам не пришлось бы напрягать наши мозги в такой степени, в какой мы принуждены делать это сегодня, и тем не менее испытывать уверенность в своей способности осуществить все осуществимое *в соответствии с имеющимися фактами*.

Более того, всякий должен был бы согласиться с тем, что мы обнаружили или к чему пришли путем логических умозаключений, потому что было бы легко проверить вычисление, либо повторив его, либо попытавшись произвести какие-то испытания наподобие вычеркивания девяток в арифметике. А буде кто усомнился бы в содеянном мной, я сказал бы ему: «Давайте займемся вычислениями, сэр», – а затем, взяв перо и чернила, мы вскоре уладили бы дело.

Я всегда добавляю: *при условии, что это может быть достигнуто путем рассуждения, в соответствии с имеющимися фактами*. Ибо хотя какие-то

опыты всегда необходимы, дабы служить основою для рассуждений, тем не менее, коль скоро эти опыты проделаны, мы могли бы извлечь из них все, что был бы в состоянии извлечь из них кто-то другой, и даже придумать опыты, которые нужно было бы проделать для разрешения остающихся сомнений. Это было бы достойное восхищения подспорье, даже в политике и в медицине, рассуждениям об имеющихся симптомах и свидетельствах надежным и безошибочным способом. Ибо пусть даже у нас не будет достаточных свидетельств, чтобы сформировать неопровержимое суждение, мы всегда сможем определить наиболее вероятное следствие *из имеющихся фактов*. И это все, что может сделать рассуждение.

Пойдем далее. Знаки, способные выразить все наши мысли, образуют новый язык, на котором можно будет писать и разговаривать; этот язык будет очень трудно построить, но очень легко выучить. Он скоро будет принят всеми в силу своего широкого использования и поразительной полезности и чудесно послужит цели общения между многими народами, что будет еще больше способствовать его принятию. Те, кто станет писать на этом языке, не будут допускать ошибок, при условии что будут избегать ошибок в вычислениях, варваризмов, нарушения правил и других ошибок грамматического и синтаксического характера. Более того, этот язык будет обладать замечательным свойством, а именно принуждать к молчанию невежественных. Ибо никто не будет способен говорить или писать на этом языке ни о чем, кроме того, что понимает, а если попытается сделать, то произойдет одно из двух: либо тщета оглашаемого будет очевидна всякому, либо предмет будет изучен в результате письменной или устной речи. Точно так же, как те, кто вычисляет, научаются посредством письма, а те, кто говорит, иногда встречают успех, о котором не помышляли, когда *язык прежде ума рыщет*. Особенно часто это будет случаться в нашем языке ввиду его точности. Настолько, что не будет никаких двусмысленностей и амфиболий, и все, что можно выразить на нем вразумительно, будет сказано с полной обоснованностью.

Осмелюсь сказать, что это высочайшее напряжение человеческого разума и что когда проект будет осуществлен, от самого человека будет зависеть, быть ли ему счастливым, потому что у людей появится средство, которое будет служить превознесению разума не в меньшей степени, чем телескоп служит совершенствованию зрения.

Я страстно желал бы завершить этот проект, если Бог дарует мне время. Я обязан им только самому себе, и первые мысли о нем появились у меня, когда мне было восемнадцать лет, что я засвидетельствовал несколько позже в печатном рассуждении (Leibniz 1666). И так как я уверен, что не существует изобретения, сколько-нибудь близкого к этому, я полагаю, что нет ничего, способного в такой же степени обессмертить имя изобретателя. Но у меня есть и более основательные причины думать об этом, потому что религия, которую я исповедую, убеждает меня, что любовь к Богу заключается в горячем желании распространять общее благо, а разум говорит мне, что нет ничего, что было бы лучшим общим благом для всех людей, чем то, что совершенствует разум.