

Выхованец Валерий Святославович

**Синтез эффективных математических моделей
дискретной обработки данных на основе
алгебраической и понятийной декомпозиции
предметной области**

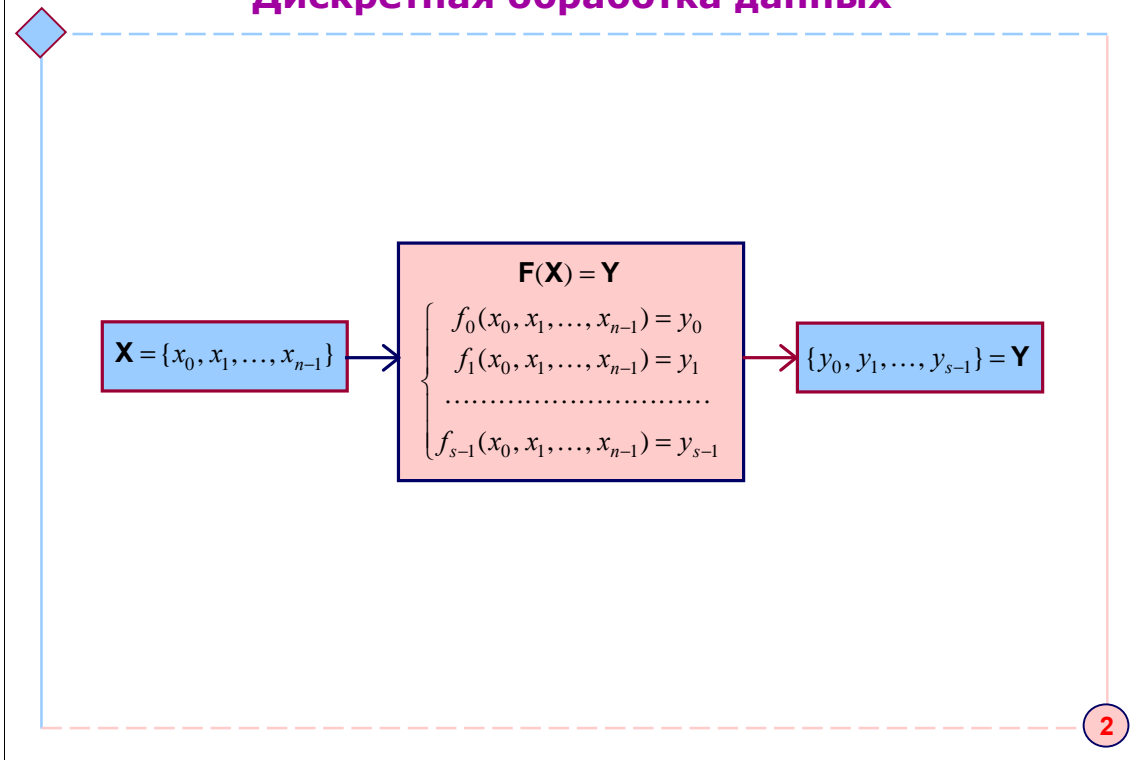
05.13.11 – «Математическое и программное обеспечение
вычислительных машин, комплексов и компьютерных сетей»

05.13.15 – «Вычислительные машины и системы»

Диссертация на соискание ученой степени
доктора технических наук

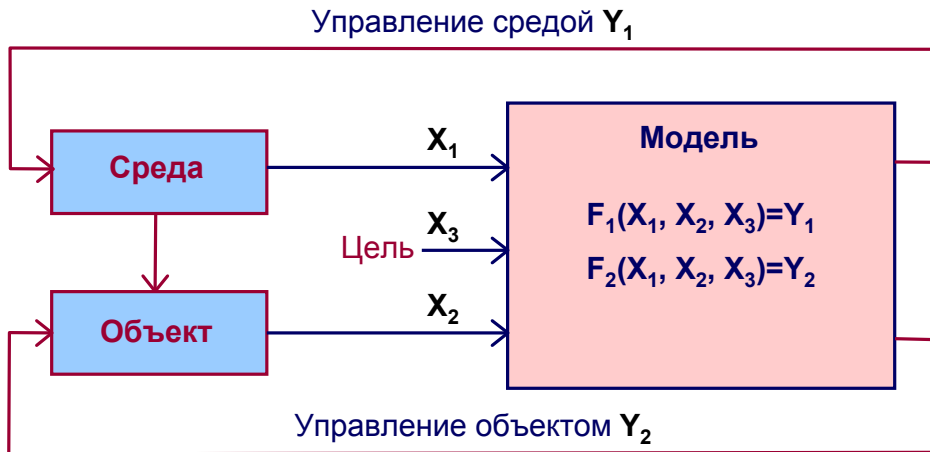
Вашему вниманию предлагаются результаты диссертационного исследования на тему «Синтез эффективных математических моделей дискретной обработки данных на основе алгебраической и понятийной декомпозиции предметной области».

Дискретная обработка данных



Данные при дискретной обработке представляются в виде конечных последовательностей знаков, а сама обработка осуществляется путем преобразования входных данных в выходные посредством разделения входных данных на части (переменные) и выполнения над ними некоторой последовательности операций. В связи с этим возникает задача представления функций большой размерности в виде композиции функций меньшей размерности.

Обработка данных в управлении



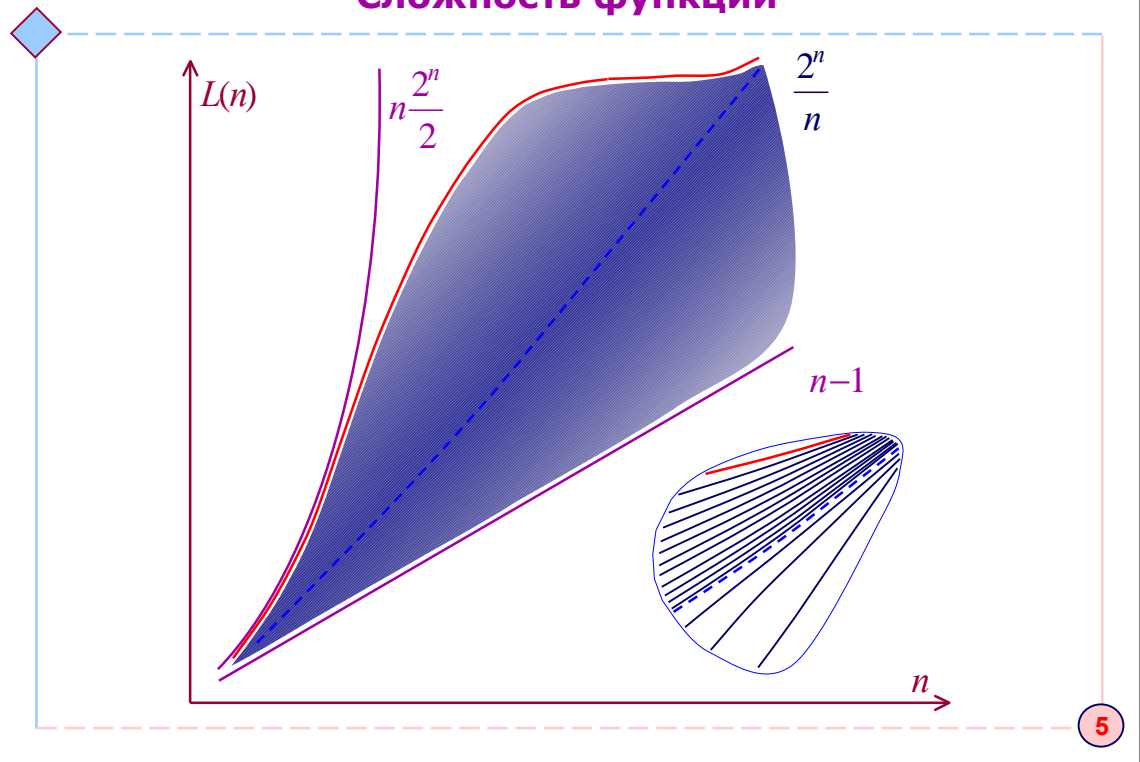
Постановка задачи в управлении базируется на автоматных моделях с памятью. Описание функционирования автомата осуществляется в виде распределенных во времени элементарных действий, каждое из которых является результатом вычисления дискретной функции. Таким образом, главной задачей дискретной обработки данных является дискретная декомпозиция.

Декомпозиции функций

Декомпозиция	Конструкция
Пересекающаяся Закревский, 1964	$f(X) = \theta(X', a(X''))$
Разделительная Ashenurst, 1952	$f(X) = \theta(X', a(X''))$
Кратная Curtis, 1956	$f(X) = \theta(X', a_1(X''), a_2(X''), \dots, a_m(X''))$
Промежуточная Лупанов, 1958	$f(X) = \Sigma(\theta_1(X', a_1(X'')), \dots, \theta_m(X', a_m(X'')))$
Алгебраическая Выхованец, 2001	$f(X) = \sum \theta_i(X') \times a_i(X'')$
Решающая Shannon, 1938	$f(X) = \sum \theta_i(X \setminus x) \times a_i(x)$
Спектральная Карповский, 1970	$f(X) = \sum \theta_i(X) \times a_i$

История декомпозиции связана, в основном, с декомпозицией булевых функций. Установлено, что почти все функции в асимптотической области реализуются со сложностью, близкой к максимальной.

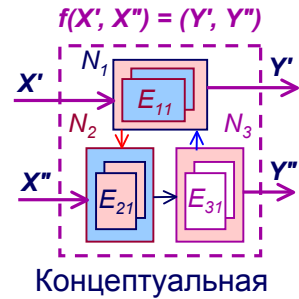
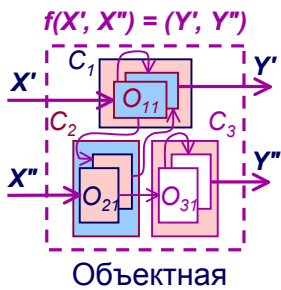
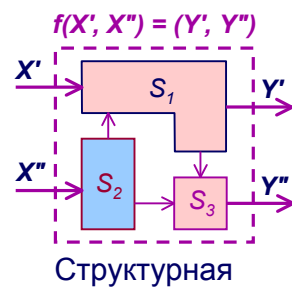
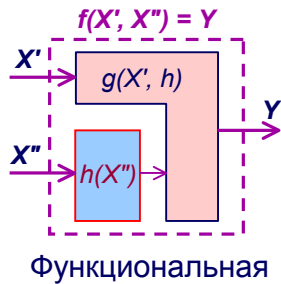
Сложность функций



5

Нерешенной остались задачи определения максимальной сложности функций при конечной размерности задачи; и нахождения методик синтеза, удовлетворяющих этой оценке.

Методологии и технологии

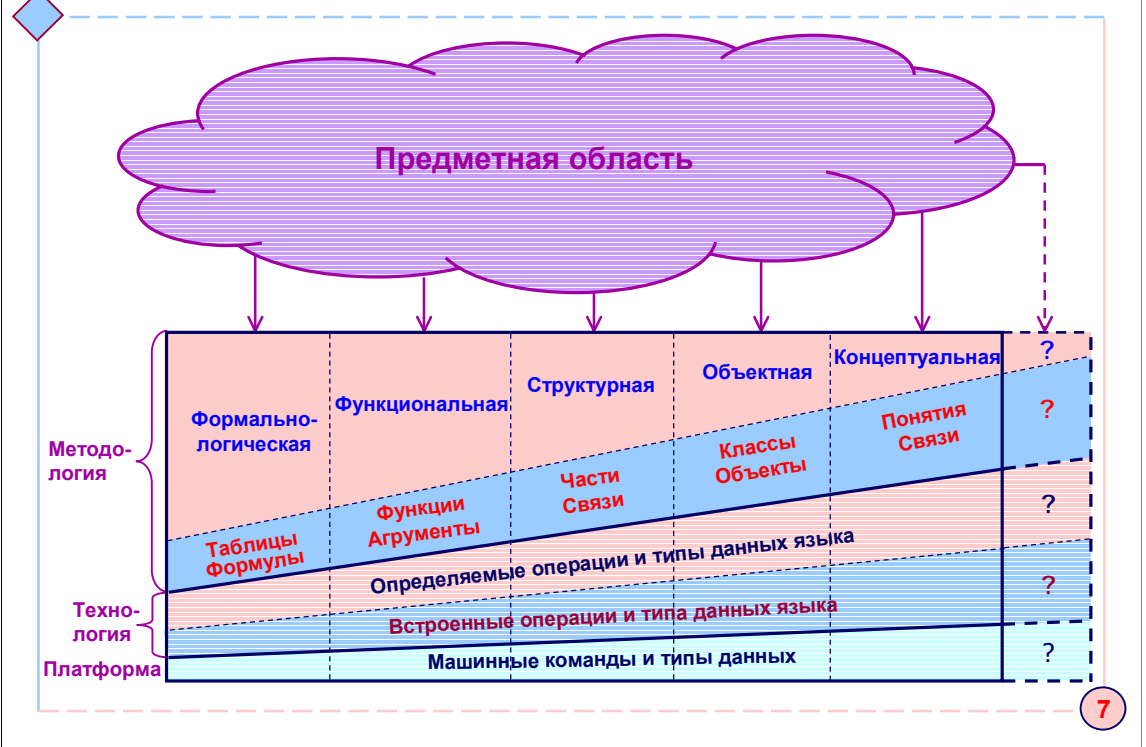


Декомпозиция общего вида оказалась нереализуемой, ибо приводит к **серьезным комбинаторным трудностям**.

В связи с чем дискретную обработку данных представляют в **алгоритмических формах**, на основе поиска и формализации частных декомпозиционных схем.

Теоретической базой для этого служат функциональная, структурная, объектная и концептуальная методологии анализа предметной области, реализуемые соответствующими технологиями программирования.

Семантический разрыв



Несмотря на развитость и завершенность перечисленных методологий, не устраненным остается семантический разрыв между содержательными представлениями о предметной области и теми средствами, которые служат для выражения этих представлений в виде формальных спецификаций.

Для сокращения семантического разрыва используют повышение уровня абстракции языков моделирования. Однако, последнее не затрагивает существенным образом синтаксис языков программирования, что связано с проблемой описания семантики таких языков.

В итоге, описание дискретной обработки данных до сих пор выполняется на интуитивном уровне, с применением неформальных методов, основанных на искусстве разработчиков, их практическом опыте, экспертных оценках и экспериментальных проверках получаемых результатов.

Научная проблема

Научная проблема – теоретическая разработка и практическое решение проблемы синтеза эффективных математических моделей дискретной обработки данных на основе формально-логической и концептуально-онтологической декомпозиции.

Цель работы – разработка теории и методов синтеза математических моделей дискретной обработки данных для ее эффективной реализации аппаратными и программно-аппаратными средствами.

Научная проблема диссертационной работы - теоретическая разработка и практическое решение проблемы синтеза эффективных математических моделей дискретной обработки данных на основе формально-логической и концептуально-онтологической декомпозиции предметной области.

Цель – разработка теории и методов синтеза математических моделей дискретной обработки для ее эффективной реализации аппаратными и программно-аппаратными средствами.

Предлагаемый подход



Особенностью выполненных исследований является применение **двух противоположных подходов**: формально-логического и концептуально-онтологического. Эти подходы применяются для решения одной и той же задачи – синтеза эффективных математических моделей дискретной обработки данных.

При **формально-логическом** подходе исходными данными являются результаты наблюдения предметной области, представленные в табличной форме.

При **концептуально-онтологическом** подходе синтез осуществляется на основе выражения содержательных представлений эксперта.

Задачи исследования

Общая задача – получение эффективных математических моделей предметной области, где эффективность модели определяется количеством операций, необходимых для реализации дискретной обработки.

Частные задачи:

- развитие общей теории дискретных функций на основе аппарата алгебраической декомпозиции;
- обоснование методики синтеза формальных моделей и нахождение оценок их сложности;
- разработка методологии анализа предметной области, позволяющей получать эффективные семантически замкнутые формальные спецификации;
- создание технологии обработки данных на основе использования декомпозиционных схем предметной области в конструкциях предметного языка.

10

Общей задачей выполненного исследования является получение эффективных математических моделей дискретной обработки данных, где эффективность модели определяется количеством операций, необходимых для ее реализации.

Отсюда следуют **частные задачи** исследования:

- развитие общей теории дискретных функций на основе аппарата алгебраической декомпозиции;
- обоснование методики синтеза формальных моделей и нахождение оценок их сложности;
- разработка методологии анализа предметной области, позволяющей получать эффективные и семантически замкнутые формальные спецификации;
- создание технологии обработки данных на основе использования декомпозиционных схем предметной области в конструкциях предметного языка, создаваемого для решения каждой задачи.

Структура работы

Глава 1 – критический анализ известных результатов в области моделирования и обработки данных.

Глава 2 – алгебраическая декомпозиция дискретных функций.

Глава 3 – алгебраический синтез эффективных математических моделей дискретной обработки данных.

Глава 4 – методология понятийного анализа и контекстная технология обработки данных.

Глава 5 – разнесенный грамматический разбор и система контекстного программирования.

Приложение 1 – синтез форм представления частичных функций.

Приложение 2 – полиномиальная факторизация функций.

Приложение 3 – решение задачи управления лифтом.

Приложение 4 – реализация исчисления предикатов 1-го порядка.

Приложение 5 – описание использованной виртуальной машины.

В **Главе 1** приводится критический анализ известных результатов в области математического моделирования и обработки данных.

В **Главе 2** посвящена алгебраической декомпозиции дискретных функций.

В **Главе 3** рассмотрен алгебраический синтез эффективных формульных представлений.

В **Главе 4** разрабатывается методология понятийного анализа и основанная на нем контекстная технология обработки данных.

В **Главе 5** описан метод разнесенного грамматического разбора и система контекстного программирования.

На защиту выносятся результаты, полученные в главах 2, 3, 4 и 5.

В работе также имеется 5 приложений, которые содержат **поясняющий материал** к основному тексту.

В **Приложении 1** изложена методика синтеза полиномиальных и неполиномиальных форм, применяемых при алгебраической декомпозиции.

В **Приложении 2** рассмотрен алгоритм полиномиальной факторизации и его применение для сжатия изображений.

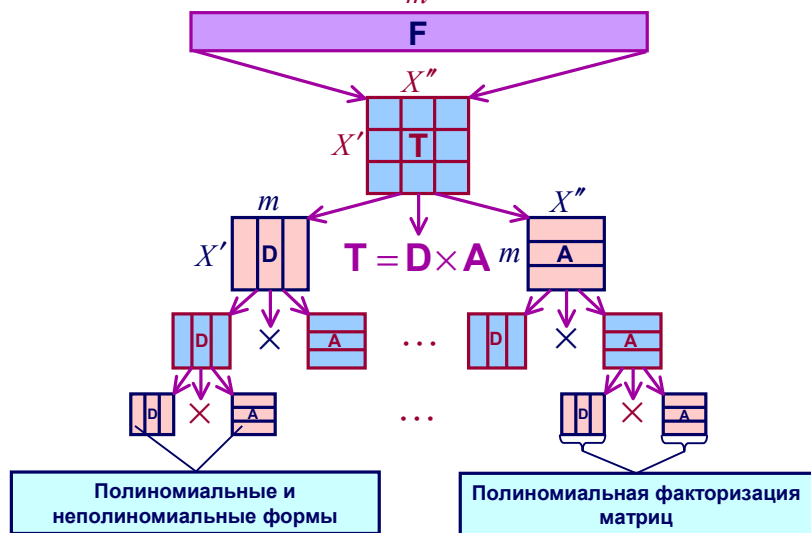
В **Приложении 3** приведен пример эффективного решения средствами контекстной технологии задачи управления лифтом.

В **Приложении 4** реализована система логического программирования, в основе которой лежит исчисление предикатов первого порядка.

В **Приложении 5** описана виртуальная машина, использованная для определения первичных семантических категорий исчисления предикатов

Алгебраическая декомпозиция

$$f(X) = \sum_{i=1}^m \theta_i(X') \times a_i(X'')$$



Рассмотрим **синтез** математических моделей дискретной обработки данных, выполняемый при алгебраической декомпозиции.

Суть **подхода** заключается в том, что после разделения переменных на два непересекающихся подмножества X' и X'' исходная функция $f(X)$ разлагается по системе частных функций θ_i , зависящих от первого подмножества переменных, в результате чего получаются частичные функции a_i , зависящие от второго подмножества.

Найденные функции θ_i и a_i на **следующем шаге** подвергаются аналогичному разложению.

Декомпозиция **завершается**, когда получены функции, которые могут быть реализованы вычислительным средством.

В **Приложении 1** приведена методика синтеза полиномиальных и неполиномиальных форм частных функций, при которой функции представляются в виде формулы в базисе унарных и бинарных операций.

В **Приложении 2** ставится и решается задача полиномиальной факторизации матрицы, при которой матрица функций представляется в виде единой формулы.

Образующие алгебры

$$f(X) = \sum_{i=1}^m \theta_i(X') \times a_i(X''), \quad \mathbf{T} = \mathbf{D} \times \mathbf{A}$$

Алгебра логики

$$R_L = \langle N_k, +, \times \rangle$$

$$+ = \begin{bmatrix} 0 & 1 & 2 & 3 \\ 1 & * & * & * \\ 2 & * & * & * \\ 3 & * & * & * \end{bmatrix} \quad \times = \begin{bmatrix} 0 & 0 & 0 & 0 \\ * & * & * & * \\ * & * & * & * \\ 0 & 1 & 2 & 3 \end{bmatrix}$$

D – матрица перестановок (двухзначные унимодальные функции)

Мультипликативная

$$R_M = \langle N_k, +, \times \rangle$$

$$+ = \begin{bmatrix} 0 & 1 & 2 & 3 \\ 1 & * & * & * \\ 2 & * & * & * \\ 3 & * & * & * \end{bmatrix} \quad \times = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 2 & 3 & 1 \\ 0 & 3 & 1 & 2 \\ 0 & 1 & 2 & 3 \end{bmatrix}$$

D – мономиальная (многозначные унимодальные функции)

Аддитивная

$$R_A = \langle N_k, +, \times \rangle$$

$$+ = \begin{bmatrix} 0 & 1 & 2 & 3 \\ 1 & 0 & 3 & 2 \\ 2 & 3 & 0 & 1 \\ 3 & 2 & 1 & 0 \end{bmatrix} \quad \times = \begin{bmatrix} 0 & 0 & 0 & 0 \\ * & * & * & * \\ * & * & * & * \\ 0 & 1 & 2 & 3 \end{bmatrix}$$

D – биполярная (двухзначные мульти-модальные функции)

Фундаментальная

$$R_F = \langle N_k, +, \times \rangle$$

$$+ = \begin{bmatrix} 0 & 1 & 2 & 3 \\ 1 & 0 & 3 & 2 \\ 2 & 3 & 0 & 1 \\ 3 & 2 & 1 & 0 \end{bmatrix} \quad \times = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 2 & 3 & 1 \\ 0 & 3 & 1 & 2 \\ 0 & 1 & 2 & 3 \end{bmatrix}$$

D – фундаментальная (многозначные мультимодальные функции)

13

Исследованы алгебры, которые могут использоваться для алгебраической декомпозиции. Такие алгебры названы **образующими**.

Классы операций сложения и умножения и ограничения на частичные функции получены исходя из условий существования алгебраического разложения.

Существуют **четыре алгебры**: алгебра логики, мультипликативная, аддитивная и фундаментальные алгебры, различающиеся требованиями к операциям и видом частичных функций θ_i .

В **алгебре логики** требования к операциям минимальны – необходимо только существование нейтральных элементов по сложению (коммутативный нуль) и умножению (левая единица), а частичные функции θ_i образуют матрицу перестановок, т.е. являются двухзначными унимодальными.

Мультипликативная алгебра отличается от алгебры логики тем, что операция умножения образует группу на множестве без нулевого элемента, а частичные функции составляю мономиальную матрицу, т.е. являются многозначными унимодальными (**новый результат**).

Аддитивная алгебра является новой и имеет группу по сложению и частичные функции, задаваемые биполярной матрицей (**новый результат**).

В **фундаментальной алгебре** на операции предъявляются самые серьезные требования – они должны образовывать конечное поле или целостное кольцо, а частичные функции составлять матрицу с ненулевым определителем.

Алгебраический синтез

$$f(X) = \sum_{i=0}^{M'-1} \theta_i(X') \times a_i(X'') = \sum_{i=0}^{M'-1} \left(\theta_i(X') \times \sum_{j=0}^{M''-1} \theta_{ij}(X'') \right)$$

τ	ϕ	ϕ	ϕ	ϕ	ϕ
ϕ	τ	ϕ	ϕ	ϕ	ϕ
ϕ	ϕ	τ	ϕ	ϕ	ϕ
ϕ	ϕ	ϕ	τ	ϕ	ϕ
$*$	$*$	$*$	$*$	τ	ϕ
$*$	$*$	$*$	$*$	ϕ	τ

×

$+$	$+$	$+$	$+$
$+$	$+$	$+$	$+$
$+$	$+$	$+$	$+$
$+$	$+$	$+$	$+$
ϕ	ϕ	ϕ	ϕ
ϕ	ϕ	ϕ	ϕ

=

$+$	$+$	$+$	$+$
$+$	$+$	$+$	$+$
$+$	$+$	$+$	$+$
$+$	$+$	$+$	$+$
\sim	\sim	\sim	\sim
\sim	\sim	\sim	\sim

$$\left\{ \begin{array}{l} M' = k''; \\ M'' \sim \frac{k''}{n''} \end{array} \right. \Rightarrow k'' \sim \frac{k'}{n''} \Rightarrow \left\{ \begin{array}{l} M = M'M'' \sim \frac{m}{n^2}; \\ L = nM \sim \frac{m}{n} \end{array} \right.$$

14

Исследован **эффективный синтез** формул при алгебраической декомпозиции. Для этого степени свободы, связанные с разделением переменных и выбором системы частичных функций использованы для получения более простых формул.

Суть **предлагаемого подхода** заключается в следующем (**новый результат**). Пусть задана прямоугольная таблица функции **T** полученная при разделении переменных на два непересекающихся множества **X'** и **X''** таким, что $k' > k''$. Представим **T** как произведения единичной матрицы **D** и матрицы **A**, равной **T**.

Тогда в любой образующей алгебре $k' - k''$ строк матрицы **A** будут линейно-зависимыми и могут быть выражены как линейная комбинация оставшихся строк. Это позволяет путем выбора элементов, помеченных звездочкой, обнулить линейно-зависимые строки **A** и получить **сложность разложения функции**, равную k'' .

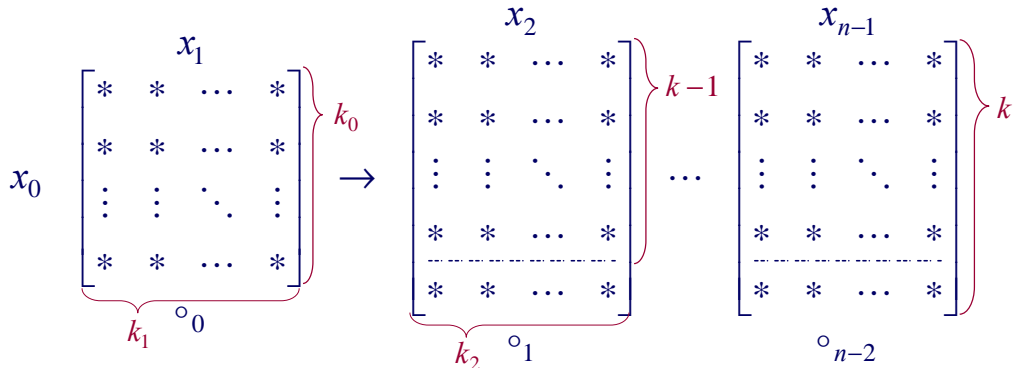
Так как замена некоторого столбца матрицы **D** на его сумму с любым другим столбцом при соответствующем суммировании строк **A** не изменяет **T**, то для уменьшения **сложности представления** построим такой набор функций θ_i , которые выражаются бесповторными бесконечными формулами.

Полученные при этом функции-коэффициенты a_i подвергнем **спектральному разложению**, сложность которого не превосходит k''/n'' , n'' – число переменных в множестве **X''**. Если при разделении переменных положить $k'' = k'/n''$, то в итоге получим максимальную сложность представления функции **L**, где m - длина ее вектора.

Эффективными будем называть такой синтез, при котором имеем количество операций, не превосходящее максимального их числа, достаточного для реализации любой функции той же размерности.

Аналитическая конструкция

$$\theta(X) = x_0 \circ_0 x_1 \circ_1 x_2 \circ_2 \dots \circ_{n-2} x_{n-1}$$



$$N_{\theta}^*(k, k_0, k_1, \dots, k_{n-1}) = k^{(k-\alpha) \left(\sum_{i=0}^{n-1} k_i - \beta \right)}$$

$$0 < \alpha < 1, \quad \beta \approx k$$

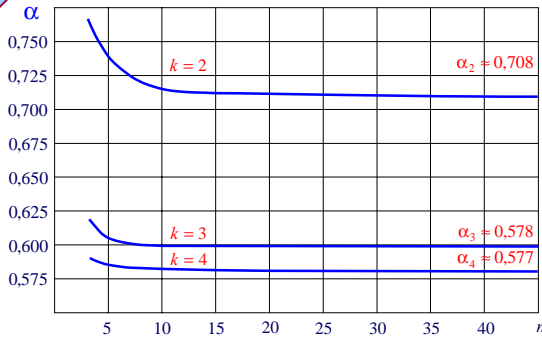
15

Для **получения точных оценок сложности** использована **бесповторная бескобочная** конструкция формул.

Найдено **количество различных функций N_{θ}^*** , порождаемых такой конструкцией, где k – значность образующей алгебры, k_i – значности переменных, α – порождающая способность, а β – параметр, учитывающий начальные условия (**новый результат**).

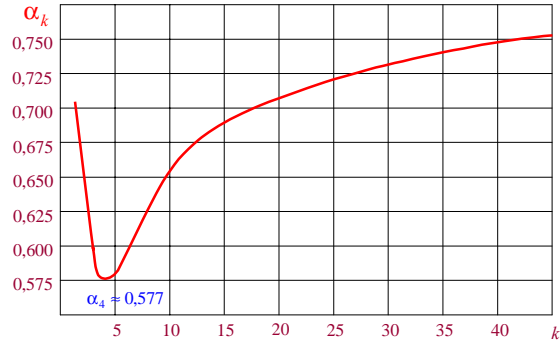
Полученная оценка допускает достаточно простую **содержательную интерпретацию**: верхняя граница степеней свободы аналитической конструкции меньше суммарной площади операций, а нижняя граница определяется при вычеркивании одной строки у каждой операции, кроме нулевой.

Порождающая способность



$$\alpha = k - \frac{\log_k N_{\theta}^* - k_0 k_1}{\sum_{i=2}^{n-1} k_i}$$

$$\alpha_k = k - \frac{1}{k} \log_k \prod_{i=0}^{k-1} \frac{k^k - i}{i+1}$$



Исследована **зависимость порождающей способности α** аналитической конструкции от числа переменных и значности образующей алгебры (**новый результат**).

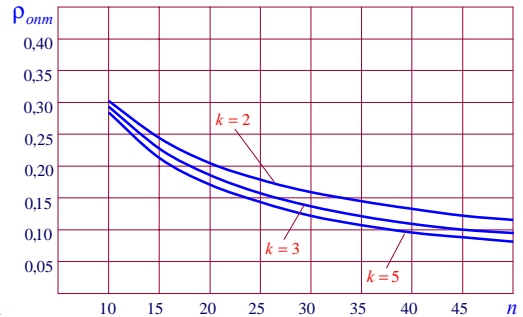
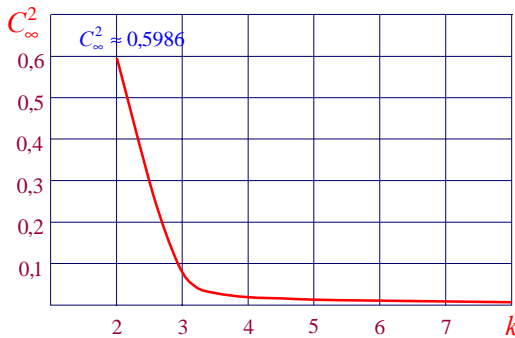
Установлено, что с ростом числа переменных порождающая способность асимптотически быстро стремится к некоторой величине – **предельной порождающей способности α_k** , а **наибольшее число функций** порождается при значности образующей алгебры, равной 4, когда алгебраический синтез является наиболее эффективным (**новый результат**).

Сложность представления

$$L(k, k_0, k_1, \dots, k_{n-1}) = \frac{C^2}{n} \prod_{i=0}^{n-1} k_i$$

$$C = \frac{1}{k - \alpha} \frac{2}{\sqrt{\left(\frac{1}{n} \left(\sum_{i=0}^{n-1} k_i - \sum_{j=0}^{\gamma-1} \beta_j \right) \right)^2 - \rho^2}}$$

$$\rho = \frac{1}{n} \left(\sum_{i=0}^{n'-1} k'_i - \sum_{j=0}^{n''-1} k''_j \right) > 0$$



17

Найдена **точная оценка сложности** представления функции при алгебраическом синтезе (**новый результат**), где, как и ранее, k – значность образующей алгебры, k_i – значности переменных, α – порождающая способность аналитической конструкции, β_j – параметры, учитывающие начальные условия подформулы в скобках, γ – количество таких подформул.

Наименьшая сложность достигается при двухступенчатой декомпозиции ($\gamma=2$) и оптимальном разделении переменных $\rho=\rho_{\text{опт}}$, для выражения которой получена соответствующая зависимость (**новый результат**).

Исследовано **асимптотическое поведение** константы C при различных k (**новый результат**).

Эффективность обработки данных

$$L_{\text{эф}} = \frac{4}{n(k-\alpha)^2} \frac{\prod_{i=0}^{n-1} k_i}{\frac{1}{n^2} (\sum_{i=0}^{n-1} k_i - 2\beta)^2 - \rho_{\text{онм}}^2}$$

$$L_{\text{эф}} \approx 4 \frac{D_y}{D_x} 2^{D_x} - 4r, \quad E = \frac{L_{\text{эф}}}{L}$$

$L_{\text{эф}}$ (L) – эффективное количество операций;

D_x (D_y) – объем входных (выходных) данных, бит;

r – разрядность вычислительного средства, бит;

E – эффективность модели.

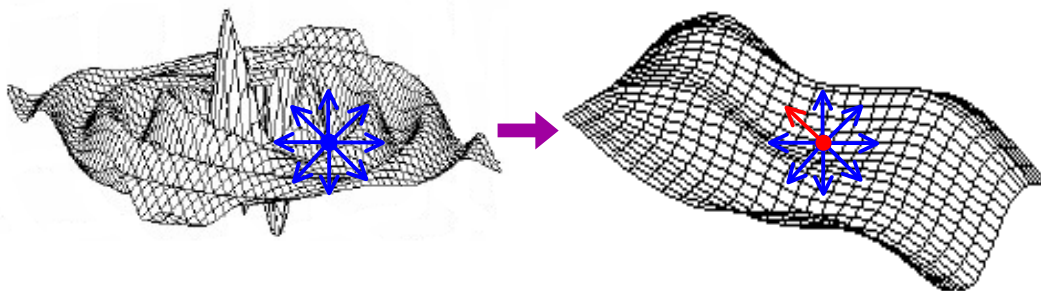
18

Полученные результаты позволяют по объему обрабатываемых данных D_x (D_y) и количеству требуемых для этого операций L определить **эффективность** этой обработки с учетом разрядности вычислительного средства r (**новый результат**).

Трудоемкость алгоритмов алгебраического синтеза достаточно высока. Это объясняется отсутствием ограничений на степени свободы декомпозируемых функций и является следствием предельного абстрагирования общей теории декомпозиции от содержательных представлений.

Для решения прикладных задач большой размерности следует использовать не общую, а **частные декомпозиционные схемы**, которые необходимо искать для каждого класса задач. Имеются определенные надежды на возможность автоматизации такого поиска.

Содержательный подход



Допущение: В процессе изучения предметной области, еще до начала формализации, создается система понятий и декомпозиционные схемы, наиболее приспособленные для решения стоящих прикладных задач.

Рассмотрим второй подход к синтезу математических моделей, при котором **мышление** будем рассматривается как способность сведения многоэкстремальной задачи, которую можно решать только методом перебора, к градиентной, одноэкстремальной. Последнее есть следствие специфического отражения окружающего мира в сознании человека, благодаря которому он решает задачу на основе обобщений и ассоциаций, двигаясь к эффективному результату по прогнозируемому градиенту эмоциональной оценки ситуации.

Таким образом, исследуемый подход основан на **допущении**, что уже в процессе изучения предметной области, еще до начала формализации, создается система понятий и декомпозиционные схемы, наиболее приспособленные для решения стоящих прикладных задач.

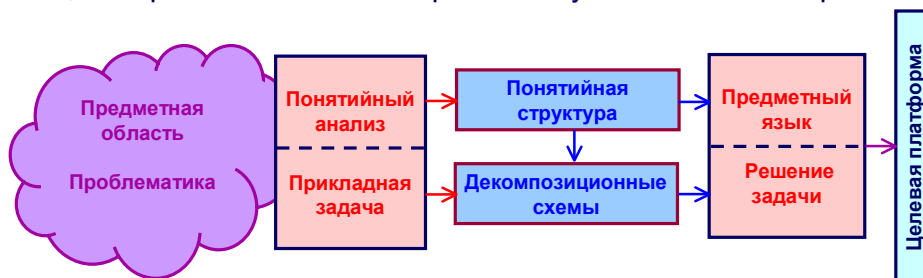
Понятийная декомпозиция

Понятийная декомпозиция – нахождение *понятийного базиса* проблемной области, позволяющего эффективно описать решение стоящей прикладной задачи.

Проблемная область – совокупность предметной области и решаемых в ней задач (проблем).

Предметная область – фрагмент реальной (виртуальной) действительности, представляемый некоторой совокупностью принадлежащих ему сущностей.

Сущность – устойчивое и уникальное представление предметной области, воспринимаемое некоторой совокупностью своих признаков.



20

В основе концептуально-онтологического подхода лежит понятийная декомпозиция, или нахождение понятийного базиса проблемной области, позволяющего в компактной форме выразить решение задачи.

Под **проблемной областью** понимается совокупность предметной области и решаемых в ней задач (проблем), а предметная область рассматривается как фрагмент реальной или виртуальной действительности, представляемый некоторой совокупностью принадлежащих ему сущностей.

Суть содержательного подхода заключается в следующем.

1). Поиск эффективных декомпозиционных схем осуществим на основе **понятийного анализа** предметной области.

2). Для формальной спецификации результатов этого анализа будем строить **специализированный предметный язык**. Для чего:

– выявленную в процессе анализа **понятийную структуру** отразим в понятиях создаваемого языка;

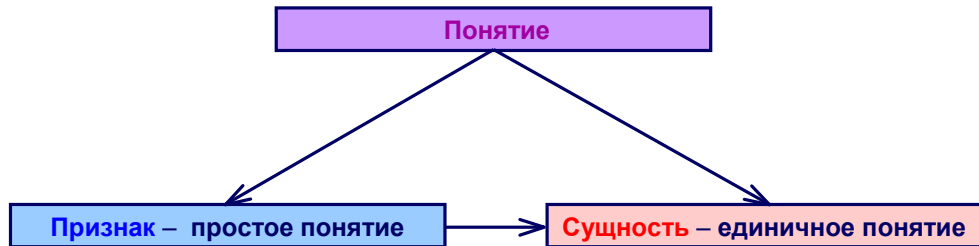
– найденные **декомпозиционные схемы** преобразуем в его языковые конструкции.

3). Определение **вычислительной семантики** специализированного языка осуществим методом семантической индукции на базе первичных семантических категорий, поставляемых **целевой вычислительной платформой**.

4). **Описание решения** задачи реализуем на созданном специализированном языке.

Фрактальность понятия

$$N = \begin{cases} \text{shm } N = (P^0, P^1, \dots, P^{n-1}); \\ \text{int } N = \{(P_j^0, P_j^1, \dots, P_j^{n-1}) \mid j = \overline{0, m-1}\}; \\ \text{ext } N = \{N_0, N_1, \dots, N_{u-1}\}. \end{cases}$$



$$P = \begin{cases} \text{shm } P = (P); \\ \text{int } P = \{(P_0), (P_1), \dots, (P_{u-1})\}; \\ \text{ext } P = \{P_0, P_1, \dots, P_{u-1}\}. \end{cases}$$

$$E = \begin{cases} \text{shm } E = (E); \\ \text{int } E = \{(E)\}; \\ \text{ext } E = \{E\}. \end{cases}$$

21

Воспользуемся известным **формализмом** понятия: понятия будем именовать и задавать схемой, интенционалом и экстенционалом.

Схема – это набор признаков, на которых понятие определено.

Интенционал – наборы значений взаимосвязанных признаков, позволяющим отличать сущности, принадлежащие понятию от других сущностей.

Экстенционал – множество сущностей принадлежащих понятию.

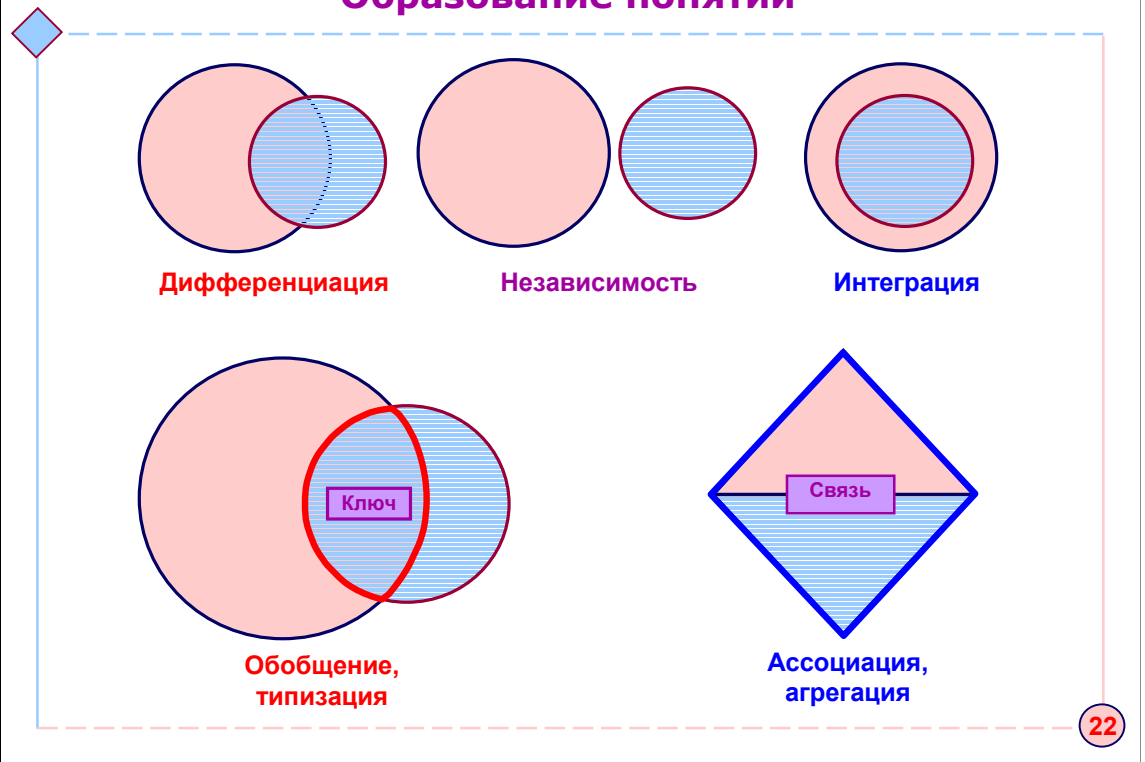
Сущности, или единичные понятия, будем рассматривать как понятия, имеющие экстенционал и схему, равную этому понятию.

Признаки, или простые понятия, представим как понятия, имеющие схему, совпадающую с самим понятием, а экстенционал - состоящий из множества единичных понятий.

Таким образом, **понятие** определяется как именованное множество других понятий, имеющих общую подсхему.

Далее откажемся от использования «сущностей» и «признаков» как специфических терминов, а вместо них будем применять термин «понятие», который в частных случаях может выражать понятие-сущность и понятие-признак (**новый результат**).

Образование понятий



Между понятиями могут быть установлены отношения независимости, дифференциации и интеграции признаков.

Понятия **независимы**, если их признаки не пересекаются.

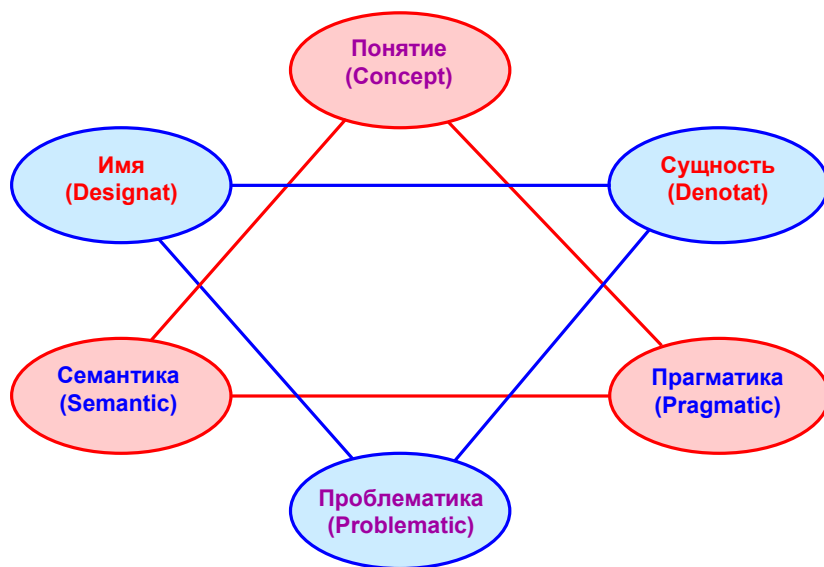
Если у двух понятий имеются общие признаки, то наблюдается **дифференциация** понятий.

Если все признаки одного понятия являются признаками другого понятия, то происходит их **интеграция**.

Понятия, полученные при **обобщении и типизации**, выражают их общность, проявляющуюся при дифференциации.

Понятия, являющиеся **агрегацией и ассоциацией**, задают их взаимосвязь, проявляющуюся при интеграции признаков.

Модель понятия



На слайде показана модель понятия, используемая при понятийном анализе (**новый результат**). Верхняя триада объясняет образование нового понятия, нижняя – выявление уже существующего.

Для образования понятия необходима некоторая проблематики, которая задает аспект анализа предметной области. Благодаря этому выявляются сущности и происходит их означивание, крайними формами которой являются сигнификативная и денотационная.

Обычно понятие используется не в абсолютном, а в некотором относительном смысле, называемом **прагматикой**. Прагматика понятия выявляется при знании **проблематики**. Если понятие рассматривать какместилище всех своих смыслов, то проблематика конкретизирует **семантику** понятия до его прагматики.

В итоге имеем два треугольника: красный, показывающий проявление уже образованного понятия, и синий – выявление новых.

Абстрагирование



Обобщение

$$\left\{ \begin{array}{l} \text{shm } C_G = \bigcap_{j=0}^{m-1} \text{shm } C_j; \\ \text{int } C_G \supseteq \bigcup_{j=0}^{m-1} \text{int } C_j; \\ \text{ext } C_G \supseteq \bigcup_{j=0}^{m-1} \text{ext } C_j. \end{array} \right.$$

Типизация

$$\left\{ \begin{array}{l} \text{shm } C_T = \bigcap_{j=0}^{m-1} \text{shm } C_j, \\ \text{int } C_T = \bigcup_{j=0}^{m-1} \text{int } C_j; \\ \text{ext } C_T = \bigcup_{j=0}^{m-1} \text{ext } C_j; \\ \text{key } C_T \subseteq \text{shm } C_T. \end{array} \right.$$

Ассоциация

$$\left\{ \begin{array}{l} \text{shm } C_B = \bigcup_{j=0}^{m-1} \text{shm } C_j; \\ \text{int } C_B \subseteq \times_{j=0}^{m-1} \text{int } C_j; \\ \text{ext } C_B \subseteq \times_{j=0}^{m-1} \text{ext } C_j; \\ \text{link } C_B \subseteq \text{shm } C_B. \end{array} \right.$$

Агрегация

$$\left\{ \begin{array}{l} \text{shm } C_A = \bigcup_{j=0}^{m-1} \text{shm } C_j; \\ \text{int } C_A = \times_{j=0}^{m-1} \text{int } C_j; \\ \text{ext } C_A = \times_{j=0}^{m-1} \text{ext } C_j. \end{array} \right.$$

Абстрагирование – форма мышления, при которой образуются сложные понятия.

Обобщение – порождение понятия на основе пересечения схем обобщаемых понятий и расширенного объединения их экстенсионалов.

Типизация – порождение понятия на основе пересечения схем типизируемых понятий и точного объединения экстенсионалов. **Ключ** используется для определения типизированного понятия для сущности из экстенсионала понятия типа.

Ассоциация – порождение понятия на основе объединения схем ассоциируемых понятий и ограниченного декартова произведения их экстенсионалов (**новый результат**). **Связь** служит для перехода от сущности одного ассоциированного понятия к связанным с ней сущностям других понятий.

Агрегация – порождение понятия на основе объединения схем агрегируемых понятий и полного декартова произведения их экстенсионалов

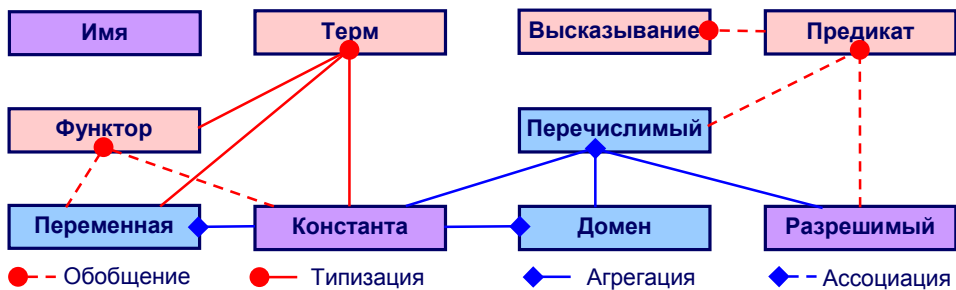
Понятийная структура

Понятийная структура S – множество понятий C , на которых заданы отображения абстрагирования: G – обобщения, T – типизации, A – ассоциации и B – агрегации;

$$S = \langle C, G, T, A, B \rangle.$$

Неполнота – используются понятия, которые явно не определены.

Противоречивость – существует сущность, описанная как принадлежащая, так и как не принадлежащая одному и тому же понятию.



25

Результаты абстрагирования понятий представим в виде **понятийной структуры** (**новый результат**), рассматриваемой как совокупность понятий, на которых заданы способы их образования. Носителем понятийной структуры является множество понятий, а ее сигнатурой – множество отображений обобщения, типизации, агрегации и ассоциации.

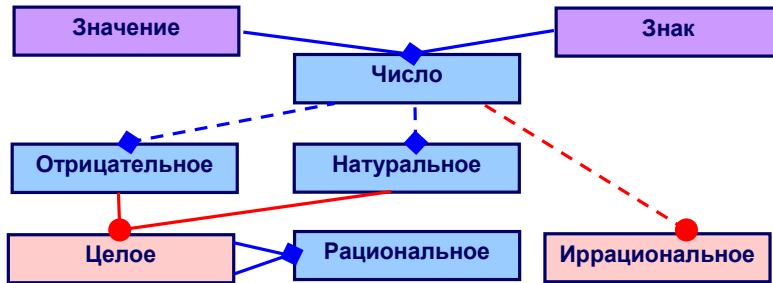
Понятийная структура называется **полной**, если описаны все используемые в ней понятия.

Понятийная структура **противоречива**, если существует сущность, описанная как принадлежащая, так и как не принадлежащая одному и тому же понятию.

В качестве **примера** на слайде показана понятийная структура исчисления предикатов, полученная в Приложении 4 при реализации специализированного языка логического программирования.

Понятийный анализ

Понятийный анализ – построение понятийной структуры предметной области: выявление признаков, сопоставление сущностей, образование понятий, описания структуры, уточнение абстракций, вычисление схем, верификация.



Shm «Рациональное» = («Знак», «Значение», «Знак», «Значение»).

26

Целью **понятийного анализа** является построение понятийной структуры (**новый результат**), которое осуществляется путем:

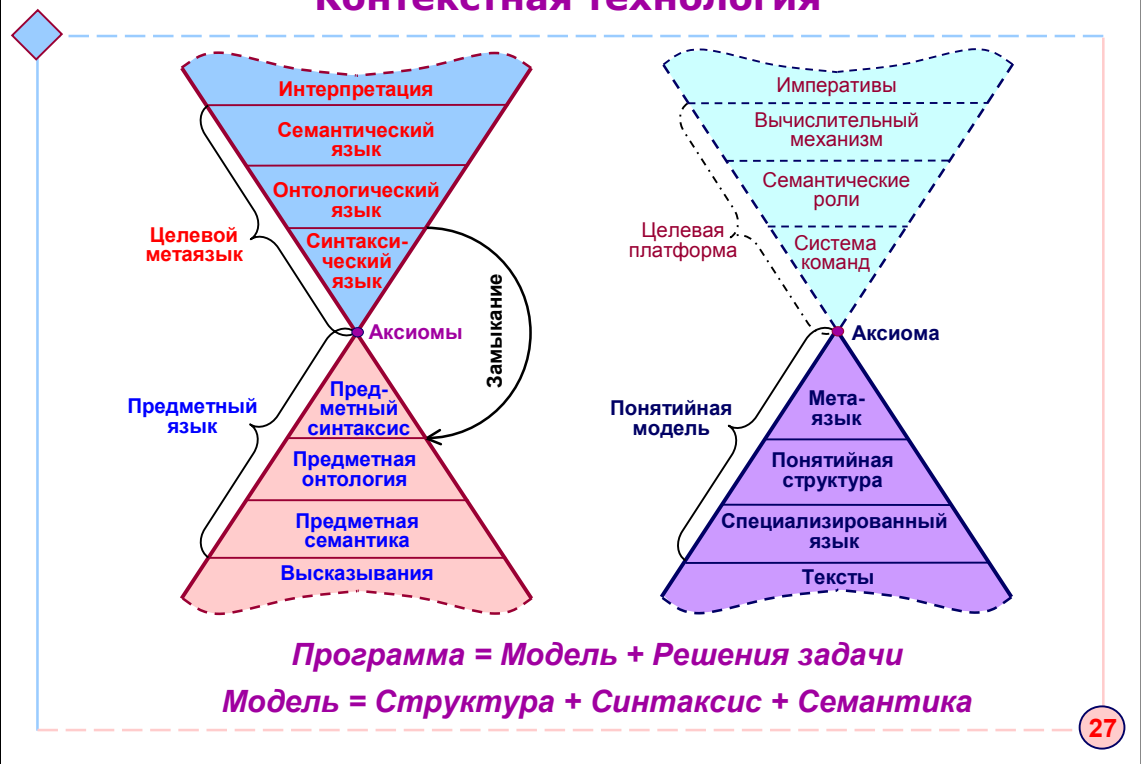
- **выявления** существенных признаков у сущностей;
- **сопоставления** сущностей и определение их общих и различающихся признаков;
- **образования** новых или выявление уже существующих понятий;
- **создания** понятийной структуры;
- **уточнения** способа абстрагирования понятий;
- **вычисления** схем понятий;
- **верификации** понятийной структуры путем проверки ее на полноту и непротиворечивость.

Верификация понятийной структуры основана на вычислении схем понятий по мере их определения.

Схема понятия вычисляется на основе рекуррентной процедуры (**новый результат**):

- схема простого понятия равна самому понятию;
- схема дифференциала равна пересечению схем дифференцируемых понятий;
- схема интеграла – их объединению.

Контекстная технология



Понятийный анализ является методологической основой **контекстная технология** обработки данных.

Особенностью контекстной технологии, отличающей ее от других известных технологий, является построение для решения задачи специализированного языка и определение его вычислительной семантики.

В любой **семиотической системе** выделяются два языка: предметный - для выражения высказываний, и метаязык - для описания семантики этих высказываний.

Предметный язык будем рассматривать как тройку, состоящую из предметного синтаксиса, или правил построения высказываний; предметной онтологии, или системы понятий языка; предметной семантики, или правил интерпретации высказываний.

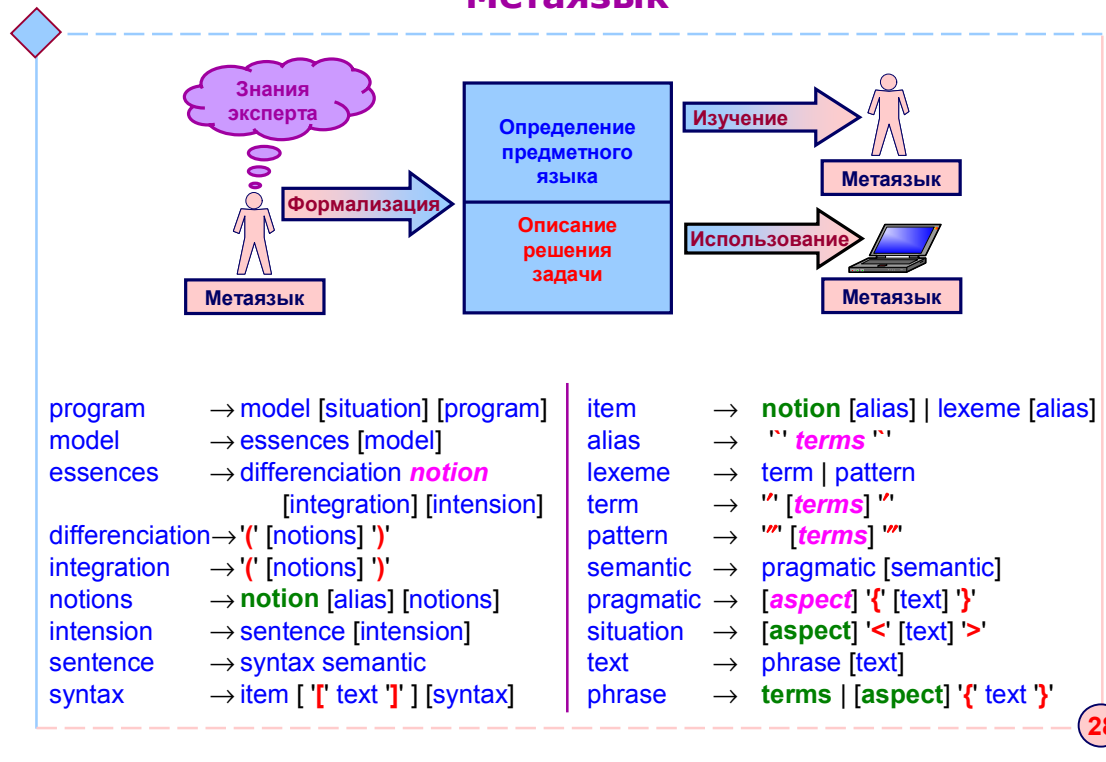
В **метаязыке** выделим три типа выразительных средств: для описания предметного синтаксиса; для определения предметной онтологии; для описания предметной семантики.

Выполним **семантическое замыкание**, т.е. отождествим составные части метаязыка с соответствующими частями предметного языка, а семантику предметного языка будем определять на самом предметном языке (**новый результат**).

Так как внутри **замкнутой семиотической системы** семантику определить нельзя, воспользуемся внешним семантическим интерпретатором – целевой вычислительной платформой, состоящей, как и метаязык, из следующих частей: алфавита метазнаков, семантических категорий и механизма интерпретации метавысказываний.

В итоге получаем **понятийную модель** предметной области, которая состоит из аксиомы, служащей для связи с целевой платформой, метаязыка, предназначенного для выражения синтаксиса специализированного языка, понятийной структуры предметной области и самого специализированного предметного языка, предназначенного для описания семантики и решения задачи (**новый результат**).

Метаязык



28

Метаязык в контекстной технологии выражает наиболее устойчивые положения понятийного анализа и служит для описания синтаксиса специализированного языка (**новый результат**).

Программа состоит из понятийной модели и ситуации, где описывается решение задачи.

Понятийная модель строится как описания сущностей предметной области.

Сущностям приписывается имя понятия, а само **понятие** задается как дифференциация и интеграция ранее объявленных понятий.

Содержание понятия выражается множеством предложений, служащих для выражения интенционала.

Каждое **предложение** состоит из двух частей – синтаксиса и семантики.

Синтаксис предложения выражается последовательностью элементов, каждый из которых либо понятие, либо лексема.

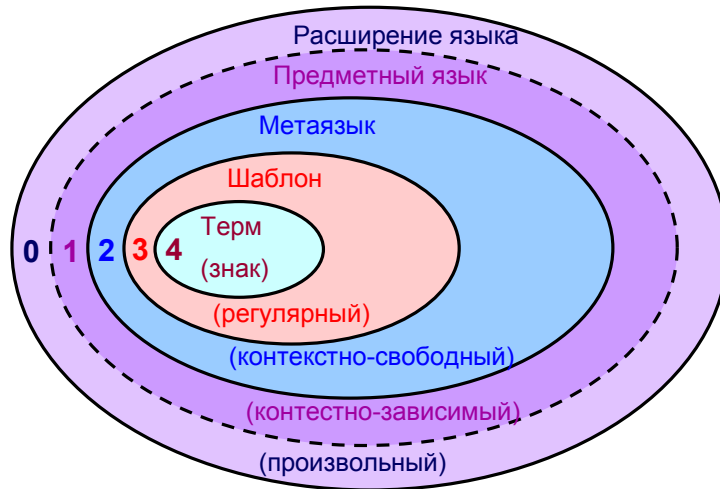
Лексемы представляются термом или множеством термов, выраженных на языке регулярных выражений.

Семантика задается множеством **прагматик**, где **аспект** выражает имя прагматики и позволяет отличать одну прагматику от другой.

Для описания процесса компиляции, прагматик и ситуации используется **текст**, построенный по правилам определяемого специализированного языка.

Привязка понятийной модели к целевой вычислительной платформы осуществляется с помощью **аксиомы**, выражаемой специальным знаком `[]`, инициирующим запись текущего элемента предложения в область формируемого кода.

Иерархия языков



Иерархия языковых средств контекстной технологии включает 5 уровней:

4 уровень – термы: непосредственное использование знаков;

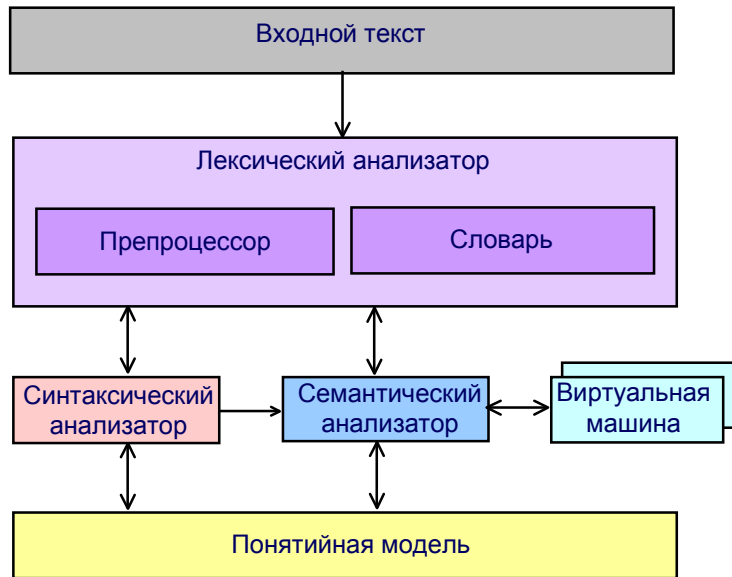
3 уровень – шаблоны: выражение перечислимых множеств термов на языке регулярных выражений;

2 уровень – метаязык: описание синтаксиса на контекстно-свободном языке;

1 уровень – предметный язык: описание семантики и решения задачи на контекстном языке;

0 уровень – расширение предметного языка: трансформационные преобразования компилируемого текста.

Система программирования



30

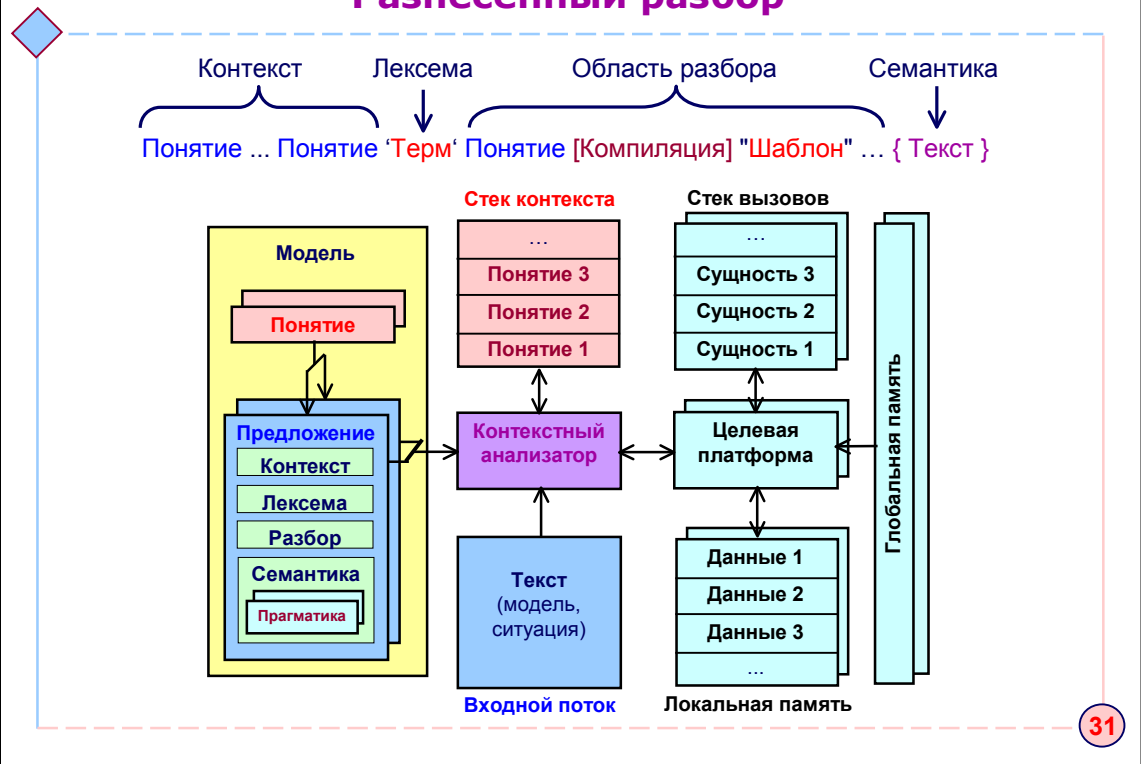
Для экспериментальной проверки положений понятийного анализа и контекстной технологии реализована **система контекстного программирования**.

Входной поток содержит текст программы, результатом обработки которого является понятийная модель. Если во входном файле имеется ситуационное описание, то после компиляции происходит его выполнение.

Лексический анализатор предназначен для контекстного выделения лексем, а **синтаксический анализатор** – для грамматического разбора той части текста, которая описывается грамматикой метаязыка. Другая часть текста, выраженная на определяемом языке, обрабатывается **семантическим анализатором**.

Для выполнения императивов используется одна или несколько целевых платформ, реализованных в виде **виртуальных машин**.

Разнесенный разбор



31

Для повышения эффективности анализа текста разработан метод **разнесенного грамматического разбора** (новый результат).

Для его реализации предложение моетли разделяется на три **области**: контекст, лексему и область разбора.

Поиск предложений, **сопоставимых** с текущем состоянии анализатора, осуществляется путем сравнения контекста предложений с текущим контекстом в стеке контекста.

Из сопоставимых предложений выбираются **применимые**, лексема которых представляется текущим термом из входного потока.

В итоге текст, подлежащий просмотру назад, уже представлен текущим контекстом, а его разбор выполняется до сопоставления предложений.

После выявления применимого предложения наступает этап разбора оставшейся его части, которая еще не сопоставлена входному потоку.

Если все элементы области разбора сопоставлены входному потоку, то предложение считается распознанным, контекст предложения в стеке контекста замещается на его понятие-результат, а полученное состояние входного потока объявляется текущим.

Понятийная модель

```
( ) Variable
  "[A-Za-z][A-Za-z0-9]*" {...}
( ) Constant
  'false' [ asm{ mov eax, 0; push eax } ] {}
  'true' [ asm{ mov eax, -1; push eax } ] {}
(Variable) Logic
  Variable [ asm{ pop ebx; mov eax, [ebx]; push eax } ] {}
  Integer [ asm{ pop eax; cmp eax, 0; je lab; mov eax, -1; lab: push eax } ] {}
  (' Boolean ') {}
(Constant Logic) Negation
  'not' Logic [ asm{ pop eax; not eax; push eax } ] {}
(Negation) Conjunction
  Negation 'and' Negation [ asm{ pop eax; pop edx; and eax, edx; push eax } ] {}
(Conjunction) Disjunction
  Conjunction 'a' 'or' Conjunction 'b' { not a and not b }
(Disjunction) Boolean
  Disjunction 'a' 'imp' Disjunction 'b' { not a or b }

<(not x or y) and z>      fuzzy <(not x or y) and z >
```

32

В качестве примера рассмотрим **понятийную модель** «Исчисление высказываний».

Понятийная структура включает понятия *Переменная*, *Константа*, *Логическое*, *Отрицание*, *Конъюнкция*, *Дизъюнкция* и *Булево*. Понятие **Логическое** обобщено от понятия *Переменная*; *Отрицание* – от *Константа* и *Логическое*; *Конъюнкция*, *Дизъюнкция* и *Булево* – соответственно от *Отрицания*, *Конъюнкции* и *Дизъюнкции*.

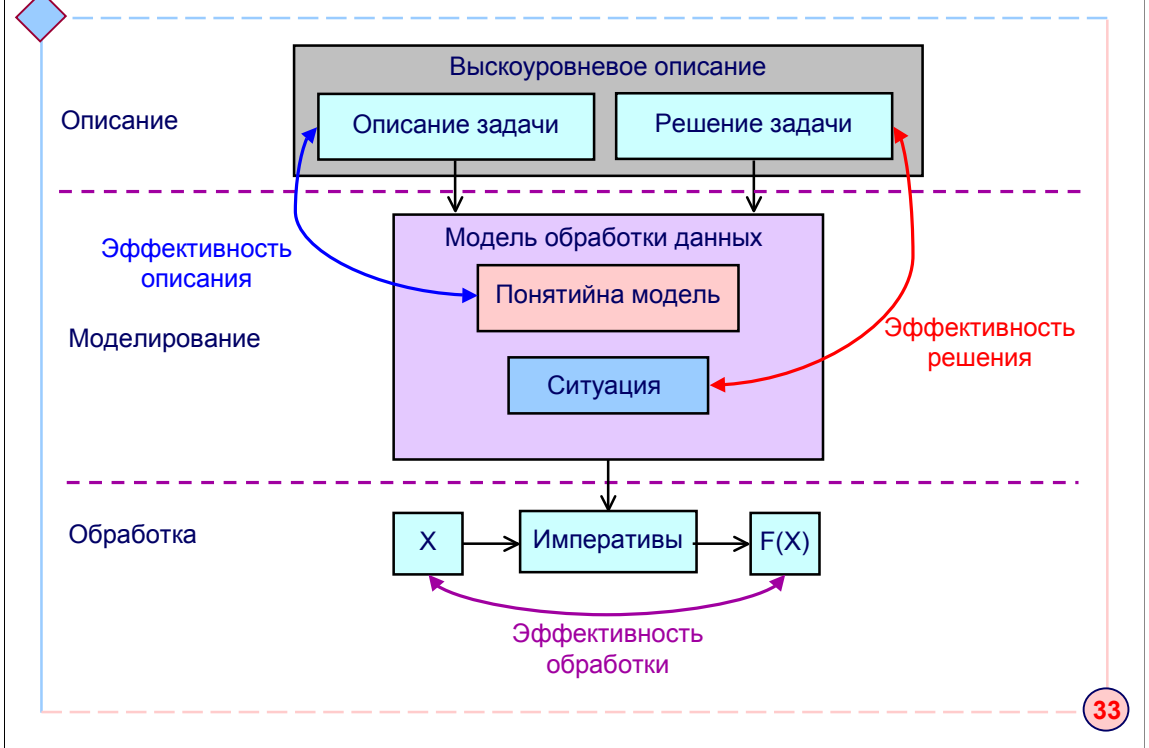
Для выражения перечисленных понятий используются **предложения**: для выражения сущностей понятия *Переменная* имеется предложение, определяющее правила именования переменных в виде шаблона (последовательность букв и цифр, начинающаяся с буквы); константы выражаются двумя предложениями, состоящими из термов *false* и *true*; и т.д. Последнее предложение используется для выражения понятия Булево и состоит из двух понятий *Дизъюнкция* и соединяющего их термина *imp*.

Семантика предложений, кроме последних двух, задана низкоуровневыми средствами. Для этого используется прагматика с именем *asm* (в примере не определена). Текст в фигурных скобках, помеченных аспектом *asm*, передается целевой системе программирования – ассемблеру аппаратной платформы.

Семантика последних двух предложений выражена на уже определенном к тому времени специализированном предметном языке, для чего использованы известные тождества, позволяющие выразить дизъюнкцию и импликацию через ранее определенные операции.

В итоге, текст первой **ситуационной части** приведет к вычислению логического выражения. В рассмотренной понятийной модели возможно задание других прагматик, например, для выполнения вычислений в нечеткой логике. Тогда следующий текст может быть проинтерпретирован как нечеткое высказывание, где *fuzzy* – аспект нечетких вычислений.

Эффективность модели



В контекстной технологии используется два типа **эффективности** – эффективность описания (задачи и ее решения) и эффективность обработки.

Эффективность описания определяет семантический разрыв, который имеет место между высокоуровневой формой постановки и решения задачи и ее выражением на специализированном предметном языке. Измеряется такая эффективность отношением числа знаков в тексте высокоуровневого описания (например, на естественном языке) и в тексте понятной модели и ситуации.

Эффективность обработки вычисляется на основе замера объемов обрабатываемых данных и количества команд, необходимых для этой обработки (**новый результат**).

Научная новизна

Основной результат – теоретическая разработка и практическое решение проблемы синтеза эффективных моделей дискретной обработки данных на основе алгебраической и понятийной декомпозиции.

Частные результаты:

- обобщена теория алгебраической декомпозиции дискретных функций в классе образующих алгебр;
- выполнен количественный и качественный анализ порождающей способности аналитических конструкций формул;
- обоснована методика алгебраического синтеза и найдены точные оценки сложности синтезируемых формул;
- предложена методология понятийного анализа, позволяющая получать семантически замкнутые формальные спецификации;
- решена задача описания семантики формальных языков на основе метода семантической индукции;
- разработана технология контекстной обработки данных, преодолевающая семантический разрыв между языком моделирования и содержательными представлениями относительно предметной области.

34

Основной результат – теоретическая разработка и практическое решение проблемы синтеза эффективных математических моделей дискретной обработки данных на основе алгебраической и понятийной декомпозиции предметной области.

Частные результаты:

- обобщена теория алгебраической декомпозиции дискретных функций в классе образующих алгебр;
- выполнен количественный и качественный анализ порождающей способности аналитических конструкций формул;
- обоснована методика алгебраического синтеза и найдены точные оценки сложности синтезируемых формул;
- предложена методология понятийного анализа, позволяющая получать семантически замкнутые формальные спецификации;
- решена задача описания семантики формальных языков на основе метода семантической индукции;
- разработана технология контекстной обработки данных, устраняющая семантический разрыв между языком моделирования и содержательными представлениями относительно предметной области.

Внедрение

ЗАО «ЛАНИТ» – при автоматической генерации документов на основе понятийного моделирования;

«МАТИ» – РГТУ им. К.Э. Циолковского – при изучении дисциплин «Организация ЭВМ и систем» «Надежность, эргономика и качество АСОИУ», «Программные средства автоматизации»;

ООО «ФРАКТАЛ-М» – для определения структурных характеристик наноматериала по изображению его поверхности;

ГКНПЦ им. М.В. Хруничева (КБ «Салют») – для представления документации в виде модели, позволяющей построить систему операционного контроля изделия с возможностью идентификации нештатных ситуаций.

Перспективы дальнейших исследований:

– в **научно-исследовательской работе** по оценке качества программных и аппаратурных средств на основе использования новой метрики – абсолютной эффективности дискретной обработки данных;

– в открытии **инновационного проекта** по разработке промышленной системы контекстного программирования.

35

Результаты диссертационной работы внедрены:

в **ЗАО «ЛАНИТ»** – при автоматической генерации документов на основе понятийного моделирования;

в **«МАТИ» – РГТУ им. К.Э. Циолковского** – при изучении дисциплин «Организация ЭВМ и систем» «Надежность, эргономика и качество АСОИУ», «Программные средства автоматизации»;

в **ООО «ФРАКТАЛ-М»** – для определения структурных характеристик наноматериала по изображению его поверхности;

в **ГКНПЦ им. М.В. Хруничева (КБ «Салют»)** – для представления документации в виде модели, позволяющей построить систему операционного контроля изделия с возможностью идентификации нештатных ситуаций.

Перспективы дальнейших исследований по теме диссертации видятся:

– в научно-исследовательской работе по оценке качества программных и аппаратурных средств на основе использования новой метрики – абсолютной эффективности дискретной обработки данных;

– в открытии инновационного проекта по разработке промышленной системы контекстного программирования.

Заключение

1). Обобщена **теория алгебраической декомпозиции** дискретных функций на основе объединения алгебраических методов, разделительной декомпозиции и ортогональных разложений в широком спектре алгебраических систем. В частности:

– показано существование четырех типов **образующих алгебр** (алгебры логики, мультипликативной, аддитивной и фундаментальной алгебр), изучены их свойства и доказаны условия существования алгебраических разложений;

– найдены четыре класса **частичных функций** (унимодальные и мультимодальные, двузначные и многозначные), имеющие эффективную реализацию на вычислительных средствах с различными операционными возможностями;

– предложен **метод многоступенчатого алгебраического синтеза** формул, позволяющий учесть нетривиальные свойства декомпозируемой функции и на основе этого получить ее эффективное представление.

Таким образом, в диссертационной работе:

1). Обобщена **теория алгебраической декомпозиции** дискретных функций на основе объединения алгебраических методов, разделительной декомпозиции и ортогональных разложений в широком спектре алгебраических систем. В частности:

– показано существование четырех типов образующих алгебр (алгебры логики, мультипликативной, аддитивной и фундаментальной алгебр), изучены их свойства и доказаны условия существования алгебраических разложений;

– найдены четыре класса частичных функций (унимодальные и мультимодальные, двузначные и многозначные), имеющие эффективную реализацию на вычислительных средствах с различными операционными возможностями;

– предложен метод многоступенчатого алгебраического синтеза формул, позволяющий учесть нетривиальные свойства декомпозируемой функции и на основе этого получить ее эффективное представление.

Заключение

2). Разработан **метод синтеза эффективных моделей** дискретной обработки данных при двухступенчатой алгебраической декомпозиции дискретных функций, базирующаяся на следующих результатах:

– выполнен **анализ порождающей способности** аналитических конструкций формул и установлено, что для эффективного синтеза достаточно использовать аналитическую конструкцию неповторных бескобочных формул с фиксированным порядком переменных;

– доказано, что при алгебраическом синтезе не требуется проверять все варианты разделения переменных, **минимизация сложности представления** обеспечивается оптимальным их разделением, гарантирующим неповторное бескобочное представление;

– получены точные, приближенные и асимптотические **оценки сложности синтезируемых формул** и показано, что требуемое количество операций не хуже известных наилучших оценок;

– найдена **методика оценки абсолютной эффективности** и степени минимизации моделей дискретной обработки данных при конечной размерности решаемых задач, что позволяет сравнивать по эффективности различные модели.

37

2). Разработан **метод синтеза эффективных моделей** дискретной обработки данных при двухступенчатой алгебраической декомпозиции дискретных функций, базирующаяся на следующих результатах:

– выполнен анализ порождающей способности аналитических конструкций формул и установлено, что для эффективного синтеза достаточно использовать аналитическую конструкцию неповторных бескобочных формул с фиксированным порядком переменных;

– доказано, что при алгебраическом синтезе не требуется проверять все варианты разделения переменных, **минимизация сложности представления** обеспечивается оптимальным их разделением, гарантирующим неповторное бескобочное представление;

– получены точные, приближенные и асимптотические оценки сложности синтезируемых формул и показано, что требуемое количество операций не хуже известных наилучших оценок;

– найдена методика оценки абсолютной эффективности и степени минимизации моделей дискретной обработки данных при конечной размерности решаемых задач, что позволяет сравнивать по эффективности различные модели.

Заключение

3). Предложена **методология понятийного анализа**, позволяющая получать эффективные декомпозиционные схемы предметной области в виде семантически замкнутых формальных спецификаций. Для этого:

– формализованы основные виды абстракции понятий и найдены **методы верификации понятийной структуры** предметной области путем ее тождественного преобразования и проверки на полноту и непротиворечивость;

– разработан **метаязык** для спецификации результатов понятийного анализа, предусматривающий выразительные средства для определения понятийной структуры, синтаксиса выражения понятий и многоаспектного описания семантики;

– найден универсальный **метод описания семантики** формальных языков на основе семантической индукции, осуществляемой посредством синтаксического и семантического замыкания понятийной модели путем определения семантических категорий по мере необходимости, в процессе описания и описанными ранее средствами.

3). Предложена **методология понятийного анализа**, позволяющая получать эффективные декомпозиционные схемы предметной области в виде семантически замкнутых формальных спецификаций. Для этого:

– формализованы основные виды абстракции понятий и найдены методы верификации понятийной структуры предметной области путем ее тождественного преобразования и проверки на полноту и непротиворечивость;

– разработан метаязык для спецификации результатов понятийного анализа, предусматривающий выразительные средства для определения понятийной структуры, синтаксиса выражения понятий и многоаспектного описания семантики;

– найден универсальный метод описания семантики формальных языков на основе семантической индукции, осуществляемой посредством синтаксического и семантического замыкания понятийной модели путем определения семантических категорий по мере необходимости, в процессе описания и описанными ранее средствами.

Заключение

4). Создана **контекстная технология обработки данных** на основе отражения понятийной структуры предметной области в понятиях создаваемого специализированного предметного языка, а получаемые при понятийном анализе декомпозиционные схемы – в его языковых конструкциях. Для реализации системы контекстного программирования:

– доказано, что **выразительные возможности** языковых средств контекстной технологии обработки данных эквивалентны формализму контекстных грамматик;

– разработан **метод разнесенного грамматического разбора**, позволяющий выполнять эффективный анализ текста, порожденного, в том числе и неоднозначными и контекстными грамматиками;

– предложен **метод определения прагматики** понятийной модели в виде многоаспектного описания семантики, позволяющий использовать различные средства: от команд процессора целевой вычислительной платформы до текстов на целевом языке программирования.

4). Создана **контекстная технология обработки данных** на основе отражения понятийной структуры предметной области в понятиях создаваемого специализированного предметного языка, а получаемые при понятийном анализе декомпозиционные схемы – в его языковых конструкциях. Для реализации системы контекстного программирования:

– доказано, что выразительные возможности языковых средств контекстной технологии обработки данных эквивалентны формализму контекстных грамматик;

– разработан метод разнесенного грамматического разбора, позволяющий выполнять эффективный анализ текста, порожденного, в том числе и неоднозначными и контекстными грамматиками;

– предложен метод определения прагматики понятийной модели в виде многоаспектного описания семантики, позволяющий использовать различные средства: от команд процессора целевой вычислительной платформы до текстов на целевом языке программирования.

Сравнительный анализ формализма

Известные формализмы:

- семиотическая модель понятия (Моррис, 1938);
- концептуальный анализ (Никаноров, 1972);
- семантическая сеть (Коллинз и Квилиан, 1969; Цейтин, 1985);
- исчисление предикатов (Кольмероз, 1975);
- теория концептуальной зависимости (Шенк, Ригер, 1974);
- концептуальное моделирование (Плесневич, 2004);
- формальный анализ понятий (Вилли и Гантер, 1999);
- концептуальные графы (Сова, 1984)
- EER-модель (Чен, 1976; Броди и Мулополос, 1984).

Отличия понятийной структуры:

- нет деления на понятия, связи и признаки;
- явное выражение типизации понятий;
- представление ассоциации как самостоятельного понятия;
- использование одной диаграммы для обобщений и ассоциаций;
- одновременное выражение понятия как обобщения и ассоциации;
- семантическая прозрачность.

Сравнительный анализ технологии

Известные технологии:

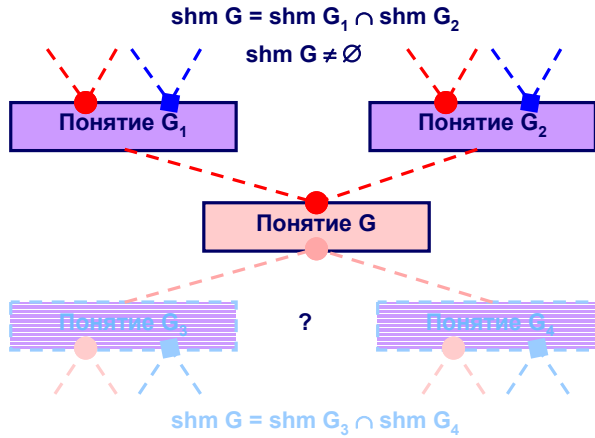
- морфологические грамматики (Джонсон, 1972);
- АТN-грамматики (Вудс, 1978; Хорошевский, 1983);
- грамматики с фазовой структурой (Хейндорн, 1975);
- трансформационные грамматики (Хомский, 1986);
- сопоставление с образцом (Турчин, 1968; Пильщиков, 1983);
- унификационный формализм (Каспер и Раундс, 1986);
- логические грамматики (Кнут, 1968; Кольмероз, 1975);
- категориальные грамматики (Ламбек, 1958);
- программирование в ограничениях (Сарасват, 1993);
- функциональные грамматики (Холидей, 1985; Тузов, 2003);

Отличия контекстной технологии:

- контекстная интерпретация фрагментов текста;
- обобщение и ассоциация нетерминальных знаков;
- расширяемость метаязыка и предметного языка;
- описание семантики на предметном языке;
- множество целевых платформ и пополняемость примитивов.

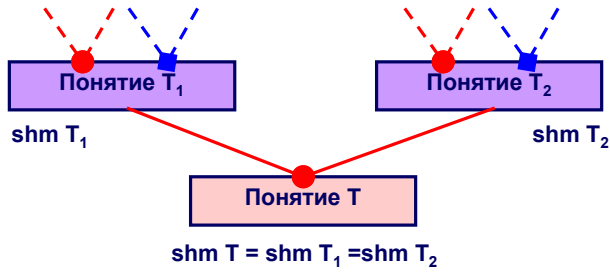
Свойства обобщения

Утверждение 1. Понятийная структура содержит не более чем одно обобщение для каждого включенного в нее понятия.



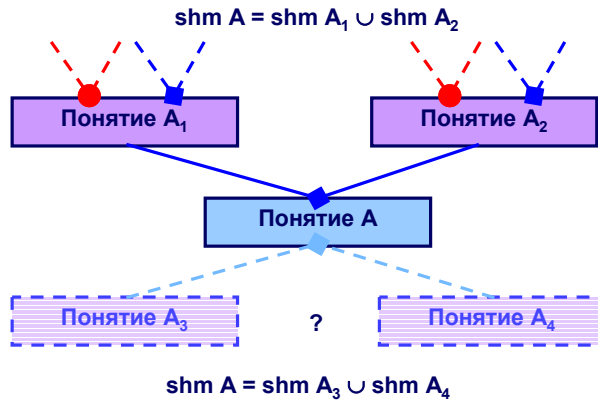
Свойства типизации

Утверждение 2. В понятийной структуре типизируемые и типизированное понятия имеют одинаковые схемы.



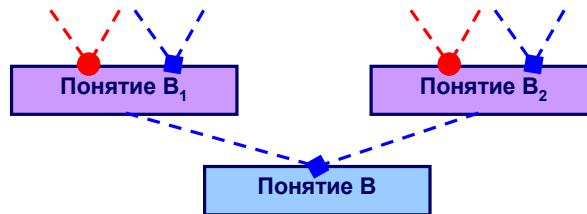
Свойства агрегации

Утверждение 3. Для непротиворечивости понятийной структуры относительно отображений агрегации необходимо и достаточно единственности агрегации у каждого понятия-агрегата.



Свойства ассоциации

Утверждение 4. Для непротиворечивости понятийной структуры относительно отображений ассоциации необходимо и достаточно существования непустого пересечения схем ассоциируемых понятий у каждого понятия-ассоциации.



$$\text{shm } B = \text{shm } B_1 \cup \text{shm } B_2$$

$$\text{link } B \neq \emptyset$$

Семантическая индукция

Семантическая индукция – определении семантических категорий по мере необходимости, в процессе определения предметного языка и средствами этого языка.

База индукции – первичные семантические категории, которые непосредственно реализуются целевой вычислительной платформой и декларируются на метаязыке перед использованием.

Предположение индукции – все ранее определенные семантические категории, выражающиеся фрагментами текста, которые построены по правилам, описываемым синтаксисом предложений.

Индуктивный переход – описание прагматики нового или уже существующего предложения на уже определенном до этого предметном языке.

Заключение индукции – определение новой семантической категории.

(Conjunction) Disjunction

Conjunction `a` 'or' Conjunction `b` { not a and not b }

(Disjunction) Boolean

Disjunction `a` 'imp' Disjunction `b` { not a or b }

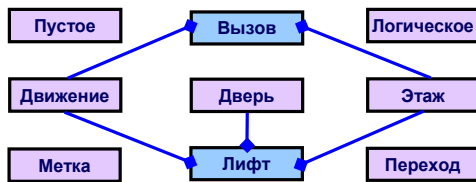
Лифт (задача)

Дано многоэтажное здание с одним лифтом.

На каждом этаже – кнопки для вызова лифта на движение вверх и вниз.

В кабине имеется панель с кнопками для перемещения на этажи.

Разработать систему управления лифтом.



1. Ожидание вызова

Если вызовов нет, то ожидать вызов (1). Если вызов со второго этажа, то перейти к открытию дверей (2), иначе – на принятие решения о движении (4).

2. Открытие дверей

Открыть двери. Если дверь открылась, то перейти на принятие решения о движении (4), иначе повторить открытие двери (2).

3. Закрытие дверей

Закрыть дверь. Если дверь не закрылась, то повторить закрытие двери (3), иначе перейти к определению направления движения (4).

4. Направление движения

Продолжение движения. Если лифт двигался вверх (вниз) и имеются вызовы на движение вверх (вниз), то начать движение вверх (5) (вниз 6). **Изменение направления.** Если вызовов на движение вверх (вниз) нет, но есть вызовы на движение вниз (вверх), то начать движение вниз (6) (вверх 5). **Возврат в начало.** Если вызовов нет и при этом лифт выше (ниже) второго этажа, то начать движение вниз (6) (вверх 5), иначе перейти в состояние ожидания (1).

5. Движение вверх

Начать движение вверх. Если при проходе этажа имеется вызов на движение вверх, то остановить лифт и открыть дверь (2), иначе продолжить движение вверх (5).

6. Движение вниз

Начать движение вниз. Если при проходе этажа имеется вызов для движения вниз, то остановить лифт и открыть дверь (2), иначе продолжить движение вниз (6).

Лифт (решение)

```
( ) Движение ( )
    "[Дд]вижение" {}
( ) Дверь ( )
    "[Дд]верь" {}
( ) Этаж ( )
    'этаж' "[0-9]" {}
    'этаж' 'вызова' "[вверх|вниз]" {}
    'этаж' 'лифта' {}
( ) Вызов (Этаж Движение)
    'вызов' {}
( ) Лифт (Этаж Движение Дверь)
    'лифт' {}
( ) Метка ( )
    "[А-Яа-я][А-Яа-я0-9]" {}
( ) Переход ( )
    Метка {}
( ) Логическое ( )
    Этаж "выше|ниже|равен" Этаж {}
    Вызов "вверх|вниз|нет" {}
    Дверь "открыта|закрыта" {}
    'не' Логическое {}
( ) ( )
    ':' {}
    Метка ':' {}
    Движение "вверх|вниз|останов" {}
    Дверь "открыта|закрыта" {}
    'Если' Логическое ',' 'то' Переход {}
    'Если' Логическое ',' 'то' Переход ','
        'иначе' Переход {}
```

Ожидание:

Если вызов нет, то Ожидание.
Если этаж вызова равен этаж 2,
то Открыть, иначе Решение.

Открыть:

Дверь открыть.
Если дверь открыта, то Решение,
иначе Открыть.

Закреть:

Дверь закрыть.
Если дверь закрыта, то Решение,
иначе Закреть.

Решение:

Если лифт вверх и вызов вверх, то Вверх.
Если лифт вниз и Вызов вниз, то Вниз.
Если не вызов вверх и вызов вниз, то Вниз.
Если не вызов вниз и вызов вверх, то Вверх.
Если вызов нет и этаж лифта выше этаж 2,
то Вниз.
Если вызов нет и этаж лифта ниже этаж 2,
то Вверх, иначе Ожидание.

Вверх:

Движение вверх.
Если этаж лифта равен этаж вызова вверх,
то Открыть, иначе Вверх.

Вниз:

Движение вниз.
Если этаж лифта равен этаж вызова вниз,
то Открыть, иначе Вниз.

Лифт (эффективность)

Длина исходного текста, написанного экспертом на естественном языке, равна **1545** знакам, причем этот текст не содержит определение своего синтаксиса и не раскрывает семантику терминов и понятий, которые в нем использованы.

Длина понятийной модели и ситуационного описания равна **1606** знакам, причем понятийная модель содержит описание синтаксиса, но не содержит описание семантики.

Предварительная эффективность реализации системы управления лифтом определяется семантическим разрывом между исходным описанием и решением задачи на специализированном языке – $1606/1545=1,04$.

Итоговая эффективность. Если описание семантики будет реализовано с той же эффективностью, что и общее описание, то эффективность всего решения существенно не изменится.

Вывод: семантический разрыв между текстом исходного описания и текстом программы практически отсутствует.