КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ И МОДЕЛИРОВАНИЕ 2017 Т. 9 № 4 С. 621−638
DOI: 10.20537/2076-7633-2017-9-4-621-638

УДК: 519.7, 612.821, 519.8

Тесты проверки параллельной организации логических вычислений, основанные на алгебре и автоматах

А. В. Коганов

Научно-исследовательский институт системных исследований (ФНУ ГНЦ НИИСИ РАН),
Россия, 117218, Москва, Нахимовский пр., 36, корп. 1

E-mail: koganow@niisi.msk.ru

Получено 21.11.2016, после доработки — 23.05.2017.
Принято к публикации 20.06.2017.

Работа продолжает опубликованные ранее исследования по способности человека повышать производительность обработки информации при параллельном выполнении нескольких логических операций заданного вида. В статье предлагаются новые тесты, позволяющие выявлять указанную способность человеческого мозга в серии предъявлений. Производительность человека определяется средним количеством информации, которую обрабатывает человек в единицу времени, решая серию тестовых задач. Сложность задачи в каждой серии тестов определяется средним количеством логических операций, которые надо выполнить для решения с учетом статистических свойств серии задач. Тесты строятся таким образом, чтобы сложность контролировалась. Изучается зависимость производительности испытуемого от сложности задач в серии. Если человек использует последовательный алгоритм решения и не меняет скорости выполнения логических операций, то производительность не зависит от сложности и среднее время решения задачи в серии примерно пропорционально сложности. Если скорость выполнения операций растет с повышением сложности (растет концентрация внимания), то увеличивается и производительность. Тот же эффект возникает, если человек при достаточно высокой сложности задачи начинает выполнять несколько логических операций одновременно (параллельные вычисления). Для контроля причин роста производительности строятся контрольные тесты на том же классе логических операций, в которых параллельная организация счета малоэффективна. Если рост производительности наблюдается как на основных, так и на контрольных тестах, то причиной роста производительности является увеличение быстродействия. Если же на контрольных тестах нет роста производительности, а на основных тестах рост имеется, то причиной роста является параллельный счет. С точки зрения теории операций это означает использование одновременной работы нескольких процессоров, каждый из которых в единицу времени перерабатывает не более некоторого известного числа элементов входных данных или промежуточных результатов (производительность процессора). В данной статье предлагается система тестов, в которой используется аппарат универсальных алгебр и теории автоматов. Работа является продолжением цикла работ по исследованию способностей человека к параллельным вычислениям. Ранее использованные тесты в экспериментах показали эффективность методики. Основные предыдущие публикации на эту тему приведены в списке литературы. Задачи в новых предлагаемых тестах можно описать как вычисление результата серии последовательных однотипных операций из некоторой специальной алгебры. Если операция ассоциативная, то с помощью удачной группировки вычислений можно эффективно применить параллельный счет. Анализируется зависимость времени решения задачи от сложности. Чтобы выявлять ситуации, когда человек увеличивает быстродействие одного процессора по мере роста сложности, требуется предъявить серии задач с похожими операциями, но в неассоциативной алгебре. Для таких задач параллельный счет малоэффективен с точки зрения отношения прироста производительности к увеличению числа процессоров. Так формируется контрольная группа тестов. В статье рассмотрен еще один класс тестов, основанных на расчете траектории состояния заданного формального автомата при задании входной последовательности. Исследован специальный класс автоматов (реле), конструкция которых влияет на эффективность параллельного расчета финального состояния. Для всех тестов оценивается эффективность параллельного счета. Эксперименты с новыми тестами не входят в данную статью.

Ключевые слова: параллельный счет, психологический тест, алгебра, ассоциативность, формальный автомат

Работа выполнена при поддержке Российского фонда фундаментальных исследований, проект 16-01-00166а.

COMPUTER RESEARCH AND MODELING
2017 VOL. 9 NO. 4 P. 621−638
DOI: 10.20537/2076-7633-2017-9-4-621-638

UDC: 519.7, 612.821, 519.8

The tests for checking of the parallel organization in logical calculation which are based on the algebra and the automats

A. V. Koganov

Scientific research institute of system analysis (FSE GSC NIISI RAN / FSE GSC SRISA RAS),
Nakhimovsky st. 36, corp. 1, Moscow, 117218, Russia

E-mail: koganow@niisi.msk.ru

Received 21.11.2016, after completion — 23.05.2017.
Accepted for publication 20.06.2017.

We build new tests which permit to increase the human capacity for the information processing by the parallel execution of the several logic operations of prescribed type. For checking of the causes of the capacity increasing we develop the check tests on the same logic operations class in which the parallel organization of the calculations is low-effectively. We use the apparatus of the universal algebra and automat theory. This article is the extension of the cycle of the work, which investigates the human capacity for the parallel calculations. The general publications on this theme content in the references. The tasks in the described tests may to define in the form of the calculation of the result in the sequence of the same type operations from some algebra. If this operation is associative then the parallel calculation is effectively by successful grouping of process. In Theory of operations that is the using the simultaneous work several processors. Each processor transforms in the time unit the certain known number of the elements of the input date or the intermediate results (the processor productivity). Now it is not known what kind elements of date are using by the brain for the logical or mathematical calculation, and how many elements are treating in the time units. Therefore the test contains the sequence of the presentations of the tasks with different numbers of logical operations in the fixed alphabet. That is the measure of the complexity for the task. The analysis of the depending of the time for the task solution from the complexity gives the possible to estimate the processor productivity and the form of the calculate organization. For the sequence calculations only one processor is working, and the time of solution is a line function of complexity. If the new processors begin to work in parallel when the complexities of the task increase than the depending of the solution time from complexity is represented by the curve which is convex at the bottom. For the detection of situation when the man increases the speed of the single processor under the condition of the increasing complexity we use the task series with similar operations but in the no associate algebra. In such tasks the parallel calculation is little affectivity in the sense of the increasing efficiency by the increasing the number of processors. That is the check set of the tests. In article we consider still one class of the tests, which are based on the calculation of the trajectory of the formal automat state if the input sequence is determined. We investigate the special class of automats (relay) for which the construction affect on the affectivity of the parallel calculations of the final automat state. For all tests we estimate the affectivity of the parallel calculation. This article do not contained the experiment results.

Keywords: parallel calculation, psychological test, algebra, associativity, formal automat

Citation: *Computer Research and Modeling*, 2017, vol. 9, no. 4, pp. 621–638 (Russian).

This work was supported by RFBR grant no. 16-01-00166a.

1. Введение

Эта статья продолжает серию работ, целью которых является выявление принципов организации работы человеческого мозга при решении задач, связанных с выполнением серии логических операций в уме без использования записей или технических устройств. Исходной установкой для человека является требование решать задачу как можно быстрее, но без ошибок. В этих исследованиях делалась попытка определить способ адаптации мозга к увеличению сложности предлагаемых задач. Под сложностью задачи понимается минимальное число операций заданного типа, которые необходимо выполнить для гарантированного получения правильного ответа. В современной технике известно несколько стратегий адаптации с целью снизить суммарное время решения серии задач при увеличении их сложности. Простейшая стратегия заключается в игнорировании времени решения задачи при концентрации на правильном вычислении необходимых операций в нужной последовательности. При такой стратегии производительность работы процессора постоянная, и время растет пропорционально сложности задачи. Вторая стратегия заключается в увеличении скорости вычислений по мере роста сложности задачи. При этом производительность растет обратно пропорционально времени выполнения одной операции. Среднее время решения задачи для этой стратегии является выпуклой снизу функцией от сложности. Третья стратегия заключается в подключении нескольких процессоров для решения сложных задач. Она называется параллельными вычислениями. Эффективность такой стратегии зависит от структуры задачи. Она измеряется отношением коэффициента выигрыша во времени решения к числу задействованных в решении процессоров. Эта зависимость называется характеристической функцией задачи. Если характеристическая функция близка к единице, то задача допускает эффективное запараллеливание. Если эта функция мала, то для данной задачи параллельный счет неэффективен. Наконец, возможна стратегия снижения надежности решения с ростом сложности. Она выражается в попытках угадать верный ответ при выполнении только некоторой части нужных операций. Такая стратегия может быть достаточно эффективной, если статистика правильных ответов допускает хорошее вероятностное прогнозирование. Таким образом, возможность применения этой стратегии зависит от выбора последовательности задач.

Предлагаемая методика выявления стратегии адаптации для отдельного человека заключается в предъявлении ему нескольких серий задач, с контролем среднего времени решения и частоты ошибочных ответов. Задачи делятся на два класса. Для одних тестов параллельный счет эффективен, а для других неэффективен. Задачи каждого класса разбиты на серии. В каждой серии сложность задач фиксирована относительно некоторого базиса операций. Тесты выбираются таким образом, чтобы базисный набор операций, необходимых для их решения, был очевиден и явно задан в постановке задачи. Число задач в каждой серии достаточно велико для оценки вероятности ошибки у данного испытуемого. Перед началом основного тестирования испытуемый проходит тренировку на специальной серии задач различной сложности.

Если испытуемый использует стратегию первого вида (последовательный счет с постоянным быстродействием), то время решения задачи у него растет пропорционально сложности для задач обоих классов. При использовании стратегии второго типа (последовательный счет с ускорением вычислений) в задачах обоих классов кривая зависимости среднего времени решения от сложности выпукла снизу. При стратегии третьего вида (подключение параллельных вычислений) указанная кривая будет выпукла снизу для класса задач, допускающих эффективную параллельность, и будет линейной для задач с неэффективной параллельностью.

Сложнее выявлять влияние четвертой стратегии (неполная обработка информации). Серии задач подбираются таким образом, чтобы угадывание ответа влекло за собой достаточно большую вероятность ошибки. Простейший способ оценки таких результатов заключается в их отбрасывании и повторном прохождении теста. Но в предыдущих работах был разработан метод оценки пропускной способности мозга при решении теста. Эта оценка учитывает не только среднее время решения задач, но и частоту ошибок. Если ошибок много, то оценка резко снижается даже при малом времени решения задач. Это позволило достаточно надежно выявлять стратегию адаптации к сложности, при этом используется следующий принцип.

Если пропускная способность не зависит от сложности для тестов обоих классов, то используется первая стратегия. Если пропускная способность растет на задачах обоих классов, то используется вторая стратегия. Если пропускная способность растет на задачах первого класса, но остается постоянной на задачах второго класса, то используется третья стратегия. Если нет определенных тенденций или происходит снижение пропускной способности с ростом сложности, то используется стратегия четвертого класса.

Ранее опубликованы работы, в которых описано тестирование группы испытуемых на способность выполнять параллельно несколько логических операций при вычислении логических функций в уме [Коганов и др., 1971; Коганов, 1972, 2001, 2010; Коганов, Злобин, Ракчеева, 2013, 2014].

В основе лежали две задачи. Первая задача состояла в вычислении максимума в числовой таблице. Эта задача допускает очень эффективное использование параллельных вычислений. Вторая задача заключалась в прослеживании траектории по числовой таблице, в которой требовалось переходить из очередной клетки в ту клетку окрестности, где стояло наибольшее число. Ответом являлось число, записанное в финальной клетке траектории. У этой задачи было несколько модификаций, отличающихся зоной поиска максимального числа для определения правильного продолжения траектории. В этой задаче эффективность подключения дополнительных процессоров очень низкая, если измерять отношение выигрыша во времени решения к числу подключенных процессоров. Поэтому наиболее вероятно последовательное вычисление траектории. Проведенные эксперименты показали, что определенный процент испытуемых использует параллельный счет, что снижает время решения первой задачи. Другая группа испытуемых при увеличении размера таблицы увеличивает скорость счета в обеих задачах, но не переходит к параллельному счету. Третья часть испытуемых действует в обеих задачах последовательно и с постоянной скоростью вычислений, независимо от размера таблиц. В проведенных тестовых исследованиях эти типы испытуемых были примерно равными. Кроме того, некоторая часть испытуемых показала высокую вероятность ошибки в расчетах, что не позволило достоверно отнести их к одному из типов организации счета. Возможно, в этом случае имела место попытка угадать ответ без полного анализа исходных данных. В обеих задачах элементарной логической операцией являлось вычисление максимума из двух или нескольких чисел.

Полученные результаты явились поводом для разработки новых тестов, где расчеты будут основаны на других элементарных операциях. При этом остается неизменной основная идея сопоставления времени решения двух серий задач, в одной из которых параллельность эффективна, а в другой — нет. Ниже предлагаются три задачи, две из которых образуют такие серии, а третья задача может рассматриваться как задача промежуточного типа. Параллельность в ней эффективна, но ценой резкого усложнения базовых вычислительных операций.

В первой задаче надо определить, сколько символов «1» содержится в таблице, заполненной цифрами и буквами. Эта задача допускает эффективное параллельное решение, когда параллельно оценивается число искомых символов в разных участках таблицы (с последующим сложением результатов). Сложение нескольких чисел тоже можно реализовать параллельно.

Во второй задаче требуется произвести вычисление траектории по прямоугольной таблице, в столбцах которой стоят перестановки номеров строк и задана начальная клетка в самом левом столбце. Надо переходить в соседний справа столбец, в клетку строки, номер которой указан в предыдущей клетке траектории. Фактически это вычисление числа, в которое будет переведен номер указанной слева позиции, для перестановки, равной последовательному произведению слева направо перестановок, стоящих в столбцах. Эта задача допускает параллельный счет только при очень низкой эффективности использования дополнительных процессоров и при большом их числе. Поэтому следует ожидать последовательное вычисление ответа человеком.

Третья задача состоит в вычислении взвешенной суммы единиц в таблице, заполненной нулями и единицами. Числа считываются по строкам таблицы, слева направо и сверху вниз (как одна объединенная строка). При этом в сумме число единиц в очередном массиве единиц умножается на длину предшествующего массива нулей. Эта задача допускает эффективное параллельное решение путем априорного разбиения таблицы на несколько зон, внутри которых происходит последовательное считывание чисел и которые обрабатываются одновременно, с последующим вычислением суммы при склейке результатов разных зон. Операция склейки достаточно сложная и требует учета расположения массивов на границе зон. Последовательное решение этой задачи не требует усложнения операций. В этом смысле данная задача занимает промежуточное положение в эффективности параллельных вычислений.

Следует заметить, что в практике построения параллельных машинных алгоритмов ситуация промежуточного типа встречается весьма часто, когда эффективность параллельного счета имеет место, но требует очень сильного усложнения алгоритма в плане объединения результатов разных процессоров. Поведение человека в подобных задачах пока изучено слабо.

Можно заметить, что задачи такого рода сводятся к вычислению некоторых выражений в алгебре чисел или в алгебре перестановок. Поэтому дополнительно будет рассмотрен общий подход к построению тестов указанного вида, с помощью вычислений выражений в алгебрах общего вида. В зависимости от наличия ассоциативности они либо допускают, либо не допускают параллельный расчет. Предлагается один из возможных классов алгебр относительно простых для обучения испытуемых. Однако сам процесс обучения вычислениям в этих экспериментах должен занимать значительную часть времени работы с испытуемым, поскольку требуется выработка автоматизма. Это усложняет организационную сторону таких исследований.

Другой класс тестов связан с вычислением финального состояния заданного формального автомата с начальным состоянием при подаче на него заданной входной последовательности. Эти задачи практически всегда требуют последовательного вычисления. Исключения составляют только некоторые простые автоматы, финальные состояния которых определяются простыми признаками входной последовательности. Поэтому данный тип задач можно использовать как для контроля скорости последовательных вычислений в зависимости от сложности задачи (число состояний автомата, длина входной последовательности и т. п.), так и для обнаружения параллельности вычислений. Будут предложены автоматы, которые можно рассматривать как обобщения реле и триггера. В одних модификациях они допускают эффективную параллельность расчета, а в других требуют последовательного вычисления.

Завершая введение, отметим, что в настоящее время имеется много работ с анализом эффективности параллельной реализации алгоритмов. Все известные автору работы посвящены исследованиям достаточно сложных задач, относящихся к практическим приложениям современной информатики и кибернетики, например [Popov et al., 2010]. Однако специфика описанной выше проблемы тестирования связана с поиском простых задач, доступных для просчета в уме человеку с минимальной подготовкой. В области инженерной и когнитивной психологии проводится большая работа по анализу поведения человека при необходимости решать несколько задач на одном отрезке времени [Fischer, Plessow, 2015]. Достоверных данных о строго параллельном способе решения получить не удалось, однако установлено, что при высоком уровне помех человек переходит на последовательную стратегию решения задач (по очереди), а при снижении опасности ошибки использует стратегию параллельной или псевдопараллельной работы, распределяя свое внимание между задачами. Интересно, что время решения при этом обычно возрастает, но снижается напряженность умственных процессов, которая регистрируется объективными методами. Возможность параллельной стратегии сильно зависит от разделения ресурсов мозга между задачами. Легче всего реализовать параллельно задачи в разных модальностях (например, слушать музыку и танцевать). Отличие наших исследований заключается в строго логическом характере тестовых задач (модальность символьной обработки) и отсутствии явного разбиения задания на несколько задач. Такое разбиение человек производит сам, когда переходит к параллельной стратегии решения. Напряжение внимания при этом явно возрастает, но время решения сокращается. Замечено, что испытуемые, показавшие параллельную работу, как правило, не могут объяснить способ решения, а при последовательной стратегии испытуемые четко объясняют свой метод. Видимо, рефлексия возможна только при одном центре активности мозга.

Ниже подробно анализируется каждая из описанных выше задач. Подготовка теста к эксперименту требует решения еще нескольких технических вопросов, которые мы в этой работе не рассматриваем.

2. Задача подсчета символов

Задача предъявляется на экране компьютера удобного для испытуемого формата.

Исходная информация предъявляется в форме таблицы, содержащей m строк и n столбцов, всего nm клеток. Клетки заполнены буквами и цифрами. Несколько (обозначим a) правых клеток последней (нижней) строки могут быть пустыми. Всего заполненных клеток $N = nm - a$.

Ответом задачи является число F в диапазоне от 0 до N, равное количеству символов «1» в таблице.

Сложностью задачи назовем число N заполненных клеток таблицы.

Серия заданной сложности — это последовательное предъявление испытуемому нескольких задач этой сложности. Серия формируется так, чтобы не было мест предпочтительного расположения искомого символа и статистической группировки ответов в некоторой узкой зоне значений.

Например, алгоритм формирования очередной задачи может быть следующим. Обозначим $rand\{M\}$ операцию выбора одного элемента из указанного конечного множества M с равномерным распределением вероятности по множеству M и статистически независимо от всех других реализаций этой операции по любым множествам.

Тогда формирование очередной задачи в серии происходит по указанным ниже этапам.

Этап 1. $F = rand\{1;\ldots;N\}$.

Этап 2. Формирование мест расположения искомого символа. Всю заполненную часть таблицы будем рассматривать как один массив длины N с последовательной нумерацией элементов, не рассматривая разбиения на строки.

$$x(1) = rand\{1;\ldots;N\};$$

$$x(l+1) = rand\left\{\{1;\ldots;N\} \setminus \{x(1);\ldots;x(l)\}\right\}, \quad l = \overline{1, N-1}.$$

Обозначим полученное множество номеров клеток: $X = \left\{x(1);\ldots;x(F)\right\}$.

Если для клетки $i = 1,\ldots,m$ — номер строки, $j = 1,\ldots,n$ — номер столбца, то номер клетки при сквозной нумерации для всех этапов —

$$x = (i-1)n + j. \tag{2.1}$$

Наоборот, если x — номер клетки (i, j) при сквозной нумерации, то соответствующие координаты в таблице определяются как

$$i = \lfloor x/n \rfloor + 1, \tag{2.2}$$

$$j = x - n(i-1). \tag{2.3}$$

Этап 3. Заполнение непустых клеток таблицы. Обозначим Alp множество всех символов алфавита, а множество символов без искомого символа обозначим как Alp', $Alp' = Alp \setminus \{1\}$. Тогда символы в клетках (i, j) таблицы определяются так:

$$y(i,j) = \begin{cases} 1 & \Leftarrow (i-1)n + j \in X, \\ rand\{Alp'\} & \Leftarrow \text{иначе}, \end{cases}$$

где $i = 1,\ldots,m$ — номер строки, $j = 1,\ldots,n$ — номер столбца, $x = (i-1)n + j$ — номер клетки при сквозной нумерации этапа 2.

Клетки с $x > N$ остаются пустыми. Всего их $a = nm - N$. Размер таблицы лучше подбирать так, чтобы все пустые клетки располагались в последней строке и их было по возможности мало. При этом надо учитывать и удобство расположения таблицы на экране.

Число предъявляемых задач в серии сложности N обозначим как $K(N)$. Можно использовать запись $K(N) = K$, если длины всех серий в эксперименте одинаковые.

При каждом предъявлении задачи регистрируются время решения задачи, выданный ответ, его совпадение или несовпадение с правильным ответом.

3. Задача произведения перестановок

Задача предъявляется на экране компьютера удобного для испытуемого формата.

Исходная информация предъявляется в форме таблицы, содержащей m строк и n столбцов, всего nm клеток. Каждая клетка заполнена двумя числами существенно различных форматов. Слева мелким шрифтом обозначено число, равное номеру строки данной клетки. Рядом крупным шрифтом обозначено число, одно из множества $\{1;\ldots;m\}$. Таким образом, заполнение клетки (i, j) имеет вид $\left[i, \omega_j(i)\right]$. Требуется, чтобы числа $\omega_j(1),\ldots,\omega_j(m)$ в указанном порядке образовывали перестановку ω_j чисел $\{1;\ldots;m\}$, что означает отсутствие повторов среди этих чисел. Одна из клеток самого левого столбца отмечена, например, яркой точкой слева от таблицы.

Ответом задачи является значение $\omega_n(i_n)$, где номер строки i_n вычисляется по таблице указанным ниже способом.

i_1 — отмеченная в первом (левом) столбце строка.

Если вычислено значение i_j, то рекуррентно

$$i_{j+1} = \omega_j(i_j), \quad j = 1,\ldots,n-1. \tag{3.1}$$

Заметим, что математически ответ задачи равен значению одной позиции произведения перестановок

$$\omega_n(i_n) = \left(\omega_n \cdot \ldots \cdot \omega_1\right)(i_1). \tag{3.2}$$

Но указанный рекуррентный процесс (3.1) вычислительно значительно экономнее, чем вычисление полного произведения перестановок (3.2).

Сложностью задачи назовем число столбцов. Число предъявляемых задач в серии сложности n обозначим как $K(n)$. Можно использовать запись $K(n) = K$, если длины всех серий в эксперименте одинаковые. Формирование каждой задачи из серии должно обеспечить статистическую независимость и равномерное распределение ответов. Это можно обеспечить следующей пошаговой процедурой. Для серии задаются параметры m и n.

Этап 1. Задание начального номера строки:

$$i_1 = rand\{1;\ldots;m\}.$$

Этап 2. Для каждого столбца $j = 1,\ldots,n$ рассчитывается перестановка ω_j:

$$\begin{aligned}
&\omega_j(1) = rand\{1;\ldots;m\};\\
&\omega_j(i+1) = rand\{\{1;\ldots;m\} \setminus \{\omega_j(1);\ldots;\omega_j(i)\}\};\\
&i = 1,\ldots,m-1.
\end{aligned}$$

При каждом предъявлении задачи регистрируются время решения задачи, выданный ответ, его совпадение или несовпадение с правильным ответом.

Первое число в клетке используется для облегчения поиска нужной строки в очередном столбце при движении по таблице слева направо. После считывания значения $\omega_j(i_j)$ надо найти клетку с координатами $\left(\omega_j(i_j), j+1\right)$ в таблице. В этой клетке записаны два числа $\left(\omega_j(i_j), \omega_{j+1}(i_{j+1})\right)$, требуется прочесть значение $\omega_{j+1}(i_{j+1})$ для продолжения решения задачи или для выдачи ответа, если выполнено условие $j+1=n$ (последний столбец).

4. Задача взвешенной суммы

Задача предъявляется на экране компьютера удобного для испытуемого формата.

Исходная информация предъявляется в форме таблицы, содержащей m строк и n столбцов, всего nm клеток. Клетки заполнены цифрами «0» и «1». Несколько (обозначим как a) правых клеток последней (нижней) строки могут быть пустыми. Всего заполненных клеток $N = nm - a$. Всю заполненную часть таблицы будем рассматривать как один массив длины N с последовательной нумерацией элементов, не рассматривая разбиения на строки, как в (2.1).

Ответом задачи является число F, равное взвешенной сумме

$$F = V_1 W_1 + \ldots + V_s W_s, \tag{4.1}$$

где s — число массивов идущих подряд единиц, W_i — длина i-го массива единиц (в порядке чтения слева направо), V_i — длина предшествующего этому массиву массива нулей. В частности, если последовательность исходных данных начинается с «1», то $V_1 = 0$.

Сложностью задачи назовем число N заполненных клеток таблицы.

Серия заданной сложности — это последовательное предъявление испытуемому нескольких задач этой сложности. Серия формируется так, чтобы не было статистической группировки ответов в некоторой узкой зоне значений.

С учетом отсутствия у испытуемых предварительной статистики ответов в данном случае можно ограничиться генерацией реализации последовательности Бернулли. Используя нумерацию клеток (2.1), можно формировать исходные данные каждого предъявления задачи по формуле

$$y(x) = rand\{0;1\}, \quad x = \overline{1,N}. \tag{4.2}$$

Алгоритм последовательного вычисления этой задачи очевиден. Время последовательного решения задачи:

$$T(n,1) \cong (B+r)N, \tag{4.3}$$

где B — коэффициент скорости распознавания символов и суммирования их числа в массивах, r — случайный коэффициент, зависящий от сложности умножения и суммирований произведений. На производительность решателя влияет его статистическое среднее в серии задач.

Для упрощения арифметических операций можно рассмотреть модификацию задачи (4.1), в которой меняется определение коэффициентов:

$$F = v_1 W_1 + \ldots + v_s W_s, \tag{4.4}$$

$$v_i = \begin{cases} 1 \Leftarrow V_i = 2K, \ K \in \overline{0,\infty}, \\ -1 \Leftarrow V_i = 2K+1, \ K \in \overline{0,\infty}, \end{cases} \tag{4.5}$$

что означает суммирование только тех единиц, перед массивом которых стоит массив нулей четной длины, и вычитание остальных единиц. Это упрощение делает случайный коэффициент r в формуле (4.3) небольшим по отношению к значению B.

Попытка решить эту задачу параллельно на нескольких (P) процессорах требует априорного разбиения считывания последовательности на соответствующее число P сегментов.

Но после этого возникает задача склейки массивов нулей и единиц, попавших в разные сегменты. Надо учесть, что один массив может быть разбит на несколько идущих подряд смежных сегментов. При этом в крайних сегментах этой совокупности массив может занимать только часть сегмента, но в центральных сегментах он занимает весь сегмент. Это подразумевает достаточно сложный обмен данными между процессорами. Можно показать, что этот процесс склейки массивов требует времени пропорционально числу процессоров, но с достаточно большим коэффициентом, учитывая сложность вычислений.

$$T(n,P) \cong (B+r)(N/P) + B'P. \tag{4.6}$$

При большой сложности задачи $N >> 1$ формула (4.6) дает выигрыш времени относительно (4.3) для значения $P \geq 2,$ но ценой очень сильного усложнения алгоритма решения.

Если использовать метод динамического разбиения на сегменты, так, чтобы избежать разбиения массивов, то сама эта процедура требует времени, пропорционального сложности задачи. Однако коэффициенты B'' и B' могут оказаться маленькими, учитывая простоту задачи поиска начала массива и отсутствие проблемы склейки.

$$T(n,P) \cong B''N + (B+r)(N/P) + B'P. \tag{4.7}$$

Какой способ решения выберет испытуемый в этой ситуации, априори не понятно.

5. Автоматные тесты параллельности

В основе этих тестов лежит понятие формального автомата. Автомат представлен схемой на экране, на которой изображен граф переходов автомата. Вершины этого графа изображают состояния автомата. Это кружки, в которые вписаны буквы. Буква в каждом кружке уникальна и обозначает идентификатор состояния. Ребра графа направленные и изображены стрелками с пометками. Пометки обозначают символ входа, которому соответствует переход из одного состояния в другое по направлению стрелки. Кроме схемы автомата задача содержит конечную последовательность входных символов, вначале которой указан идентификатор начального состояния автомата. Ответом задачи является идентификатор того состояния, в которое придет автомат после отработки этой входной последовательности из указанного начального состояния. При фиксированной схеме автомата сложность задачи будем измерять длиной входной последовательности.

Серия заданной сложности — это последовательное предъявление испытуемому нескольких задач этой сложности. Серия формируется так, чтобы не было статистической группировки ответов в некоторой узкой зоне значений.

В общем случае эффективного параллельного алгоритма для решения такой задачи не существует. Это связано с невозможностью заранее определить, из какого состояния будет действовать неначальный отрезок входной последовательности. Задача вычисления траектории по таблице перестановок, описанная выше, и задача вычисления траекторий, использованная в предыдущих экспериментах [1, 2], могут быть описаны как вычисление траектории по пространству состояний автомата.

Однако для некоторых автоматов вычисление финального состояния может быть получено как вычисление определенных характеристик входной последовательности, которые допускают эффективный параллельный расчет. Простейшим примером является автомат, который переходит в поглощающее финальное состояние при длине последовательности больше некоторого фиксированного числа. Тогда обнаружение достаточно большой длины заданной последовательности входов позволяет сразу указать ответ задачи. Имеются и более сложные примеры автоматов с эффективной параллельностью расчета состояния. Использование автоматных тестов требует от разработчика анализа множества последовательностей, приводящих в каждое из состояний. Соответствующая теория называется представлением событий состояниями автомата. Варьируя автоматы, можно строить формально похожие тесты как на способность к параллельному счету, так и на контроль скорости последовательного расчета. Как указывалось выше, для полного анализа способа решения задачи требуются тесты обоих типов.

Рассмотрим два примера таких пар автоматов. Их графы представлены на рис. 1. Входная последовательность $x(1),\ldots,x(N)$ состоит из символов «1» и «0». Число состояний равно K (на рис. 1 это 4 или 6). Пусть задано начальное состояние A. Пронумеруем состояния так:

$$A=0,\ B=1,\ C=2,\ D=3,\ E=4,\ F=5. \tag{5.1}$$

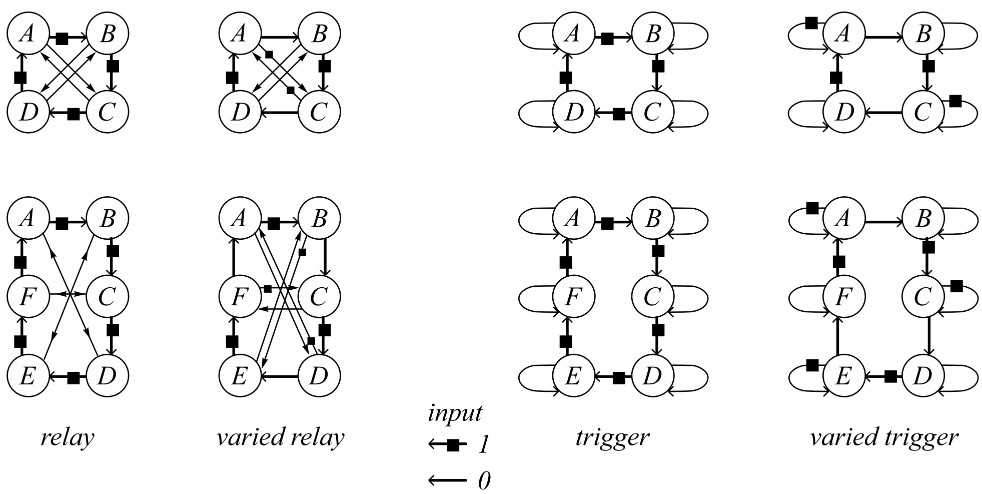

Рис. 1. Показаны схемы восьми автоматов четырех типов: реле, вариативное реле, триггер и вариативный триггер. Каждый тип представлен двумя автоматами с четырьмя и шестью состояниями. Вариативное реле с четырьмя состояниями и скачком 2 неудобно для параллельного расчета траектории состояний. Остальные автоматы, вариативный и однородный, в данном случае допускают эффективный параллельный анализ входных последовательностей

Для однородного автомата-триггера номер финального состояния определяется формулой

$$f=\sum_{i=1}^{N}x(i) \quad (\operatorname{mod} K). \tag{5.2}$$

Для однородного автомата-реле номер финального состояния определяется формулой

$$f=\sum_{i=1}^{N}\big(2-x(i)\big) \quad (\operatorname{mod} K),$$

$$f=2N-\sum_{i=1}^{N}x(i) \quad (\operatorname{mod} K). \tag{5.3}$$

Обе формулы, (5.2) и (5.3), допускают эффективное параллельное вычисление прямо по последовательности входов.

Рассмотрим вариативные модификации этих автоматов. Они отличаются чередованием типов переходов, определяемых входным символом, по номерам состояний (рис. 1). Для триггера это либо переход состояния в себя, либо переход в следующее состояние по циклу. Для реле это либо переход в следующее состояние по циклу, либо скачок через несколько (S) состояний по циклу.

Для **вариативного триггера** достаточно раздельно анализировать входные последовательности, которые начинаются с нуля или единицы. Если задано начальное состояние A, то начальный массив входов «1» оставляет автомат в этом состоянии. Далее, начало каждого нового массива символов «0» или «1» вызывает переход в следующее состояние, с последующей фиксацией его до начала следующего массива альтернативного символа. Таким образом, если входная последовательность содержит m массивов, то верна формула

$$f = \begin{cases} m \,(\mathrm{mod}\, K) \Leftarrow x(1) = 0, \\ m-1 \,(\mathrm{mod}\, K) \Leftarrow x(1) = 1. \end{cases} \tag{5.4}$$

Вычисление числа массивов легко выполнять параллельно по формуле

$$m = 1 + \sum_{i=1}^{N-1} \big(x(i+1) - x(i) \big). \tag{5.5}$$

Вариативные реле надо разбить на два класса: с четным или нечетным скачком s по циклу состояний. В каждом состоянии один вход вызывает переход в следующее по циклу состояние, а альтернативный вход вызывает скачок по циклу состояний на s номеров. Входные символы каждого из этих действий по состояниям чередуются.

Если скачок нечетный, то скачок не меняет действие входных символов по отношению к переходу в следующее состояние. Это следует из того, что переход в следующее состояние можно рассматривать как скачок единичной длины, и из того, что типы влияния каждого входного символа чередуются. Поэтому все нечетные скачки при формировании типа влияния входа эквивалентны, а все перемещения на четное число состояний меняют эти типы на противоположные: скачки на одно и на s состояний меняются местами.

Для нечетного значения s надо различать только четные и нечетные места входного символа в управляющей последовательности. Формула различна для четных и нечетных длин входной последовательности.

Для четных $N = 2L$

$$f = \sum_{i=1}^{L} \big(x(2i) + s(1 - x(2i)) \big) + \sum_{j=0}^{L-1} \big(sx(2j+1) + (1 - x(2j+1)) \big) \,(\mathrm{mod}\, K),$$

$$f = Ns + (s-1) \left(\left(\sum_{j=0}^{L-1} x(2j+1) \right) - \left(\sum_{i=1}^{L} x(2i) \right) \right) (\mathrm{mod}\, K).$$

Для нечетных $N = 2L + 1$

$$f = \sum_{i=1}^{L} \big(x(2i) + s(1 - x(2i)) \big) + \sum_{j=0}^{L} \big(sx(2j+1) + (1 - x(2j+1)) \big) \,(\mathrm{mod}\, K),$$

$$f = Ns + (s-1) \left(\left(\sum_{j=0}^{L} x(2j+1) \right) - \left(\sum_{i=1}^{L} x(2i) \right) \right) (\mathrm{mod}\, K).$$

Формулы можно объединить, используя нижнее округление чисел $\lfloor . \rfloor$ до целого:

$$f = Ns + (s-1) \left(\left(\sum_{j=0}^{N - \lfloor N/2 \rfloor} x(2j+1) \right) - \left(\sum_{i=1}^{\lfloor N/2 \rfloor} x(2i) \right) \right) (\mathrm{mod}\, K). \tag{5.6}$$

Эта формула допускает эффективное параллельное вычисление.

Для четных значений скачка s такой формулы нет. Действие входного символа зависит от всей предыдущей последовательности входов. Поэтому для этого случая минимальное число дополнительных процессоров, дающих гарантированный эффект сокращения времени решения за счет анализа одновременно двух отрезков входной последовательности, равно числу состояний автомата: по одному процессору на каждую гипотезу о финальном состоянии после отработки начального отрезка входной последовательности. Такие вариативные реле с четырьмя и шестью состояниями показаны на рис. 1 и рис. 2 соответственно.

Кроме того, можно рассмотреть нерегулярные типы триггеров и реле. Они отличаются достаточно произвольным чередованием типов переходов по входным символам, но с сохранением самих типов переходов (примеры показаны на рис. 2).

Для вариативных реле с четной длиной скачка и для нерегулярных реле и триггеров параллельная реализация малоэффективна, поскольку расчет хвоста последовательности существенно зависит от того состояния, в которое приведет автомат начало последовательности. Человек, практически наверняка, будет решать такую задачу последовательно.

В серии задач входную последовательность можно генерировать алгоритмом

$$x(i) = rand\{0;1\}, \quad i = \overline{1, N}. \tag{5.7}$$

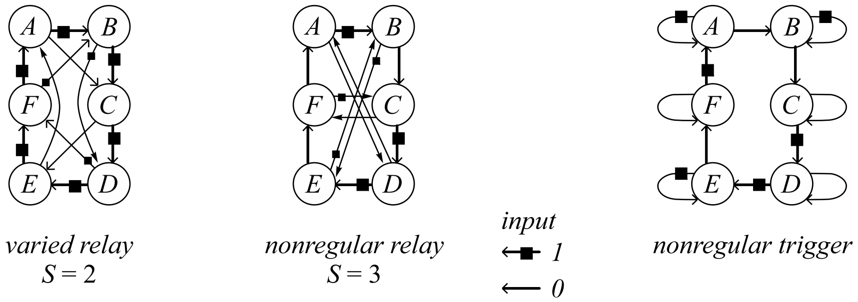

Рис. 2. Примеры нерегулярных и вариативных реле и триггеров, для которых параллельный анализ состояний малоэффективен

Использование соответственных пар автоматов позволяет построить необходимые пары тестов со сходными логическими операциями. В этих экспериментах необходимо дать возможность испытуемому привыкнуть к графу переходов используемого автомата в обучающей серии задач.

Проведем анализ эффективности параллельного счета для общего случая нерегулярного реле. Введем терминологию. Будем называть реле общего вида любой автомат с двумя входными символами, у которого есть последовательность входов, которая проводит автомат по всем состояниям без повторов. Эту последовательность состояний назовем главным циклом состояний, а соответствующую последовательность входов — главной последовательностью входов. Периодическая подача главной последовательности входов приводит к циклическому обходу состояний по главному циклу. Поскольку мы считаем фиксированным начальное состояние, то каждому состоянию соответствует ровно один входной символ из главной последовательности входов, который продолжает движение по главному циклу из этого состояния. Этот символ назовем циклическим входом в данном состоянии. Другой входной символ назовем входом скачка для этого состояния. Главный цикл позволяет дать состояниям автомата циклическую нумерацию в порядке обхода. Начав нумерацию с нуля, мы получим номера по модулю числа состояний автомата. Эту нумерацию назовем главной циклической нумерацией. В каждом состоянии циклический вход увеличивает номер состояния на единицу по модулю числа состояний. Изменение номера при подаче входа скачка назовем величиной скачка. В общем случае величина скачка различна в разных состояниях.

Замечание. Автомат *триггер* является частным случаем реле, в котором во всех состояниях величина скачка нулевая. Если во всех состояниях используется один и тот же циклический вход и величины скачков одинаковы, то автомат назовем *однородным* реле или однородным триггером. Если циклические входы чередуются по номерам состояний и величины скачков одинаковы, то автомат назовем *вариативным*. В остальных случаях автомат назовем *нерегулярным*.

Разобьем состояния автомата на классы.

Два состояния относятся к одному классу *сходства* (*сходные* состояния), если у них совпадают циклические входы и величины скачков.

Два состояния относятся к одному *функциональному* классу (*эквивалентные* состояния), если они сходные и любая входная последовательность переводит их в сходные состояния.

Обозначения.

M — число состояний автомата.

Φ — число функциональных классов состояний.

$\varphi(A)$ — функциональный класс состояния с номером A; само это состояние обозначим как $[A]$.

$\xi(A)$ — класс сходства состояния $[A]$.

$L(\xi,x)$ — величина изменения номера состояния из класса сходства ξ при подаче входа $x\in\{0;1\}$. Если это циклический вход, то $L=1$, а если это вход скачка, то L равно величине скачка.

Нам потребуется один представитель состояния из каждого функционального класса φ, номер которого мы обозначим как $A(\varphi)$.

N — длина входной последовательности в тестовой задаче.

$x(1),\ldots,x(N)$ — входная последовательность.

$A*x(1)*\ldots*x(t)$ — номер состояния, куда перейдет состояние с номером A после отработки автоматом входной последовательности $(x(1),\ldots,x(t))$.

Если задача представлена схемой автомата (реле общего вида), начальным состоянием и конечной входной последовательностью, то параллельное решение можно осуществить только разбиением входной последовательности на несколько отрезков, $x(1\div N_1),\ldots,x(N_r+1\div N)$, которые заданы границами $N_0=0<N_1<\ldots<N_r<N_{r+1}=N$. Тогда можно произвести параллельный расчет смещения номера состояния после отработки каждого отрезка. Это соответствует уровню параллельности расчета $r+1$. Но поскольку начальное состояние $[0]$ задано только для первого отрезка, то для отработки каждого из последующих отрезков в параллельном режиме требуется одновременно рассчитывать ровно Φ гипотез о начальном состоянии для этого отрезка. Таким образом, параллельность уровня $r+1$ осуществляется $1+\Phi r$ процессорами. С точки зрения уменьшения времени параллельного счета выгодно выбирать эти отрезки примерно равными по длине, насколько позволяет целочисленное деление. На каждом отрезке расчет ведется последовательно по рекуррентной формуле из начального состояния, которое будет свое для каждого из процессоров. Будем считать, что процессоры пронумерованы как $P_{\varphi,j}$, где j — номер отрезка входной последовательности $x(N_j+1\div N_{j+1})$, на котором работает данный процессор, $j=1,\ldots,r$, а начальное состояние для расчета $A(\varphi)$ соответствует функциональному классу φ, в качестве гипотезы о реально возникающем состоянии в конце предыдущего отрезка $[0*x(1)*\ldots*x(N_j)]$. Кроме того, имеется процессор $P_0=P_{\varphi([0]),0}$, который для первого отрезка один, поскольку точно известно начальное состояние. Процессор вычисляет смещение $\Delta_{\varphi,j}$ своего гипотетического начального состояния после отработки своего участка входной последовательности. Рекуррентная формула для произвольного процессора с номером j следующая:

$$\Delta_{\varphi,j}(N_j)=0, \tag{5.8.1}$$

$$B_\varphi(N_j)=A(\varphi), \tag{5.8.2}$$

$$\Delta_{\varphi,j}(t+1)=\Delta_{\varphi,j}(t)+L(\xi(B_\varphi(t)),x(N_j+t+1))\,(\operatorname{mod} M), \tag{5.8.3}$$

$$B_\varphi(N_j+t+1)=B_\varphi(N_j+t)*x(N_j+t+1), \tag{5.8.4}$$

$$t=0,\ldots,N_{j+1}-N_j-1,$$

где на каждом шаге с номером t получаем номер текущего гипотетического состояния

$$B_{\varphi}(N_j+t)=A(\varphi)*x(N_j+1)*\ldots*x(N_j+t), \tag{5.8.5}$$

что соответствует гипотезе о начальном состоянии отрезка.

После окончания этих параллельных расчетов начинается последовательная сборка результатов. Она тоже проводится рекуррентной процедурой.

$$f(1)=\Delta_{\varphi(0),0}(N_1)=B_{\varphi(0),0}(N_1), \tag{5.9.1}$$

$$f(j+1)=f(j)+\Delta_{\varphi(f(j)),j+1}(N_{j+2})\,(\operatorname{mod} M), \tag{5.9.2}$$

$$j=0,\ldots,r-1.$$

Ответ задачи:

$$f=f(r). \tag{5.9.3}$$

Если длины отрезков входной последовательности примерно равны $N/(r+1)$, а точнее, они принимают значения либо $\lfloor N/(r+1)\rfloor$, либо $\lceil N/(r+1)\rceil$, так, чтобы в сумме дать значение N, то суммарное число шагов итераций в процессах (5.8) и (5.9) равно

$$T=\left(1+\frac{N}{r}\right)+\left(1+r\right)=2+r+\frac{N}{r}. \tag{5.10}$$

Если $N+1-(N/r)>r+2$, то параллельность дает выигрыш во времени решения задачи. Но при этом на этапе (5.8) используется $1+\Phi r$ процессоров. А в ответ задачи по формулам (5.9) входят результаты счета только от $1+r$ процессоров. Результаты работы остальных процессоров просто отбрасывается. Поэтому КПД процессоров не выше чем

$$\eta\le\frac{r+1}{\Phi r+1}. \tag{5.11}$$

Эти формулы, (5.10) и (5.11), показывают, что параллельный счет оправдан только для автоматов с малым числом классов функциональной эквивалентности и только при достаточно больших длинах заданной входной последовательности. Поэтому для нерегулярных реле параллельный расчет неэффективен, а для регулярных реле параллельность может быть целесообразна только для весьма длинных входных последовательностей. Причем для вариативного реле, где $\Phi=2$, КПД использования процессоров не выше $2/3$ при параллельности 2 и снижается до $1/3$ с ростом параллельности. Поэтому крайне маловероятен параллельный способ решения задачи человеком в неоднородном случае.

6. Тесты, основанные на алгебре

Алгебры универсального типа с конечным числом элементов также удобный инструмент для создания тестов параллельности. Алгебра определяется таблицей операций над своими элементами. Удобно использовать бинарные и унарные операции с двумя или одним элементом соответственно. Бинарную операцию $\varphi: A\times A\to A$ будем записывать в привычной форме со знаком операции между операндами $x\varphi y=z$, где оба операнда и результат принадлежат носителю алгебры A (конечное множество). В качестве теста можно задавать цепочки элементов $xyz\ldots v$, к которым операция применяется последовательно, например, слева направо. Требуется рассчитать ответ w:

$$xyz\ldots v\mapsto(\ldots((x\varphi y)\varphi z)\varphi\ldots)\varphi v=w. \tag{6.1}$$

Унарную операцию $\psi : A \mapsto A$, запись $\psi(x) = y$, где $x, y \in A$, можно использовать в цепочках итераций, в которых требуется рассчитать результат последней итерации:

$$\psi^{\times N}(x) \mapsto \psi((\psi(\ldots(\psi(x))\ldots))) = y. \tag{6.2}$$

В обоих случаях, в зависимости от таблицы, которой задается операция, задача может допускать или не допускать эффективную параллельность вычисления. Для тестирования удобнее использовать бинарную операцию, поскольку она позволяет создать практически неограниченный набор исходных данных при фиксированной длине вычислительной цепочки. Для формулы (6.2) при заданном значении N различных исходных данных ровно столько, сколько имеется элементов носителя алгебры.

Формула (6.1) допускает эффективную параллельность вычислений, если операция ассоциативна: $(x\varphi y)\varphi z = x\varphi(y\varphi z)$. Еще больше возможностей параллельности счета возникает, если к ассоциативности добавляется коммутативность $x\varphi y = y\varphi x$, которая позволяет менять порядок переменных при счете. Если же нет ассоциативности, то организовать параллельный расчет очень трудно, и удельная эффективность работы дополнительных процессоров очень мала в пересчете на один процессор. Доказательство практически повторяет рассмотрение для нерегулярных автоматов из предыдущего раздела. Большинство дополнительных процессоров будут рассчитывать нереализуемые варианты.

Рассмотрим несколько вариантов таких алгебр. При отборе примеров будем исходить из простоты обучения испытуемых операциям алгебры.

В качестве коммутативной и ассоциативной алгебры можно рассмотреть алгебры сложения или умножения по конечному модулю $[0;\ldots;K \mid +(\mathrm{mod}\, K+1)]$ или $[0;\ldots;K \mid \bullet(\mathrm{mod}\, K+1)]$. Вычисления значительно упрощаются, если в качестве модуля взять число 10, что означает $K = 9$. Тогда можно выполнять обычные сложение и умножение однозначных чисел и при этом отбрасывать все разряды результата, кроме младшего. Эти алгебры имеют вид

$$[0;\ldots;9 \mid +(\mathrm{mod}\, 10)], \tag{6.2.1}$$

$$[0;\ldots;9 \mid \bullet(\mathrm{mod}\, 10)]. \tag{6.2.2}$$

Для построения аналогичной, но не ассоциативной и не коммутативной, алгебры можно заметить, что сами операции сложения и умножения в цепочке вычислений переставлять и перегруппировывать нельзя, если цепочка содержит операции обоих видов. Рассмотрим носитель алгебры, содержащий не только цифры, но и символ арифметической операции, которую надо применить. Такую алгебру можно записать в форме $[+0;\ldots;+K;\bullet 0;\ldots\bullet K \mid *(\mathrm{mod}\, K+1)]$, где

$$+x * \bullet y = +(x \cdot y(\mathrm{mod}\, K+1)), \tag{6.3.1}$$

$$\bullet x * \bullet y = \bullet(x \cdot y(\mathrm{mod}\, K+1)), \tag{6.3.2}$$

$$+x * +y = +(x + y(\mathrm{mod}\, K+1)), \tag{6.3.3}$$

$$\bullet x * +y = \bullet(x + y(\mathrm{mod}\, K+1)). \tag{6.3.4}$$

Практически удобно рассмотреть алгебру

$$[+0;\ldots;+9;\bullet 0;\ldots\bullet 9 \mid *(\mathrm{mod}\, 10)]. \tag{6.4}$$

При предъявлении тестовой задачи знак внутренней операции алгебры (*) можно опустить, поскольку сам элемент алгебры несет на себе знак нужной арифметической операции. Если обозначить элемент алгебры в форме αX, где $\alpha \in \{+;\bullet\}$, $X \in \{0;\ldots;K\}$, то такое упрощение записи можно записать так:

$$\alpha_1 X_1 * \alpha_2 X_2 * \ldots * \alpha_N X_N \equiv \alpha_1 X_1 \alpha_2 X_2 \ldots \alpha_N X_N (\mathrm{mod}\, 10). \tag{6.5}$$

Здесь знак тождества означает переход к другой форме записи той же задачи. Само вычисление ведется с помощью операций, входящих в символы элементов алгебры, выполняемых подряд слева направо, без учета порядка действий суммы и умножения:

$$\alpha_1 X_1 \alpha_2 X_2 \ldots \alpha_N X_N (\mathrm{mod}\, 10) = \alpha_1 (X_1 \alpha_2 X_2 \ldots \alpha_N X_N (\mathrm{mod}\, 10)) =$$
$$= \alpha_1 ((\ldots((X_1 \alpha_2 X_2)\alpha_3 X_3)\ldots)\alpha_N X_N (\mathrm{mod}\, 10)). \tag{6.6}$$

Например,

$$\bullet 4 * +7 * +4 * \bullet 6 \equiv \bullet 4 + 7 + 4 \bullet 6 (\mathrm{mod}\, 10) =$$
$$= ((\bullet 4 + 7) + 4)\bullet 6 (\mathrm{mod}\, 10) = (\bullet 1 + 4)\bullet 6 (\mathrm{mod}\, 10) = \bullet 5 \bullet 6 (\mathrm{mod}\, 10) = \bullet 0.$$

Если все тесты используют один модуль, то его также можно исключить из предъявляемой записи. Покажем на примерах отсутствие коммутативности и ассоциативности. При расстановке скобок иногда будет удобно частично возвращать в запись знак операции $(*)$.

$$+5\bullet 7 = +5,$$
$$\bullet 7 + 5 = \bullet 2,$$
$$(+7\bullet 8) + 3 = +6 + 3 = +9,$$
$$+7 * (\bullet 8 + 3) = +7\bullet 1 = +7.$$

Испытуемому нужно объяснить, что «школьный» порядок действий в этой записи не действует и вычислять выражение надо последовательно слева направо (польская запись операций).

Пример

Правильный порядок действий:

$$\bullet 5 + 4\bullet 4 = ((\bullet 5 + 4)\bullet 4) = \bullet 9\bullet 4 = \bullet 6.$$

Неправильный порядок действий:

$$\bullet 5 + 4\bullet 4 = (\bullet 5 * (+4\bullet 4)) = \bullet 5 + 6 = \bullet 1.$$

Заметим, что задачи, в которых все операнды имеют один знак арифметической операции, допускают эффективную параллельность счета, поскольку каждая операция циклической арифметики и ассоциативна и коммутативна.

Примеры

$$+6 + 9 = +9 + 6 = +5,$$
$$\bullet 7\bullet 4 = \bullet 4\bullet 7 = \bullet 8,$$
$$(+6 + 2) + 7 = +6 * (+2 + 7) = +5,$$
$$(\bullet 6\bullet 2)\bullet 7 = \bullet 6 * (\bullet 2\bullet 7) = \bullet 4.$$

Это позволяет использовать данную алгебру в тестах обоих типов. Использование одной алгебры в обоих тестах полезно с точки зрения инженерной психологии, поскольку работа испытуемого проверяется на одних и тех же операциях, как в параллельном, так и в последовательном случае.

7. Заключение

В данной работе рассмотрены принципы построения тестов параллельной обработки информации человеком на основе ассоциативных и неассоциативных алгебр или с помощью расчета финального состояния конечного автомата после отработки заданной входной последовательности. В основе методики лежит сопоставление времени решения задач двух типов, один из которых допускает эффективный параллельный счет, а для другого типа параллельность малоэффективна. Каждый тип задач представлен несколькими сериями задач одной сложности по числу необходимых для решения логических операций. Зависимость времени решения задачи от сложности дает информацию о параллельном или последовательном способе решения задачи. Если время решения параллельных задач растет медленнее, чем линейно, а на последовательных задачах время растет линейно, то можно регистрировать параллельность обработки информации. Если на обоих типах задач время растет по сложности медленнее, чем линейно, то можно говорить только об ускорении обработки информации с ростом сложности, но параллельность регистрировать нельзя. Если же на задачах обоих типов время растет линейно по сложности, то это последовательный способ решения с равномерной скоростью обработки информации. Другие варианты зависимости времени от сложности не позволяют определенно утверждать некоторый способ организации счета. Ранее поставленные эксперименты показали эффективность такой методики. Данная работа позволяет расширить набор тестов. В этой работе не рассматривались вопросы обработки ошибок, допущенных испытуемыми, и подробности способа предъявления информации и регистрации времени решения. Для решения этих вопросов можно использовать методы, подробно описанные и исследованные в ранее изданных работах [Коганов, Злобин, Ракчеева, 2013, 2014]. Следует отметить, что новые тесты могут потребовать дополнительной проработки этих вопросов при постановке инженерно-психологических экспериментов. Однако математический анализ, проведенный в этой работе, гарантирует адекватность новых тестов поставленной проблеме.

Список литературы (References)

Злобин А. И., Коганов А. В., Ракчеева Т. А. Исследование скорости переработки информации человеком в серии задач растущей сложности // IX Курдюмовские чтения: Международная междисциплинарная научная конференция «Синергетика в общественных и естественных науках», труды, 17–21 апреля 2013 г., Тверской государственный университет, Тверь. — С. 57–60.

Zlobin A. I., Koganov A. V. Rakcheeva T. A. Issledovanie skorosti pererabotki informacii chelovekom v serii zadach vysokoj slozgnosti [The research of the speed of the information processing in the serios of the tasks of the high complexity] // IX Kurdumovskie chteniya: Mezgdunarodnaya nauchnaya konferenciya “Sinergetika v obtshestvennyh i estestvennyh naukah” [9-th Kurdumov reading: international interdisciplinary science conference “Synergetic in social and natural sciences”, works], 17–21 april 2013, TvGU. — P. 57–60.

Злобин А. И., Коганов А. В., Ракчеева Т. А. Метод исследования пропускной способности человеческого мозга при обработке символьной информации // XX международная конференция «Математика. Компьютер. Образование», Пущино МО, 2013. — Тезисы докладов. — М.–Ижевск: НИЦ «Регулярная и хаотическая динамика». — С. 171.

Zlobin A. I., Koganov A. V., Rakcheeva T. A. Metod issledovaniya propusknoj sposobnosni chelovecheskogo mozga pri obrabotke simvol’noj informacii [The method of research of the channel capacity of the human brain in the symbol information processing] // XX mezgdunarodnaya konferenciya “Matematika. Komputer. Obrazovanie” [20-th International conferencing “Mathematics, Computer. Education.”], Putshino at Moscow region, 2013, thesisy. — Moscow–Izhevsk: NIC “Regulyarnaya i haoticheskaya dinamika”. — P. 171.

Коганов А. В. Исследование возможности параллельного выполнения логических операций человеком. Параллельные вычисления и задачи управления // Труды международной конференции PACO-2001, Москва, 2–4 октября 2001 г., на компакт-диске. ИПУ РАН. — М., 2001.

Koganov A. V. Issledovanie vozmozgnosti parallel’nogo vypolneniya logicheskih operacij chelovekom. Parallel’nye vychsleniya i zadachi upravleniya [The research of the possibility of the parallel calculations by the man. The parallel calculations and the tasks of the control] // Trudy mezgdunarodnoj konferencii PACO-2001, Moskva, 2–4 oktyabrya 2001 g., na compact-diske. IPU RAN [The works of The international conference PACO-2001, Moscow, 2–4 October 2001, on compact-disk, IPU RAN]. — Moscow, 2001.

Коганов А. В. Коллективы автоматов в детерминированных и случайных средах и приложение к психологическим тестам: Диссертация на соискание ученой степени к. ф-м. н. — М., 1972.

Koganov A. V. Kollektivy avtomatov v determinirovannyh i sluchajnyh sredah i prilozgenie k sluchajnym sredam [The collectives of the automates in the determine and probability mediums and the application to the psychological testes]: Dissertaciya na soiskanie uchenoj stepeni k. a.-m. n. [The dissertation at phys. math. sciences]. — Moscow, 1972.

Коганов А. В. Растущие индукторные пространства и анализ параллельных алгоритмов // Программные продукты и системы. Приложение к международному журналу «Проблемы теории и практики управления». — 2010. — № 2. — С. 33–38.

Koganov A. V. Rastucshie inductornye prostranstva i analis parallelnyh algoritmov [The growing inductor spaces and the analysis of the parallel algorithms] // Programmnye producty i sistemy. Prilozgenie k mezgdunarodnomu zgurnalu "Problemy teorii i practiki upravleniya" [The programming products and systems, the application to the international journal "The problems of the theory and practicum of the control"]. — 2010. — No. 2. — S. 33–38.

Коганов А. В., Злобин А. И., Ракчеева Т. А. Задача вычисления траектории с равномерным распределением ответов // Компьютерные исследования и моделирование. — 2014. — Т. 6, № 5. — С. 803–828.

Koganov A. V., Zlobin A. I., Rakcheeva T. A. Zadacha vychisleniya traectorii s ravnomernym raspredeleniem otvetov [The task of the calculation of the trajectory with homogenous the distribution of solutions] // Komputernye issledovaniya i modelirovanie [The Computer research and modeling]. — 2014. — Vol. 6, no. 5. — P. 803–828.

Коганов А. В., Злобин А. И., Ракчеева Т. А. Исследование возможности параллельной переработки информации человеком в серии задач высокой сложности // Компьютерные исследования и моделирование. — 2013. — Т. 5, № 5. — С. 845–861.

Koganov A. V., Zlobin A. I., Rakcheeva T. A. Issledovanie vozmozgnosti parallel'noj pererabotki informacii chelovekom v serii zadach vysokoj slozgnosti [The research of the possibility of the parallel processing of the information by man in the series of the tasks of the high complexity] // Komputernye issledovaniya i modelirovanie [The Computer research and modeling]. — 2013. — Vol. 5, no. 5. — P. 845–861.

Коганов А. В., Пятецкий-Шапиро И. И., Фейгенберг И. М. Зависимость скорости решения от сложности и способа кодирования исходных данных // Вопросы экспериментального исследования скорости реагирования: Сборник. — Тарту, 1971.

Koganov A. V., Pyatecky-Shapiro I. I., Fejgenberg I. M. Zavisimost' skorosti resheniya ot slozgnosti i sposoba kodirovaniya ishodhyh dannyh [The depending of the speed of the soluteion from the complexity and the coding method of initial dates] // Voprosy Heksperimental'nogo issledovaniya skorosti reagirovaniya [Compendium "The equations of the experimental investigation of the reacting speed"]: Sbornik. — Tartu, 1971.

Котик М. Л. Курс инженерной психологии. — Таллин: Валгус, 1978.

Kotik M. L. Kurs inzgenernoj psihologii [The course of the engineering psychology]. — Tallin: Valgus, 1978.

Крылов А. А. Организация целостной деятельности функциональных механизмов обработки информации. Хрестоматия по инженерной психологии. — М.: Высшая школа, 1991.

Krylov A. A. Organizaciya celostnoj deyatel'nosty funkcional'nyh mehanozmov obrabotki informacii. Hrestomatiya po inzgenernoj psihologii [The organization of the whole activity of the functional mechanisms in the information processing. The Reader at the engineering psychology]. — Moscow: Vysshaya shkola, 1991.

Суходольский Г. В. К вопросу о формировании у человека-оператора навыка слежения за движущейся целью // Проблемы общей и инженерной психологии. — Л.: Изд-во ЛГУ, 1964. — С. 80–89.

Suhodolsky G. V. K voprosu o formirovanii u cheloveka-operatora navyka slezgeniya za dvizgutshejsya cel'yu [The problem of the forming the habit of the moving target watch at the man-operator] // Problemy obtshej i ingenernoj psihologii [The problems of the common and engineering psychology]. — Leningrad: LGU, 1964. — P. 80–89.

Fischer R., Plessow F. Efficient multitasking: parallel versus serial processing of multiple tasks // Front Psychol. — 2015. — 6:1366. — DOI: 10.3389/fpsyg.2015.01366

Popov G., Mastorakis N., Mladenov V. Calculation of the acceleration of parallel programs as a function of the number of threads. — https://www.researchgate.net/publication/228569958. — Article January 2010.

Zlobin A. I., Koganov A. V., Rakcheeva T. A. About Engineering-Psychological Test at Trajectory calculation kind, which has the uniformly distribution of intermediate and final results // 21-th International conferencing "Mathematics, Computer. Education", Dubna at Moscow region, 2014, thesis's. — Moscow–Izhevsk: NIC "The regular and chaotic dynamics". — P. 112.