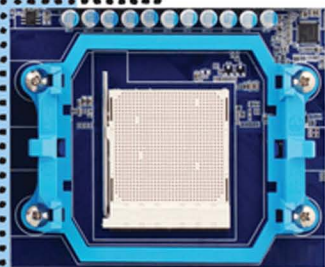




Д. И. Черемисинов

# ПРОЕКТИРОВАНИЕ И АНАЛИЗ ПАРАЛЛЕЛИЗМА В ПРОЦЕССАХ И ПРОГРАММАХ



НАЦИОНАЛЬНАЯ АКАДЕМИЯ НАУК БЕЛАРУСИ  
Объединенный институт проблем информатики

**Д. И. Черемисинов**

**ПРОЕКТИРОВАНИЕ  
И АНАЛИЗ  
ПАРАЛЛЕЛИЗМА  
В ПРОЦЕССАХ  
И ПРОГРАММАХ**



Минск  
«Беларуская навука»  
2011

УДК 621.382.049.77:004.42/43

**Черемисинов, Д. И.** Проектирование и анализ параллелизма в процессах и программах / Д. И. Черемисинов. – Минск : Беларус. навука, 2011. – 300 с. ISBN 978-985-08-1285-8.

В монографии обсуждается широкий круг вопросов, связанных с проектированием микроэлектронных устройств с программным управлением. Рассматриваются задачи и методы разработки программ и аппаратуры с использованием языка ПРАЛУ, область применения которого расширена задачами верификации. Описывается разработка программы для кластерного компьютера, решающей комбинаторно-сложные задачи.

Книга адресована специалистам в области проектирования дискретных устройств и может быть полезна для студентов и аспирантов, специализирующихся в данном направлении.

Табл. 5. Ил. 54. Библиогр.: 115 назв.

**Р е ц е н з е н т ы:**

доктор технических наук, профессор А. А. Прихожий  
кандидат физико-математических наук, доцент Ю. В. Поттосин

**ISBN 978-985-08-1285-8**

© Черемисинов Д. И., 2011

© Оформление. РУП «Издательский дом  
«Беларуская навука», 2011

## ОГЛАВЛЕНИЕ

<b>Предисловие</b> .....	6
<b>Введение</b> .....	11
<b>ГЛАВА 1. Параллельные алгоритмы</b> .....	33
1.1. Определение .....	33
1.2. Классическая модель вычисления.....	35
1.3. Модели параллелизма .....	38
1.3.1. Формализация параллелизма в программировании.....	38
1.3.2. Формализация параллелизма при проектировании аппаратуры.....	49
1.4. Концепция времени в моделях распределенных систем .....	60
1.4.1. Обобщенный конечный автомат .....	61
1.4.2. Сети Петри.....	64
1.4.3. Временная логика.....	70
1.4.4. Интерпретация временной логики на модели Кripke .....	72
1.5. О системе классификации формализмов.....	76
<b>ГЛАВА 2. Язык ПРАЛУ как логика для описания взаимодействия</b> .....	78
2.1. Физическая трактовка временной логики .....	81
2.2. Логика взаимодействия параллельных процессов .....	84
2.2.1. Структура события взаимодействия .....	85
2.2.2. Альтернативные предположения о связи событий .....	90
2.2.3. Понятие окружающей среды.....	91
2.3. Интерпретация ПРАЛУ как временной логики .....	93
2.3.1. Минимальная семантика ПРАЛУ .....	94
2.3.2. Эквивалентность минимальной семантики ПРАЛУ и CCS .....	102
2.3.3. Необходимость измеряемого времени .....	105
2.3.4. Уточнения минимальной семантики, обеспечивающие реали- зуемость .....	107
2.3.5. Однопроцессорная реализация ПРАЛУ .....	109

<b>ГЛАВА 3. Проектирование распределенных систем</b> .....	116
3.1. Моделирование локальной шины PCI .....	117
3.1.1. Функционирование шины PCI .....	119
3.1.2. Стандарт шины PCI .....	124
3.1.3. Пример устройства PCI .....	124
3.1.4. Оценка работоспособности алгоритма методом симуляции .....	126
3.1.5. Логическая структура устройства PCI .....	128
3.2. Самосинхронные схемы .....	132
3.3. ПРАЛУ и высокоуровневые языки описания аппаратуры .....	134
3.3.1. Компиляция конструкций языков программирования в комбинационную схему .....	136
3.3.2. Схемная семантика подмножества языка Verilog .....	138
3.3.3. Сравнение ПРАЛУ и Verilog .....	144
3.4. Верификация поведенческих моделей .....	148
3.4.1. Построение генератора и монитора .....	152
3.4.2. Задача верификации .....	154
3.4.3. Инструменты для символической верификации .....	156
3.4.4. ПРАЛУ-верификатор .....	163
<b>ГЛАВА 4. Мультиагентные системы</b> .....	168
4.1. Задача оптимизации программы агента .....	172
4.1.1. Представление протоколов конечными автоматами .....	172
4.1.2. Представление протоколов логическими формулами .....	173
4.1.3. Синтез программ, реализующих конечные автоматы .....	176
4.1.4. Модели программ .....	179
4.1.5. Синтез программ по структуре описания автомата .....	184
4.1.6. Управление синтезом программ .....	193
4.2. Системы с произвольным числом агентов .....	195
4.2.1. VDI-архитектура мультиагентных систем .....	196
4.2.2. Формализмы задания протоколов взаимодействия агентов .....	197
4.3. Спецификация протоколов взаимодействия агентов на ПРАЛУ .....	200
4.3.1. Протокол английского аукциона .....	201
4.3.3. Онтология архитектуры VDI для ПРАЛУ .....	204
4.3.4. Методология программирования агентов на ПРАЛУ .....	207
4.3.5. Программная реализация алгоритмов на ПРАЛУ .....	210
<b>ГЛАВА 5. Программирование кластерного компьютера</b> .....	214
5.1. Среда программирования кластера через удаленный терминал .....	215
5.2. Управление вычислениями на кластерном компьютере .....	217
5.2.1. MPI-программа .....	218
5.2.2. Интерфейс для управления MPI-программой .....	224
5.3. Параллельные вычисления в задачах проектирования СБИС .....	228
5.3.1. Задачи, решаемые программным комплексом .....	229
5.3.2. Сложность комбинаторно-логических задач .....	234
5.3.3. Закон Амдала .....	236

5.3.4. Классы параллельных алгоритмов .....	238
5.4. Методика программирования MPI-программ .....	243
5.4.1. Специфика отладки параллельных программ.....	243
5.4.2. Предотвращение ошибок в параллельных вычислениях .....	246
5.4.3. Верификация на модели в MPI-программе.....	249
<b>ГЛАВА 6. Защита программ</b> .....	<b>252</b>
6.1. Модели угроз нападения на защищаемую программу.....	253
6.1.1. Враждебная программа в легальной среде исполнения.....	255
6.1.2. Легальная программа во враждебной среде исполнения .....	256
6.1.3. Защита распределенных программ.....	258
6.2. Сохранение доверенности среды и обнаружение угроз.....	260
6.3. Формальные модели, используемые при защите программ .....	263
6.3.1. Защита автоматизированным преобразованием кода.....	263
6.3.2. Предотвращение изменения кода программы .....	265
6.3.3. Защита, обоснованная результатами, связанными со сложностью задачи.....	267
6.3.4. Использование аппаратных средств.....	268
6.4. Анализ стойкости защиты программы с помощью электронного ключа.....	270
6.4.1. Защита программ посредством электронного ключа.....	271
6.4.2. Виды атак на защищенную программу .....	272
6.4.3. Противодействие атаке методом удаления блокировки .....	274
6.4.4. Атака злоумышленника, имеющего ключ.....	276
6.5. Защита программы для кластерного компьютера .....	278
6.5.1. Протоколы для защищенной передачи данных .....	280
6.5.2. Анализ защищенности интерфейса для управления MPI-программой.....	283
<b>Приложение</b> .....	<b>286</b>
<b>Литература</b> .....	<b>294</b>

...для посполитого доброго и розмножения мудрости, умения, опатренности, разуму и науки приложиль есмъ працу выложити книгу сию на русский языкъ...

*Ф. Скорина*

## **ПРЕДИСЛОВИЕ**

В этой книге рассматривается «анатомия» параллелизма. Наша работа в этой области началась в 80-х годах прошлого века, когда А. Д. Закревский и В. К. Василенок предложили язык ПРАЛУ. Этот язык ориентирован на описание параллельных процессов с асинхронным взаимодействием. Язык предлагался как расширение традиционной теории дискретных автоматов. Несмотря на разработку программных средств моделирования и синтеза структуры по поведению дискретного устройства, заданного на этом языке, ПРАЛУ все же не получил широкого распространения в среде проектировщиков. Основной причиной отсутствия спроса в то время являлось преобладающее среди теоретиков и практиков мнение о неочевидности необходимости усложнения простой модели дискретного автомата параллелизмом.

Наша задача в разработках, связанных с ПРАЛУ, состояла в создании моделирующей программы. В теоретическом плане здесь требовалось установить семантическую эквивалентность поведения моделирующей программы (последовательного алгоритма) и дискретного устройства (аппаратура которого работает параллельно). В результате работы над этой проблемой стало ясно, что роль формализма ПРАЛУ не ограничивается инструментом описания поведения дискретных устройств. Разработка семантики ПРАЛУ, не связанной с работой аппаратуры, потребовала рассмотрения роли времени в описании поведения дискретных устройств.

В те годы это понятие было сферой интересов философов и физиков уровня Эйнштейна. Хотя уже написана была книга Хоара с изложением формализма CSP, и предложен формализм временной логики Прайора. В программировании параллелизм находился на периферии исследований. В разработке аппаратуры приходилось учитывать параллелизм при разработке моделирующих программ (симуляторов) для языков описания аппаратуры (VHDL и Verilog). Архитектура и теория этих программ считалась узкоспециальной отраслью программирования. Результаты, связанные с разработкой общей семантики ПРАЛУ в 1980-х годах, были опубликованы в журнале «Программирование» и в препринте.

В настоящее время «пророки» программирования (Herb Sutter – «Столпы параллелизма» или «The Landscape of Parallel Computing Research: A View from Berkeley») утверждают, что использование параллелизма становится основной задачей массового программирования.

Эта книга не претендует на роль учебника или справочника по существующим моделям параллельных систем. Хотя она может быть основой для спецкурсов и справочным пособием для всех, интересующихся прикладными аспектами взаимодействия процессов в параллельных системах. Наша цель состоит в содействии продвижению формальных методов в разработку и программирование устройств, характеризующихся параллелизмом поведения, концентрируясь на проблеме взаимодействия. Идеи, подходы и методы ориентированы на создание устройств, в которых программы и аппаратура не имеют жесткого разделения.

Книга начинается с вводной главы, посвященной истории разработки понятия «событие», которое является базовым в современном проектировании как программ, так и аппаратуры. Изложение представляет собой компиляцию идей, связанных со временем, первые из которых появились еще в эллинскую эпоху, и заканчивающихся основами современного представления о параллелизме в программировании и технике. Это достаточно компактный текст об идеях, кажущихся хорошо известными и понятными.



В первой главе приводится обзор способов описания поведения на основе понятия параллельного алгоритма. В программировании параллельный алгоритм – это форма задания процесса вычислений, предназначенная для более эффективного использования оборудования по сравнению с последовательным представлением. При проектировании систем различают трансформационные системы, работающие конечное число шагов, и реактивные, которые предназначены обеспечить определенное поведение. Для реактивных систем параллельный алгоритм – это способ задания требуемого поведения. В обеих областях ключевой проблемой формализации является проблема взаимодействия процессов параллельного алгоритма. Описание поведения системы формально представляет собой структуру над событиями. С точки зрения проблемы взаимодействия рассматриваются все основные формализмы для задания структуры событий в поведении дискретных устройств.

Все известные методики использования временной логики исходят из предположения, что она является формализмом для описания *свойств* дискретного устройства, в то время как само дискретное устройство задается средствами другого формализма. Во второй главе предлагается логическое исчисление, в котором формулы временной логики интерпретируются как алгоритмы, описывающие поведение. Формулами этого исчисления служат алгоритмы языка ПРАЛУ, генерирующие события как в модели Крипке. Время в этой модели дискретно – состоит из мгновенных и длительных событий – и плотно, но не измеряемо. В этой семантике поведение взаимодействующих линейных алгоритмов ПРАЛУ недетерминировано. Возможность произвольно объявлять пары событий как событие взаимодействия соответствует пониманию непредсказуемости задержки выполнения операции как недетерминированного выбора величины этой задержки. Детерминизм и реализуемость ПРАЛУ могут быть обеспечены несколькими вариантами уточнения минимальной семантики, состоящими в использовании свойства измеряемости времени. Сущность каждой реализуемой семантики состоит в предположениях о длительности протекания операции действия.

Следующая глава посвящена циклу проектирования устройства распределенной системы с использованием ПРАЛУ. Этот цикл демонстрируется на примере разработки простейшего устройства для шины PCI. Рассматриваются синтез структуры устройства, проверка работоспособности функционирования и верификация соответствия спроектированного устройства стандарту шины.

В четвертой главе рассматривается разработка агентов мультиагентных систем. Агент представляет собой программу или устройство, действует *автономно* и выполняет задачи для своего владельца *в сотрудничестве* с другими агентами. Сообщество агентов является реактивной системой и функционирует на основе общего знания. Рассматривается случай, когда общим знанием служит протокол совместного поведения. На примере английского аукциона демонстрируется применения ПРАЛУ для описания поведения агентов в соответствии с протоколом аукциона и разработка программ агентов покупателя и аукциониста. Разработка программы используется в качестве повода для рассмотрения задачи оптимизации программ. Известный результат теории программирования – теорема Райса – утверждает, что оптимизация каждого нетривиального семантического свойства программ – это алгоритмически неразрешимая задача. Структура управления программы может быть представлена конечным автоматом. Даже для программ, реализующих конечный автомат, не существует связи между автоматом управления программы и реализуемым автоматом. В общепринятой технологии управляющий автомат программы является тем автоматом, который нужно реализовать. Однако структура программы может определяться формой представления автомата. Подходящий язык представления автомата может обеспечить по сравнению со стандартной реализацией лучшую зависимость параметров программы от параметров описания, задача оптимизации в этом случае – выбор языка описания реализуемого автомата.

Основная тема следующей главы – случай программирования для трансформационных систем. Описывается проектирование распределенной системы логико-комбинационных вычис-

лений для кластерного компьютера. Эффективность параллельной программы – это коэффициент ускорения по сравнению с последовательным алгоритмом. В том случае, когда подзадачи, на которые разбивается исходная задача, решаются в режиме соревнования, возможно «сверхлинейное» ускорение. Этот результат показывает, что закон Амдала, утверждающий, что ускорение решения трансформационной задачи на параллельном компьютере всегда меньше линейного, является свойством метода программирования, а не архитектуры компьютера. Язык ПРАЛУ играет важную роль в проверке корректности структуры информационного взаимодействия параллельной программы.

В последней главе описываются две задачи защиты программ. Для анализа способов защиты защищаемую программу вместе с ее окружением удобно рассматривать как некоторую распределенную систему, состоящую из самой программы, ее операционного окружения и злоумышленника. Часть, в которой наличие злоумышленника не предполагается, заслуживает доверия, поэтому модели защиты сразу можно разделить на два класса: благонадежная программа в неблагонадежной среде и неблагонадежная программа в «доверенной среде». Для случая неблагонадежной среды проанализирована известная технология защиты программы с помощью электронного ключа. Этот способ обеспечивает гарантированную защиту от атаки злоумышленника, имеющего код программы, но не имеющего ключа. В случае легальной покупки программы, злоумышленник приобретает ключ, и этот метод становится не лучше других методов защиты. Вторая рассмотренная задача заключается в защите распределенной программы для выполнения комбинаторно-логических вычислений на кластерном компьютере. Если считать кластер доверенной средой, угрозой является несанкционированный доступ к вычислительным ресурсам кластерного компьютера. Предложено решение этой проблемы путем создания защищенных каналов взаимодействия программы кластерного компьютера с окружающей средой.

С целью сделать материал самодостаточным в приложении приведено описание синтаксиса ПРАЛУ.

Непочтительно объединять пространство и время, потому что мы можем абстрагироваться лишь от пространства, но не от времени.

Наше сознание постоянно переходит от одного состояния к другому, а это и есть время: последовательность. Кажется, Анри Бергсон сказал, что время является главной проблемой метафизики. Разрешив эту проблему, мы разрешили бы все загадки, но, к счастью, это нам не грозит. Мы вечно будем жаждать решения. Мы всегда сможем вслед за Святым Августином сказать: «Что такое время? Пока меня не спрашивают, я это знаю. А если спросят, я теряюсь».

*Х. Л. Борхес*

## ВВЕДЕНИЕ

Эта книга о проектировании систем, характеризующихся параллелизмом поведения, о формализмах для представления условий функционирования и о средствах проектирования, которые делают это возможным. Системы, характеризующиеся параллелизмом поведения, состоят из взаимодействующих компонентов. В таких системах правила взаимодействия составляют самую существенную и трудную для проектирования часть поведения. Так как мы интересуемся не только анализом таких систем, но также и их проектированием, существенное значение приобретает язык описания, применяемый для анализа систем, и диктующий путь, которым они будут проектироваться.

Прежде чем практически работать с параллельными системами, требуется иметь четкое представление, что подразумевается под параллелизмом и почему приходится его использовать в проектировании дискретных устройств и разработке программ. На самом простом, «бытовом» уровне, параллелизм – это два или больше отдельных действия, происходящих в одно время. Так как параллелизм постоянно присутствует в обычной жизни людей, возникает ощущение, что это понятие можно отнести к разряду первичных (неопределимых), поскольку оно понятно на интуитивном уровне. Тем не менее, научный анализ

понятий, непосредственно входящих в это «определение», – время, действие – привлекал внимание выдающихся мыслителей начиная с древнейших времен. Эта тема относится к разряду вечных, так как интересует не только ученых, но и писателей, и философов, и обычных людей.

Удивительно, насколько разнообразным оказывается понимание терминов, относящихся к проблеме параллелизма. Этот терминологический разнобой невозможно объяснить наличием научных школ редуccionистов и примитивистов.

Параллельное поведение с точки зрения редуccionиста сводится к недетерминированному последовательному поведению, которое рассматривается как чередование. Примитивная школа считает параллельное поведение элементарной формой поведения, которое отлично и более обще, чем последовательное поведение.

Основной аргумент редуccionизма состоит в том, что последовательное поведение лучше всего понято в терминах последовательного, потому что последнее мы понимаем естественно и инстинктивно. Каждый из нас осознает себя как последовательного агента. Наши органы, мускулы и клетки могут действовать одновременно, но наше управление ими в основном последовательно, а параллельное управление происходит ниже уровня нашего сознания, иначе было бы столь же просто вращать большими пальцами на руках в противоположных направлениях, как в одном. Примитивисты возражают тем, что сведение параллельного поведения к последовательному делает понимание более трудным. По аналогии, форма представления телевизионного изображения в виде одномерной последовательности, в которой оно достигает телевизора, делает это изображение не воспринимаемым зрением. Очевидно, эта форма не воспринимаема не из-за потери информации при квантовании картинки. Даже если редуccionизм – это единственный метод восприятия параллельного поведения человеком, лучше изучать параллелизм как явление природы, а не субъективное представление о нем.

Ясно, что разница в подходе не связана с особенностями мышления лидеров школ или их последователей. Стремление

к простоте модели здесь вступает в противоречие с объективной сложностью явления параллелизма, которая к тому же для многих не очевидна.

В последнее время популярность этой тематики возросла. Проблема параллелизма и времени стала интересовать широкие массы. Ряд идей в изучении этой проблемы имеет своим источником философию.

**Что такое время?** В результате попыток ответить на этот вопрос появилась знаменитая «проблема времени». Первое научное представление о времени возникло в эпоху эллинской научной мысли. Для эллинского решения «проблемы времени» характерны две черты: во-первых, ясное представление о времени физическом или математическом *как мере движения*, во-вторых, убеждение в безграничности времени. Эти результаты были сформулированы в виде знаменитых «апорий» (парадоксов) Зенона. Проблема времени анализировалась эллинами с помощью логики, не прибегая к «здоровому смыслу» или интуиции.

Метод Зенона состоял не в защите результата анализа, а в демонстрации того, что из утверждений отрицания результата возникают еще большие нелепости. В связи с этим Зенон выработал метод опровержения противников посредством серии вопросов. Отвечая на них, собеседник был вынужден прийти к самым необычным парадоксам, с необходимостью следовавшим из его взглядов. Этот метод, получивший название *диалектического* (греч. «разговаривать»), впоследствии применял Сократ.

Предположительно около 465 года до н. э. Зенон изложил свои идеи в книге, которая не сохранилась до нашего времени и известна лишь по трудам Платона, который родился на 60 лет позже Зенона, и по сообщениям ученика Платона Аристотеля. Результаты анализа Зеноном проблемы времени известны в виде апории «Стрела».

Согласно Аристотелю, в парадоксе о летящей стреле Зенон утверждает: любая вещь либо движется, либо стоит на месте. Однако ничто не может пребывать в движении, занимая пространство, равное ему по протяженности. В определенный момент движущееся тело (в данном случае стрела) постоянно находится на

одном месте. Следовательно, летящая стрела не движется. Симплиций формулирует парадокс в сжатой форме: «Летящий предмет всегда занимает пространство, равное себе, но то, что всегда занимает равное себе пространство, не движется. Следовательно, оно покоится». Таким образом, доказывается нереальность времени и движения. В более близкое нам время парадокс нереальности времени анализировали многие философы и логики, используя аппарат современной математической логики.

Критическим моментом, к которому привлекает внимание эта апория, является понятие места. Если движение и время существуют, то эти понятия существенно зависят от понятия места (в случае пространства) или события (в случае времени). Эти понятия анализируются в «метрической» атории. Аристотель излагал эту аторию следующим образом: «Если что-нибудь, будучи прибавлено к какой-нибудь вещи или отнято от нее, не делает эту вещь больше, соответственно меньше, тогда, по словам Зенона, оно не принадлежит к числу существующего, причем существующая, очевидно, понимается как величина телесная: ведь именно такая величина обладает бытием в полной мере». В изложении П. У. Бриджмена (1882–1961, американский физик, лауреат Нобелевской премии по физике 1946 г.) парадокс состоит в следующем: «если бы линию понимали так, что она буквально состоит из совокупности точек нулевой длины, а интервал времени представляет собою сумму неделимых мгновений событий, тогда уже само это понимание было бы парадоксальным».

Из-за трудностей, связанных с «метрической» аторией, предлагалось, сохраняя как базу для анализа понятие событий, в качестве основного временного отношения между событиями использовать прошлое, настоящее, будущее. Это так называемая *A*-теория времени Д. Э. МакТаггарта (1866–1925, английский философ). Понятия «прошлое, настоящее, будущее» задают способ, с помощью которого могут быть упорядочены события во времени.

События оказываются упорядоченными, если каждому из них приписать одно из свойств – «быть прошлым», «быть на-

стоящим», «быть будущим». Событиям, упорядоченным с помощью этих свойств, МакТаггарт дает название «*A*-ряда». Сущность времени может быть адекватно выражена *A*-рядом, поскольку она (эта сущность) состоит в изменении. Однако *A*-ряд является внутренне противоречивым. С одной стороны, различные *A*-свойства несовместимы друг с другом (никакое событие *t* не может быть и прошлым, и настоящим, и будущим), а с другой – каждое событие должно обладать всеми этими различными *A*-свойствами (например, событие *t*, которое является будущим, будет настоящим и прошлым). *A*-ряд часто упоминается как *стрела времени*, в котором прошлое является неизменным, будущее – неизвестным, настоящее неуловимо, но налицо. Любопытно, что проблема деления времени на прошлое, настоящее и будущее рассматривалась и эллинами. Известно высказывание Аристотеля: «Времени почти нет, ибо прошлого уже нет, будущего еще нет, а настоящее длится мгновение». Прошлого действительно нет, оно – было, так же, как и будущее – будет.

События во времени могут быть упорядочены другим способом с помощью отношений «раньше», «позже». Если события считать множеством, то понятиям «раньше», «позже», которые упорядочивают события, соответствуют антирефлексивные и транзитивные бинарные отношения (отношения частичного порядка). Событиям, упорядоченным с помощью свойств «раньше», «позже», МакТаггарт дает название «*B*-ряда». Математически *B*-ряд – это некоторая структура на событиях.

Эти исследования проблемы времени в античные времена выполнялись средствами логики. Само понятие логики с тех пор значительно уточнилось. Современные исследования в логике (IX Международный конгресс по логике, методологии и философии науки, Упсала, Швеция, 1991) разбивают логику на три основных раздела: математическая (символическая) логика; философская логика; неклассические (нестандартные) логики.

В этой номенклатуре «проблема времени» исследуется в последние годы в рамках неклассической (модальной и временной) логики. Обзор достигнутых результатов будет изложен в следующей главе.



**Физическое время.** В предыдущем параграфе проблема времени рассматривалась в разрезе меры изменений. Убеждение в безграничности времени было вновь в неизвестном для древности совершенстве введено в науку Галилео Галилеем (1564–1642). Через столетие после Галилея кембриджский профессор Исаак Ньютон ввел то понятие времени, которое наложило печать на всю научную мысль и научную работу в проблеме времени вплоть до наших дней. Ньютон подошел к проблеме времени как физик, потому что рассмотрение динамики механических устройств требовало учета времени. Для этого время нужно было считать физической величиной.

Физика относится к числу наук, экспериментальная основа которых – измерение. Измерение, как известно, представляет собой количественное сравнение некоторой величины с эталоном, который не изменяется от измерения к измерению. В физике физическая величина всегда определяется методом измерения и эталоном единицы измерений. А для измерения времени давно существовал известный прибор – часы. Таким образом, время для физика – это то, что измеряется часами («определение» времени, данное Эйнштейном). В 1686 г. Ньютон сформулировал представление о времени следующим образом: «Абсолютное, настоящее и математическое время само по себе и по своей природе равномерно *течет* относительно ко всему окружающему».

Для физика время – одна из числовых переменных в уравнениях динамики. Называя это время абсолютным, Ньютон подчеркивал «одинаковость» поведения этой переменной в уравнениях динамики *различных* физических тел. Такое время – это одна из координат траектории движения тела. От остальных (пространственных) координат ее отличает только то, что время необратимо, то есть нельзя изменить знак времени  $t$  на  $-t$ , обратить абсолютное время  $t$  вспять. Время похоже на пространство в том отношении, что события содержатся во времени почти так же, как объекты содержатся в пространстве.

Однако элементарное сознательное восприятие времени – поток, или перемещение момента «теперь» в будущее, – не имеет места в физическом описании реального мира, и это является

одной из наиболее трудных загадок физики. Совершенно не ясно, является ли это недостатком тех физических теорий, которые нам известны, или же ощущение течения времени представляет собой иллюзию. Предыдущий логический анализ понятия времени, подталкивает к мысли, что обратимость абсолютного времени представляет собой недостаток ньютоновской модели времени.

**Событие, действие.** Действие и событие представляют собой неотъемлемую часть действительности, и философы уже давно изучают их. Идея, что событие может быть в каком-то смысле более первичным, чем пространство и время, естественна для логического анализа времени. Бертран Рассел (1872–1970, английский математик и философ) при рассмотрении пространства и времени, использовал события как «исходный материал». Событие – это место времени, каждое из которых можно представить локализованным в конечной непрерывной области пространства–времени. При этом допускается наложение (пространственное) двух событий, но никакое событие не происходит больше чем один раз. Событие в трактовке Рассела является по сути своей понятием из области теории множеств. Как принято в теории множеств, все эти понятия статичны, и реально не включают понятия действия. Понятие действия как причины события необходимо, чтобы время, состоящее из событий, было не только описанием – статикой, но включало в себя движение. Когда мы говорим о действии, мы говорим также и об *агенте*, совершающем это действие. Если не интересоваться причинами событий, то действие и событие – синонимы. В. Ф. Турчин (1931, физик и кибернетик, создатель языка Рефал) пишет об этом так: «Событие – это действие, абстрагированное от агента». Введение агентов необходимо для описания динамики событий. Составное действие называется *процессом*.

В физической картине времени события вторичны и присутствуют для того, чтобы сделать явным течение времени, то есть, метафорически, события подобны поплавкам на текущей воде. Считая, что действия расположены в пространстве и времени, мы тем самым полагаем, что они менее фундаментальны, чем пространство и время.

Физика сосредоточена на *описании* мира во времени и пространстве, она оставляет, по крайней мере до настоящего времени, понятие агента скрытым. Понятие действия здесь производно от движения и является физической величиной количества *действия*, которая квантуется постоянной Планка  $h$ .

Шуточный логический парадокс «*Что было раньше — курица или яйцо?*» в физической теории разрешается в пользу первичности времени, в то время как в логической теории первичными предполагаются события – время состоит из событий.

**Поведение.** Если же мы интересуемся причинами событий, то описание мира относительно агента, вызывающего эти события, называется *поведением* агента. В психологическом словаре поведение определяется как «присущее живым существам взаимодействие с окружающей средой, опосредованное их внешней и внутренней активностью». «Опосредованное» – значит представленное в виде совокупности действий и соответствующих им событий.

Всеми формами *поведения*, поскольку они являются регулярными, или детерминированными, или воспроизводимыми, занимается *кибернетика*. В 1959 г. академик А. Н. Колмогоров писал: «...Сейчас уже поздно спорить о степени удачи Винера, когда он... в 1948 году выбрал для новой науки название кибернетика. Это название достаточно установилось и воспринимается как новый термин, мало связанный с греческой этимологией. Кибернетика занимается изучением систем любой природы, способных воспринимать, хранить и перерабатывать информацию и использовать ее для управления и регулирования. При этом кибернетика широко пользуется математическим методом и стремится к получению конкретных специальных результатов, позволяющих как анализировать такого рода системы (восстанавливать их устройство на основании опыта обращения с ними), так и синтезировать их (рассчитывать схемы систем, способных осуществлять заданные действия). Благодаря этому своему конкретному характеру кибернетика ни в какой мере не сводится к философскому обсуждению природы целесообразности в машинах и философскому анализу изучаемого ею круга явлений...». В киберне-

тике и информатике вслед за психологами под поведением понимается взаимодействие агента с внешним миром, определяемое индивидуальными особенностями и внутренней активностью агента, имеющее форму совокупности преимущественно внешних событий. «Под поведением понимается любое изменение объекта по отношению к окружающей среде» (А. Розенблют, Н. Винер, Дж. Бигелоу «Поведение, целенаправленность и телеология»). Теория информации играет важную роль в проблемах кибернетики, так как *«теория информации»* характеризуется по сути тем, что она всегда имеет дело с некоторым множеством возможностей; как ее исходные данные, так и ее окончательные выводы относятся всегда к *множеству* как таковому, а не к какому-либо отдельному элементу в нем. Часто не имеет значения даже замкнутость или открытость системы в энергетическом отношении – важна лишь та степень, в которой система подчиняется детерминирующим и управляющим факторам. Никакая информация, или сигнал, или детерминирующий фактор не могут пройти из одной части системы в другую, не будучи отмечены как *значимое событие*» (Эшби У. Р. «Введение в кибернетику»). В представлении и рассуждении об изменяющихся объектах время фундаментально, а термин «поведение» является эвфемизмом для смягчения связи понятия времени с философией.

Практическим инструментом проектирования поведение, состоящее из событий, стало благодаря С. К. Клини («Representation of Events in Nerve Nets and Finite Automata»). Клини пишет: «Наша теоретическая задача сформулирована в предположениях, не точно соответствующих действительности. Известная стратегия науки, сталкивающейся с реальной задачей, слишком сложной, чтобы ей можно было овладеть в целом, состоит в том, чтобы выбрать некоторую ограниченную область простых ситуаций и попытаться создать модель, соответствующую ей по крайней мере приблизительно. Считая эту модель заданной, следующий шаг состоит в возможно более полном овладении этой моделью. Не следует ожидать, что все особенности модели будут одинаково подходящими для действительности. Овладев пониманием модели, мы находимся в лучшем положении в ис-

следовании действительности, так как можем увидеть, как изменить модель и приспособить ее для более широкой ситуации».

Событием (элементарным) – *event* – Клини считает любое *свойство* внешних сигналов. Из элементарных событий строятся другие события, которые Клини называет «имеющими длительность» – *events definite of length (or duration)*. Множество всех событий образует алгебру, которая сейчас называется алгеброй Клини, или формальный язык, в котором буквами алфавита являются элементарные события, называемый в современной литературе регулярным. Длительные события могут рассматриваться как действия, то есть причины возникновения некоторых элементарных событий. Клини эту ситуацию изображал равенством. Равенства Клини задают *поведение* агента в виде набора пар вида «действие = событие». Из условий физической реализуемости *действие* должно произойти раньше *события*, то есть они упорядочены во времени. Пример системы равенств Клини приведен ниже. Это известный язык регулярных выражений.

$$\begin{aligned} \{x\} \{y\} &= R_1 \\ \{x \vee y\} \{x\} &= R_2. \end{aligned}$$

Следует отметить, что современная теория поведения дискретных устройств построена на основе состояний. Понятие *состояния* мира или его части знакомо всем, кто хоть как-то занимался наукой. В теории поведения дискретных устройств это понятие считается столь основополагающим, что нет необходимости и, вероятно, возможности его определения. В информатике преобладающий подход к моделированию событий состоит в том, чтобы рассматривать событие как изменение состояния. В этом представлении время представлено последовательностью состояний, каждое состояние представляет мгновенный «снимок» мира. Действия считаются функциями, отображающими одно состояние в другие.

Система равенств Клини полностью эквивалентна модели конечного автомата. Для вышеприведенной системы представление поведения в состояниях показано в таблице. Выражения в клетках таблицы следует понимать как абстрактные символы состояний «0», «1 ∨ 3 ∨ 4» и т. д.

**Таблица переходов и выходов автомата Мили,  
эквивалентного системе равенств Клини**

	$(R_1)$	$(R_1, R_2)$	$(R_1)$	$(R_2)$	$( )$
	0	$1 \vee 3 \vee 5$	$2 \vee 4$	$3 \vee 5$	
$x$	$1 \vee 3 \vee 5$	$1 \vee 3 \vee 5$	$3 \vee 5$	$3 \vee 5$	$3 \vee 5$
$y$	$2 \vee 4$	$2 \vee 4$	$2 \vee 4$	4	4

Алгебры Клини позволяют описывать поведение так, что понятие *одновременно* не используется. В этом формализме мы абстрагируемся от возможности одновременности событий (т. е. не рассматриваем такие ситуации). Следовательно, при описании поведения единственного объекта параллелизм не является необходимым. Параллелизм неизбежен при анализе и проектировании изменяющихся объектов, обладающих структурой. В этом случае мы вынуждены рассматривать поведение более чем одного агента.

**Взаимодействие.** Так как параллелизм связан со структурой, то возникает подозрение в нефундаментальности этого понятия. В описании поведения параллелизм появляется при выборе структурной формы описания. Так как это одна из *возможных* форм, то для обоснования фундаментальности требуется показать *необходимость* параллелизма при анализе и проектировании реальных устройств. Формально-логически необходимость доказывается легко, так как структурная форма описания является результатом проектирования устройств и, следовательно, необходима в процессе проектирования.

В качестве примера реальной системы с параллелизмом рассмотрим работу интерфейса RS-232 – СОМ-порта персонального компьютера типа IBM PC. Система состоит из двух подсистем: приемника и передатчика, каждая из которых работает строго последовательно. Приемник и передатчик связаны двухпроводной электрической линией для передачи информации от передатчика приемнику (рис. 1).

При функционировании системы для передачи одного байта в передатчике происходит последовательность событий передачи данных, а в приемнике – последовательность приемов. Из

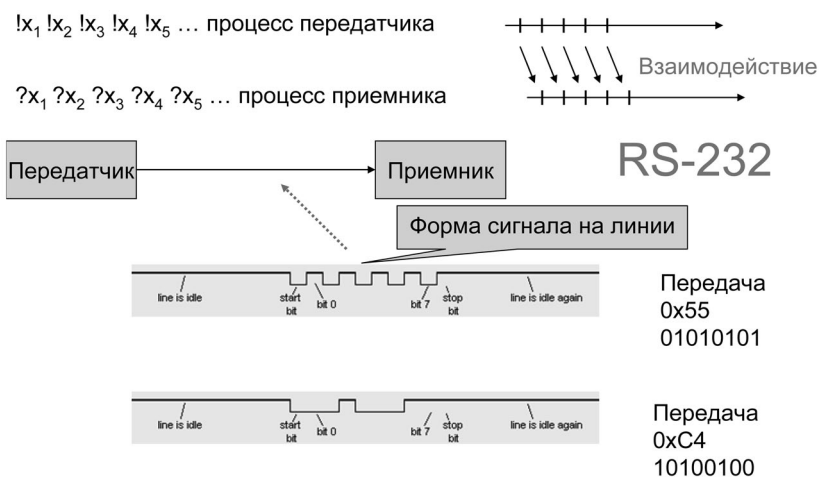


Рис. 1. Передача данных в интерфейсе RS-232

сути задачи возникает необходимость рассмотрения совместно-го поведения именно двух агентов, так как функционирование всей системы в целом тривиально – внешние проявления процесса передачи отсутствуют. Для передачи данных нужно, чтобы событие передачи бита в передатчике происходило *одновременно* с событием приема в приемнике.

Как понимать одновременность событий? Если время составлено из событий, то как события могут быть одновременными? Возможность одновременности прямо ведет к утверждению, что время не существует. Может быть мы здесь столкнулись с логическим парадоксом, который практического значения не имеет, и нужно просто попробовать предложить реально осуществимые устройства? Каждый, кто немного разбирается в электронике, может предложить устройство, осуществляющее передачу данных в интерфейсе RS-232. Задачу решает регистр сдвига длиной в два байта, одна половина и вход регистра представляют собой передатчик, а вторая половина и выход регистра – приемник. Трудность здесь в том, что регистр сдвига – синхронная схема. Для того чтобы через разряды регистра информация продвигалась, требуется на каждый триггер регистра подавать сигнал синхронизации. Однако половины регистра по

условиям задачи находятся в разных устройствах, следовательно, для обеспечения общего сигнала синхронизации требуется или среда для его передачи (еще одна пара проводов) или возможность синхронизировать часы (синхрогенераторы) приемника и передатчика. COM-порт персонального компьютера типа IBM PC использует микросхему UART (Universal Asynchronous Receiver/Transmitter – универсальный асинхронный приемопередатчик). Каждая такая микросхема содержит свой синхрогенератор, стабилизируемый стандартным для RS-232 кварцевым резонатором. Синхронизации фазы независимых генераторов, осуществляется за счет специального поведения приемника и передатчика перед и после приема байта. Описание этого поведения носит название протокола интерфейса RS-232. Возникает задача: возможна ли передача данных (взаимодействие) без использования часов и как этого достигнуть? Как показывает этот пример, одновременность вызывает трудности и на понятийном и на техническом уровне.

Взаимодействие в этом примере рассматривалось как событие. Если события считаются мгновенными и неделимыми, то в этом случае акт взаимодействия рассматривается как одно событие, требующее участия всех взаимодействующих в этом акте процессов. Рассматривая взаимодействие как события, которые реализуются агентами совместно, приходится признать, что оно должно иметь продолжительность, так как агенты могут быть пространственно разделены, и коммуникация между ними не может происходить мгновенно.

*Параллельные системы* традиционно разбиваются на системы *без взаимодействия*, в которых процессы независимы и конкурируют за ресурсы и системы *с взаимодействием* (реактивные), состоящие из ряда процессов, которые сотрудничают, чтобы достигнуть общей цели. В качестве типичного примера параллельных систем без взаимодействия на рис. 2 приведена логика работы процессора Intel Pentium 4 технологии Hyper-Threading. Диаграммы потоков *A* и *B* указывают занятость устройств процессора при выполнении последовательности команд (выполнение команды требует одного цикла).