УДК 681.3.06

П.П. Приходько

О ТЕРМАЛЬНОМ АСПЕКТЕ АВТОМАТИЗАЦИИ ПРОГРАММИРОВАНИЯ

Уточняются основные положения немонотонного синтеза программ с переприсваиваниями. Такой синтез рассматривается с использованием теоретико-модельного подхода. Для изучения программистской и декларативной семантики предлагается использовать многоосновные алгебры термов при представлении операций над данными в синтезируемых программах. Для изучения путей эффективного автоматического синтеза программ применяются сети Петри.

Введение

В исследованиях и практических разработках по автоматизации программирования приходится обращать особое внимание на его термальный аспект [1, с. 25 -28]. Ведь практическая ценность какой-либо программы определяется прежде всего ее соответствием некоторой проблемной области. Всякий практически ценный результат выполнения программы является конструктивным объектом в такой области. Из [2, с.15-22] следует, что конструктивность объекта может истолковываться как возможность его представления некоторым термом.

Рассмотрим выполнение программы как процесс построения такого терма. Будем отслеживать соответствие между составляющими указанной проблемной области и конструкциями образуемого терма. Это даст возможность исследовать механизм соотнесения программистской семантики и семантики проблемной области. Программа образуется как следствие такого соотнесения. Эффективность автоматизации программирования определяется эффективностью воспроизведения указанного механизма. Покажем возможные подходы к такому воспроизведению.

1. Основные понятия и определения

Используем определения и обозначения, упоминавшиеся или предложенные в [3]. Составляющие какой-либо предметной области конкретной программы будем представлять тройкой $(X, \varepsilon, \mathcal{X})$, где X - множество программных объектов; $\varepsilon = \{E_x : x \in X\}$ - семейство частично упорядоченных множеств значений этих объектов $(E_x, \subseteq_x)_{x \in X}$; $\mathcal{X}$ - булева алгебра множеств - наборов программных объектов.

Состояния наборов программных объектов представляют элементы множества

$$W \equiv W(X, \varepsilon, \mathcal{X}) = \cup\left\{\left(\prod \varepsilon\right)_C : C \in \mathcal{X}\right\},$$

для которых установлено отношение частичного порядка $\subseteq$: для $u, v \in W(X, \varepsilon, \mathcal{X})$

$$u \subseteq v \Leftrightarrow \left(X_u \subseteq X_v \,\&\, \forall x \in X_u \; u(x) \subseteq_x v(x)\right).$$

Это отношение индуцирует на W операции взятия точных верхней и нижней граней, а также отношения равенства и строгого порядка. Возможные состояния моделируемого объекта представляют элементы множества

$$W_{возм} \equiv W_{возм}(X, \varepsilon, \mathcal{X}) \subseteq W(X, \varepsilon, \mathcal{X}), \qquad (1)$$

удовлетворяющего следующему условию:

$$\begin{aligned} &\forall w \in W_{возм} \; \forall U \subseteq \{v : v \in W \,\&\, v \subseteq w\} \\ &\exists \cup U, \cap U \in W_{возм} (\cup U, \cap U \subseteq w). \end{aligned} \qquad (2)$$

При этом для $A \subseteq X$

$$\begin{aligned} W(A) &\equiv W(X, \varepsilon, \mathcal{X}, A) =_{Df} \\ &= \{w : w \in W(X, \varepsilon, \mathcal{X}) \,\&\, X_w \subseteq A\}, \\ W^*(A) &\equiv W^*(X, \varepsilon, \mathcal{X}, A) =_{Df} \\ &\{w : w \in W(X, \varepsilon, \mathcal{X}) \,\&\, X_w = A\}, \\ \underline{W}(A) &\equiv \underline{W}(X, \varepsilon, \mathcal{X}, A) =_{Df} \\ &\{w : w \in W(X, \varepsilon, \mathcal{X}) \,\&\, X_w \supseteq A\}, \\ W_{возм}(A) &\equiv W_{возм}(X, \varepsilon, \mathcal{X}, A) =_{Df} \\ &= W(X, \varepsilon, \mathcal{X}, A) \cap W_{возм}, \end{aligned}$$

$$W^*_{возм}(A) \equiv W^*_{возм}(X,\varepsilon,X,A) =_{Df}$$
$$= W^*(X,\varepsilon,X,A) \cap W_{возм},$$
$$\underline{W}_{возм}(A) \equiv \underline{W}_{возм}(X,\varepsilon,X,A) =_{Df}$$
$$\underline{W}(X,\varepsilon,X,A) \cap W_{возм}.$$

Средой поддержки программ (СПП), как и в [3], будем называть пятерку $(X,\varepsilon,X,\mu,W_{возм})$, где (X,ε,X) – представление составляющих предметной области описанного выше вида; $W_{возм}$ – множество возможных состояний наборов программных объектов вида (1)-(2); μ – конечное множество процедурных представлений отношений, выражающих знания о предметной области функциями вида

$$f: W^*(A_f) \to W(B_f),$$

где $A_f, B_f \in X$, а

$$\forall w', w'' \in W^*_{возм}(A_f) w' \subseteq w'' f(w') \subseteq f(w''),$$

причем для B_f установлено его единственное разбиение на наборы $B'_f, B''_f \in X$, для которых $B_f = B'_f \cup B''_f$, $B'_f \cap B''_f = \varnothing$ и

$$\forall w \in \underline{W}_{возм}(A_f) \quad \exists w|_{\overline{B''_f}} \cup f(w|_{A_f}) \in W_{возм}.$$

Будем использовать следующие обозначения. Пусть для любой СПП $(X,\varepsilon,X,\mu,W_{возм})$, функции $f \in \mu$ и состояния $w \in W(X,\varepsilon,X)$

$$F_f \equiv \begin{cases} w|_{\overline{B''_f}} \cup f(w|_{A_f}), \text{ если } w \in \underline{W}_{возм}(A_f) \\ w \text{ для всех остальных } w, \end{cases}$$

а для множества μ $\mu^* =_{Df} \bigcup \{\mu^n : n \in \mathbf{Z}_+\}$, $\mu^+ =_{Df} \bigcup \{\mu^n : n \in \mathbf{N}\}$, $\mu^0 =_{Df} \{\lambda\}$, где λ – отсутствие символа, и для цепочки символов $\boldsymbol{f}$ $\boldsymbol{f} =^{df} (f_{i,i=\overline{1,j}})$ и $|\boldsymbol{f}| =^{df} j$, а для натуральных m,n

$[\boldsymbol{f}]_m =^{df} (f_{i,i=\overline{m,j}})$, если $m \le j$, и

$[\boldsymbol{f}]^n =^{df} (f_{i,i=\overline{1,n}})$, если $n \le j$, а

$[\boldsymbol{f}]^n_m =^{df} (f_{i,i=\overline{m,n}})$, если $m \le n \le j$, и

$(\boldsymbol{f})_i =_{Df} f_i$ для натурального $i \le |\boldsymbol{f}|$, а

$r(\boldsymbol{f}) =_{Df} \{f : f = (\boldsymbol{f})_i \,\&\, i \in \mathbf{N} \,\&\, i \le |\boldsymbol{f}|\}$.

Пусть при этом для $w \in W(X,\varepsilon,X)$

$$G(\lambda, w) =_{Df} w, \qquad (3)$$

а для $\boldsymbol{f} \in \mu^*$

$$G(\boldsymbol{f}, w) =_{Df} G([\boldsymbol{f}]_2, F_{f_1} w) \qquad (4)$$

и

$$G(\boldsymbol{f}, w) =_{Df} F_{f_n} G([\boldsymbol{f}]^{n-1}, w), \qquad (5)$$

если $|\boldsymbol{f}| \le 2$, и

$$G(\boldsymbol{f}, w) =_{Df} F_f w \qquad (6)$$

для $\boldsymbol{f} \in \mu$. Для какой-либо цепочки функциональных символов $\boldsymbol{f} \in \mu^*$

$$\underline{A}_{\boldsymbol{f}} =_{Df} \bigcup \{A_{f_i} : i \in \mathbf{N} \& i \le |\boldsymbol{f}|\},$$
$$\underline{B}_{\boldsymbol{f}} =_{Df} \bigcup \{B_{f_i} : i \in \mathbf{N} \& i \le |\boldsymbol{f}|\},$$
$$B''_{\boldsymbol{f}} =_{Df} \bigcup \{B''_{f_i} : i \in \mathbf{N} \& i \le |\boldsymbol{f}|\},$$
$$B'_{\boldsymbol{f}} =_{Df} B_{\boldsymbol{f}} \setminus B''_{\boldsymbol{f}},$$
$$A_{\boldsymbol{f}} =_{Df} \bigcup \left\{A_{f_i} \setminus B_{[\boldsymbol{f}]^{i-1}} : i \in \mathbf{N} \& i \le |\boldsymbol{f}|\right\},$$

причем $\underline{A}_\lambda = A_\lambda = B_\lambda = \varnothing$. Все данные о состоянии объекта, моделируемого СПП $(X,\varepsilon,X,\mu,W_{возм})$, которые можно получить исходя из определенного состояния набора программных объектов $w \in W(X,\varepsilon,X)$, представляет выражение

$$F_\mu =_{Df} \bigcup \left\{G(\boldsymbol{f}, w)|_{\overline{B''_{\boldsymbol{f}}}} : \boldsymbol{f} \in \mu^*\right\}.$$

Выполнение любой программы, поддерживаемой в какой-либо СПП $(X,\varepsilon,X,\mu,W_{возм})$, представляется цепочкой функциональных символов-элементов μ, соответствующих действиям программы в их очередности.

Для каких-либо $\varphi, \psi \in \mu^*$ и $n \in \mathbf{N}$ при $n \le |\varphi|$

$$\psi \underline{[}_n \varphi \Leftrightarrow \Big(\psi = \big(f_{m_k}, {}_{k=\overline{1,L}}\big) \& m_1 = n \&$$
$$\& L \le |\varphi| - n + 1 \& \forall k \in \mathbf{N}: 1 < k \le L$$
$$\big(m_{k-1} < m_k \,\&\, \&\big(A_{g_k} \cap B_{g_{k-1}}\big) \setminus B''_{[\varphi]^{m_k - 1}_{m_{k-1}+1}} \neq$$
$$\neq \varnothing) \& B_{g_L} \setminus B''_{[\varphi]_{m_L+1}} \neq \varnothing\Big).$$

Результат выполнения последовательности действий, представляемых цепочкой функциональных символов $\varphi \in \mu^*$, заведомо не зависит от выполнения *n*-го действия, если не существует цепочки $\psi \in \mu^*$, для которой $\psi \underline{[}_n \varphi$ (теорема 3 в [3]).

Обозначим Σ совокупность всех цепочек из μ^*, в которых нет звеньев, представляющих действия, заведомо не влияющие на конечный результат выполнения соответствующих последовательностей действий:

$$\Sigma \underset{Df}{=} \Big\{\varphi : \varphi \in \mu^* \,\&\, \forall n \in \mathbf{N}: n < |\varphi| \exists \psi \in \mu^* \; \psi \underline{[}_n \varphi\Big\}.$$

Для каких-либо $A, B \in X$ обозначим $Z(A,B)$ совокупность всех цепочек из Σ, представляющих последовательное определение состояния набора программных объектов *B* исходя из данных о состоянии набора *A*:

$$Z(A,B) \underset{Df}{=} \{\varphi : \varphi \in \Sigma \;\&\; A_\varphi \subseteq A \& B \subseteq B'_\varphi \;\&$$
$$\& \forall n \in \mathbf{N}: n \le |\varphi| \exists k \in \mathbf{N}: \big(k \le |\varphi| \&$$
$$\& A_{f_k} \setminus B''_{[\varphi]^{k-1}} \neq \varnothing\big) \exists \psi \in \mu^*: \big(\exists m \in \mathbf{N}:$$
$$f_n = g_n\big)\big(\psi \underline{[}_k \varphi \;\&\; B'_\psi \cap B \neq \varnothing\big)\},$$

через $Z(A,B)$ обозначим совокупность цепочек из $Z(A,B)$, каждая из которых включает все функциональные символы, входящие в цепочки из $Z(A,B)$, т.е.

$$Z(A,B) \underset{Df}{=} \{\varphi : \varphi \in z(A,B) \&$$
$$\& \forall \psi \in z(A,B) r(\psi) \subseteq r(\varphi)\},$$

а $Z_1(A,B)$ – совокупность цепочек из μ^*, представляеющих первые или единственные вхождения функциональных символов в соответствующую цепочку из $Z(A,B)$:

$$Z_1(A,B) \underset{Df}{=} \{\varphi : \varphi \in \mu^* \,\&\, \psi \in Z(A,B) \&$$
$$\& r(\varphi) = r(\psi) \& |\varphi| = |r(\varphi)| \& \varphi \overset{df}{=} \big(f_j, {}_{j=\overline{1,|\psi|}}\big) \&$$
$$\& \forall j \in \mathbf{N}: j \le |\psi| \Big(\big(\exists i \in \mathbf{N}: i < j \,\&\, g_i = g_j\big) \Rightarrow$$
$$\Rightarrow f_j = \lambda\Big) \& \Big(\big(\forall i \in \mathbf{N}: i < j \;\; g_i \neq g_j\big) \Rightarrow$$
$$f_j = g_j\big)\big)\Big\}.$$

В [3] показано, что знания о проблемной области, представляемые СПП, можно использовать для построения программ, а существование непустого множества $Z_1(A,B)$ составляет необходимое условие возможности построения программы определения значений параметров, соответствующих набору программных объектов *B*, исходя из значений входных параметров, соответствующих набору *A*.

2. Математические представления программистской семантики

Суть предыдущего раздела заключается в том, что в СПП функциональные символы представляются вместе с их интерпретациями, выражающими все знания о свойствах соответствующих исполнимых программных составляющих.

В [3] привлекаются только важнейшие свойства, которые определяют очередность выполнения действий и соотнесение с управляющими операторами. Соответственно программистская семантика представлена лишь функциональностью отношений и наличием “затираемых” или “дописываемых” значений. Соответственно, абстрагируясь от второстепенных обстоятельств, были получены ответы на основные вопросы. Сохраняя приемлемый уровень абстрагирования, используем адекватные формализмы для представления программистской семантики в ее термальном аспекте. Отработанным и удобным средством для формализованного представления преобразований данных в программах признаны

многоосновные алгебры [4].

Как и в [4], под семейством α объектов-элементов какого-либо класса ❀ будем понимать функциональное отношение между элементами некоторого множества S и объектами-элементами ❀ При этом α называется S-индексированным семейством:

$$\alpha = (\alpha_s, s \in S).$$

Под сигнатурой понимают пару (S, Ω), где S – множество имен основ; Ω – $S^* \times S$ -индексированное семейство множеств знаков операций. Тогда многоосновной алгеброй сигнатуры $\Sigma = (S, \Omega)$ можно назвать тройку (M, Σ, r), где $M \overset{df}{=} (M_\alpha, \alpha \in S)$ – семейство множеств, называемое носителем алгебры; $r = \left(\alpha_{us}: \Omega_{us} \to M_s^{M_u}, (u,s) \in S^* \times S\right)$ – семейство функций, а $M_u \overset{df}{=} \prod\left(M_{u_i}, i = \overline{1,|u|}\right)$.

Сигнатура в значительной степени определяет важные свойства многоосновной алгебры. Для каждой заданной сигнатуры $\Sigma = (S, \Omega)$ определяются алгебра базисных термов W_Σ с этой же сигнатурой Σ, в которой для каждой основы $s \in S$

$$\Omega_s \equiv \Omega_{\lambda s} \subset \left(M\left(W_\Sigma\right)\right)_s,$$

а для кортежа основ $u \in S^*$ и функционального символа $\omega \in \Omega_{us}$, кортежа базисных термов $t \in \left(M\left(W_\Sigma\right)\right)_u$

$$'\omega(t)' \in \left(M\left(W_\Sigma\right)\right)_s,$$

$$\left(\alpha_{us}(\omega)\right)(t) = '\omega(t)'.$$

Если задано еще и S-индексированное семейство множеств переменных X, то определяется алгебра термов с переменными $W_\Sigma(X)$ также с сигнатурой Σ и для каждой основы $s \in S$

$$X_s \cup \Omega_s \subset \left(M\left(W_\Sigma(X)\right)\right),$$

а для кортежа основ $u \in S^*$ и функционального символа $\omega \in \Omega_{us}$, кортежа термов с переменными $t \in \left(M\left(W_\Sigma(X)\right)\right)_u$

$$'\omega(t)' \in \left(M\left(W_\Sigma(X)\right)\right)_s,$$

$$\left(\alpha_{us}(\omega)\right)(t) = '\omega(t)'.$$

Наличие функций оценки, сопоставляющих переменным их значения, позволяет определить интерпретации для таких термов, “интерпретированные” алгебры термов, устанавливающие для термов с переменными значения базисные термы и элементы множеств семейства-носителя. Для какой-либо определенной сигнатуры $\Sigma = (S, \Omega)$ и S-индексированного семейства множеств X к множеству $I_W(\Sigma, X)$ относится всякая многоосновная алгебра (M, Σ, r), для которой

$$M = M\left(W_\Sigma(X)\right),$$

$$r \overset{df}{=} \left(\alpha_{us}: \Omega_{us} \to M_s^{M_u}, (u,s) \in S^* \times S\right)$$

и существует семейство функций

$$r = \left(r_s: \left(M\left(W_\Sigma(X)\right)\right)_s \to \left(M\left(W_\Sigma\right)\right)_s, s \in S\right),$$

для которых $r_s(X_s) \subseteq \left(M\left(W_\Sigma\right)\right)_s$ и $r_s(t) = t$ для $t \in \left(M\left(W_\Sigma\right)\right)_s$, а для $u \in S^*$, функционального символа $\omega \in \Omega_{us}$ и кортежа $t \in \left(M\left(W_\Sigma\right)\right)_u$

$$\alpha_{us}(\omega)(t) = r_s(\omega(t)) =$$
$$= (\alpha_{us}(W_\Sigma))(\omega)\left(r_{u_1}(t_1), \ldots, r_{u_{|u|}}\left(t_{|u|}\right)\right).$$

Для определенной сигнатуры $\Sigma = (S, \Omega)$ и определенной многоосновной алгебры (M, Σ, r), где

$$r = \left(\alpha_{us}: \Omega_{us} \to M_s^{M_u}, (u,s) \in S^* \times S\right),$$

интерпретацию термов реализует многоосновная алгебра $i(M, \Sigma, r) = \left(M^i, \Sigma, r^i\right)$, где

$$r^i = \left(\alpha_{us}^i: \Omega_{us} \to M_s^{iM_u^i}, (u,s) \in S^* \times S\right),$$

для $s \in S$

$$M_s \cup \left(M\left(W_\Sigma\right)\right)_s \subseteq M_s^i$$

и существует S-индексированное семейство

функций r^i, для которых $r^i : M_s^i \to M_s$ и для $t \in M_s$

$$r_s^i(t) = t;$$
$$r_s^i(t) = \alpha_s(t) \equiv \alpha_{\lambda s}(t),$$

если $t \in \Omega_s$, для $\boldsymbol{u} \in S^*$, функционального символа $\omega \in \Omega_{\boldsymbol{u}s}$ и набора термов $\boldsymbol{t} \in M_{\boldsymbol{u}}^i$

$$'\omega(\boldsymbol{t})' \in M_s^i,$$
$$(\omega(\boldsymbol{t}))^{(M,\Sigma,r)} \underset{Df}{=} \alpha_{\boldsymbol{u}s}^i(\omega)(\boldsymbol{t}) = r_s^i(\omega(\boldsymbol{t})) =$$
$$= \alpha_{\boldsymbol{u}s}(\omega)\left(r_{u_1}^i(t_1),\ldots,r_{u_{|\boldsymbol{u}|}}^i\left(t_{|\boldsymbol{u}|}\right)\right).$$

Для какой-либо определенной сигнатуры $\Sigma = (S, \Omega)$, многоосновной алгебры (M, Σ, r) с этой сигнатурой и S- индексированного семейства множеств переменных X к множеству $I(M, \Sigma, r, X)$ относится всякая многоосновная алгебра $(\underline{M}, \Sigma, \underline{r})$, где

$$\underline{r} = \left(\underline{\alpha}_{\boldsymbol{u}s} : \Omega_{\boldsymbol{u}s} \to M_s^{\underline{M}_{\boldsymbol{u}}}, (\boldsymbol{u}, s) \in S^* \times S\right),$$

такая, что существует S-индексированное смейство множеств $\underline{r}$ такое, что для $s \in S$

$$M_s^i \cup \left(M\left(W_\Sigma\right)\right)_s \subseteq \underline{M}_s,$$
$$\underline{r}_s\big|_{M_s^i} \equiv r_s^i,$$
$$\underline{r}_s\left(X_s\right) \subseteq M_s,$$

а для $\boldsymbol{u} \in S^*$, функционального символа $\omega \in \Omega_{\boldsymbol{u}s}$ и кортежа термов $\boldsymbol{t} \in \underline{M}_{\boldsymbol{u}}$

$$'\omega(\boldsymbol{t})' \in \underline{M}_s,$$
$$\underline{\alpha}_{\boldsymbol{u}s}^i(\omega)(\boldsymbol{t}) = \underline{r}_s^i(\omega(\boldsymbol{t})) =$$
$$= \alpha_{\boldsymbol{u}s}(\omega)\left(\underline{r}_{u_1}^i(t_1),\ldots,\underline{r}_{u_{|\boldsymbol{u}|}}^i\left(t_{|\boldsymbol{u}|}\right)\right).$$

Введение логического типа позволяет устанавливать для термов общие условия. Всякую сигнатуру $\Sigma = (S, \Omega)$ можно расширить “логическим” основанием и перейти к сигнатуре $\Sigma_{лог} = (S_{лог}, \Omega_{лог})$, где

$$S_{лог} = S \cup \{'лог'\},$$
$$\Omega_{«»} = \{истина, фальш\},$$
$$\Omega_{лог\,лог} = \{'\neg', '\&', '\vee'\},$$
$$\Omega_{лог^2\,лог} = \{'=', '\&', '\vee', '\to'\},$$

$'='\in \Omega_{s^2\,лог}$ для любого $s \in S_{лог}$ и для какого-либо натурального $n > 2$

$$\Omega_{лог^n,лог} = \{'\&', '\vee'\}.$$

Кроме того, можно условиться реализовывать “логические” функциональные символы лишь соответствующим образом. Иными словами, если с какой-либо сигнатурой $\Sigma = (S, \Omega)$ соотнести “логическую” сигнатуру $\Sigma лог = (S_{«»}, \Omega_{«»})$, то с какой-либо многоосновной алгеброй (M, Σ, r) можно соотнести “логическую” многоосновную алгебру $\left(M^{лог}, \Sigma^{лог}, r^{лог}\right)$, для которой

$$M_{лог}^{лог} = \Omega_{лог},$$

а для $s \in S$

$$M_s^{лог} = M_s$$

и

$$\forall t_1, t_2 \in M_s \left(\alpha_{s^2\,лог}('=')(t_1, t_2) \Leftrightarrow t_1 = t_2\right),$$

а

$$\forall n \in Z_+ \forall \boldsymbol{t} \in \left(M_{лог}^{лог}\right)^n \forall \omega \in \Omega_{лог^n лог}$$
$$\left(\alpha_{лог^n лог}(\omega)(\boldsymbol{t}) \Leftrightarrow \omega(\boldsymbol{t})\right).$$

Поэтому, не теряя общности, каждую сигнатуру можно считать “логической” и рассматривать только “логические” многоосновные алгебры.

Таким образом, для каждой “логической” сигнатуры $\Sigma = (S, \Omega)$ можно ввести “логическую” алгебру базисных термов $W_{лог}(\Sigma)$, для которой

$$M\left(W_{лог}(\Sigma)\right) = M\left(W_\Sigma\right),$$
$$\Sigma\left(W_{лог}(\Sigma)\right) = \Sigma$$

и существует S-индексированное семейство функций r таких, что для $s \in S$

$$r_s : \left(M\left(W_\Sigma\right)\right)_s \to \left(M\left(W_\Sigma\right)\right)_s$$

и для $\boldsymbol{u} \in S^*$, функционального символа $\omega \in \Omega_{\boldsymbol{u}s}$, набора базисных термов $\boldsymbol{t} \in \left(M\left(W_\Sigma\right)\right)_{\boldsymbol{u}}$

$$\alpha_{\boldsymbol{u}s}(W_{лог}(\Sigma))(\omega)(\boldsymbol{t}) = r_s(\omega(\boldsymbol{t})) =$$
$$= \alpha_{\boldsymbol{u}s}(W_{лог}(\Sigma))(\omega)\left(r_{u_1}(t_1),\ldots,r_{u_{|u|}}\left(t_{|\boldsymbol{u}|}\right)\right),$$

причем

$$\left(\left(s \neq 'лог' \vee \boldsymbol{t} \notin \Omega^*_{лог}\right) \& \omega \neq '='\right) \Rightarrow$$
$$\Rightarrow r_s(\omega(\boldsymbol{t})) = '\omega\left(r_{u_1}(t_1),\ldots,r_{u_{|u|}}\left(t_{|\boldsymbol{u}|}\right)\right), \qquad (7)$$

а если $\boldsymbol{t} \in \Omega^*_{лог}$ и $s = 'лог'$, то

$$r_{лог}(\omega(\boldsymbol{t})) = \omega(\boldsymbol{t}); \qquad (8)$$

для базисных термов t_1, t_2 основы s, если $t_1 \in \Omega_s$ и $t_2 \in \Omega_s$, то

$$r_{лог}('t_1 = t_2') \Leftrightarrow t_1 = t_2, \qquad (9)$$

иначе если $'t_1' = 't_2'$, то

$$r_{лог}('t_1 = t_2') = 'істина', \qquad (10)$$

а если $'t_1' \neq 't_2'$, то

$$r_{лог}('t_1 = t_2') = 't_1 = t_2' \in \left(M\left(W_\Sigma\right)\right)_{лог}. \qquad (11)$$

Для какой-либо логической сигнатуры $\Sigma = (S, \Omega)$ и S-индексированного семейства множеств переменных X можно определить алгебру термов с переменными той же сигнатуры $W_{лог}(\Sigma, X)$ такую, что для любой основы $s \in S$

$$\left(M\left(W_\Sigma(X)\right)\right)_s \subseteq \left(M\left(W_{лог}(\Sigma, X)\right)\right)_s,$$

а для семейства функций r

$$r_S : \left(M\left(W_{лог}(\Sigma, X)\right)\right)_s \to \left(M\left(W_{лог}(\Sigma, X)\right)\right)_s,$$

причем для какого-либо терма $\varphi \in \left(M\left(W_{лог}(\Sigma, X)\right)\right)_s$ и переменной $x \in X_s$

$$'\forall x{:}s\ \varphi', '\exists x{:}s\ \varphi' \in \left(M\left(W_{лог}(\Sigma, X)\right)\right),$$

а

$$r_{лог}('\forall x; s\ \varphi') = r_{лог}\left(\&\left(\varphi(x), {}_{x \in \left(M\left(W_\Sigma\right)\right)_s}\right)\right),$$

$$r_{лог}('\exists x; s\ \varphi') = r_{лог}\left(\vee\left(\varphi(x), {}_{x \in \left(M\left(W_\Sigma\right)\right)_s}\right)\right),$$

причем для каких-либо термов $t_1, t_2 \in \left(M\left(W_{лог}(\Sigma, X)\right)\right)_s$ выполняются условия (9)-(11); для $\boldsymbol{u} \in S^*$, функционального символа $\omega \in \Omega_{\boldsymbol{u}S}$, набора термов $\boldsymbol{t} \in$

$$\in \left(M\left(W_{лог}(\Sigma, X)\right)\right)_{\boldsymbol{u}}$$

$$'\omega(\boldsymbol{t})' \in \left(M\left(W_{лог}(\Sigma, X)\right)\right)_s$$

и

$$r_s(\omega(\boldsymbol{t})) = \alpha_{\boldsymbol{u}s}(W_{\cdot\cdot\cdot}(\Sigma, X))(\omega)\left(r_{u_1}(t_1),\ldots\right.$$
$$\left.\ldots, r_{u_{|u|}}\left(t_{|\boldsymbol{u}|}\right)\right)$$

причем выполняются условия (7)-(8).

Для какой-либо сигнатуры $\Sigma = (S, \Omega)$, многоосновной алгебры (M, Σ, r) и S-индексированного семейства множеств переменных X можно определить многоосновную алгебру логических интерпретаций термов

$$i_X(M, \Sigma, r) \overset{df}{=} \left(M^{i_X}, \Sigma, r^{i_X}\right),$$

где

$$r \overset{df}{=} \left(\alpha_{\boldsymbol{u}S} : \Omega_{\boldsymbol{u}S} \to M_s^{M_{\boldsymbol{u}}}, {}_{(\boldsymbol{u}, s) \in S^* \times S}\right),$$

а

$$r^{i_X} \overset{df}{=} \left(\alpha^{i_X}_{\boldsymbol{u}S} : \Omega_{\boldsymbol{u}S} \to M_s^{i_X M_{\boldsymbol{u}}^{i_X}}, {}_{(\boldsymbol{u}, s) \in S^* \times S}\right).$$

Для этой алгебры $i_X(M, \Sigma, r)$

$$\forall s \in S\ M_s^i \subseteq M_s^{i_X}$$

и существует S-индексированное семейство функций r^{i_X} такое, что для $s \in S$

$$r_s^{i_X} : M_s^{i_X} \to M_s,$$
$$r_s^{i_X}\big|_{M_s^i} = r_s^i,$$

а для какого-либо логического терма $\varphi \in \left(M\left(W_{лог}(\Sigma, X)\right)\right)_{лог}$ со свободной переменной типа s, иначе говоря, для такого терма φ, что

$$\forall a \in \left(M\left(W_\Sigma\right)\right)_s r_{лог}(W_{лог}(\Sigma, X))(\varphi[x := a]) \in$$
$$\left(M\left(W_\Sigma\right)\right)_{лог},$$

и для $x \in X_s$

$$'\forall x{:}s\ \varphi', '\exists x{:}s\ \varphi' \in M_{лог}^{i_X},$$

а

$$r_{лог}^{i_X}('\forall x : s\varphi') = '\&'\left(r_{лог}^{i_X}(\varphi[x := a]), {}_{a \in (M\ (W_\Sigma))_s}\right),$$
$$r_{лог}^{i_X}('\exists x : s\varphi') = '\vee'\left(r_{лог}^{i_X}(\varphi[x := a]), {}_{a \in (M\ (W_\Sigma))_s}\right),$$

для $\boldsymbol{u} \in S^*$, функционального символа $\omega \in \Omega_{\boldsymbol{us}}$, набора термов $\boldsymbol{t} \in M_{\boldsymbol{u}}^{i_X}$

$$'\omega(\boldsymbol{t})' \in M_{s}^{i_X}$$

и

$$(\omega(\boldsymbol{t}))^{(M,\Sigma,r)} \underset{Df}{=} \alpha_{\boldsymbol{us}}^{i_X}(\omega)(\boldsymbol{t}) = r_{s}^{i_X}(\omega(\boldsymbol{t})) =$$

$$= \alpha_{\boldsymbol{us}}(\omega)\left(r_{u_1}^{i_X}(t_1),\ldots,r_{u_{|\boldsymbol{u}|}}^{i_X}\left(t_{|\boldsymbol{u}|}\right)\right).$$

Представленные построения основываются на воззрениях, изложенных в [4]. Они дают возможность рассматривать знания о какой-либо предметной области, отражаемые в форме "логических" базисных термов с переменными, не затрагивая их конкретизаций в определенных многоосновных алгебрах. Если, как в [4], обозначить представление таких знаний (Σ,Φ), где $\Sigma = (S,\Omega)$ – сигнатура, а $\Phi \subseteq M\left(W_{лог}(\Sigma, X)\right)_{лог}$ для некоторого S-индексированного семейства множеств переменных X, то $\mathcal{M}(\Sigma,\Phi)$ можно обозначить множество всех моделей для представления (Σ,Φ):

$$\mathcal{M}(\Sigma,\Phi) \underset{Df}{=} \{(M,\Sigma,r) \in MBA : \forall \varphi \in \Phi \quad \varphi^{(M,\Sigma,r)}\},$$

где $\mathfrak{MBA}$ – класс всех многоосновных алгебр. $\mathcal{M}^*(\Sigma,\Phi)$ обозначим замыкание представления (Σ,Φ):

$$\mathcal{M}^*(\Sigma,\Phi) \underset{Df}{=} \{\varphi \in (M\ (W_{лог}(\Sigma, X)))_{лог} :$$

$$\forall (M,\Sigma,r) \in \mathcal{M}(\Sigma,\Phi) \quad \varphi^{(M,\Sigma,r)}\}.$$

Теорией в [4] называют представление, в котором множество формул совпадает с замыканием. Иначе говоря, если обозначить $\mathfrak{D}_S$ совокупность всех представлений, то

$$\mathcal{T} \underset{Df}{=} \{(\Sigma,\Phi) \in Ds : \Phi = \mathcal{M}^*(\Sigma,\Phi)\} \qquad (12)$$

обозначает совокупность всех теорий. На основании изложенного выше можно представить выражения для различных толкований семантического отношения выводимости $\vdash$. Пусть для какой-либо сигнатуры $\Sigma = (S,\Omega)$, представления (Σ,Φ), S-индексированного семейства множеств переменных X и логического терма $\varphi \in M\left(W_{лог}(\Sigma, X)\right)_{лог}$

$$(\Sigma,\Phi) \vdash \varphi \Leftrightarrow \varphi \in \mathcal{M}^*(\Sigma,\Phi),$$

а для какой-либо алгебры (M,Σ,r)

$$(M,\Sigma,r) \vdash \varphi \Leftrightarrow \varphi^{(M,\Sigma,r)}.$$

Для решения задач автоматизации программирования А.Я.Диковский и М.И.Канович в [5] предложили "теоретико-модельный подход", заключающийся в учреждении "теоретико-модельной семантики". Для теории вида (12), отражающей взаимосвязь свойств данных и операций, используемых в строящейся программе, эта семантика выражается моделью с носителем, представляющим "логическую" связь данных, которые могут обрабатываться этой программой или возникать при её выполнении. В [6] эту связь данных выражают "предложения вычислимости" вида $P \xrightarrow[f]{} Q$, соответствующие логическим формулам вида

$$\forall x: P(x) \quad Q(f(x)).$$

В [5, с.51] о причинах использования таких моделей сказано следующее: "Одно из основных достоинств теоретико-модельной семантики состоит в том, что появляется возможность конструктивного понимания ситуации, когда решение отсутствует: если задача неразрешима, то можно построить её опровержение...". Необходимо отметить: дальнейшее развитие воззрений, представленных в [5-7], связано со сложностями, возникающими при автоматизации построения циклических предложений. На это в [7] указывал С.С.Лавров. Источником сложностей является монотонность используемых логико-математических исчислений. Она вызывает необходимость привлечения дополнительных построений, таких, как "правила с забыванием" [5], "индексированные предикатные символы" [6]. Но даже отказ от монотонности, использование неклассических логик и исчислений "нелогического" типа не устраняют всех сложностей. Ведь сами исчисления возникли как формализация умозаключений. Объекты, возникающие в процессе вывода в каком-либо исчислении, не выделяются ка-

ким-либо особым образом. Само исчисление часто представляют как способ перечисления элементов множества.

Признанным средством формализованного описания организации каких-либо взаимосвязанных действий являются транзиционные системы. Они используются для исследования вопросов безопасного поведения сложных технических систем. Под транзиционной системой [8] понимают пятерку $(S, \Delta, S_0, S_{acc}, S_{rec})$, где S – множество состояний; $\Delta \subseteq S \times \Sigma \times S$ – отношение перехода; $S_0, S_{acc}, S_{rec} \subseteq S$ – множества соответственно начальных, допустимых и повторяющихся состояний.

Для формализованного представления взаимосвязи вычислений и данных, используемых при их выполнении, можно применять сети Петри [9]. Под сетью Петри понимают четверку $C = (P, T, I, O)$, где P,T – соответственно конечные множества позиций и переходов; $I: T \to k(P)$, $O: T \to k(P)$ – функции, устанавливающие для каждого перехода соответственно его входные и выходные позиции, а также определяющие их кратность.

Для какого-либо множества A

$$k(A) =_{Df} \{B : t_B : A \to \mathbf{Z}_+\}$$

представляет множество его комплектов, а каждый комплект B представляет его функция количества экземпляров t_B. Эта функция определяет кратность принадлежности элементов A комплекту B. Комплект можно истолковывать как множество, элементы которого не задаются “абсолютно однозначно” их именами, иначе говоря, когда имеется несколько экземпляров элементов с одним именем. В таком случае A представляет множество используемых имен.

Состояние системы в его представлении сетью Петри, меру наличия условий возможного выполнения переходов в этой сети представляет её текущая маркировка – функция вида

$$\mathrm{M} : P \to \mathbf{Z}_+ .$$

Значение функции маркировки для каждой позиции называют “количеством фишек” в ней. Переход в сети Петри может запускаться, если кратность каждой входной позиции этого перехода не превышает количества фишек, находящихся в этой позиции в текущий момент. В результате запуска перехода для каждой позиции из числа её фишек вычитается её кратность как входной позиции выполняемого перехода и добавляется её кратность как выходной позиции этого перехода. Таким образом сеть Петри переходит в свое следующее состояние.

Сети Петри довольно давно и широко используются для анализа и поддержки выполнения программ (см., например, [10]). Однако очевидна и ограниченность их выразительных возможностей. Она, в частности, проявляется в “ разрешительном характере ” выполнения переходов. В таких случаях при выборе перехода к выполнению приходится использовать дополнительную информацию, изначально соотнеся её с именами позиций и переходов.

3. Сети Петри для сред поддержки программ

Сети Петри – отработанное, удобное средство для формализованного представления какой-либо системы взаимообусловленных действий. Представив такую систему с помощью этого средства, можно установить для нее свойства, вытекающие из общих соображений, установленных для сетей Петри как формальной системы. Если для среды поддержки программ, определение которой приводится в разд. 1, установить соответствующую сеть Петри, то задачи автоматизации программирования можно будет соотнести с задачами достижимости для такой сети.

Пусть $(X, \varepsilon, X, \mu, W_{возм})$ – среда поддержки программ. Очевидно, функции из μ следует соотнести с переходами, а с позициями – программные объекты-элементы X. Однако необходимо учитывать особенности представления состояний выполнения взаимообусловленных действий маркировками сети. Ведь если какие-либо две функции из μ имеют общий входной объект, то при их непосредственном соотнесении с переходами и

позициями сети Петри это приводит к конфликтной ситуации, когда запуск одного перехода исключает запуск другого. Избежать конфликта можно, установив в требуемой сети Петри отдельную позицию для каждой из функций, имеющих этот общий входной объект, а также общий для них переход от исходного общего входного объекта к вновь учреждаемым позициям. Таким образом, каждый переход, соответствующий одной из этих функций, имеет возможность независимого запуска.

Определенные сложности возникают и при учете "дописываемых" значений входных объектов. Количество фишек для позиции, соответствующей такому программному объекту, истолковывается как количество "дописанных" элементов данных. Но кратность такой позиции как входной для какого-либо перехода, соответствующего функции из μ, должна совпадать с этим количеством фишек. Представленную проблему можно решить следующим образом: ограничить максимальное количество " дописываний " для каждого такого объекта; учредить в требуемой сети Петри новую позицию, соответствующую значению этого объекта, используемому при обращении к функции из μ; учредить для каждого такого объекта в требуемой сети Петри переходы, количество которых совпадает с максимальными "дописываниями", причем все эти переходы имеют общие входную и выходную позиции: входная соответствует рассматриваемому объекту, а выходной является упомянутая вновь учрежденная, кратность входной позиции учреждаемых переходов будет различной у разных переходов и охватывать все возможные " дописывания "; для всех переходов, соответствующих тем функциям из μ, для которых рассматриваемый объект с "дописываемым" значением – входной в сети, вместо него входной устанавливается вновь учрежденная позиция. Таким образом в требуемой сети Петри представляется возможность отработки значений структурированных данных.

Рассмотренные построения сетей Петри отражены в следующих трех утверждениях, проверяемых непосредственно.

Лемма 1. Для какой-либо СПП $(X, \varepsilon, X, \mu, W_{возм})$ можно построить единственную сеть Петри $C_1 \overset{df}{=} (P', T', I', O')$, для которой $P' = X$, $T' = \mu$, а для какой-либо функции $f \in \mu$ её входные и выходные объекты представляют входные и выходные позиции соответствующего перехода сети C_1:

$$A_f = \left\{x : t_{I'(f)}(x) = 1\right\},$$

$$B_f = \left\{x : t_{O'(f)}(x) = 1\right\},$$

соответственно

$$t_{I'(f)}\Big|_{\overline{A_f}} \equiv 0,$$

$$t_{O'(f)}\Big|_{\overline{B_f}} \equiv 0.$$

Лемма 2. Для какой-либо СПП $(X, \varepsilon, X, \mu, W_{возм})$ и любого натурального числа K можно построить единственную сеть Петри $C_2 \overset{df}{=} (P'', T'', I'', O'')$, для которой

$$P'' = P' \cup P_1, \; T'' = T' \cup T_1$$

и

$$P_1 = \left\{(x,1) : x \in Q_1\right\},$$

$$T_1 = \left\{(n, x, 1) : (x,1) \in P_1 \;\&\; n \in \mathbf{N} \;\&\; n \le K\right\},$$

а

$$Q_1 = \left\{x \in X : x \in B_f' \;\&\; f \in \mu\right\},$$

причем для какого-либо $t \in T'$

$$t_{I''(t)}\Big|_{P' \setminus Q_1} \equiv t_{I'(t)}\Big|_{P' \setminus Q_1},$$

$$t_{I''(t)}\Big|_{Q_1} \equiv 0,$$

$$t_{O''(t)}\Big|_{P_1'} \equiv t_{O'(t)},$$

$$t_{O''(t)}\Big|_{P_1} \equiv 0$$

и для каждого натурального $n \le K$ и всякого $x \in Q_1$ если $t_{I'(t)}(x) = 1$, то

$$t_{I''((n,x,1))}(x) = n,$$

$$t_{I''((n,x,1))}\Big|_{\overline{\{x\}}} \equiv 0,$$

$$t_{O''((n,x,1))}\big((x,1)\big) = 1,$$

$$t_{O''((n,x,1))}\Big|_{\overline{\{(x,1)\}}} = 0;$$

при этом

$$t_{I'(t)}\Big|_{\{(x,1):t_{I'(t)}(x)=1\,\&\,x\in Q_1\}} \equiv 1,$$

и

$$t_{I''(t)}\Big|_{P_1\setminus\{(x,1):t_{I'(t)}(x)=1\,\&\,x\in Q_1\}} \equiv 0,$$

а (P',T',I',O') – сеть Петри из формулировки леммы 1.

Лемма 3. Для какой-либо СПП $(X,\varepsilon,X,\mu,W_{возм})$ и какой-либо сети Петри $c_2 \overset{df}{=} (P'',T'',I'',O'')$ можно построить единственную сеть Петри $c_3 \overset{df}{=} (P''',T''',I''',O''')$, для которой

$$P''' = P''\cup P_2,$$

$$T''' = T''\cup T_2$$

и

$$P_2 = \{(p,t):\ p\in Q_2\,\&\,t\in T''\,\&\,t_{I''(t)}(p)=1\},$$

$$T_2 = \{(0,p):\ p\in Q_2\},$$

а

$$Q_2 = \{p\in P''\mid t_{I''(t_1)}(p)=t_{I''(t_2)}(p)=1\,\&\,t_1\neq t_2\,\&\,t_1\in T''\,\&\,t_2\in T''\},$$

причем для какого-либо $t\in T'$

$$t_{I'''(t)}\Big|_{Q_2} \equiv 0,$$

$$t_{I'''(t)}\Big|_{P''\setminus Q_2} \equiv t_{I''(t)}\Big|_{P''\setminus Q_2},$$

$$t_{O'''(t)}\Big|_{P''} \equiv t_{O''(t)},$$

а если же $t\in T_1$, то

$$t_{I'''(t)}\Big|_{P''} \equiv t_{I''(t)},$$

$$t_{O'''(t)}\Big|_{P''} \equiv t_{O''(t)},$$

$$t_{I'''(t)}\Big|_{P_2} \equiv 0,$$

$$t_{O'''(t)}\Big|_{P_2} \equiv 0;$$

при этом для какого-либо $p\in Q_2$

$$t_{I'''((0,p))}(p) = 1,$$

$$t_{I'''((0,p))}\Big|_{\overline{\{p\}}} \equiv 0,$$

$$t_{O'''((0,p))}\Big|_{\{(p,t):\,t\in T''\,\&\,t_{I''(t)}(p)=1\}} \equiv 1,$$

$$t_{O'''((0,p))}\Big|_{\overline{\{(p,t):\,t\in T''\,\&\,t_{I''(t)}(p)=1\}}} \equiv 0.$$

Адекватность полученной сети Петри c_3 рассматриваемым задачам представляет следующее утверждение.

Теорема 1. Пусть $(X,\varepsilon,X,\mu,W_{возм})$ – СПП, а $A,B\in X$ и для любого $w\in W^*_{возм}(A)$ найдется цепочка $\boldsymbol{f}\in\mu^*$ такая, что существует

$$F_\mu w \equiv G(\boldsymbol{f},w)\Big|_{\overline{B''_{\boldsymbol{f}}}} \in \underline{W}_{возм}(B).$$

Тогда для какой-либо сети Петри $c_3 = (P''',T''',I''',O''')$ в формулировке леммы 3 подмаркировка $\Lambda p\in B'''.1$, где $B''' \overset{df}{=} (B\setminus Q_1)\cup((B\cap Q_1)\times\{1\})$, а Λ – металямбдаабстракция [11], достижима с начальной маркировки $\Lambda p\in A.1$, причем для каждого $\boldsymbol{g}\in Z_1(A,B)$ в обобщенном дереве достижимости 9 с. 90-106] для сети $(c_3,\Lambda p\in A.1)$ существует путь от начальной вершины к вершине, соответствующей подмаркировке $\Lambda p\in B'''.1$, в котором переходы, соответствующие элементам μ, размещаются в той же последовательности, что и в $\boldsymbol{g}$.

Справедливость представленного утверждения вытекает из следствия 3 в [3 с.15].

4. Синтез программ на сетях Петри

Представленные в предыдущем разделе утверждения обосновывают подход к использованию сетей Петри для автоматизации программирования с переприсваиваниями. Из теоремы 1 следует, что необходимым условием положительного решения задачи анализа [5, с. 36] является положительное решение соответствующей задачи достижимости подмаркировки, а очередность расположения исполнимых программных предложений в тексте синтезируемой программы связана с соответствующим деревом достижимости.

Отраниченность выразительных возможностей сетей Петри проявляется, кроме прочего, и в "разрешительном характере" теоремы 1. Однако наличие определенного конечного дерева достижимости представляет механизм, используя который в любом состоянии выполнения какой-либо системы взаимообусловленных действий, представляемых СПП $\left(X, \varepsilon, \chi, \mu, W_{возм}\right)$, можно однозначно определить действие или несколько равноценных действий, которые следует попытаться выполнить для достижения поставленных целей. Кроме того, исполнимые программные предложения, представляемые в сети элементами μ, реализуют термы вполне определенной простой структуры.

Корректно поставленная задача автоматизации программирования [5, с. 36] предполагает наличие "вычислительной модели" или "модели предметной области", в которой полностью представляются все знания, достаточные для построения требуемых программ. Исходя из этого уточним мнения функциональных зависимостей, задаваемых в СПП $\left(X, \varepsilon, \chi, \mu, W_{возм}\right)$ элементами μ. Исследуя исторически сложившиеся представления о вычислительных моделях, А. Я. Диковский и М. И. Канович пришли к выводу [5, с. 37], что в вычислительных моделях представляются функциональные, условно функциональные, операторные и условно операторные зависимости. При этом под функциональной зависимостью понимают зависимость вида

$$f: W^{*}_{возм}\left(A_f\right) \to W^{*}_{возм}\left(B_f\right),$$

иначе говоря, зависимость, реализуемую собственно исполнимым программным предложением (безусловным оператором). Под условной зависимостью понимают какую-либо зависимость φ, реализуемую с использованием условного оператора, т.е. зависимостью вида

$$\underline{если}\ P\ \underline{то}\ \varphi .$$

Операторные зависимости – зависимости, реализуемые процедурами с парамерами-процедурами. Такие процедуры могут потребоваться, например, для вычисления значений определенных интегралов.

Очевидно, действие всякой функциональной зависимости, определенной в СПП $\left(X, \varepsilon, \chi, \mu, W_{возм}\right)$ элементом $f \in \mu$, можно реализовать мультиветвлением вида

ВЫБОР f
СЛУЧАЙ $X_{f(w)} \cap B_f = C'$
Реализация $w|_{\overline{B''_f}} \cup \left(f\left(w|_{A_f} \right) \right)\Big|_{C'}$
СЛУЧАЙ $X_{f(w)} \cap B_f = C''$
Реализация $w|_{\overline{B''_f}} \cup \left(f\left(w|_{A_f} \right) \right)\Big|_{C''}$
. . . .
СЛУЧАЙ $X_{f(w)} \cap B_f = C'''$
Реализация $w|_{\overline{B''_f}} \cup \left(f\left(w|_{A_f} \right) \right)\Big|_{C'''}$
ВСЁ-ВЫБОР

где $\left\{C', C'', \ldots, C'''\right\} = \left\{ B_f \cap C : C \in X \right\}$. При этом следует учитывать, что при практическом программировании "затирание" неопределенных значений B''_f требует привлечения специальных средств.

Зависимости, функциональные по Диковскому - Кановичу, – простейший случай для представления программных предложений в сети Петри. Если все переходы в такой сети соответствуют этим зависимостям, то программа получается непосредственно из

пути в дереве достижимости отбрасыванием переходов из T_1 и T_2. Сложности возникают при соотнесении переходов сети Петри с условно функциональными зависимостями. Сеть Петри позволяет определить возможную последовательность действий для достижения требуемой цели, но она не представляет средств для учета условий выполнения таких действий. Поэтому обращение к условному оператору в виде одного перехода представить невозможно. Оно задается совокупностью переходов, в которой каждый из них соответствует отдельной альтернативе. Но тогда и все последующие переходы в каждом из отдельных путей в дереве достижимости вплоть до узла, соответствующего требуемой маркировке, представляет тело условного оператора, соответствующего условию выполнения упомянутого начального перехода. Положение может спасти совпадение всех альтернативных путей начиная с какого-либо общего для них узла в дереве достижомости. В этом случае такому узлу соответствует конец условного оператора, а упомянутые несовпадающие альтернативные пути представляют тела альтернатив в этом операторе.

Итак, наличие условнофункциональных зависимостей, соответствующих переходам, имеющим в дереве достижимости общий начальный узел, может истолковываться как мультиветвление. Однако, чтобы программа, включающая такое мультиветвление, не была частичной, необходимо, чтобы указанные альтернативы образовывали полный набор. Иначе говоря, необходимо, чтобы конъюнкция соответствующих условий всегда была истинной.

В [9, с. 90-98] показано, что логическую взаимосвязь всех возможных состояний какой-либо системы взаимообусловленных действий представляет конечное дерево достижимости соответствующей сети Петри. Для представления узлов такого конечного дерева используются маркировки, в которых позиции с неограниченно большой достижимой кратностью получают новое, обобщенное значение "∞". В соответствии с теоремой 1 такое дерево можно использовать для определения очередности следования исполнимых программных предложений. Но, кроме того, используя такое дерево достижимости, для требуемой программы можно построить циклические предложения.

Если какой-либо из путей от начальной маркировки до какой-либо маркировки, включающей требуемую подмаркировку, содержит позицию, для которой в дереве достижимости есть замкнутый путь, то этому пути соответствует тело цикла-рекурсии. Итерации соответствует участок упомянутого пути до маркировки, в которой каждая из позиций по количеству фишек меньше, чем в конечной.

Как указано в [5, с.37]: "Из известных фактов теории алгоритмов следует, что если вид зависимостей не ограничить разумным образом, то алгоритмы анализа и синтеза либо невозможны, либо неизбежно крайне неэффективны. Поэтому в практически реализуемых системах синтеза программ ограничиваются зависимостями сравнительно простого вида." Исходя из представленных выше соображений рассмотрим возможности синтеза программ из модулей, реализующих зависимости, функциональные и условно функциональные по Диковскому - Кановичу. Для этого расширим язык цепочек функциональных символов, представленный в [1] и уже использованный подобным образом в [3]. Семантику учреждаемых конструкций будет отражать функция, представляющая значения для образуемых функциональных термов.

Пусть $\left(X, \varepsilon, X, \mu, W_{возм}\right)$ – СПП. Рассмотрим выразительные возможности такой среды, когда в μ представлены зависимости, функциональные и условно функциональные по Диковскому - Кановичу. Иначе говоря, пусть μ включает непустые множества $\mu_F^{''}$ и $\mu_{CF}^{''}$ такие, что

$$\forall f \in \mu_F^{''}\ f : W_{возм}^{*}\left(A_f\right) \to W_{возм}^{*}\left(B_f^{''}\right),$$
$$\forall f \in \mu_{CF}^{''}\ \forall w \in W_{возм}^{*}\left(A_f\right)$$
$$\left(X_{f(w)} = B_f^{''}\ \&\ X_{f(w)} = \varnothing\right).$$

Таким образом, для $f \in \mu_{CF}^{''}$ существует предикат

$$P_f \in \{истина, фальш\}^{W^*_{возм}(A_f)},$$
$$\forall w \in W^*_{возм}(A_f)\big(P_f(w) \Leftrightarrow B''_f \;\&$$
$$\&\neg P_f(w) \Leftrightarrow X_{f(w)} = \varnothing\big),$$

и функция $\varphi_f : W^*_{возм}(A_f, P_f) \to W^*_{возм}(B^*_f)$, где

$$W^*_{возм}(A_f, P_f) \overset{df}{=} \{w : w \in W^*_{возм}(A_f) \& P_f(w)\}.$$

Представим условно функциональные зависимости из μ''_{CF} парами вида (P, φ). Семантику языка цепочек функциональных символов-элементов μ представляет функция $G(\boldsymbol{f}, w)$ вида (3)-(6). Дополним этот язык предложениями вида $(P, \boldsymbol{f})$, где P – предикат, $\boldsymbol{f}$ – предложение языка (его императивно интерпретируемый нетерминальный символ). Пусть

$$G((P, \boldsymbol{f}), w) =$$
$$= \begin{cases} G(\boldsymbol{f}, w), если\ P(w) \& A_f \subseteq X_w, \\ w\ во\ всех\ остальных\ случаях, \end{cases}$$

где $w \in W(X, \varepsilon, X)$. Несложно ввести предложения более общего вида, представляющие мультиветвление. Пусть для предикатов $P', P'', \ldots, P'''$ и предложений $\boldsymbol{f}'$, $\boldsymbol{f}''$,…, $\boldsymbol{f}'''$

$$G((P', \boldsymbol{f}'; P'', \boldsymbol{f}''; \ldots, P''', \boldsymbol{f}'''), w) \underset{Df}{=}$$
$$\underset{Df}{=} \begin{cases} G(\boldsymbol{f}', w), если P'(w) \& A_{f'} \subseteq X_w, \\ G(\boldsymbol{f}'', w), если P''(w) \& A_{f''} \subseteq X_w, \\ \vdots \\ G(\boldsymbol{f}''', w), если P'''(w) \& A_{f'''} \subseteq X_w, \\ w\ во\ всех\ остальных\ случаях. \end{cases}$$

Очевидна справедливость следующего утверждения.

Лемма 4. Пусть $(X, \varepsilon, X, \mu, W_{возм})$ – СПП и для некоторой цепочки функциональных символов $\boldsymbol{f} \in \mu^*$ известно, что для $w \in W^*_{возм}(\underline{A}_{\boldsymbol{f}})$

$$P'(w) \Rightarrow G(\boldsymbol{f}, w) \equiv G(\boldsymbol{f}', w),$$
$$P''(w) \Rightarrow G(\boldsymbol{f}, w) \equiv G(\boldsymbol{f}'', w),$$
$$\ldots$$
$$P'''(w) \Rightarrow G(\boldsymbol{f}, w) \equiv G(\boldsymbol{f}''', w)$$

для определенных предикатов $P', P'', \ldots, P''' \in \{истина, фальш\}^{W^*_{возм}(\underline{A}_f)}$ и цепочек функциональных символов $\boldsymbol{f}', \boldsymbol{f}'', \ldots, \boldsymbol{f}''' \in \mu^*$ при

$$A_{\boldsymbol{f}'}, A_{\boldsymbol{f}''}, \ldots, A_{\boldsymbol{f}'''} \subseteq \underline{A}_{\boldsymbol{f}}.$$

Тогда

$$G(\boldsymbol{f}, w) \equiv G((P', \boldsymbol{f}'; P'', \boldsymbol{f}''; \ldots; P''', \boldsymbol{f}'''), w)$$

для $w \in W^*_{возм}(\underline{A}_{\boldsymbol{f}})$, если

$$\forall w \in W^*_{возм}(\underline{A}_f) P'(w) \& P''(w) \& \ldots \& P'''(w).$$

Также нетрудно убедиться в справедливости следующего утверждения.

Теорема 2. Пусть $(X, \varepsilon, X, \mu, W_{возм})$ – СПП. Для каких-либо $\boldsymbol{f}', \boldsymbol{f}''$,… …, $\boldsymbol{f}''', \boldsymbol{h} \in (\mu''_F \cup \mu''_{CF})^*$ и предикатов $P', P'', \ldots, P''' \in \{истина, ложь\}^{W_{\text{‰»"Ќ"}}}$

$$G((P', \boldsymbol{f}'\boldsymbol{h}; P'', \boldsymbol{f}''\boldsymbol{h}; \ldots; P''', \boldsymbol{f}'''\boldsymbol{h}), w) \equiv$$
$$\equiv G(\boldsymbol{h}, G((P', \boldsymbol{f}'; P'', \boldsymbol{f}''; \ldots; P''', \boldsymbol{f}'''), w))$$

для $w \in W_{возм}$, если

$$\forall w \in W_{возм} \quad P'(w) \& P''(w) \& \ldots \& P'''(w).$$

В [3] рассмотрен синтез циклических предложений, причины и условия их построения в синтезируемой программе. Главные условия применения циклических предложений – наличие набора программных объектов, состояние которого определяется и изменяется (возможно, многократно) в процессе выполнения требуемых действий, а также наличие набора программных объектов, в состоянии которого фиксируется конечный результат выполнения действий, связанных с циклическим изменением состояния упомянутого выше набора.

Дополним рассматриваемый язык цепочек функциональных символов предложениями вида $(\boldsymbol{f}, A)$, где $\boldsymbol{f}$ – предложение языка, A – описанный выше набор программных объектов с циклически изменяемым состоя-

нием. Пусть для $w \in W(X, \varepsilon, X)$

$$G((\boldsymbol{f}, A), w) =$$

$$= \begin{cases} G((\boldsymbol{f}, A), G(\boldsymbol{f}, w)), \text{если } w|_A \neq G(\boldsymbol{f}, w)|_A, \\ G(\boldsymbol{f}, w), \text{если } w|_A \equiv G(\boldsymbol{f}, w)|_A. \end{cases}$$

Из следствия 1 в [3 с.13] вытекает справедливость следующего утверждения.

Теорема 3. Пусть $(X, \varepsilon, X, \mu, W_{возм})$ – СПП. Для каких-либо $A, B \in X$ и какой-либо цепочки функциональных символов $\boldsymbol{f} \in (\mu_F'' \cup \mu_{CF}'')^*$ такой, что $\boldsymbol{f} \in Z_1(A, B)$, для $h \in \mu_{CF}''$, если $A_h \subseteq A_{\boldsymbol{f}} \cup B_{\boldsymbol{f}}$ и $A' \overset{df}{=} A_{\boldsymbol{f}_1} \cap B_h'' \neq \varnothing$, то

$$F_\mu(G((\boldsymbol{f}h, A'), w)) \subseteq F_\mu w.$$

Лемма 4 и теорема 2 обосновывают приведенные выше соображения о синтезе условных операторов на дереве достижимости, а теорема 3 – подход к построению циклических предложений на дереве достижимости.

Таким образом, если для СПП $(X, \varepsilon, X, \mu, W_{возм})$ функции, представляемые элементами μ, выражают функциональные или условно функциональные зависимости, то для каких-либо входного A и выходного B наборов, $A, B \in X$, можно решить задачу анализа, используя дерево достижимости соответствующей сети Петри. Положительное решение этой задачи представляется наличием в таком дереве достижимости пути от маркировки $\Lambda p \in A.1$ до подмаркировки $\Lambda p \in B.1$, в котором нет переходов $f \in \mu$, приводящих к маркировкам, где для некоторого $p \in B_f''$ $M(p) > 1$.

Выводы

Термальный аспект автоматизации программирования с переприсваиваниями рассмотрен с теоретико-модельных позиций. Такой подход позволил учесть и программистскую, и декларативную семантику синтезируемых термов, используя модели, основанные на семантике как маркировок, так и состояний. На этой концептуальной основе предложен уточненный подход к практическому автоматическому синтезу реальных программ.

1. *Приходько П.П.* Функциональный подход к концептуальному программированию // Дис. ... канд.физ.-мат. наук.– Киев,1991. – 177 с.
2. *Успенский В.А., Семенов А.Л.* Теория алгоритмов: основные открытия и приложения. – М.: Наука,1987.– 288 с.
3. *Приходько П.П.* О возможных основаниях немонотонного дедуктивного синтеза программ // Пробл. программирования. – 2003. – №4. – С. 5-23.
4. *Замулин А.В.* Категории типов данных / Новосибирск: ВЦСО АН СССР . - Препр. № 667. 1986. – 24 с.
5. *Диковский А.Я., Канович М.И.* Вычислительные модели с разделяемыми подзадачами // Изв. АН СССР. Техн. Кибернетика. – 1985. – №5. – С. 36-59.
6. *Тыугу Э.Х.* Концептуальное программирование. – М.: Наука, 1984. – 256 с.
7. *Лавров С.С.* Синтез программ // Кибернетика. – 1982.– №6 . – С.11-16.
8. Model checking / Edmund M. Clarke, Bernd-Holger Schlingloff // HANDBOOK OF AUTOMATED REASONING Edited by Alan Robinson and Andrei Voronkov © Elsevier Science Publishers B.V., 2001. – P. 1369-1522.
9. *Питерсон Дж.* Теория сетей Петри и моделирование систем: Пер. с англ. – М.: Мир, 1984. – 264 с.
10. *Shapiro R., Saint H.,* A New Approach to Optimization of Sequencing Decisions, *Annual Review in Automatic Programming,* 6, Part 5, 1970, p. 257-288.
11. *Барендрегт Х.* Ламбда-исчисление. Его синтаксис и семантика. – М. : Мир, 1985. – 606 с.

Получено 04.10.05

Об авторе
Приходько Павел Петрович
канд. физ.- мат. наук
тел (044) 5324690

Место работы автора:
Институт программных систем
НАН Украины
просп. Академика Глушкова, 40,
Киев-187,03680
тел (044) 5324690