УДК 510.23+510.25+510.54+512.567

В.И. ШИНКАРЕНКО, В.М. ИЛЬМАН

КОНСТРУКТИВНО-ПРОДУКЦИОННЫЕ СТРУКТУРЫ И ИХ ГРАММАТИЧЕСКИЕ ИНТЕРПРЕТАЦИИ. I. ОБОБЩЕННАЯ ФОРМАЛЬНАЯ КОНСТРУКТИВНО-ПРОДУКЦИОННАЯ СТРУКТУРА

Аннотация. Обобщены возможности различных модификаций формальных грамматик, предложен аппарат конструктивно-продукционных структур. Это позволяет формализовать процессы и результаты формирования конструкций на основе элементов с атрибутами. Рассмотрены возможности специализации, конкретизации конструктивно-продукционных структур, а также интерпертации на основе алгоритмических структур, которые моделируют исполнителя.

Ключевые слова: конструктивизм, формальная грамматика, конструктивно-продукционная структура, конструкция, атрибут, алгоритм.

ВВЕДЕНИЕ

Формальные грамматики являются одним из универсальных средств моделирования структурированных объектов. Этот инструментарий применяется в теории и практике разработки алгоритмических языков и трансляторов [1–6], алгорит-

© В.И. Шинкаренко, В.М. Ильман, 2014

мов [7–9], автоматов [2, 4, 5], распознания образов [10–12], формализации сложных систем [13], компьютерной графике [14, 15] и других направлениях программирования [5, 14 и др.].

Традиционно классические мультисимвольные грамматики, как частный случай исчисления Поста, были предложены в середине прошлого века для исследований в лингвистике и определялись как упорядоченная четверка [2, 4, 12]:

$$G = \langle V_T, V_N, Q, \sigma \rangle,$$

где V_T — терминальный и V_N — нетерминальный конечные алфавиты, $Q = \{(x_i \to y_i); x_i \in (V_T \cup V_N)^* \setminus \{\varepsilon\}, y_i \in (V_T \cup V_N)^*\}$ — множество подстановок (правил продукций); σ — начальный символ (аксиома). Здесь ε — пустой элемент, $(V_T \cup V_N)^*$ — свободный язык, определенный на терминальном и нетерминальном алфавитах.

Простой анализ развития грамматик показывает их гибридизацию на основе различных подходов и формализаций, что приводит к разнородности представления и отсутствию единообразия. Так, некоторые положения не указываются, а применяются на основе интуитивных представлений, в частности это касается операции конкатенации, связывающей терминалы и нетерминалы в цепочки, допустимого порядка применения операций подстановки и т.п.

Еще одна особенность — применение множества модификаций классических грамматик: матричные [8], индексные [10, 16], разнообразные графические [10–12, 15], стохастические [12], программные [1, 10, 17], атрибутивные [6], деревьев [12] и др. Кроме того, есть ряд грамматико-подобных систем, таких как R-системы [18], L-системы [14], которые по сути являются модификациями грамматик, хотя не представляются грамматическими средствами.

Вопросы целостности, конструктивизма, структурированности, унификации грамматик для проектируемых систем предлагается решать с помощью конструктивно-продукционных структур (КПС). Формальные КПС являются обобщающими аналогами соответствующих грамматик и грамматико-подобных систем.

Формальная структура определяется на конструктивных атрибутивных алфавитах и множествах операций, нагруженных атрибутами. Только классические мультисимвольные грамматики можно представить без аппарата атрибутов. Остальные модификации грамматик явно или неявно используют атрибутику. И это естественно, так как все конструируемые элементы строятся на основе элементарных составляющих, свойства которых и их возможное поведение часто существенны для конструкции и ее применения.

Предлагаемые формальные структуры являются средством задания множества конструкций. Определяющим признаком объединения конструкций во множество является особенность их строения. Под конструкциями будем понимать составные, структурированные объекты или конструируемые процессы с заданными свойствами их составляющих. Конструктивный процесс — процесс, который протекает вследствие формируемой последовательности действий или управляемой смены состояний.

В предыдущих работах авторов с помощью средств конструктивных структур решались задачи конструирования множества алгоритмов, предназначенных для решения определенных задач [9], восстановления графов [19, с. 88–118] и отдельных классов грамматик [20]. Была разработана модель исполнителя в виде алгоритмических структур [21]. В данной статье исследуются и обобщаются возможности применяемых ранее конструктивных структур, а также других известных грамматик и грамматических систем. При интерпретации специализированных конструктивных структур применяются разработанные модели исполнителя [21]. Выполнено обобщение конструкторов (КПС) структур различной природы и на ее основе систематизированы продукционные модели.

Принципиально важным и продуктивным оказалось представление формальных структур на основе алгоритмической атрибутивности операций.

Унификация представлений КПС достигается детализацией обобщенной формальной структуры посредством специализации, интерпретации, конкретизации (содержательности) и реализации.

Формирование конструкций подразумевает наличие исполнительных средств (исполнителя) различной природы естественного или искусственного происхождения: компьютеры, компьютерные сети, мобильные телефоны, автоматизированные станки, производственные линии и другие электронные и механические устройства, биохимические и биофизические процессы формирования растений и животных, оркестр (как исполнитель, а воспроизводимая музыка — как конструкция) и многие другие.

В первой части статьи приведены общие положения по составу КПС, во второй части будут описаны типичные детализации структур, основанные на известных модификациях грамматик.

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.

Определим абстрактную обобщенную грамматическую структуру.

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

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

где M — носитель структуры, Σ — сигнатура, состоящая из множеств возможных имен многоместных операций и отношений: связываний, подстановок, выводов и вспомогательных операций; Λ — конструктивная аксиоматика, которая включает множество определений, аксиом, правил, свойств, инструкций и пр. Множества M и Σ определяются аксиоматикой.

Выделим в аксиоматике Λ семантически завершенные части — подаксиоматики (частичные аксиоматики): носитель грамматической структуры, сигнатуру, операцию 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_G состоит из конструктивных элементов с набором атрибутов.

У каждого элемента может быть свой набор атрибутов. Атрибуты могут быть связаны как со статическими свойствами элементов (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 ${}_{w_i}m_i$ (идентификатор m_i с атрибутом w_i), иначе $\overline{w}_{i_j} \lrcorner m_i$ обозначает, что $\overline{w}_{i_j}$ является атрибутом идентификатора m_i. Набор атрибутов задается кортежем длины k, $w_i = \langle \overline{w}_{i_1}, \overline{w}_{i_2}, \ldots , \overline{w}_{i_k} \rangle \in W_1 \times W_2 \times \ \ldots \ \times W_k = W$. Таким образом, носителем является $M = \{ {}_{w_i}m_i \}$.

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 атрибутом 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). Далее m — идентификатор элемента, $\lrcorner m$ — элемент с идентификатором m.

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

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;
- $T \cup N \subseteq M$, $T \cap N = \varnothing$, $\varepsilon \in T$, $\varepsilon \notin N$.

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. Во множестве N должно быть выделено множество начальных нетерминалов $U \subset N$.

Как правило [2, 4, 12], 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 ${}_{w_i} \circ \in \Sigma$.

Сигнатура Σ включает множество 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: $\Xi \cap \Theta = \varnothing$; $\Xi \cap \Phi = \varnothing$; $\Theta \cap \Phi = \varnothing$, $\varepsilon \in \Phi$, $\rightarrow \notin \Xi \cup \Theta \cup \Phi$.

Аксиома 6. Обязательным атрибутом операций сигнатуры являются 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 [2, 4, 12] используется одна бинарная операция связывания (конкатенации) над элементами терминального и нетерминального алфавитов. В специализированных грамматиках используются разнообразные операции связывания: по условию [10–12, 16], многоместные [10–12, 15], графических элементов [10, 12] и др.

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

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

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

— ${}_{w_l} l = {}_{w_0} \otimes({}_{w_1} m_1, {}_{w_2} m_2, \ldots, {}_{w_k} m_k)$ для $\forall {}_{w_i} m_i \in M$;

— ${}_{w_l} l = {}_{w_j} m_j$, если $l = {}_{w_0} \otimes(\varepsilon, \ldots, \varepsilon, {}_{w_j} m_j, \varepsilon, \ldots, \varepsilon)$;

— ${}_{w_l} l = {}_{w_0} \otimes({}_{w_1} l_1, {}_{w_2} l_2, \ldots, {}_{w_k} l_k)$, 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, сконструированным 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 ${}_{w_l} l = {}_{w_0} \otimes ({}_{w_1} l_1, {}_{w_2} l_2, \ldots, {}_{w_k} l_k)$, то ${}_{w_1} l_1, {}_{w_2} l_2, \ldots, {}_{w_k} l_k$ назовем подформами формы ${}_{w_l} l$ и обозначим ${}_{w_i} l_i \prec {}_{w_l} l$.

Определение 5. Формы, в которых отсутствуют нетерминальные элементы, будем называть конструкциями и обозначать ${}_{\Delta,w}l$.

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

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

Свойства операций связывания:

- $\forall w \in W : {}_{w_l}l = {}_{w_0}\otimes(\varepsilon, \varepsilon, \ldots, \varepsilon) = \varepsilon$;
- $\forall\, {}_{w_i}l_i, {}_{w_j}l_j \in F^* : {}_{w_i}l_i = {}_{w_0}\otimes({}_{w_1}m_1, {}_{w_2}m_2, \ldots, {}_{w_q}m_q, {}_{w_h}m_h, \ldots, {}_{w_k}m_k)$,
  ${}_{w_j}l_j = {}_{w_0}\otimes({}_{w_1}m_1, {}_{w_2}m_2, \ldots, {}_{w_h}m_h, {}_{w_q}m_q, \ldots, {}_{w_k}m_k)$, ${}_{w_i}l_i \neq {}_{w_j}l_j$.

Свойство вложенности множеств: $T \subset T^* \subset F^*$ 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 [1, 8, 10–12, 14, 15, 17, 18].

Будем различать отношение подстановки и операцию подстановки.

Определение 8. Отношение подстановки — двухместное отношение с атрибутами ${}_{w_p}\to({}_{w_i}l_i, {}_{w_j}l_j)$.

Аксиома 9. Обязательный атрибут отношения подстановки — атрибут доступности.

В грамматиках отношение подстановки принято обозначать в инфиксной форме $l_i \to l_j$. Для наглядности будем придерживаться аналогичных обозначений: ${}_{w_i}l_i\ {}_{w_p}\!\!\to {}_{w_j}l_j$.

Определение 9. Пусть $s = \langle {}_{w_1}l_1\ {}_{w_2}\!\!\to {}_{w_3}l_3, {}_{w_4}l_4\ {}_{w_5}\!\!\to {}_{w_6}l_6, \ldots, {}_{w_m}l_m\ {}_{w_{m+1}}\!\!\to {}_{w_{m+2}}l_{m+2}\rangle$ — последовательность отношений подстановки или $s = \varepsilon$ и $g = \langle {}_{\nu_1}\oplus_1^{k_1}(w_{1,1}, w_{2,1}, \ldots, w_{k_1,1}), {}_{\nu_2}\oplus_2^{k_2}(w_{1,2}, w_{2,2}, \ldots, w_{k_2,2},), \ldots, {}_{\nu_n}\oplus_n^{k_n}(w_{1,n}, w_{2,n}, \ldots, w_{k_n,n})\rangle$ — 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 ${}_{w_l}l = {}_{w_0}\otimes({}_{w_1}l_1, {}_{w_2}l_2, \ldots, {}_{w_h}l_h, \ldots, {}_{w_k}l_k)$ и доступное отношение подстановки ${}_{w_p}\to({}_{w_h}l_h, {}_{w_q}l_q)$ такое, что ${}_{w_h}l_h \prec {}_{w_l}l$, тогда результатом трехместной операции подстановки ${}_{w_l^*}l^* = {}_{w_p}\!\!\Rightarrow({}_{w_h}l_h, {}_{w_q}l_q, {}_{w_l}l)$ будет форма ${}_{w_l^*}l^* = {}_{w_0}\otimes({}_{w_1}l_1, {}_{w_2}l_2, \ldots, {}_{w_q}l_q, \ldots, {}_{w_k}l_k)$, где $\Rightarrow \in \Theta$.

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

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

— выполнение на его основе операций подстановки;

— выполнение операций над атрибутами g_r в соответствующей последовательности.

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).

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

При пустом отношении подстановки могут выполняться только операции над атрибутами.

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

Основное назначение конструктивно-продукционных структур — формирование конструкций с допустимыми структурой, составом и связями.

Определение 12. Операция полного вывода или просто вывода ($||\Rightarrow \in \Theta$) 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: $_{\Delta, w_l^*} l^* = ||\Rightarrow (\Psi, {}_{w_l} l)$, где $_{w_l} l \in U$.

В результате вывода не всегда может быть сформирована конструкция, удовлетворяющая условию окончания вывода. Если на $i-1$-м шаге сформирована форма $_{w_f} f_{i-1} = {}_{w_0} \otimes ({}_{w_1} l_1, {}_{w_2} l_2, \ldots, {}_{w_h} l_h, \ldots, {}_{w_k} l_k)$ и для $\forall {}_{w_i} l_i \prec {}_{w_f} f_{i-1}$ нет ни одного доступного правила $s = ({}_{w_i} l_i \; {}_{w_{p_j}}\!\!\rightarrow {}_{w_j} l_j)$ и операция частичного вывода невыполнима, такой вывод будем считать тупиковым.

Операция полного вывода неоднозначная.

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. $\forall {}_{\Delta, w_l^*} l^* \in \Omega(C)$: $_{\Delta, w_l^*} l^* = ||\Rightarrow (\Psi, {}_{w_l} l) \,\&\, {}_{w_l} l \in U$ и $\forall {}_{\Delta, w_l^*} l^* \notin \Omega(C)$: $_{\Delta, w_l^*} l^* \neq ||\Rightarrow (\Psi, {}_{w_l} l) \& {}_{w_l} l \in U$.

Множество $\Omega(C)$ в определенном смысле является многозначной функцией одного аргумента. Каждая конструкция связана с «историей» своего конструирования.

Последовательность форм вывода (список вывода) будем представлять

$$_{w_{f_1}} f_1 \; {}_{w_{v_1}}\!\!\Rightarrow {}_{w_{f_2}} f_2 \; {}_{w_{v_2}}\!\!\Rightarrow \ldots, \; {}_{w_{v_{k-1}}}\!\!\Rightarrow {}_{w_{f_k}} f_k, \text{ где } {}_{w_{f_1}} f_1 \in U, \quad {}_{w_{f_k}} f_k \in \Omega(C).$$

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

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

ПРЕОБРАЗОВАНИЯ КОНСТРУКТИВНО-ПРОДУКЦИОННЫХ СТРУКТУР

Аксиоматики ОКПС (1) недостаточно для формирования конкретных конструкций ввиду неопределенности, связанной с носителем, атрибутикой, операциями над атрибутами, а также отсутствия конкретных правил подстановки. Для устранения этих неопределенностей необходимо дополнить аксиоматику структуры (1). Выделим четыре доопределяющих преобразования ОКПС: специализация, интерпретация, конкретизация и реализация.

Определение 15. Специализацией конструктивно-продукционной структуры назовем формирование модифицированной структуры

$$C = \langle M, \Sigma, \Lambda \rangle \; {}_S\!\!\mapsto {}_S C = \langle M_1, \Sigma_1, \Lambda_1 \rangle : \Lambda_1 = \Lambda \cup \widetilde{\Lambda},$$

при котором аксиоматикой $\widetilde{\Lambda}$ задаются атрибуты носителя, определяющие семантическую природу его элементов, конкретное конечное множество имен операций $\aleph_1 \subseteq \Sigma_1$, атрибутику операций, порядок их выполнения и ограничения на правила подстановки. Здесь S — атрибут специализации.

Так, в мультисимвольных формальных структурах терминальная составляющая носителя ограничена символами некоторого формального языка (например, программирования).

Возможна последовательность из нескольких специализаций.

Определение 16. Интерпретация КПС на основании алгоритмической структуры заключается в расширении аксиоматики: $\forall \circ \in \Sigma$ задается $(A|_X^Y \lrcorner \circ)$, т.е. операции $\circ$ присваивается значение атрибута в виде $A|_X^Y$ — алгоритма некоторой базовой алгоритмической структуры C_A [19, 21]. Таким образом, операции из сигнатуры Σ связываются с алгоритмами выполнения

$$\langle_S C = \langle M_1, \Sigma_1, \Lambda_1 \rangle, C_A = \langle M_A, V_A, \Sigma_A, \Lambda_A \rangle\rangle_I \mapsto {}_{S,I,C_A}C = \\ = \langle M_1, \Sigma_1, \Lambda_2 \rangle : \Lambda_2 = \Lambda_1 \cup \overline{\Lambda},$$

где $V_A = \{A_i^0|_{X_i}^{Y_i}\}$ — множество образующих алгоритмов базовой алгоритмической структуры (множество элементарных алгоритмов, реализуемых некоторым исполнителем), X_i, Y_i — множество определений и значений алгоритма $A_i^0|_{X_i}^{Y_i}$, $M_A = \bigcup\limits_{A_i^0 \in V_A} (X(A_i^0) \cup Y(A_i^0))$ — носитель алгоритмической структуры, Σ_A — множество операций связывания алгоритмов (в [19, 21] определены операции композиции и условного выполнения), Λ_A — аксиоматика алгоритмической структуры (типичная аксиоматика приведена в [19, 21]), $\overline{\Lambda} = \{(A_i|_{X_i}^{Y_i} \lrcorner \circ_i):$ $\forall \circ_i \in \Sigma, A_i|_{X_i}^{Y_i} \in \Omega(C_A)\}$, $i = 1 \ldots i_{\text{end}}$, где i_{end} — количество операций в Σ, $\Omega(C_A)$ — множество алгоритмов, конструируемых в C_A.

При интерпретации выполняется связывание информационной модели способа построения конструкций и модели исполнителя.

Если в ОКПС частично интерпретированы операции подстановки и вывода, то в интерпретированной структуре должны быть интерпретированы все без исключения операции.

Определение 17. Повторную интерпретацию КПС с заменой одной алгоритмической структуры на другую в интерпетированной КПС назовем переинтерпретацией.

Определение 18. Интерпретация КПС несколькими алгоритмическими структурами является множественной интерпретацией. При этом каждой операции сигнатуры ставится в соответствие несколько алгоритмов, соответствующих разным исполнителям.

Реализация КПС с множественной интерпретацией позволяет формировать одновременно несколько взаимосоответствующих конструкций различной природы, например изображение и звуковое сопровождение.

Определение 19. Конкретизация конструктивно-продукционной структуры заключается в расширении аксиоматики множеством конкретных правил продукций и задании конкретных множеств $T_1 \subset M_1$ и $N_1 \subset M_1$:

$$C = \langle M, \Sigma, \Lambda \rangle_K \mapsto {}_K C = \langle M_1, \Sigma, \Lambda_1 \rangle : \Lambda_1 = \Lambda \cup \breve{\Lambda}, \quad \breve{\Lambda} = \{p_i : \langle s_i, g_i \rangle\}.$$

Определение 20. Реализация конструктивно-продукционной структуры заключается в формировании конструкции (или конструкций) из элементов носителя конструктивно-продукционной структуры путем выполнения алгоритмов, связанных с операциями сигнатуры, и соответствует аксиоматике

$${}_{S,I,C_A,K}C = \langle M, \Sigma, \Lambda \rangle_R \mapsto \overline{\Omega}({}_{S,I,C_A,K}C),$$

где $\overline{\Omega}({}_{S,I,C_A,K}C) \subset \Omega({}_{S,I,C_A,K}C)$.

Реализация КПС возможна лишь применительно к предварительно специализированной, интерпретированной и конкретизированной структуре. Порядок специализации, интерпретации и конкретизации может быть произвольным.

Аксиома 10. Специализация, интерпретация и конкретизация ОКПС выполняется внешним исполнителем, реализация конкретной КПС — внутренним. Модель внутреннего исполнителя представлена набором алгоритмов алгоритмической структуры, которые «умеет» выполнять исполнитель. При этом базовые алгоритмы присущи исполнителю от природы, а конструируемые построены внешним исполнителем на основе базовых.

При любом уточняющем преобразовании КПС может быть доопределена расширением, изменением либо сужением носителя, сигнатуры и аксиоматики. Доопределение может включать объединение двух и более КПС (носителя, сигнатуры, аксиоматики). В результате доопределения формируется новая КПС. Частично этот вопрос рассмотрен в [20].

Если аксиоматика структуры включает аксиоматику другой структуры, то вторая является подструктурой первой. Подструктура может быть получена как полная либо частичная обратная специализация, интерпретация или конкретизация.

Решение задач, связанных с эффективным формированием и обработкой конструкций, зачастую предполагает построение эквивалентных структур.

Определение 21. Эквивалентными назовем структуры $C_1 \cong C_2 : \Omega(C_1) = \Omega(C_2)$.

НАЗНАЧЕНИЕ КОНСТРУКТИВНО-ПРОДУКЦИОННЫХ СТРУКТУР И СВЯЗАННЫЕ С НИМИ ЗАДАЧИ

Исходя из того, что КПС могут моделировать процессы и конструкции, сформированные на основании разнообразной элементной базы с наличием атрибутики элементов и конструкций, можно рассчитывать на широкое применения предложенных средств. КПС могут применяться при решении самых разнообразных задач в области машиностроения, строительства, здравоохранения, математики, программирования и других направлений науки и техники.

Применение КПС в прикладном программировании предполагает решение двух классических задач:

— формирование конструкции (или конструкций) из элементов носителя конструктивно-продукционной структуры путем выполнения алгоритмов, связанных с операциями сигнатуры и соответствующей аксиоматики $\omega \in \Omega({}_K C)$;

— проверка возможности формирования заданной конструкции средствами конкретной конструктивно-продукционной структуры, т.е. проверки принадлежности $\omega \in \Omega({}_K C)$.

Решение второй из приведенных задач, как и других задач, может основываться на двух подходах: доопределения КПС с расширением сигнатуры либо построения новых структур с носителем, включающим $\Omega({}_K C)$.

Представленные средства являются основой математико-алгоритмического конструктивизма (МАК).

К основным задачам МАК можно отнести: определение множества конструкций; определение условий формирования конструкций (без тупиков и зацикливания); определение степени сходства конструкций; анализ возможности формирования конструкции (и подобных ей) на основе конкретной КПС; поиск оптимального вывода конструкции; формирование конструкций другой природы на основе существующей (кода программ на основе текста на языке программирования, звучания музыки на основе партитуры и т.п.); анализ и оптимизация свойств конструкций и его частей.

Новое научное направление МАК включает новые положения теории множеств (определение и операции над множествами конструкций), функционального анализа (установление соответствия между конструкциями — по существу функция с конструкциями в качестве аргумента и значения). В то же время, если математический подход рассматривает множества и функции как нечто данное, для МАК существенен процесс формирования конструкций с учетом алгоритмических возможностей исполнителя.

Основные особенности предлагаемого подхода: расширяемый носитель; связь операций с алгоритмами их выполнения; применение моделей исполнителя-конструктора в виде набора его базовых алгоритмов; атрибутивность элементов операций, форм и конструкций; конгломерат математики и алгоритмики.

Отметим также проблему, связанную с предлагаемым представлением КПС. Интерпретация КПС выполняется связыванием операций сигнатуры с алгоритмами, а алгоритмы, в свою очередь, являются конструкциями. Здесь есть своеобразная рекурсия. И хотя она достаточно убедительно разорвана понятием базовых алгоритмов исполнителя, остается ряд невыясненных моментов. Речь идет о приобретении этих алгоритмов (самообучение человека или компьютера), их преобразовании и др. Решение этих проблем сложное. Можно лишь отметить, что оно может быть ключом к формализации процесса появления интеллекта человека и искусственного интеллекта.

ЗАКЛЮЧЕНИЕ

Предложенный аппарат конструктивно-продукционных структур позволяет формализовать процессы и результаты формирования конструкций различной природы, связывая элементы конструкций и учитывая свойства элементов и их агрегатов (форм).

В первой части статьи представлена ОКПС, которая является основой формирования различных конструкций. Во второй части будут рассмотрены различные модификации КПС, соответствующие известным аналогам формальных грамматик, показаны различные возможности специализации и интерпретации, которые воспроизводят возможности грамматик и предопределяют перспективные направления их развития.

СПИСОК ЛИТЕРАТУРЫ

1. Братчиков И.Л. Синтаксис языков программирования. — М.: Наука, 1975. — 232 с.
2. Ахо А., Ульман Дж. Теория синтаксического анализа, перевода и компиляции. Т. 1: Синтаксический анализ. — М.: Мир, 1978. — 612 с.
3. Ахо А. Сети Р., Ульман Дж. Компиляторы: принципы, технологии и инструменты. — М.: Издательский дом «Вильямс», 2003. — 768 с.
4. Хантер Р. Основные концепции компиляторов. — М.; СПб.; Киев: Издательский дом «Вильямс», 2002. — 252 с.
5. Гинзбург С. Математическая теория контекстно-свободных языков. — М.: Книга по требованию, 2012. — 286 с.
6. Пратт Т., Зелковиц М. Языки программирования, разработка и реализация. — СПб.: Питер, 2002. — 688 с.
7. Марков А.А., Нагорный Н.М. Теория алгорифмов. — М.: Наука, 1984. — 432 с.
8. Андон Ф.И., Дорошенко А.Е., Цейтлин Г.Е., Яценко Е.А. Алгеброалгоритмические модели и методы параллельного программирования. — Киев: Академпериодика, 2007. — 634 с.
9. Шинкаренко В.И., Ильман В.М., Кроль Г.Г. Грамматико-алгоритмические структурные модели метаалгоритмов // Математические машины и системы. — 2010. — № 1. — С. 3–16.
10. Фу К. Структурные методы распознавания образов. — М.: Мир, 1977. — 318 с.
11. Павлюк О.В., Савчинський Б.Д. Ефективний синтаксичний аналіз та розпізнання структурованих зображень // Управляющие системы и машины. — 2005. — № 5. — С. 13–24.
12. Ту Дж., Гонсалес Р. Принципы распознавания образов. — М.: Мир, 1978. — 411 с.
13. Глубочанский А.Д., Нехамкин Е.Б. Формально-языковое описание сложных динамических систем // Динамические системы. — 1983. — № 2. — С. 109–115.
14. Prusinkiewicz P., Lindenmayer A. The algorithmic beauty of plants. — New York: Springer Corp., 1990. — **12**. — 228 p.
15. Шлезингер М.И., Главач В. Десять лекций по статистическому и структурному распознаванию. — Киев: Наук. думка, 2004. — 546 с.
16. Ахо А. Индексные грамматики — расширение контекстно-свободных грамматик // Сб. «Языки и автоматы». — М.: Мир, 1975. — С. 130–165.
17. Розенкранц Д. Программные грамматики и классы формальных языков // Сб. переводов по вопросам информационной теории и практики, ВИНИТИ, 1970. — № 16. — С. 117–146.
18. Лисовик Л.П., Карнаух Т.А. Об одном методе задания фрактальных множеств // Кибернетика и системный анализ. — 2009. — № 3. — С. 42–50.
19. Ільман В.М., Скалозуб В.В., Шинкаренко В.І. Формальні структури та їх застосування. — Дніпропетровськ: Вид-во Дніпропетр. нац. ун-ту залізн. транспорту, 2009. — 205 с.
20. Ільман В.М., Шинкаренко В.І. Структурний підхід до проблеми відтворення граматик // Проблеми програмування. — 2007. — № 1. — С. 5–16.
21. Шинкаренко В.И., Ильман В.М., Скалозуб В.В. Структурные модели алгоритмов в задачах прикладного программирования. I. Формальные алгоритмические структуры // Кибернетика и системный анализ. — 2009. — № 3. — С. 3–14.

Поступила 02.12.2013