УДК 004.422+510.23+510.25+510.54+512.567

КОНСТРУКЦИОННО-ПРОДУКЦИОННАЯ МОДЕЛЬ СТРУКТУР ДАННЫХ НА ЛОГИЧЕСКОМ УРОВНЕ

В.И. Шинкаренко, В.М. Ильман, Г.В. Забула

Днепропетровский национальный университет железнодорожного транспорта
им. академика В. Лазаряна 49010, Днепропетровск, ул. акад. Лазаряна, 2
E-mail: shinkarenko_vi@ua.fm, zabulus12@gmail.com

Разработана обобщенная конструктивно-продукционная структура, которая аккумулирует возможности различных грамматик и грамматико-подобных систем по формированию конструкций с элементов различной природы. Отличительная особенность формируемых конструкций заключается в применении аппарата атрибутов к элементам, их связям, частям конструкций, конструкций в целом. Рассматривается специализация обобщенной конструкционно-продукционной структуры по формированию структур данных на логическом уровне. Приведена конкретизация конструкционно-продукционной структуры на примере логической структуры BMP-файлов.

Developed generalized constructive-synthesizing structures that accumulate scope of various grammars and grammar-like systems intended to produce constructions consist of different elements. Distinctive feature of produced constructions is attribute applying framework for elements, their connections, construction parts and entire constructions. Specialization of generalized constructive-synthesizing structure to produce data structure on the logical level considered. Concretization of generalized constructive-synthesizing structure for logical structure of BMP-file shown.

Введение

Для теоретического обоснования результатов выполненных экспериментальных исследований [1, 2] по повышению временной эффективности, и адаптации структур данных появилась необходимость формализации представления структур данных. Известные способы формализации на основе систем алгоритмических алгебр [3, 4] не позволяют в полной мере формализовать структуры на логическом и физическом уровне и их взаимосвязи.

Предлагаются формальные структуры как средство задания множества конструкций. Определяющий признак объединения конструкций в множество – это особенность их строения. Под конструкциями будем понимать составные, структурированные объекты или конструируемые процессы с заданными свойствами их составляющих.

На основе анализа известных модификаций классических грамматик: матричных [5], индексных [6, 7], разнообразных графических [6, 8–10], стохастических [9], программных [11, 6, 12], атрибутивных [5], деревьев [9], и других разработана обобщенная конструктивно-продукционная структура. Учтены так же возможности грамматико-подобных систем, таких как R-системы [13], L-системы [10], которые, по сути, являются модификациями грамматик, хотя не представляются грамматическими средствами.

Обобщенная конструктивно-продукционная структура

Особенность конструктивно-продукционных структур состоит в формировании множеств конструкций с помощью операций связывания, подстановки и вспомогательных операций, задаваемых правилами аксиоматики. Конструкции формируются в результате выполнения операции вывода.

Определение 1. Обобщенной конструктивно-продукционной структурой (ОКПС) назовем:

$$C = \langle M, \Sigma, \Lambda \rangle, \tag{1}$$

где M – носитель структуры, Σ – сигнатура, состоящая из множеств возможных имен многоместных операций и отношений: связывания, подстановки, вывода и вспомогательных операций; Λ – конструктивная аксиоматика, которая включает множество определений, аксиом, правил, свойств, инструкций и пр. Множества M и Σ определяется аксиоматикой.

Выделим в аксиоматике Λ семантически завершенные части – подаксиоматики: носителя грамматической структуры, сигнатуры, операций связывания, подстановки и вывода. Рассмотрим отдельно каждую подаксиоматику.

Подаксиоматика носителя грамматической структуры. Чтобы не усложнять структуру формализацией относительно ее атрибутики будем считать, что имеет место следующие аксиомы.

Аксиома 1. Носитель M структуры C состоит из конструктивных элементов с набором атрибутов.

У каждого элемента может быть свой набор атрибутов.

Атрибуты могут быть связаны как со статическими свойствами элементов (такими как цвет, объем и т. п.), так и динамическими (способность перемещаться определенным образом и т. п.), а также составными свойствами (например, в терминах объектно-ориентированного программирования объект представляется набором свойств и методов, при этом свойства в свою очередь могут быть объектами).

Аксиома 2. Обязательными атрибутами элементов носителя являются:

- идентифицирующий атрибут – имя, указатель местоположения, набор этих или других свойств;
- атрибуты, связанные с семантикой.
- атрибут со значением «терминал» или «нетерминал».

Атрибуты, связанные с семантикой и местоположением кроме конкретных значений, могут иметь значения «не определено» и «любое»;

Наличие набора атрибутов w_i у элемента носителя m_i будем обозначать как ${}_{w_i}m_i$ (идентификатор m_i с атрибутом w_i), а то, что w_{i_j} является атрибутом идентификатора m_i – $w_{i_j} \leftarrow\!\lrcorner m_i$. Набор атрибутов задается кортежем длины k – $w_i = \langle w_{i_1}, w_{i_2}, \ldots, w_{i_k} \rangle \in W_1 \times W_2 \times \ldots \times W_k = W$. Таким образом носитель – $M = \{{}_{w_i}m_i\}$.

Здесь и далее идентифицирующие атрибуты (по местоположению) будем обозначать индексом справа, другие атрибуты – индексом слева.

Аксиома 2 устанавливает наличие взаимно-однозначного соответствия между элементом и идентифицирующим атрибутом. Можно считать, что элемент обладает идентифицирующим атрибутом или идентификатор имеет атрибут «элемент» («значение»).

В формальных теориях элементы носителя принято обозначать их идентифицирующим атрибутом, т.е. под обозначением m понимается идентификатор элемента носителя, а не сам элемент (значение). В дальнейшем изложении будем обозначать: m – идентификатор элемента, $\leftarrow\!\lrcorner m$ – элемент с идентификатором m.

Следствие 1. Из аксиомы 2 имеем свойства пустого элемента ε: $w_i = (\varepsilon, \ldots, \varepsilon, w_{i_j}, \varepsilon, \ldots, \varepsilon) = w_{i_j}$; $w_i = (\varepsilon, \ldots, \varepsilon) = \varepsilon$; ${}_{\varepsilon}m = m$; ${}_{\varepsilon}\varepsilon = \varepsilon$; $\varepsilon \in M$; $\{m_i\} \subset \{{}_{w_i}m_i\}$.

Следствие 2.

- $\{{}_{w_i}m_i\}$ – мультимножество по идентификаторам с различными атрибутами;
- согласно аксиомам 1, 2 можно выделить подмножества носителя: $T = \{{}_{w_i}t_i\}_{i=1}^{I}$ – терминалов, $N = \{{}_{w_j}\beta_j\}_{j=1}^{J}$ – нетерминалов;
- значение атрибута терминал/нетерминал будем определять по принадлежности к множеству T или N;
- $T \bigcup N = M$, $T \bigcap N = \varnothing$, $\varepsilon \in T$, $\varepsilon \notin N$.

Аксиома 3. $\varepsilon \in W_i$, $i = 1, \ldots, k$.

Определение 2. Два элемента ${}_{w_i}m_i, {}_{w_j}m_j \in M$ равны ${}_{w_i}m_i = {}_{w_j}m_j$, если $m_i = m_j \,\&\, w_i = w_j$ и равны по идентификатору ${}_{w_i}m_i \cong {}_{w_j}m_j$, если $m_i = m_j$.

Аксиома 4. В множестве N должно быть выделено множество начальных нетерминалов $U \subset N$.

Замечание. Как правило [8, 14, 9] множество начальных нетерминалов состоит из одного элемента $U = \{\sigma\}$.

Подаксиоматика сигнатуры.

Аксиома 5. Сигнатура состоит из имен операций, обладающих набором атрибутов.

Произвольная операция сигнатуры представляется как ${}_{\nu}\circ \in \Sigma$, где $\nu = \langle \nu_1, \nu_2, \ldots, \nu_k \rangle \in V_1 \times V_2 \times \ldots \times V_k = V$ (ν – набор атрибутов из множества V).

Сигнатура Σ состоит из множества операций: Ξ – связывания, Θ – подстановки и вывода, Φ – операций над атрибутами, а также отношения подстановки – «$\rightarrow$». $\Sigma = \langle \Xi, \Theta, \Phi, \{\rightarrow\} \rangle$.

Свойства подмножеств сигнатуры. $\Xi \bigcap \Theta = \varnothing$; $\Xi \bigcap \Phi = \varnothing$; $\Theta \bigcap \Phi = \varnothing$, $\varepsilon \in \Phi$.

Аксиома 6. Обязательным атрибутом операций сигнатуры является интерпретация – правила и порядок ее выполнения.

Подаксиоматика операций связывания. Операции связывания элементов ОКПС связывают отдельные элементы в конструкции или их части.

В классических формальных грамматиках [8, 9, 14] используется одна бинарная операция связывания (конкатенации) над элементами терминального и нетерминального алфавитов. В специализированных грамматиках используются разнообразные операции связывания: по условию [6–9], многоместные [6, 8, 9, 15], графических элементов [6, 9] и др.

Пусть ${}_{\nu}\otimes$ – произвольная k-местная операция связывания из множества Ξ с набором атрибутов ν.

Аксиома 7. Для операции ${}_{\nu}\otimes$ справедливо ${}_{\varepsilon}\otimes = \otimes$.

Определение 3. Будем называть формой ${}_{w_l}l$ с набором атрибутов w_l:

${}_{w_l}l = {}_{\nu_0}\otimes({}_{w_1}m_1, {}_{w_2}m_2, \ldots, {}_{w_k}m_k)$ для $\forall {}_{w_i}m_i \in M$;

${}_{w_l}l = {}_{w_j}m_j$, если $l = {}_{\nu_0}\otimes(\varepsilon, \ldots, \varepsilon, {}_{w_j}m_j, \varepsilon, \ldots, \varepsilon)$;

${}_{w_l} l = {}_{\nu_0}\otimes({}_{w_1} l_1,\ {}_{w_2} l_2, \ldots,\ {}_{w_k} l_k)$, если ${}_{w_1} l_1,\ {}_{w_2} l_2, \ldots,\ {}_{w_k} l_k$ – формы.

Из определения 3 следует, что операция связывания применяется как к элементам носителя так и к формам, сконструированных с ее помощью на основе элементов носителя.

Аксиома 8. Носитель M расширяется сконструированными формами.

В классических алгебраических операциях результат операции не содержит никаких информативных признаков об операндах. Определение 3 выделяет класс операций, результат которых содержит информацию об операндах.

Определение 4. Если ${}_{w_l} l = {}_{\nu_0}\otimes({}_{w_1} l_1,\ {}_{w_2} l_2, \ldots,\ {}_{w_k} l_k)$, то ${}_{w_1} l_1,\ {}_{w_2} l_2, \ldots,\ {}_{w_k} l_k$ назовем подформами формы ${}_{w_l} l$, что будем обозначать ${}_{w_i} l_i \prec {}_{w_l} l$.

Определение 5. Формы, в которых отсутствуют нетерминальные элементы, будем называть конструкциями и обозначать ${}_{\Delta, w} l$, где Δ – определяющий форму атрибут.

Определение 6. Назовем множество всех форм, конструируемых на носителе этой структуры операциями связывания, свободным множеством форм $(T \bigcup N)^* = F^*$.

Определение 7. Свободным множеством конструкций формальной структуры назовем все допустимые конструкции, сформированные на носителе структуры операциями связывания T^* .

Свойства операций связывания:

– $\forall w \in W : {}_{w_l} l = {}_{\nu_0}\otimes(\varepsilon, \varepsilon, \ldots, \varepsilon) = \varepsilon$;

– $\forall {}_{w_i} l_i, {}_{w_j} l_j : {}_{w_i} l_i = {}_{\nu_0}\otimes({}_{w_1} m_1,\ {}_{w_2} m_2, \ldots,\ {}_{w_q} m_q, {}_{w}\ m_h, \ldots,\ {}_{w_k} m_k)$, ${}_{w_j} l_j = {}_{\nu_0}\otimes({}_{w_1} m_1,\ {}_{w_2} m_2, \ldots,$ ${}_{w_h} m_h, {}_{w_q} m_q, \ldots,\ {}_{w_k} m_k)$ ${}_{w_i} l_i \neq {}_{w_j} l_j$.

Свойство вложенности множеств: $T \subset_C T^* \subset F^*$ и $N \subset F^*$.

Подаксиоматика подстановки и вывода. Подстановки и выводы в грамматиках предназначены для выделения из свободных языков формальных языков. Существует большое разнообразие операций подстановки и вывода в различных грамматических средствах [5, 6, 8-13, 15].

Будем различать отношение подстановки и операцию подстановки.

Определение 8. Отношение постановки – двуместное отношение с атрибутами ${}_{\nu_p}({}_{w_i} l_i, {}_{w_j} l_j)$.

Аксиома 9. Обязательный атрибут отношения подстановки – атрибут доступности ${}_{d}\nu$.

Замечание. В грамматиках отношение подстановки принято обозначать в инфиксной форме $l_i \to l_j$ и в классических грамматиках всегда ${}_{d}\nu = 1$. Для наглядности будем придерживаться аналогичных обозначений ${}_{w_i} l_i \ \ {}_{\nu_p}\!\!\to \ {}_{w_j} l_j$.

Определение 9. Пусть $s = \left\langle {}_{w_1} l_1 \ \ {}_{\nu_2}\!\!\to \ {}_{w_3} l_3,\ {}_{w_4} l_4 \ \ {}_{\nu_5}\!\!\to \ {}_{w_6} l_6, \ldots, {}_{w_m} l_m \ \ {}_{\nu_{m+1}}\!\!\to \ {}_{w_{m+2}} l_{m+2} \right\rangle$ – последовательность отношений подстановки или $s = \varepsilon$, и $g = \left\langle {}_{\nu_1}\oplus_1^{k_1}(w_{1,1}, w_{2,1}, \ldots, w_{k_1,1}),\ {}_{\nu_2}\oplus_2^{k_2}(w_{1,2}, w_{2,2}, \ldots, w_{k_2,2},), \ldots,\right.$ $\left. {}_{\nu_n}\oplus_n^{k_n}(w_{1,n}, w_{2,n}, \ldots, w_{k_n,n}) \right\rangle$ – последовательность операций над атрибутами. Назовем правилом продукции $p : \langle s, g \rangle$. Здесь $\oplus$ – произвольная операция над атрибутами ($\oplus \in \Phi$).

Множество правил продукций будем обозначать $\Psi = \{\psi_i : \langle s_i, g_i \rangle\}$.

Определение 10. Пусть задана форма ${}_{w_l} l = {}_{\nu_0}\otimes({}_{w_1} l_1,\ {}_{w_2} l_2, \ldots,\ {}_{w_h} l_h, \ldots,\ {}_{w_k} l_k)$ и доступное отношение подстановки ${}_{\nu_p}({}_{w_h} l_h,\ {}_{w_q} l_q)$ такое, что ${}_{w_h} l_h \prec {}_{w_l} l$, тогда результатом трехместной операции подстановки ${}_{w_l^*} l^* = {}_{\nu_p}\Rightarrow({}_{w_h} l_h, {}_{w_q} l_q,\ {}_{w_l} l)$ будет форма ${}_{w_l^*} l^* = {}_{\nu_0}\otimes({}_{w_1} l_1,\ {}_{w_2} l_2, \ldots,\ {}_{w_q} l_q, \ldots,\ {}_{w_k} l_k)$, где $\Rightarrow \in \Theta$.

Определение 11. Двухместная операция частичного вывода ${}_{w_l^*} l^* = {}_{\nu_p}|\Rightarrow(\Psi,\ {}_{w_l} l)$ ($|\Rightarrow \in \Theta$) заключается в:

– выборе одного из доступных правил подстановки $p_r : \langle s_r, g_r \rangle$ с отношениями подстановки s_r ;

– выполнении на его основе операций подстановки;

– выполнении операций над атрибутами g_r в соответствующей последовательности.

Замечание 1. Выполнение операций над атрибутами g_r – часть операции частичного вывода. Один из атрибутов операций над атрибутами определяет последовательность выполнения операций над атрибутами: в начале операции частичного вывода или после его окончания, перед поиском подформы ${}_{w_h} l_h$ в ${}_{w_l} l$, при сравнении подформ (для сравнения атрибутов), перед удалением ${}_{w_h} l_h$ из формы ${}_{w_l} l$, после удаления, перед вставкой ${}_{w_q} l_q$ на место ${}_{w_h} l_h$ в форме ${}_{w_l} l$ без ${}_{w_h} l_h$, после вставки (после операции подстановки).

Замечание 2. Выбор правила из числа доступных может быть произвольным, а может задаваться некоторыми условиями, алгоритмами или атрибутами.

Замечание 3. При пустом отношении подстановки могут выполняться только операции над атрибутами.

Замечание 4. Доступность правила определяется атрибутикой правила и аксиоматикой структуры. Доступность правила может изменяться операциями из g_r.

Основное назначение конструктивно-продукционных структур – формирование конструкций с допустимым структурой составом и связями.

Определение 12. Операция полного вывода или просто вывода ($|\models\in\Theta$) заключается в пошаговом преобразовании форм, начиная с начального нетерминала и заканчивая конструкцией, удовлетворяющей условию окончания вывода, что подразумевает циклическое выполнение операций частичного вывода. Операция двухместная ${}_{\Delta,w_l^*}l^* = |\models(\Psi,\ {}_{w_l}l)$, где ${}_{w_l}l \in U$.

Замечание 1. В результате вывода не всегда может быть сформирована конструкция удовлетворяющая условию окончания вывода. Если на *i*-1 шаге сформирована форма ${}_{w_f}f_{i-1}={}_{\nu_0}\otimes({}_{w_1}l_1,\ {}_{w_2}l_2,\ \ldots,\ {}_{w_h}l_h,\ \ldots,\ {}_{w_k}l_k)$ и для $\forall_{w_i}l_i \prec_{w_f} f_{i-1}$ нет ни одного доступного правила $s=({}_{w_i}l_i\ \ {}_{\nu_{pj}}\rightarrow\ \ {}_{w_j}l_j)$ и операция частичного вывода невыполнима, такой вывод будем считать тупиковым.

Замечание 2. Операция полного вывода не однозначная.

Определение 13. Множество конструкций, которые могут быть сформированы в результате вывода в соответствии с аксиоматикой этой структуры, с атрибутами элементов и самой конструкции определенными в результате вывода назовем множеством выводимых (правильных) конструкций $\Omega(C)$ структуры C, т.е. $\forall_{\Delta,w_l^*}l^* \in \Omega(C):\ {}_{\Delta,w_l^*}l^* = |\models(\Psi,\ {}_{w_l}l)\ \&\ {}_{w_l}l \in U$ и $\forall_{\Delta,w_l^*}l^* \notin \Omega(C):\ {}_{\Delta,w_l^*}l^* \neq |\models(\Psi,\ {}_{w_l}l)\ \&\ \ {}_{w_l}l \in U$.

Замечание. $\Omega(C)$ является многозначная функцией одного аргумента. Каждая конструкция связана с «историей» своего конструирования.

Последовательность форм вывода (список вывода) будем представлять ${}_{w_{f1}}f_1\ \ {}_{w_{v1}}\Rightarrow\ \ {}_{w_{f2}}f_2\ \ {}_{w_{v2}}\Rightarrow\ \ \cdots\ \ \ {}_{w_{vk-1}}\Rightarrow\ \ \ {}_{w_{fk}}f_k$, где ${}_{w_{f1}}f_1 \in U$, ${}_{w_{fk}}f_k \in \Omega(C)$.

Определение 14. Любая из форм вывода является сентенциальной формой $f \in F$.

Множество сентенциальных F форм включает все начальные, конечные и промежуточные формы операции вывода.

Формальная структура для проектирования данных на логическом уровне

Рассмотрим специализацию ОКПС для формального представления СД на логическом уровне.

Формальной структурой для проектирования СД на логическом уровне, назовем специализированную ОКПС C_{LD}:

$$C =< M, \Sigma, \Lambda >\ {}_S\mapsto\ C_{LD} =< M_{LD}, \Sigma_{LD}, \Lambda_{LD} >, \qquad (2)$$

где ${}_S\mapsto$ – операция специализации формальных структур (операция выполняется внешним исполнителем), $\Sigma_{\mathrm{LD}} =< \Xi_{\mathrm{LD}}, \Theta_{\mathrm{LD}}, \Phi_{\mathrm{LD}}, \{\rightarrow\} >$, $\Xi_{\mathrm{LD}} = \{"\otimes", "\oplus"\}$, $\Lambda_{LD} = \Lambda_1 \bigcup \Lambda_2$.

$\Lambda_1 = \{M_{LD} \supset (T \bigcup N), T = \{{}_{\downarrow x_i, t_i, s_i, m_i}x_i\}, N = \{{}_{s_i}\alpha_i\}\}$, где $\{x_i\}$ – множество простейших элементов данных, со значением $\downarrow x_i$, типом t_i, семантикой s_i, и необязательным порядковым номером m_i; $\{\alpha_i\}$ – множество составных элементов, с семантикой s_i.

Частичная аксиоматика Λ_2 включает аксиомы 10-11 и инструктивные дополнения 1-4.

Аксиома 10. Операция связывания терминалов "⊗" обладает свойством симметричности: $\forall_{s_i}x_i \otimes_{s_i} x_j \Rightarrow s_i \neq s_j\ \ \&\ \ (x_i \otimes x_j = x_j \otimes x_i)$.

Аксиома 11. Операция связывания терминалов "⊕" обладает свойством антисимметричности: $\forall_{s_i}x_i \oplus_{s_i} x_j \Rightarrow s_i = s_j\ \&(x_i \oplus x_j \neq x_j = x_i)$.

Дополнение 1. Операции связывания имеют атрибут $\rho_i \in P = \{\rho_1, \rho_2, \ldots \rho_n\}$, идентифицирующий операцию связывания, *n* – общее количество групп операций связывания в конкретизированной грамматике.

Дополнение 2. Порядок применения операции над атрибутами в процессе выполнения операции частичного вывода задается атрибутом $t_j \in \mathrm{T}; \mathrm{T} = \{t_0, t_1\}$, где t_0 – операция над атрибутом выполняется перед операцией подстановки, t_1 – после операции подстановки.

Дополнение 3. Операции подстановки имеют атрибут доступности ${}_d\nu_i$ – доступность операции подстановки і-го правила продукции, такая что: $\downarrow_d \nu_i \in \{true, false\}$.

Дополнение 4. Нетерминалы правил продукции имеют атрибут λ_i – количество элементов в форме.

Для интерпретации C_{LD} построим модель исполнителя в виде базовой алгоритмической структуры (БАС):

$$C_{A,LD} = \langle M_{A,LD}, V_{A,LD}, \Sigma_{A,LD}, \Lambda_{A,LD} \rangle,$$

где $M_{A,LD}$ – неоднородный носитель, $V_{A,LD}$ – множество образующих алгоритмов, базовых (элементарных) для некоторого исполнителя, $\Sigma_{A,LD}$ – сигнатура и $\Lambda_{A,LD}$ – аксиоматика. Носитель $M_{A,LD} \supset T \bigcup N \bigcup \Omega(C_{A,LD}) \bigcup W$, где $\Omega(C_{A,LD})$ все сформированные алгоритмами алгоритмической структуры конструкции; W – множество допустимых значений атрибутов. Множество базовых алгоритмов $V_{A,LD} \supset \{A_1^0|_{A_i,A_j}^{A_i \cdot A_j}, A_2^0|_{Z_1,Z_2,A_i}^{A_i}, A_3^0|_{l_i,l_j}^{l_i \otimes l_j}, A_4^0|_{l_i,l_j}^{l_i \oplus l_j}\}$ и сконструированных $\{A_5|_{f_h,f_q,f_i}^{f_j}, A_6|_{f_i,\Psi}^{f_j}, A_7|_{f_i,\Psi}^{f_j}\} \bigcup V_W \subset \Omega(C_{A,LD})$:

– алгоритм выполнения операции композиции алгоритмов $A_1^0|_{A_i,A_j}^{A_i \cdot A_j}$ ($A|_X^Y$ – алгоритм над данными из входного множества X со значениями из множества Y, A^0 – образующий алгоритм), $A_i, A_j \in \Omega(C_{A,MS})$, $A_i \cdot A_j$ – последовательное выполнение алгоритма A_j после алгоритма A_i;

– алгоритм условного выполнения $A_2^0|_{Z_1,Z_2,A_i}^{A_i}$, который заключается в выполнении алгоритма A_i при условии $Z_1 \supseteq Z_2$;

– алгоритм связывания элементов $A_3^0|_{l_i,l_j}^{l_i \otimes l_j}$, $l_i, l_j \in F^*$, согласно аксиоме 10;

– алгоритм связывания элементов $A_4^0|_{l_i,l_j}^{l_i \oplus l_j}$, $l_i, l_j \in F^*$, согласно аксиоме 11;

– алгоритм выполнения операции подстановки $A_5|_{l_h,l_q,f_i}^{f_j}$, $f_i, f_j \in F, l_h, l_q \in S$ спецификация которого задана определением 10 аксиоматики ОКПС;

– алгоритмы выполнения операций частичного и полного вывода $A_6|_{f_i,\Psi}^{f_j}$, $A_7|_{f_i,\Psi}^{f_j}$ $f_i, f_j \in F$, спецификация которых заданы определением 11 и 12 (соответственно) аксиоматики ОКПС;

– множество алгоритмов, реализующих операции над атрибутами V_W.

Аксиоматика алгоритмической структуры $\Lambda_{A,LD}$ представлена в [19].

Интерпретация формальной структуры проектирования СД на логическом уровне:

$$C_{LD} = \prec M_{LD}, \Sigma_{LD}, \Lambda_{LD} \succ \ {}_I\mapsto \ {}_I C_{LD} = \prec M_{I,LD}, \Sigma_{I,LD}, \Lambda_{I,LD} \succ,$$

где ${}_I\mapsto$ – операция интерпретации; $\Lambda_{I,LD} = \Lambda_{LD} \bigcup \Lambda_3 \bigcup \Lambda_4$;

$$\Lambda_3 = \{(A_3^0|_{l_i,l_j}^{l_i \otimes l_j} \leftarrow\!\!\lrcorner \otimes); (A_4|_{l_h,l_q,f_i}^{f_j} \leftarrow\!\!\lrcorner \Rightarrow); (A_5|_{f_i,\Psi}^{f_j} \leftarrow\!\!\lrcorner |\Rightarrow); (A_6|_{f_i,\Psi}^{f_j} \leftarrow\!\!\lrcorner |\models);$$

$$\forall \circ_i \in \Phi : \exists (A_i|_{X_i}^{Y_i} \leftarrow\!\!\lrcorner \text{"}\circ_i\text{"}),\ A_i|_{X_i}^{Y_i} \in \Omega(C_{A,MS})\}\ \forall \circ_i \in \Phi : \exists (A_i|_{X_i}^{Y_i} \leftarrow\!\!\lrcorner \text{"}\circ_i\text{"}),\ A_i|_{X_i}^{Y_i} \in \Omega(C_{A,MS})\},\ \Lambda_4 = \varnothing. \quad (3)$$

Рассмотрим одну из конкретизаций ${}_I C_{LD}$ на примере логической структуры BMP файла [1, 2]

$${}_I C_{LD} = \prec M_{I,LD}, \Sigma_{I,LD}, \Lambda_{I,LD} \succ \ {}_K\mapsto \ C_{L_{BMP}} = \prec M_{L_{BMP}}, \Sigma_{L_{BMP}}, \Lambda_{L_{BMP}} \succ, \quad (4)$$

где ${}_K\mapsto$ операция конкретизации; $M_{L_{BMP}} = M_{I,LD} \bigcup \mathbb{N}$; $\Sigma_{L_{BMP}} = \Sigma_{I,LD} \bigcup \Sigma_{BMP}$; $\Sigma_{BMP} = \{\text{"}\le\text{"},\text{"}\ge\text{"},\text{"}<\text{"},\text{"}>\text{"},\text{":="},\text{"+"}\}$; $\Lambda_{L_{BMP}} = \Lambda_{I,LD} \bigcup \Lambda_5$.

Частичная аксиоматика Λ_3:

$$\Lambda_5 = \{\ s_1 = \langle {}_{s_\alpha}\alpha \quad {}_p v_0 \rightarrow \ {}_{s_\beta}\beta_{\rho_0} \otimes_{s_\delta} \delta \rangle;\ \ {}_{t_0 \leftarrow\!\!\lrcorner t} g_1 := \langle {}_p v_0 := true \rangle;$$

$$s_2 = \langle {}_{s_\beta}\beta \quad {}_p v_1 \rightarrow \ {}_{t_a s_a m_a} a_{\rho_1} \otimes_{t_b s_b m_b} b_{\rho_1} \otimes_{t_c s_c m_c} c \rangle;\ \ {}_{t_0 \leftarrow\!\!\lrcorner t} g_2 := \langle {}_p v_1 := true \rangle;$$

$$s_3 = \langle {}_{s_{1,\delta}}\delta \quad {}_p v_2 \rightarrow \ {}_{\lambda_1 s_\chi}\chi \rangle;\ \ {}_{t_0 \leftarrow\!\!\lrcorner t} g_3 := \left\langle \lambda_1 := 0;\ {}_p v_2 := \left\{ \begin{matrix} true, z \leftarrow\!\!\lrcorner a > 256 \\ false, z \leftarrow\!\!\lrcorner a \le 256 \end{matrix} \right\} \right\rangle;$$

$$s_4 = \langle {}_{s_{2,\delta}}\delta \quad {}_p v_3 \rightarrow \ {}_{\lambda_2 s_\phi}\phi_{\rho_2} \otimes_{\lambda_3 s_\theta} \theta \rangle;\ \ {}_{t_0 \leftarrow\!\!\lrcorner t} g_4 := \left\langle \lambda_2 := 0;\ \lambda_3 := 0;\ {}_p v_3 := \left\{ \begin{matrix} true, z \leftarrow\!\!\lrcorner a \le 256 \\ false, z \leftarrow\!\!\lrcorner a > 256 \end{matrix} \right\} \right\rangle;$$

$$s_5 = \langle {}_{\lambda_1 s_\chi}\chi \quad {}_p v_4 \rightarrow \ {}_{t_e s_e m_e} e_{\rho_3} \otimes_{\lambda_1 s_\chi} \chi \rangle;\ \ g_5 = \{g_{5,1}, g_{5,2}\};$$

$$_{t_0\text{-}lt}\, g_{5,1} = \left\langle {}_{p}\nu_4 := \lambda_1 < z\text{-}lb \cdot z\text{-}lc; m_e := \lambda_1 \right\rangle;\quad _{t_1\text{-}lt}\, g_{5,2} = \left\langle \lambda_1 := \lambda_1 + 1 \right\rangle;$$

$$s_6 = \left\langle {}_{\lambda_2 s_\phi}\phi \quad {}_{p}\nu_5 \rightarrow \; {}_{t_f s_f m_f} f\, {}_{\rho_4 \otimes_{\lambda_2 s_\phi}} \phi \right\rangle;\quad g_6 = \{g_{6,1}, g_{6,2}\};$$

$$_{t_0\text{-}lt}\, g_{6,1} = \left\langle {}_{p}\nu_5 := \lambda_2 < z\text{-}lb \cdot z\text{-}lc;\; m_f := \lambda_2 \right\rangle;\quad _{t_1\text{-}lt}\, g_{6,2} = \left\langle \lambda_2 := \lambda_2 + 1 \right\rangle;$$

$$s_7 = \left\langle {}_{\lambda_3 s_\theta}\theta \quad {}_{p}\nu_6 \rightarrow {}_{t_p s_p m_p} p\, {}_{\rho_5 \otimes_{\lambda_3 s_\theta}} \theta \right\rangle;\quad g_7 = \{g_{7,1}, g_{7,2}\};$$

$$_{t_0\text{-}lt}\, g_{7,1} = \left\langle {}_{p}\nu_6 := \lambda_3 < 256;\;\; m_p := \lambda_3 \right\rangle;\quad _{t_1\text{-}lt} g_{7,2} = \left\langle \lambda_3 := \lambda_3 + 1 \right\rangle\}.$$

Будем считать, что при интерпретации $C_{L_{BMP}}$ частичная аксиоматика Λ_4 включает конструктивные дополнения 5…7.

Дополнение 5. Будем считать, что алгоритмическая структура $C_{A,LD}$ содержит следующие алгоритмы: $A_8\,|_a^b$ – алгоритм реализующий присваивания $b := a$; $A_9\,|_{a,b}^c$ – сложение $c := a + b$; $A_{10}\,|_{a,b}^c$ – сравнение $c := a \le b$; $A_{11}\,|_{a,b}^c$ – сравнение $c := a > b$.

Дополнение 6. **П**ри интерпретации (3) операций $\circ_i$ имеется $(A_8\,|_a^b \text{-}|:=)$; $(A_9\,|_{a,b}^c \text{-}|+)$; $(A_{10}\,|_{a,b}^c \text{-}|\le)$; $(A_{11}\,|_{a,b}^c \text{-}|>)$.

Дополнение 7. Значения атрибутов семантики приведено в таблице.

Таблица

Атрибут	Значение	Атрибут	Значение
$s\text{-}\alpha$	BMP файл без файлового заголовка	$s\text{-}e$	цвет пикселя
$s\text{-}\beta$	информационный заголовок BMP	$s_2\text{-}\delta$	индексированный цвет
$s_1\text{-}\delta$	цвет в модели RGB	$s\text{-}\delta$	данные изображения
$s\text{-}a$	количество битов на пиксель	$s\text{-}\phi$	индекс цвета пикселей
$s\text{-}b$	ширина изображения	$s\text{-}\theta$	цвета RGB
$s\text{-}c$	высота изображения	$s\text{-}p$	цвет RGB
$s\text{-}\chi$	цвет пикселей	$s\text{-}f$	индекс цвета пикселя

На рисунке показана визуализация логической структуры BMP-файла соответствующей структуре модели (4). Данное представление выполнено средствами нотации Джексона [20] с модифицированными связями нетерминал-значение и идентификацией правил подстановки.

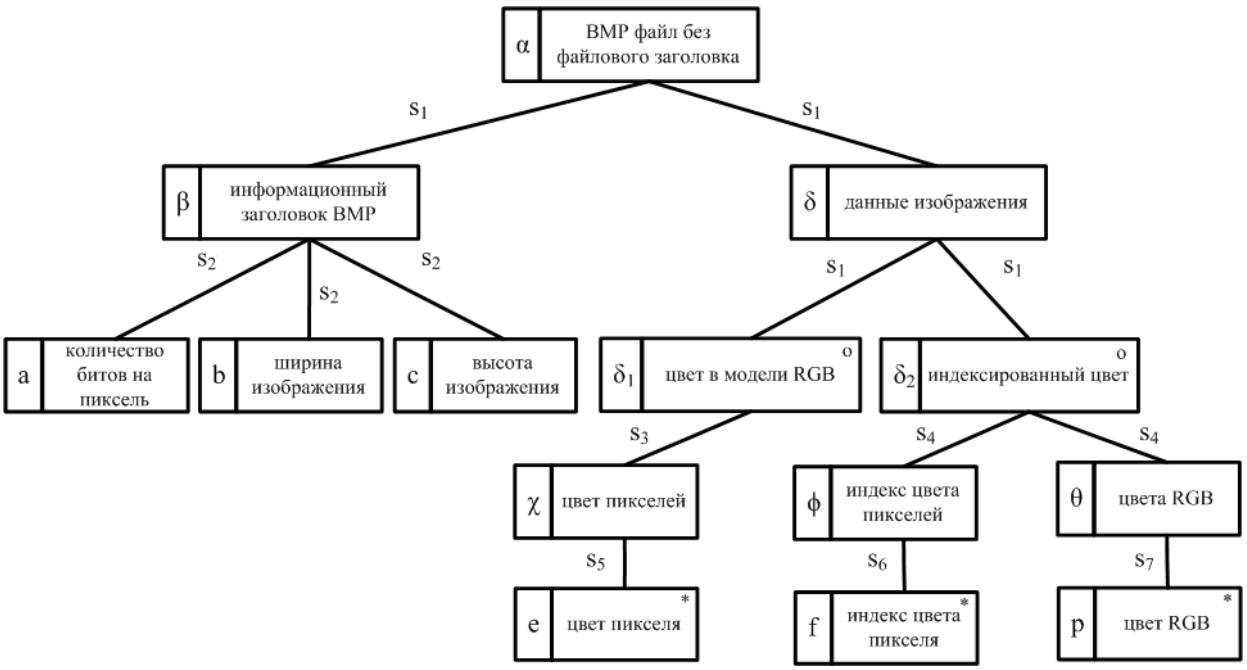

Рисунок. Логическая структура данных

Выводы

Применение средств КПС позволяет формализовать процессы и результат проектирования СД на логическом уровне. Реализация конкретизированной КПС представляет модели логически связанных конкретных элементов данных. Однако логический уровень не предполагает физической реализации данных и связей, т. е. представление данных на физическом носителе и порядка их размещения.

В последующих работах предполагается разработка КПС СД на физическом носителе, связывание между собой КПС разных уровней.

Как и любая формализация, разрабатываемые КПС, позволяют автоматизировать процесс разработки и совершенствования структур данных, повышать их временную эффективность.

1. *Шинкаренко В.И., Забула Г.В.* Повышение временной эффективности структур данных в оперативной памяти на основе адаптации // Проблеми програмування. – 2012. – № 2–3. – С. 211–218.
2. *Шинкаренко В.И., Забула Г.В.* Применение генетического алгоритма в задачах адаптации структур данных // Искусственный интеллект. – 2012. – 3. – С. 323–331.
3. *Цейтлин Г.Е.* Алгоритмические алгебры структур данных и многоуровневое проектирование программ // Программирование. – 1986. – № 3. – С. 8–16.
4. *Акуловский В.Г.* Алгебра для описания данных в композиционных схемах алгоритмов // Проблеми програмування. – 2012. – № 2–3. – С. 234-240.
5. *Андон Ф.И., Дорошенко А.Е., Цейтлин Г.Е., Яценко Е.А.* Алгеброалгоритмические модели и методы параллельного программирования. – Киев: Академпериодика, 2007. – 634 с.
6. *Фу К.* Структурные методы распознавания образов. – М.: Мир, 1977. – 318 с.
7. *Ахо А.* Индексные грамматики – расширение контекстно-свободных грамматик // Сб. «Языки и автоматы». – М.: Мир, 1975. – С. 130–165.
8. *Павлюк О.В., Савчинський Б.Д.* Ефективний синтаксичний аналіз та розпізнання структурованих зображень // Управляющие системы и машины. – 2005. – № 5. – С. 13–24.
9. *Ту Дж., Гонсалес Р.* Принципы распознавания образов. – М.: Мир, 1978. – 411 с.
10. *Prusinkiewicz P., Lindenmayer A.* The algorithmic beauty of plants. New York etc.: Springer, Cop. 1990. – XII. – 228 p.
11. *Братчиков И.Л.* Синтаксис языков программирования. – М.: Наука, 1975. – 232 с.
12. *Розенкранц Д.* Программные грамматики и классы формальных языков. // Сборник переводов по вопросам информационной теории и практики, ВИНИТИ, 1970. – № 16. – С. 117–146.
13. *Лисовик Л.П., Карнаух Т.А.* Об одном методе задания фрактальных множеств // Кибернетика и системный анализ. – 2009. – № 3. – С. 42–50.
14. *Хантер Р.* Основные концепции компиляторов. – М.; СПб. ; К.: Издательский дом "Вильямс", 2002. – 252 с.
15. *Шлезингер М.И., Главач В.* Десять лекций по статистическому и структурному распознаванию. – Киев: Наук. думка, 2004. – 546 с.
16. *Шинкаренко В.И.* Экспериментальные исследования алгоритмов в программно-аппаратных средах : монография. – Д.: Изд-во Днепропетр. нац. ун-та ж.-д. трансп. им. акад. В. Лазаряна, 2009. – 279 с.
17. *Ахо А., Ульман Дж.* Теория синтаксического анализа, перевода и компиляции. Т. 1: Синтаксический анализ. – М.: Мир, 1978. – 612 с.
18. *Пратт Т., Зелковиц М.* Языки программирования, разработка и реализация. – СПб.: Питер, 2002. – 688 с.
19. *Шинкаренко В.И., Ильман В.М., Скалозуб В.В.* Структурные модели алгоритмов в задачах прикладного программирования Часть I. Формальные алгоритмические структуры // Кибернетика и системный анализ. – 2009. – № 3. – С. 3–14.
20. *Зиглер К.* Методы проектирования программных систем: Пер. с англ. – М.: Мир, 1985. – 328 с.