

# КОМПЬЮТЕРНАЯ ИНЖЕНЕРИЯ И ТЕХНИЧЕСКАЯ ДИАГНОСТИКА



УДК681.326:519.713

## ЭВОЛЮЦИЯ КИБЕРНЕТИЧЕСКОГО ПРОСТРАНСТВА

*ХАХАНОВ В.И., ХАХАНОВА А.В., ЗАКАРЯН В.В.*

Предлагаются быстродействующие технологии описания и решения логических задач, на основе использования булевых графов ассоциативных таблиц, путем аппаратной реализации параллельных средств синтеза и анализа алгебраических, графовых и табличных структур предикатных отношений для получения детерминированного многозначного вывода в  $n$ -мерном дискретном пространстве. Рассматриваются проблемы создания теоретических основ для мозгоподобных вычислений в форме: метрики ассоциативных отношений, архитектур описания ассоциаций, оптимизации логических структур данных, предикатных моделей вычислительных процессов системного уровня, архитектуры и системы команд логического ассоциативного мультипроцессора; синтеза и анализа прилагательных [1-5].

### 1. Актуальность создания мозгоподобных вычислителей

Мотивация – создание инфраструктуры для описания, анализа и синтеза логических ассоциативных отношений, моделирующих мозгоподобные (Brain-Like – BL) вычисления. К ним можно отнести: 1. Анализ и синтез синтаксических и семантических языковых конструкций для решения многочисленных естественно-языковых задач (реферирование, исправление ошибок, анализа качества текстов). 2. Определение степени принадлежности объекта существующим ассоциативным компонентам BL-системы на основе введенных критериев качества выбора. 3. Распознавание видео- и аудио-образов путем их представления вектором существенных параметров. 4. Оперативное диагностирование неисправностей, технического состояния объекта и ремонт в процессе функционирования изделия. 5. Тестирование знаний и экспертное обслуживание объектов или субъектов для определения их валидности. Например, анализ кредитной истории для выдачи банком кредита. 6. Идентификация объекта или процесса для принятия решений в условиях неопределенности.

Существует также многообразие практически интересных для рынка задач, которые можно свести к выбору оптимального решения на основе использования логической структуры ассоциаций. 1. Точный поиск заданной вектором параметров информации в

Internet, где по запросу пользователя очень часто выдаются два сообщения: отсутствуют данные или слишком много информации, слабо ассоциируемой с входным запросом. Здесь нужна правильная метрика оценивания и валидный запрос. 2. Коррекция текста в процессе его набора, когда автоматически исправляются только тривиальные ошибки, типа повторения буквы в слове, но можно корректировать более сложные ошибки, связанные с неверным окончанием, а также предлагать более приемлемые варианты порядка слов в предложении или в его части. Данная задача актуальна для 100% пользователей компьютеров. 3. Более серьезная проблема относится к критическим технологиям. Наведение на цель (выбор цели), будь то целеуказатель в истребителе или в автоматической системе посадки лайнера. Здесь очень существенно функционирование системы наведения в реальном масштабе времени в микросекундном диапазоне измерения. 4. Обратной задачей выбора цели в критических технологиях является разведение объектов во времени и в пространстве, например, в диспетчерской службе аэропорта или оптимальной организации городского транспорта для исключения коллизий в воздушном пространстве и пробок на дорогах. Практически все упомянутые задачи должны решаться в реальном времени, они сходны по логической структуре ассоциативных многозначных векторов, которые составляют модель процесса или объекта. Они нуждаются в быстродействующей и специальной аппаратной платформе, которой может выступать логический ассоциативный мультипроцессор, ориентированный на параллельное выполнение поиска в ассоциативных структурах памяти и принятия (выбора) решения на основе использования интегрального критерия качества.

В плане предлагаемых исследований интересной представляется опубликованный в EE Times (12.2009) десяток перспективных технологий от Gartner Research Group (Gary Smith) для ближайших лет. В ней нашли отражение только технологии, связанные с разработкой специализированных цифровых изделий на кристаллах, хотя развитие программных технологий также будет оказывать сильное влияние на состояние рынка электроники. Глобально важными остаются технологии снижения потребляемой мощности и решения, направленные на уменьшение содержания ценных материалов в продукте. Данные технологии выступают двигателями многих направлений развития электроники, перечисленных ниже: 1. Биологическая обратная связь или электроника, управляемая мыслью. 2. Печатная электроника на основе использования органических материалов. 3. Пластиковая память на основе полимеров, проявляющих ферроэлектрические свойства. 4. Безмасочная литография на основе использовании электронного луча для создания топологии схемы. 5. Параллельная обработка данных для многоядерных гетерогенных графических процессоров. 6. Сбор энергии от механических и электрических процессов в окружающей среде. 7. Биоэлект-

роника и wetware, сочетающие биологические объекты и электронику для медицины. 8. Резистивное ОЗУ или мемристор с эффектом памяти как четвертый пассивный элемент электронной схемы, дополняющий резистор, конденсатор и индуктивность. 9. Переходные отверстия в кремнии (Through-Silicon-Via – TSV) для создания реальных 3D-SiP и кристаллов. 10. Различные технологии батарей на основе сочетания никеля и лития.

*Цель* – существенное повышение быстродействия анализа и синтеза логических ассоциативных отношений путем аппаратной реализации параллельных средств обработки аналитических, графовых и табличных структур предикатных данных для детерминированного (многозначного) вывода в n-мерном дискретном пространстве.

*Задачи:* 1) Актуальность создания мозгоподобных вычислителей. 2) Метрика ассоциативных отношений. 3) Архитектуры ассоциаций: таблицы, графы, уравнения. 4) Оптимизация логических структур данных. 5) Условные и безусловные эксперименты синтеза и анализа. 6) Предикатная модель вычислительного процесса системного уровня. 7) Архитектура и система команд логического ассоциативного мультипроцессора. 8) Практические примеры и применения (синтез и анализ прилагательных).

*Источники.* 1. Нейро-фаззи структуры, ориентированные на использование ассоциативной памяти [1-2]. 2. Структурные организации ассоциативных памятей для решения информационно-логических задач [3-6]. 3. Аппаратно-ориентированные методы и средства для решения практически ориентированных информационно-логических задач, включая анализ и синтез словоформ и предложений русского языка [7-15].

Специализация и универсализация – две фазы спирального развития технологий компьютерной индустрии. Универсализация имеет стремление на создание еще более мощных изделий, или на предоставление пользователю новых функциональностей. Специализация тяготеет к разработке более быстродействующих и дешевых устройств, ориентированных на эффективное обслуживание задач определенного класса. Затем разработчик начинает увеличивать функциональные мощности специального изделия, что означает новый виток на пути к универсализации.

Для решения информационно-логических задач акцент делается на следующие характеристики: 1) Высокое быстродействие параллельного выполнения минимального множества логических команд. 2) Исключение из процессора мощной системы арифметических вычислений, как функциональности, несвойственной человеку. 3) Логический секвенсор – элементарный процессор – должен иметь не более 16 команд, которые обеспечиваются регистром, разрядностью 4 бита. 4) Устройство управления логическим ассоциативным мультипроцессором должно обеспечивать параллельное выполнение задач логического анализа и синтеза. 5) Каждый секвенсор имеет ассо-

циативную память, а также регистры для хранения результатов логических вычислений и связи с другими секвенсорами. 6) Компилятор для языка описания аппаратуры или программирования, внешняя программа по отношению к мультипроцессору, обеспечивает квазиоптимальное планирование вычислительного процесса во времени и в пространстве секвенсоров с учетом ограничений на размерность блоков ассоциативной памяти. 7) Память прямого доступа обслуживает мультипроцессор и хранит программу вычислительного процесса, полученную от компилятора, для решения логической задачи. 8) Создание гибкой инфраструктуры ассоциативной памяти для размещения таблиц неравной размерности. 9) Использование GUI (Guide User Interface) – эффективно и дружественного интерфейса для общения с пользователем. Выделение курсором слова или буквы вызывает высвечивание таблицы со всеми параметрами словоформы или части слова.

## **2. Генерирование мозгоподобных процесс-моделей на основе спецификации функциональности**

Предлагается стратегия автоматического создания процесс-моделей рыночно-ориентированных функциональностей, основанная на использовании спецификации высокого уровня описания с последующим генерированием кодов программно-аппаратных средств и их имплементации в аппаратуру. Таблица рассматривается как система отношений, создающая примитив. Связи между таблицами формируются посредством регистровых или булевых переменных.

Генерация схемной структуры есть итеративный причинно-следственный mapping спецификации библиотечными примитивами. Спецификация представлена вектором входных и выходных условий, подлежащих покрытию примитивными таблицами, которые имеют соответствующие переменные. Итеративный процесс мэппинга имеет иерархию, подобно Y-технологии, где каждый компонент, в свою очередь, также может быть покрыт примитивами более нижнего уровня. Оценка качества генерирования схемной структуры или покрытия определяется векторно-логическим критерием качества по результатам моделирования структуры, которая формирует решение, сравниваемое с вектором исходной спецификации. Приближенная оценка качества позволяет сделать вывод о покрытии всех переменных (входных и выходных) библиотечными примитивами. Но здесь отсутствует понятие адекватности и точности примитивов для решения задачи. Учитывая, что может существовать несколько решений для одной функциональности, то в соответствии с таким условием может быть сгенерировано несколько схем, качество функционирования которых относительно получаемой информации можно оценить после проведения модельного эксперимента на представительной выборке тестовых примеров (testbench). Итерации для получения решения могут осуществляться с двух сторон: от входов и выходов.

Если нет примитива, создающего покрытие спецификации, его следует сгенерировать по требуемой в процессе генерирования спецификации. Формальная модель генерирования функциональной структуры по спецификации имеет вид:

$$\begin{aligned}
 S &= (X, Z, Y), \\
 X &= (X_1, X_2, \dots, X_i, \dots, X_n); \\
 Z &= (Z_1, Z_2, \dots, Z_i, \dots, Z_m); \\
 Y &= (Y_1, Y_2, \dots, Y_i, \dots, Y_k); \\
 X \cap Z \cap Y &= \emptyset; \\
 n+m+k \\
 \cup_{i=1} (S \cap P_i) &= S; \\
 Q_{cov} &= S \oplus P; \\
 Q_{sim} &= \min(S \oplus P_i).
 \end{aligned}$$

Для создания схемы, реализующей полезную функциональность, следует генерировать примитивы  $P = \{P_1, P_2, \dots, P_i, \dots, P_n\}$  самого нижнего уровня. Для этого необходимо создавать фильтры  $F = \{F_1, F_2, \dots, F_j, \dots, F_m\}$ , формирующие таблицы примитивных отношений, взятых из информационного пространства планеты (рис. 1).

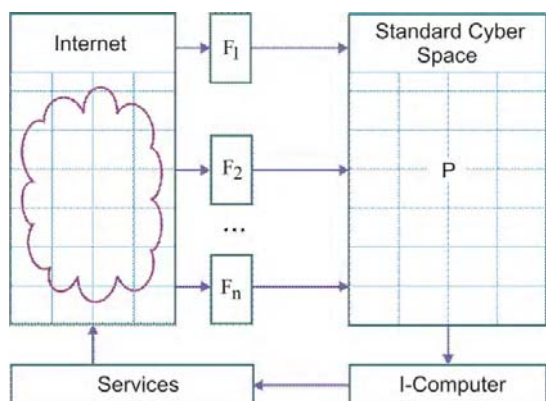


Рис. 1. Эволюция Cyber Space и Internet

Имея стандартизованные структуры данных для отдельных порталов и браузеров, доставляющих новые сервисы с более высоким быстродействием, следует ожидать постепенного качественного улучшения всех компонентов Cyber Space. Конечная цель такого взаимного и положительного влияния элементов инфраструктуры кибернетического пространства – выработка единых стандартов по интерфейсам и его превращение в саморазвивающуюся интеллектуальную информационно-компьютерную экосистему. Существенное значение будут иметь первичные фильтры или преобразователи для создания новых стандартизованных примитивов, создающих технологичную инфраструктуру для скоростного драйва по Cyber Space с использованием специализированного неарифметического двигателя (I-Computer). Со временем аморфная или «мусорная» часть Internet будет уменьшаться, а стандартизованная инфраструктура – увеличиваться. К 2020 году информационное пространство планеты должно принять цивилизованные

форматы структур данных со стандартизованными интерфейсами, подобно тому, как это происходило с развитием планетарной инфраструктуры транспортных сообщений с терминалами, отелями, заправками, ориентированными на сервисы, удовлетворяющие любые запросы пользователя.

В настоящее время отсутствуют стандарты формирования и хранения структур однотипных данных на сайтах и порталах Internet. Это затрудняет понимание информации поисковыми системами для распознавания и принятия правильных решений. Трудно ожидать, что информационное пространство в одночасье перейдет на рельсы принятых всеми форматов и интерфейсов. Единственным выходом может служить эволюционирование структур данных. Для этого необходимо разрабатывать преобразователи (фильтры) форматов данных. Наличие первичных фильтров позволяет автоматизировать времязатратные процессы создания библиотек базовых примитивов. Например, нетрудно создать генераторы логических отношений (фильтры, см. рис. 1) для каждой части речи русского языка, ориентированные на анализ существующих электронных словарей или признанных эталонных текстов в целях формирования первоначальных таблиц отношений в виде <часть речи> <словообразующие компоненты>. Имея такого рода базовые примитивы, можно автоматически создавать более сложные конструкции и процесс-модели анализа, синтеза предложений, коррекции ошибок (рис. 2), что всегда будет актуально и востребовано на рынке информационно-компьютерных сервисов.

Имея спецификацию, представленную после обработки вербального описания в форме вектора входных и выходных переменных, нетрудно записать стратегию создания новой функциональности как задачу поиска покрытия библиотечными элементами обобщенного вектора  $\langle X, Y \rangle$ . Общее решение задачи похоже на синтез автоматной модели, определяющей взаимодействие компонентов во времени и в пространстве. Однако многообразие примитивов, заранее не заданных, исключает такую возможность, что означает – необходим переход из строгого детерминизма цифровых автоматов в область эволюционных и квазиоптимальных решений.

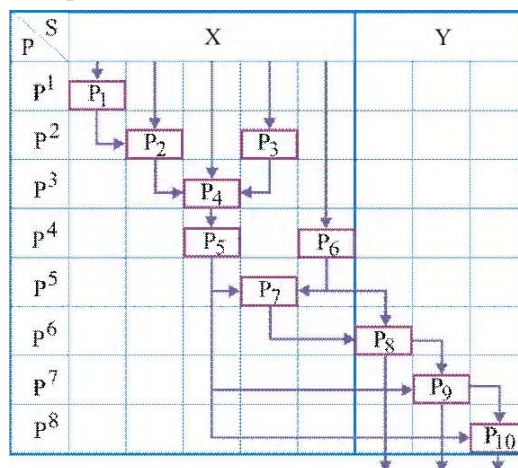


Рис. 2. Синтез покрытия спецификации примитивами

Условие задачи: имеется спецификация в виде вектора существенных переменных, которые необходимо покрыть минимальным множеством примитивов из библиотеки и сформировать выходной вектор. Красивое решение задачи синтеза функциональной структуры по спецификации есть ключ к самостоятельному генерированию компьютером новых решений. После ее решения остается решить только две задачи на пути к созданию интеллекта компьютера – самогенерирование оригинальной функциональности, необходимой для решения задачи покрытия, и спецификаций новых полезных для человека или компьютера сервисов.

Относительно структуры информационного пространства планеты. Оно не привязано к топологии поверхности и в этом смысле является виртуальным. Но в части покрытия поверхности и пространства телекоммуникациями физическая основа Cyber Space должна быть мажорированной в целях обеспечения отказоустойчивости и надежности функционирования каналов связи при возникновении катаклизмов. Информационная структура Cyber Space должна быть иерархической и замкнутой, как глобально, так и локально, на любом уровне иерархии. Элементарная ячейка структуры пространства должна быть треугольной. Это обеспечит существенное уменьшение информационных объемов Cyber Space, в пределе – на треть, что означает повышение производительности всех приемо-передатчиков и хранилищ контента планеты на 33%. Данное утверждение связано с аксиомой транзитивного замыкания конечного числа  $(1, 2, 3, 4, \dots, n)$  точек в информационном векторно-логическом пространстве:

- 1)  $d_1 = 0$ ;
- 2)  $d_1 \oplus d_2 = 0$ ;
- 3)  $d_1 \oplus d_2 \oplus d_3 = 0$ ;
- 4)  $d_1 \oplus d_2 \oplus d_3 \oplus d_4 = 0$ ;
- 5)  $d_1 \oplus d_2 \oplus \dots \oplus d_i \oplus \dots \oplus d_n = 0$ .

Рассматривая все возможные варианты транзитивного замыкания с позиции минимизации условий для восстановления информации, очевидным представляется факт восстановления третьей стороны треугольника по двум известным. В этом случае достаточно, например, передать по каналу связи две стороны, чтобы восстановить третью. Сокращение передаваемых объемов в данном случае равно 33%. Во всех других случаях уменьшение объемов информации будет меньшим. Например, для замкнутого четырехугольника необходимо передать три расстояния или стороны, чтобы восстановить четвертую, выигрыш – 25%. Таким образом, плоскостная интерпретация Cyber Space будет иметь вид, представленный на рис. 3.

Для того чтобы восстановить всю упомянутую структуру, необходимо иметь (передать) уже не 2/3 объема информации, которая соответствует каждому двум сторонам из трех всех треугольников, представленных зеленым (более светлым) цветом. Для данной структуры достаточно передать стороны, заданные

зеленым цветом, что составляет 44% от объема кодов-расстояний. Остальные 56% компонентов треугольников могут быть определены с помощью аксиомы:  $d_1 \oplus d_2 \oplus d_3 = 0 \rightarrow d_3 = d_1 \oplus d_2$ . В общем случае функциональная зависимость отношения восстанавливаемых сторон замкнутого в треугольник пространства к общему числу  $n$  слоев треугольной структуры определяется выражением:

$$\eta = \frac{\sum (i+1)}{\sum (3 \times i)}, (i = \overline{1, n}).$$

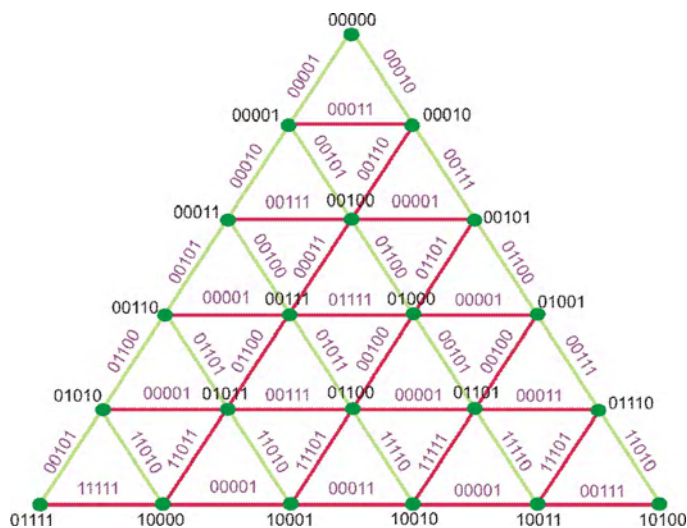


Рис. 3. Triangle Cyber Space

Данное выражение можно записать в виде функциональной зависимости, которая более технологична для вычислений:

$$\eta = \frac{1}{3} \left( \frac{2}{n+1} + 1 \right) = \frac{1}{3} \cdot \frac{2+n+1}{n+1} = \frac{1}{3} \cdot \frac{n+3}{n+1} = \frac{n+3}{3(n+1)}$$

График отношения количества передаваемой информации к ее полному восстанавливаемому объему расстояний в треугольном кибернетическом пространстве представлен на рис. 4. Предельное значение графика стремится к 33%. Это означает – для больших информационных объемов треугольного пространства достаточно передать третью часть всех кодов-расстояний, чтобы затем восстановить полное пространство кодов.

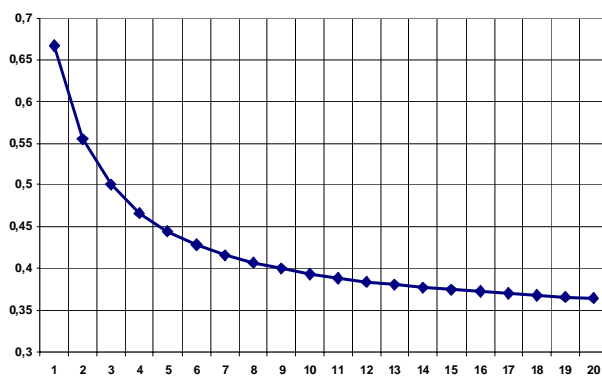


Рис. 4. Функция числа расстояний для восстановления пространства

Треугольная метрика пространства есть самая экономичная, поскольку она создает самые короткие расстояния и пути между объектами, благодаря наличию транзитивного замыкания. Соседство объекта с шестью точками делает метрику более предпочтительной по сравнению с «Манхэттен» в части выбора путей для достижения объекта в пространстве. Оптимальная структура мультипроцессорной системы должна быть составлена из треугольников. Здесь оптимальность определяется соотношением длины пути между любыми двумя точками и общим числом сторон, формирующих треугольное пространство. Длина пути здесь всегда будет не хуже, чем в «Манхэттене», но общее число связей больше на одну диагональ в каждом четырехугольнике. Это дает возможность существования прямых контактов для 6 соседей, что есть существенно для мультипроцессора. «Манхэттен» с диагональными связями обладает той же длиной пути, что и треугольное пространство, но имеет при этом на одну связь больше. Здесь задействована вторая диагональ четырехугольника, что обеспечивает соседство с восемью вершинами пространства. Поскольку треугольник самая примитивная фигура, формирующая плоскость, то вес одной стороны, которую не следует хранить для идентификации пространства, будет максимальным в треугольнике. Далее собранные в систему треугольники дают уже 66% структурных компонентов, которые можно не описывать при формировании пространства, но которые при желании можно восстановить. Все другие примитивные плоскостные фигуры дают меньший выигрыш при формировании ими Cyber Space.

Что касается отношений двух точек в пространстве, то здесь также действует правило «треугольника»: сумма по модулю 2 двух n-мерных векторов дает расстояние (отрезок), которое соединяет в отношении два объекта:

$$\begin{aligned} a \oplus b &= ab; \\ a \oplus b \oplus ab &= 0; \\ a &= b \oplus ab; \\ b &= a \oplus ab. \end{aligned}$$

Следовательно, пространство можно структурировать не только плоскостными фигурами (треугольник, квадрат), но и отношением двух точек, формирующим линию. Тогда структура пространства может быть представлена, например, ломаной, но можно уже незамкнутой линией, последовательно соединяющей точки пространства (рис. 5).

Имея одну точку и все расстояния, можно вычислить все точки структуры. При этом передача расстояний через канал связи гарантирует обнаружение однократных (многократных) искажений кодов без контрольного кода CRC, поскольку сумма всех кодов-расстояний замкнутой фигуры равна нулю. Так же, как и сумма всех  $2^n$  кодов, составляющих полное множество комбинаций, равна нулю.

### 3. Увеличение значности алфавита задания переменных

Рассмотренная алгебра векторной логики и ее практические применения оперирует двоичными переменными на самом примитивном алфавите  $A = \{0,1\}$ . Более сложные алфавиты представляют не меньший интерес для практики, поскольку они, обладая избыточностью, способны сжимать информацию путем применения двух замечательных свойств алгебры логики, основанных на использовании оператора со-граней и поглощения:  $ab \vee a\bar{b} = a$ ;  $a \vee ab = a$ . Первое тождество позволяет исключить несущественную переменную, второе – не учитывать часть, если есть целое.

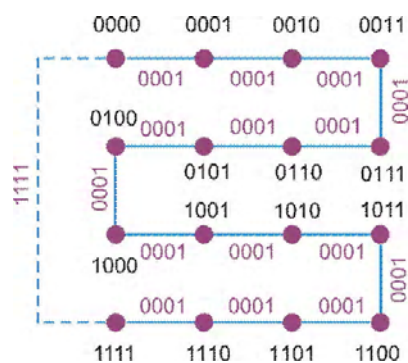


Рис. 5. Lane Cyber Space

Практически ориентированные на решение задач синтеза и анализа дискретной информации алфавиты можно записать в следующем виде:

$$A = \{A^2, A^{2^2}, A^{2^4}, A^{2^{10}}, A^{2^{33}}\}.$$

Все многозначные алфавиты, подобно двужанному множеству, могут создавать векторно-логические пространства, где каждая переменная вектора имеет область определения, равную булеану на универсуме примитивов, который создает множество всех подмножеств. Это значит, что любой алфавит, который является булеаном, может создавать векторное логическое пространство, в котором действует операция симметрической разности (к тому же она симметрична относительно главной диагонали на любом конечном множестве символов):

$\Delta$	0	1	x	$\emptyset$
0	$\emptyset$	x	1	0
1	x	$\emptyset$	0	1
x	1	0	$\emptyset$	x
$\emptyset$	0	1	x	$\emptyset$

Она формирует метрику транзитивного замыкания  $a\Delta b = c$ , что также дает возможность сворачивать замкнутое пространство в вектор пустого множества  $a\Delta b\Delta c = 0$  с помощью симметрической разности. Таким образом, для любого конечного алфавита можно синтезировать табличную операцию симметрической разности, которая способна оценивать расстояние

между объектами, сворачивать пространство расстояний замкнутых структур в  $\emptyset$ -вектор, восстанавливать структуры по остову графа. Сказанное относится и к десяти цифрам, которые выступают в качестве универсума примитивов для образования булеана, содержащего 1024 символа. В этом случае появляется возможность находить дополнение к каждому символу десятичной системы счисления и определять метрику только на основе симметрической разности, которая является аналогом функции хог для теоретико-множественного алфавита. Практический интерес такого алфавита заключается в новой технологии сжатия информации, за счет существенной избыточности алфавита,  $k=1024$ . Кроме того, можно целенаправленно добиваться понижения модуля числа, обозначающего характеристику пикселя в целях последующего применения существующих технологий кодирования (стандарты MPEG, JPEG 2000) для сжатия информации и последующей ее передачи по каналам связи.

В общем случае полная логическая модель функционирования, используя принцип дополнения  $F_i \cup \bar{F}_i = U$  до полного замкнутого множества  $U$ , может быть представлена в виде:

$$\begin{aligned} F &= \{F_1, F_2, \dots, F_1, \dots, F_n\}, \\ F_i \cap F_j &= \emptyset, \cup F_i = U = F, \\ F_i \cup \bar{F}_i &= U, F_i \cap \bar{F}_i = \emptyset, \\ \bar{F}_i &= U \cap \bar{F}_i. \end{aligned}$$

Здесь важную роль играет форма задания функциональности. Таблица истинности (переходов) есть покрытие функциональности конечным множеством кубов или векторов логических переменных, которые в совокупности формируют полное множество состояний объекта:

$$\begin{aligned} C &= \{C_1, C_2, \dots, C_i, \dots, C_n\}, \\ C_i \cap C_j &= \emptyset, \cup C_i = U = C, \\ C_i \cup \bar{C}_i &= U, C_i \cap \bar{C}_i = \emptyset, \\ C_i &= U \cap \bar{C}_i, U = 2^m, \\ m &= m + p + q. \\ Y(t) &= \begin{cases} f[X(t)] \rightarrow f(X_1, X_2, \dots, X_i, \dots, X_m), \\ f[X(t), Z(t-1)], Z(t) = g[X(t), Z(t-1)], \end{cases} \\ X &= (X_1, X_2, \dots, X_i, \dots, X_m), \\ Y &= (Y_1, Y_2, \dots, Y_i, \dots, Y_p), \\ Z &= (Z_1, Z_2, \dots, Z_i, \dots, Z_q), \end{aligned}$$

Для синтеза функциональной структуры необходимо записать спецификацию цифрового устройства в терминах примитивных библиотечных элементов. Последние формируют строки таблицы истинности в виде дизъюнктивной нормальной формы (ДНФ), заданной единичными значениями функции выходов:

$$Y = \bigvee_{i=1}^n \bigwedge_{j=1}^k P_j^a, \quad P = (P_1, P_2, \dots, P_j, \dots, P_k), \quad a = \{0, 1\}.$$

Минимизация дизъюнктивной нормальной формы покрытия функциональности, реализуемой комбинационной схемой, дает решение в виде минимального количества библиотечных примитивов. Процедура минимизации такой частично определенной единичными значениями функции есть решение задачи покрытия.

Пример. Пусть имеется восемь примитивов в библиотеке, которые формируют функциональность в виде следующей таблицы истинности:

P <sub>1</sub>	P <sub>2</sub>	P <sub>3</sub>	P <sub>4</sub>	P <sub>5</sub>	P <sub>6</sub>	P <sub>7</sub>	P <sub>8</sub>	Y
1	.	1	.	.	.	.	1	1
.	1	.	1	.	.	.	1	1
.	.	.	1	.	1	1	.	1
.	.	1	1	.	1	.	.	1
.	.	1	.	.	.	1	1	1
.	1	.	1	.	.	1	.	1
.	1	.	.	1	1	.	.	1
1	.	.	.	1	.	.	1	1

Здесь единичными значениями отмечены не состояния переменных, а необходимость использования примитива для формирования состояния выхода. В данном случае ДНФ функциональности будет иметь вид:

$$\begin{aligned} Y &= P_1 P_3 P_8 \vee P_2 P_4 P_8 \vee P_4 P_6 P_7 \vee P_3 P_4 P_6 \vee \\ &\vee P_3 P_7 P_8 \vee P_2 P_4 P_7 \vee P_2 P_5 P_6 \vee P_1 P_5 P_8. \end{aligned}$$

Данная форма слабо ориентирована на получение оптимального решения – минимального покрытия примитивами заданной функциональности. Если же записать КНФ таблицы по столбцам

$$\begin{aligned} Y &= (P_1 \vee P_3 \vee P_8) \vee (P_2 \vee P_4 \vee P_8) \vee (P_4 \vee P_6 \vee P_7) \vee \\ &\vee (P_3 \vee P_4 \vee P_6) \vee (P_3 \vee P_7 \vee P_8) \vee \\ &\vee (P_2 \vee P_4 \vee P_7) \vee (P_2 \vee P_5 \vee P_6) \vee (P_1 \vee P_5 \vee P_8) \end{aligned}$$

и выполнить переход к ДНФ, то каждый терм полученной функции будет решением задачи покрытия. Однако минимальным покрытием будет в данном случае один из термов минимальной длины, который создает функциональность комбинацией из трех библиотечных примитивов:  $Y = P_6 P_7 P_8 \vee P_3 P_4 P_5 \vee P_2 P_6 P_8$ .

#### 4. Выводы

Компьютер, как субъект экосистемы, имеет свой собственный путь (само-) развития во взаимодействии с остальным миром. Человеку не следует заниматься проблемой очеловечивания компьютера. Никто и никогда уже не станет человеком: ни лошадь, ни дельфин, ни обезьяна, ни компьютер. Но сейчас компьютер, как дитя, нуждается в помощи человека в части надления его способностью к саморазвитию (творчеству) путем создания стандартизированной инфраструктуры информационно-компьютерного пространства, включающей опыт человечества, оформленный в виде всемирной библиотеки математической, технологической и интеллектуальной культуры.

Информационно-компьютерная экосистема планеты есть материальная структура, ориентированная на повышение качества жизни каждого человека. При этом персонализация компьютера трансформируется в индивидуализацию, которая имеет целью создать интеллектуального и верного «друга», способного обслуживать человека 24 часа в сутки по всем сервисам, связанным с бытом, бизнесом, отдыхом, здоровьем.

**Литература:** 1. *Бондаренко М.Ф.* О мозгоподобных ЭВМ / М.Ф. Бондаренко, З.В. Дударь, И.А. Ефимова, В.А. Лещинский, С.Ю. Шабанов-Кушнарченко // Радиоэлектроника и информатика. Харьков: ХНУРЭ. 2004. № 2. С. 89–105. 2. *Cohen A.A.* Addressing architecture for Brain-like Massively Parallel Computers / Euromicro Symposium on Digital System Design (DSD'04). 2004. P. 594-597. 3. *Кузнецов О.П.* Быстрые процессы мозга и обработка образов / Новости искусственного интеллекта. 1998. №2. 4. *Васильев С.Н., Жерлов А.К., Федосов Е.А., Федунцов Б.Е.* Интеллектуальное управление динамическими системами. М.: Физико-математическая литература. 2000. 352 с. 5. *Лунаев В.В.* Программная инженерия. Методологические основы. Учебник. М.: Теис. 2006. 608 с. 6. *А.С.* №1439682. 22.07.88. Регистр сдвига / Какурин Н.Я., Хаханов В.И., Лобода В.Г., Какурина А.Н. 4с. 7. *Гайдук С.М., Хаханов В.И., Обризан В.И., Каменюка Е.А.* Сферический мультипроцессор PRUS для решения булевых уравнений // Радиоэлектроника и информатика. 2004. № 4(29). С.107-116. 8. *Проектирование* и тестирование цифровых систем на кристаллах / В.И. Хаханов, Е.И. Литвинова, О.А. Гузь. Харьков: ХНУРЭ, 2009. 484с. 9. *Проектирование* и верификация цифровых систем на кристаллах. Verilog & System Verilog / В.И. Хаханов, И.В. Хаханова, Е.И. Литвинова, О.А. Гузь. Харьков: Новое слово, 2010. 528с. 10. *Акритас А.* Основы компьютерной алгебры с приложениями: Пер. с

англ. / А. Акритас. М.: Мир. 1994. 544 с. 11. *Аттетков А.В.* Методы оптимизации / А.В. Аттетков, С.В. Галкин, В.С. Зарубин. М.: Изд-во МГТУ им. Н.Э. Баумана, 2003. 440 с. 12. *Abramovici M.* Digital System Testing and Testable Design / M. Abramovici, M.A. Breuer and A.D. Friedman. Comp. Sc. Press. 1998. 652 p. 13. *Densmore D.* A Platform-Based taxonomy for ESL Design / Douglas Densmore, Roberto Passerone, Alberto Sangiovanni-Vincentelli // Design & Test of computers. 2006. P. 359–373. 14. *Автоматизация* диагностирования электронных устройств / Ю.В. Малышенко и др. / Под ред. В.П. Чипулиса. М.: Энергоатомиздат, 1986. 216с. 15. *Трахтенгерц Э.А.* Компьютерные методы реализации экономических и информационных управленческих решений. СИНТЕГ. 2009. 396 с.

Поступила в редколлегию 12.06.2010

**Рецензент:** д-р техн. наук, проф. Кривуля Г.Ф.

**Хаханов Владимир Иванович**, декан факультета КИУ ХНУРЭ, д-р техн. наук, профессор кафедры АПВТ ХНУРЭ. Научные интересы: техническая диагностика цифровых систем, сетей и программных продуктов. Увлечения: баскетбол, футбол, горные лыжи. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 70-21-326. E-mail: hahanov@kture.kharkov.ua.

**Хаханова Анна Владимировна**, канд. техн. наук, ассистент кафедры АПВТ ХНУРЭ. Научные интересы: функционально-ориентированные устройства, частотно-импульсные вычислительные устройства. Адрес: Украина, 61166, Харьков, пр. Ленина, 14.

**Закарян Ваган Ваганович**, студент ХНУРЭ. Научные интересы: техническая диагностика цифровых систем, сетей и программных продуктов. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 70-21-326.