УДК 519.68:510

С. А. Нигяян, А. В. Нигяян

К логической трактовке недетерминированных конечных автоматов

(Представлено академиком Н.У. Аракеляном 18/III 2008)

Ключевые слова: *недетерминированный, конечный, автомат, логический, язык, программирование, интерпретатор*

В работе показано, каким образом недетерминированные конечные автоматы (см. [1]) могут быть определены как логический язык программирования (определение логического языка программирования дано в [2]). Доказано, что интерпретатор системы Пролог — U_0 , не является логически полным для данного языка. Доказано также, что: существует недетерминированный конечный автомат, не имеющий оптимального стандартного представления для U_0 ; каждый детерминированный конечный автомат имеет оптимальное стандартное представление для U_0 ; каждое стандартное представление конечного автомата является оптимальным для U_0 .

1. **Недетерминированные конечные автоматы с логической точки зрения.** Зафиксируем три непересекающихся множества V , Π и Φ .

V — счетное множество переменных. $\Pi = \Pi_1 \cup \Pi_2 \cup \Pi_3$, где Π_1 — множество 1-местных предикатных символов, Π_2 — множество 2-местных предикатных символов, Π_3 — множество 3-местных предикатных символов:

$\Pi_1 = \{state, letter, word, initial_state, final_state, accept\},$

$\Pi_2 = \{nil_transition, accept_from\},$

$\Pi_3 = \{transition\}.$

$\Phi = States \cup Letters \cup \{nil, append\}$ — множество функциональных символов, где $States$ и $Letters$ — счетные множества 0-местных функциональных символов, $States \cap Letters = \emptyset$, nil — 0-местный функциональный символ, $nil \notin States \cup Letters$, $append$ — 2-местный функциональный символ.

Из элементов множеств V и Φ строятся термы. Терм nil условимся обозначать также $|$, а терм $append(t_1, t_2) = [t_1|t_2]$, где t_1, t_2 — термы. Множество всех термов, не использующих переменных, обозначим через M . Множество M называется универсумом Эрбрана. Используя предикатные символы множества Π и термы, традиционным образом строятся атомы. Традиционным образом определяется формула логики предикатов первого порядка, использующая логические операции $\neg, \&, \vee, \supset$ и кванторы $\forall$ и $\exists$ (см. [3]).

Опишем рассматриваемые нами интерпретации. Предметным множеством рассматриваемых интерпретаций будет множество M . Функциональные символы интерпретируются следующим образом: каждому 0-местному символу из Φ сопоставляется он сам. Символу $append \in \Phi$ сопоставляется отображение $M^2 \rightarrow M$, которое каждой паре $(t_1, t_2) \in M^2$ ставит в соответствие терм $append(t_1, t_2)$. Каждому k -местному ($k = 1, 2, 3$) символу из Π сопоставляется некоторое отображение: $M^k \rightarrow \{true, false\}$. Обозначим описанное множество интерпретаций через H . Отметим, что интерпретации из H могут отличаться одна от другой лишь значениями, сопоставляемыми символам множества Π .

Пусть E замкнутая формула и I интерпретация из H . Через $I(E)$ условимся обозначать значение формулы E на интерпретации I . Если $I(E) = true$, то интерпретация I называется моделью формулы E . Пусть E и E' замкнутые формулы. Будем говорить, что формула E' логически следует из формулы E , и обозначать это $E \models E'$, если любая интерпретация из H является моделью формулы $E \supset E'$.

Логическая программа P (далее просто программа) есть последовательность предложений $D_1, \dots, D_n, n > 0$. Предложение $D \in \{D_1, \dots, D_n\}$ является либо фактом B , либо правилом $B:-B_1, \dots, B_m$, которое является импликацией $B_1 \& \dots \& B_m \supset B$, где $B, B_1, \dots, B_m$ — атомы, $m > 0$. Программе P сопоставляется формула

$$\forall x_1 \dots \forall x_v (D_1 \& \dots \& D_n),$$

где $x_1, \dots, x_v$ — переменные, использованные в предложениях $D_1, \dots, D_n$, $v \geq 0$.

Запрос Q имеет вид $?-C_1, \dots, C_k$, где C_i — атом, $i = 1, \dots, k, k > 0$. Запросу Q сопоставляется формула

$$\exists y_1 \dots \exists y_s (C_1 \& \dots \& C_k),$$

где $y_1, \dots, y_s$ — переменные, использованные в атомах $C_1, \dots, C_k, s \geq 0$ (см. [3]).

Определим множество недетерминированных конечных автоматов *Automata*. Недетерминированный конечный автомат A есть пятерка $(\Xi, \Sigma, \delta, q_0, F)$, где:

Ξ — конечное множество состояний, $\Xi \subset States$, $\Xi \neq \emptyset$;

Σ — конечный алфавит входных символов, $\Sigma \subset Letters$, $\Sigma \neq \emptyset$;

$\delta : \Xi \times (\Sigma \cup \{nil\}) \rightarrow 2^\Xi$ — функция переходов, где 2^Ξ — множество всех подмножеств множества Ξ ;

q_0 — начальное состояние, $q_0 \in \Xi$;

F — множество заключительных состояний, $F \subset \Xi$.

Определим множество слов $Words_\Sigma$. Множество $Words_\Sigma$ есть наименьшее множество, удовлетворяющее следующим условиям:

1. $[\] \in Words_\Sigma$;

2. $a \in \Sigma, \omega \in Words_\Sigma \Rightarrow [a|\omega] \in Words_\Sigma$.

Пару (q, ω) , где $q \in \Xi, \omega \in Words_\Sigma$, назовем конфигурацией автомата A . Будем говорить, что из конфигурации (q, ω) непосредственно выводима конфигурация (q', ω) , где $q, q' \in \Xi, \omega \in Words_\Sigma$, и обозначать это $(q, \omega) \rightarrow (q', \omega)$, если $q' \in \delta(q, nil)$. Будем говорить, что из конфигурации $(q, [a|\omega])$ непосредственно выводима конфигурация (q', ω) , где $q, q' \in \Xi, a \in \Sigma, \omega \in Words_\Sigma$, и обозначать это $(q, [a|\omega]) \rightarrow (q', \omega)$, если $q' \in \delta(q, a)$.

Будем говорить, что из конфигурации K выводима конфигурация K' , и обозначать это $K \rightarrow\rightarrow K'$, если существует конечная последовательность конфигураций $K_0, K_1, \dots, K_r$ ($r \geq 0$) такая, что $K_0 = K, K_r = K'$, и $K_i \rightarrow K_{i+1}$, $i = 0, \dots, r - 1$.

Определим язык $L(A)$, $L(A) \subset Words_\Sigma$, распознаваемый недетерминированным конечным автоматом A :

$$L(A) = \{\omega | \omega \in Words_\Sigma \text{ и } (q_0, \omega) \rightarrow\rightarrow (q, nil), \text{ где } q \in F\}.$$

Опишем программу P_A , которую сопоставим недетерминированному конечному автомату A . Программа P_A является последовательностью, состоящей из следующих предложений:

если $a \in \Sigma$, то $letter(a) \in P_A$;

если $q \in \Xi$, то $state(q) \in P_A$;

если $q \in F$, то $final_state(q) \in P_A$;

$initial_state(q_0) \in P_A$;

если $q' \in \delta(q, a)$, где $q, q' \in \Xi, a \in \Sigma$, то $transition(q, a, q') \in P_A$;

если $q' \in \delta(q, nil)$, где $q, q' \in \Xi$, то $nil_transition(q, q') \in P_A$;

следующие предложения принадлежат P_A для каждого автомата $A \in Automata$:

$word([\])$

$word([L|W]) :- letter(L), word(W)$

$accept_from(S, []):-final_state(S)$

$accept_from(S, [L|W]):-state(S), letter(L), word(W), transition(S, L, S1), state(S1),$
 $accept_from(S1, W)$

$accept_from(S, W):-state(S), word(W), nil_transition(S, S1), state(S1),$
 $accept_from(S1, W)$

$accept(W):-word(W), initial_state(S), accept_from(S, W),$

где $L, S, S1, W \in V$.

Программа P_A называется стандартным представлением автомата A . Очевидно, что два стандартных представления автомата A могут отличаться одно от другого только порядком их предложений.

Логический язык программирования (определение см. в [2]), который соответствует недетерминированным конечным автоматам, определяется шестеркой $M, V, \Pi, \Phi, Prog, \Delta$, где множества M, V, Π, Φ определены выше, $Prog = \{P_A | A \in Automata\}$ есть множество программ, состоящее из всех стандартных представлений всех недетерминированных конечных автоматов множества $Automata$; Δ — отображение, которое по каждой программе $P_A \in Prog$, где $A \in Automata$, определяет множество соответствующих ей запросов, $\Delta(P_A) = \{?-accept(\omega) | \omega \in Words_\Sigma\}$.

Теорема 1. Пусть $A = (\Xi, \Sigma, \delta, q_0, F)$ — недетерминированный конечный автомат, P_A — его некоторое стандартное представление и $\omega \in Words_\Sigma$. Тогда:

$$\omega \in L(A) \Leftrightarrow P_A \models ?-accept(\omega).$$

Доказательство теоремы 1 опускается.

2. Недетерминированные конечные автоматы и интерпретатор системы Пролог. Пусть U_0 — интерпретатор системы Пролог (см. [4, 5]). Если интерпретатор U_0 останавливается на программе P и запросе Q (с положительным или отрицательным ответом), то будем говорить, что $U_0(P, Q)$ определено, иначе $U_0(P, Q)$ не определено.

Теорема 2. а) Существует такой недетерминированный конечный автомат A и такое $\omega \in L(A)$, что для любого стандартного представления P_A будем иметь: $U_0(P_A, ?-accept(\omega))$ не определено.

б) Существует такой недетерминированный конечный автомат A и такое $\omega \notin L(A)$, что для любого стандартного представления P_A будем иметь: $U_0(P_A, ?-accept(\omega))$ не определено.

Доказательство. (а) Рассмотрим недетерминированный конечный автомат A :

$$A = (\{q_0, q_1, q_2, q_3\}, \{a, b\}, \delta, q_0, \{q_3\}),$$

где δ определяется следующим образом:

$$\begin{aligned}
\delta(q_0, a) &= \{q_1\}, & \delta(q_0, b) &= \emptyset, & \delta(q_0, nil) &= \emptyset, \\
\delta(q_1, a) &= \emptyset, & \delta(q_1, b) &= \emptyset, & \delta(q_1, nil) &= \{q_2, q_3\}, \\
\delta(q_2, a) &= \{q_3\}, & \delta(q_2, b) &= \emptyset, & \delta(q_2, nil) &= \{q_1\}, \\
\delta(q_3, a) &= \emptyset, & \delta(q_3, b) &= \{q_1\}, & \delta(q_3, nil) &= \{q_3\}.
\end{aligned}$$

Покажем, что $|a, b, a| \in L(A)$, т.е. покажем, что из конфигурации $(q_0, |a, b, a|)$ выводима конфигурация (q_3, nil) . Действительно,

$$(q_0, |a, b, a|) \rightarrow (q_1, |b, a|) \rightarrow (q_3, |b, a|) \rightarrow (q_1, |a|) \rightarrow (q_2, |a|) \rightarrow (q_3, nil).$$

Можно показать, что для любого стандартного представления P_A автомата A $U_0(P_A, ?-accept(|a, b, a|))$ не определено.

(b) Здесь рассматривается тот же автомат A , что и в пункте (a) и $|a, b, a, a| \notin L(A)$. Можно показать, что для любого стандартного представления P_A автомата A $U_0(P_A, ?-accept(|a, b, a, a|))$ не определено. Теорема 2 доказана.

Стандартное представление $P_{A_{opt}}$ недетерминированного конечного автомата $A = (\Xi, \Sigma, \delta, q_0, F)$ назовем оптимальным стандартным представлением для U_0 , если для любого стандартного представления P_A автомата A и любого $\omega \in Words_\Sigma$ имеем: $U_0(P_A, ?-accept(\omega))$ определено $\Rightarrow U_0(P_{A_{opt}}, ?-accept(\omega))$ определено.

Теорема 3. Существует недетерминированный конечный автомат A , не имеющий оптимального стандартного представления для U_0 .

Доказательство. Рассмотрим следующий недетерминированный конечный автомат A :

$$A = (\{q_0, q_1, q_2, q_3\}, \{a\}, \delta, q_0, \{q_3\}),$$

где δ определяется следующим образом:

$$\begin{aligned}
\delta(q_0, a) &= \{q_1\}, & \delta(q_0, nil) &= \emptyset, \\
\delta(q_1, a) &= \emptyset, & \delta(q_1, nil) &= \{q_2, q_3\}, \\
\delta(q_2, a) &= \{q_3\}, & \delta(q_2, nil) &= \{q_1\}, \\
\delta(q_3, a) &= \emptyset, & \delta(q_3, nil) &= \{q_3\}.
\end{aligned}$$

Легко видеть, что $L(A) = \{|a|, |a, a|\}$. Можно показать, что для любого стандартного представления P_A автомата A имеет место только одно из трех:

a)	$U_0(P_A, ?-accept(a))$	определено,
	$U_0(P_A, ?-accept(a, a))$	не определено;
b)	$U_0(P_A, ?-accept(a))$	не определено,
	$U_0(P_A, ?-accept(a, a))$	определено;
c)	$U_0(P_A, ?-accept(a))$	не определено,
	$U_0(P_A, ?-accept(a, a))$	не определено.

Отсюда будет следовать, что автомат A не имеет оптимального стандартного представления для U_0 . Действительно, если бы оптимальное стандартное представление $P_{A_{opt}}$ существовало, то для него должно было выполняться следующее:

$$\begin{aligned} U_0(P_{A_{opt}}, ?-accept(\{a\})) & \text{ определено,} \\ U_0(P_{A_{opt}}, ?-accept(\{a, a\})) & \text{ определено,} \end{aligned}$$

чего не может быть. Теорема 3 доказана.

Пусть $A = (\Xi, \Sigma, \delta, q_0, F)$ – недетерминированный конечный автомат.

Будем говорить, что состояние $q \in \Xi$ достижимо из начального состояния q_0 , если существует такое $\omega \in Words_{\Sigma}$, что $(q_0, \omega) \rightarrow \rightarrow (q, nil)$.

Будем говорить, что автомат A имеет nil -цикл, если существует такая последовательность состояний $q, q_1, \dots, q_n, n \geq 0$, автомата A , что:

$$\begin{aligned} q_1 \in \delta(q, nil), q_2 \in \delta(q_1, nil), \dots, q_n \in \delta(q_{n-1}, nil), q \in \delta(q_n, nil), \text{ если } n > 0, \text{ и} \\ q \in \delta(q, nil), \text{ если } n = 0. \end{aligned}$$

Будем говорить, что nil -цикл автомата A достижим, если состояние q достижимо из начального состояния q_0 .

Теорема 4. Пусть $A = (\Xi, \Sigma, \delta, q_0, F)$ – недетерминированный конечный автомат без достижимых nil -циклов. Тогда для любого его стандартного представления P_A и любого $\omega \in Words_{\Sigma}$ имеем: $U_0(P_A, ?-accept(\omega))$ определено.

Доказательство теоремы 4 опускается.

3. Детерминированные конечные автоматы и интерпретатор системы Пролог. Недетерминированный конечный автомат $A = (\Xi, \Sigma, \delta, q_0, F)$ называется детерминированным конечным автоматом, если:

1. для любых $q \in \Xi$ и $a \in \Sigma \cup \{nil\}$ имеем: мощность множества $\delta(q, a) \leq 1$;
2. для любого $q \in \Xi$ имеем: $\delta(q, nil) \neq \emptyset \Rightarrow$ для любого $a \in \Sigma, \delta(q, a) = \emptyset$.

Теорема 5. а) Существуют такой детерминированный конечный автомат A , такое $\omega \in L(A)$ и такое его стандартное представление P_A , что $U_0(P_A, ?-accept(\omega))$ не определено.

б) Существуют такой детерминированный конечный автомат A и такое $\omega \notin L(A)$, что для любого стандартного представления P_A будем иметь: $U_0(P_A, ?-accept(\omega))$ не определено.

Доказательство. (а) Рассмотрим следующий детерминированный конечный автомат A :

$$A = (\{q_0, q_1, q_2\}, \{a, b\}, \delta, q_0, \{q_2\}),$$

где δ определяется следующим образом:

$$\begin{aligned} \delta(q_0, a) &= \{q_2\}, & \delta(q_0, b) &= \{q_1\}, & \delta(q_0, nil) &= \emptyset, \\ \delta(q_1, a) &= \emptyset, & \delta(q_1, b) &= \emptyset, & \delta(q_1, nil) &= \{q_1\}, \\ \delta(q_2, a) &= \emptyset, & \delta(q_2, b) &= \emptyset, & \delta(q_2, nil) &= \{q_2\}. \end{aligned}$$

Легко видеть, что $[a] \in L(A) = \{[a]\}$. Покажем, что, если $\omega = [a]$, то существует такое стандартное представление P_A автомата A , что $U_0(P_A, ?\text{-accept}(\omega))$ не определено. Пусть P_A удовлетворяет следующему условию:

правило

$$\text{accept_from}(S, W) :- \text{state}(S), \text{word}(W), \text{nil_transition}(S, S1), \text{state}(S1), \text{accept_from}(S1, W)$$

встречается раньше правила

$$\text{accept_from}(S, []):- \text{final_state}(S)$$

в стандартном представлении P_A автомата A .

Можно показать, что интерпретатор U_0 на P_A и $?\text{-accept}([a])$ функционирует бесконечно.

(b) Здесь рассматривается автомат A из доказательства пункта (a) и $[b] \notin L(A)$. Можно показать, что для любого стандартного представления P_A автомата A будем иметь: $U_0(P_A, ?\text{-accept}([b]))$ не определено. Теорема 5 доказана.

Теорема 6. *Всякий детерминированный конечный автомат A имеет оптимальное стандартное представление P_{Aopt} для U_0 , и, если $\omega \in L(A)$, то $U_0(P_{Aopt}, ?\text{-accept}(\omega))$ определено.*

Доказательство. Можно показать, что всякое стандартное представление P_A детерминированного конечного автомата A , в последовательности предложений которого правило

$$\text{accept_from}(S, []):- \text{final_state}(S)$$

встречается раньше правила

$$\text{accept_from}(S, W) :- \text{state}(S), \text{word}(W), \text{nil_transition}(S, S1), \text{state}(S1), \text{accept_from}(S1, W),$$

является оптимальным стандартным представлением для U_0 и, если $\omega \in L(A)$, то $U_0(P_{Aopt}, ?\text{-accept}(\omega))$ определено. Теорема 6 доказана.

Недетерминированный конечный автомат $A = (\Xi, \Sigma, \delta, q_0, F)$ называется конечным автоматом, если:

1. для любых $q \in \Xi$ и $a \in \Sigma$ имеем: мощность множества $\delta(q, a)$ равна 1;
2. для любого $q \in \Xi$ имеем: $\delta(q, nil) = \emptyset$.

Теорема 7. *Пусть $A = (\Xi, \Sigma, \delta, q_0, F)$ – конечный автомат. Тогда для любого его стандартного представления P_A и любого $\omega \in \text{Words}_\Sigma$, имеем: $U_0(P_A, ?\text{-accept}(\omega))$ определено.*

Справедливость теоремы 7 следует из теоремы 4.

Ереванский государственный университет

С. А. Нигиян, А. В. Нигиян

К логической трактовке недетерминированных конечных автоматов

В работе показано, каким образом недетерминированные конечные автоматы могут быть определены как логический язык программирования. Доказано, что интерпретатор системы Пролог - U_0 , не является логически полным для данного языка. Доказано также, что существует недетерминированный конечный автомат, не имеющий оптимального стандартного представления для U_0 ; каждый детерминированный конечный автомат имеет оптимальное стандартное представление для U_0 ; каждое стандартное представление конечного автомата является оптимальным для U_0 .

Ս. Ա. Նիգիյան, Ա. Վ. Նիգիյան

Ոչ դեթերմինացված վերջավոր ավտոմատների փրամաբանական սեկնարանության մասին

Աշխատանքում ցույց է տրված, թե ինչպես ոչ դեթերմինացված վերջավոր ավտոմատները կարող են սահմանվել որպես ծրագրավորման փրամաբանական լեզու: Ապացուցված է, որ Պրոլոգ համակարգի U_0 ինտերպրետատորը փրամաբանորեն լրիվ չէ այդ լեզվի համար: Ապացուցված է նաև, որ գոյություն ունի ոչ դեթերմինացված վերջավոր ավտոմատ, որը չունի օպտիմալ ստանդարտ ներկայացում U_0 -ի համար, իսկ ամեն մի դեթերմինացված վերջավոր ավտոմատ ունի օպտիմալ ստանդարտ ներկայացում U_0 -ի համար, եւ վերջավոր ավտոմատի յուրաքանչյուր ստանդարտ ներկայացում օպտիմալ է U_0 -ի համար:

S. A. Nigiyan, A. V. Nigiyan

On Logical Interpretation of Nondeterministic Finite Automata

In this paper it is shown how the nondeterministic finite automata can be defined as a logic programming language. It is proved that the interpreter of PROLOG system U_0 is not logically complete for this language. It is also proved that there exists such a nondeterministic finite automaton which has not the optimal standard representation for U_0 , but every deterministic finite automaton has optimal standard representation for U_0 , and every standard representation of finite automaton is optimal for U_0 .

Литература

1. *Aho A.V., Sethi R., Ullman J.D. Compilers Principles, Techniques, and Tools. Addison-Wesley Pub. Comp. Inc. 1985. (рус. пер. Ахо А., Сети Р., Ульман Дж. Компиляторы: принципы, технологии, инструменты. М. Изд. дом "Вильямс". 2001. 768 с.)*
2. *Нигиян С.А., Хачоян Л.О., Нигиян А.В. - ДНАН Армении. 2007. Т. 107. N1. С. 20-25.*
3. *Lloyd J.W. Foundations of Logic Programming. Springer-Verlag. 1984. 120 p.*
4. *Нигиян С.А. - Программирование. 1994. N2. С. 64-73 (анг. пер. Nigiyan S.A. - Programming and Computer Software. 1994. V. 20. N2. P. 69-75).*
5. *Bratko I. Prolog Programming for Artificial Intelligence. Addison-Wesley Pub. Comp., Inc. 1986. (рус. пер. Братко И. Программирование на языке Пролог для искусственного интеллекта. М. Мир.1990. 560 с.)*