



ОБ ОДНОЙ ПОЛУГРУППОВОЙ МОДЕЛИ ПОСЛЕДОВАТЕЛЬНЫХ ПРОГРАММ, ОПРЕДЕЛЯЕМОЙ ПРИ ПОМОЩИ ДВУХЛЕНТОЧНЫХ АВТОМАТОВ

В. В. ПОДЫМОВ
В. А. ЗАХАРОВ

*Московский
государственный
университет
им. М.В. Ломоносова*

*e-mail:
zakh@cs.msu.su*

В настоящей работе рассматриваются полугруппы программных операторов, подчиняющихся тождествам коммутативности и поглощения. Полугруппы такого вида возникают при решении задач верификации пропозициональных моделей последовательных программ, построенных на основании результатов анализа потоков данных в императивных программах. Установлены необходимые и достаточные условия, при которых отношения равенства операторных цепочек в рассматриваемых полугруппах может быть описано посредством детерминированных двухленточных односторонних автоматов. На основании этих условий получены оценки сложности решения проблемы эквивалентности программ, семантика операторов которых определяется тождествами коммутативности и поглощения.

Ключевые слова: программа, эквивалентность, полугруппа операторов, перестановочные операторы, поглощаемые операторы, двухленточный автомат.

1. Введение. Проблема эквивалентности программ состоит в том, чтобы для произвольной пары программ выяснить, имеют ли они одинаковое поведение. Обычно поведение вычислительной программы полностью определяется функцией преобразования входных данных программы в выходные данные. В этом случае программы считаются эквивалентными, если они вычисляют одну и ту же функцию. Проблема функциональной эквивалентности программ возникает во многих задачах системного и прикладного программирования. Однако в универсальных системах программирования эта проблема алгоритмически неразрешима. Поэтому для получения эффективно проверяемых достаточных условий функциональной эквивалентности программ приходится обращаться к абстрактным моделям программ, в которых разрешимое отношение эквивалентности программ аппроксимирует отношение функциональной эквивалентности.

Одной из таких моделей является пропозициональная модель программ (см. [1]); семантика программ в этой модели определяется на основе динамического шкала Крипке [2] или полугрупп [3, 4]. Полугрупповые модели программ интересны тем, что в терминах полугрупповых тождеств удобно описывать алгебраические свойства программных операторов, которые необходимо учитывать при проверке эквивалентности программ. Если в процессе работы любой программы ни одно состояние данных не вычисляется дважды (все вычисления являются строго прогрессивными), то семантика таких программ может быть описана посредством детерминированных двухленточных автоматов [3]. В серии работ [1, 3, 5-7] показано, что этот способ описания семантик программ удобен для построения алгоритмов проверки эквивалентности программ. В связи с этим возникает следующая задача: для полугруппы программных операторов с заданным множеством определяющих соотношений (тождеств) выяснить, существует ли двухленточный детерминированный автомат, проверяющий равенство слов в этой полугруппе. В настоящей работе проведено полное исследование этой задачи для полугрупп, определяемых тождествами двух видов: $ab = ba$ (закон коммутативности) и $ab = b$ (закон поглощения).



2. Полугруппы операторов. Мы будем рассматривать полугруппы $F = \langle S, \lambda, \circ \rangle$ с множеством образующих $A = \{a_1, \dots, a_r\}$. Элементы этого множества можно рассматривать как программные операторы. Нейтральному элементу полугруппы λ отвечает пустой оператор, а полугрупповая операция \circ соответствует последовательной композиции операторов. Конечную последовательность операторов будем называть *A-цепочкой*. Множество всех *A-цепочек* обозначим A^* . Для каждой *A-цепочки* $h = a_1 a_2 \dots a_k$ запись $|h|$ будет обозначать длину k цепочки h , запись $h[i]$ будет обозначать i -ый член a_i этой цепочки, а запись $h|_{zi}$ будет обозначать суффикс $a_i a_{i+1} \dots a_k$ этой цепочки, начинающийся i -ым ее членом. Значением указанной *A-цепочки* является элемент полугруппы $[h]_F = a_1 \circ a_2 \circ \dots \circ a_k$. В полугруппе F определим бинарное отношение $<$: будем полагать, что $[h_1]_F < [h_2]_F$ в том и только том случае, когда существует *A-цепочка* h , для которой верно равенство $[h_1 h]_F = [h_2]_F$. Если отношение $<$ является отношением частичного порядка, то такую полугруппу будем называть упорядоченной.

Наиболее распространенным способом задания полугрупп служат определяющие соотношения (тождества). В свете операторной интерпретации полугрупп мы будем рассматривать системы тождеств специального вида, отражающие некоторые алгебраические свойства композиций программных операторов. С каждым программным оператором a ассоциируются два множества переменных $Use(a)$ и $Def(a)$: первое множество состоит из всех тех переменных, значения которых должны быть определены для успешного выполнения оператора a , а второе – из всех тех переменных, значения которых изменяются в результате выполнения a . Например, $Use(y = x++) = \{x\}$, $Def(y = x++) = \{x, y\}$. В том случае, если верны равенства $Use(a) \cap Def(b) = \emptyset$, $Def(a) \cap Use(b) = \emptyset$, $Def(a) \cap Def(b) = \emptyset$, результат выполнения операторов a и b не зависит от того порядка, в котором они выполняются. Такие операторы называются коммутативными, и это свойство задается тождеством $ab = ba$. В том случае, если $Def(a) \subseteq Def(b)$, $Def(a) \cap Use(b) = \emptyset$, последовательное выполнение операторов a и b дает такой же результат, что и выполнение оператора b . Тогда говорят, что оператор b поглощает оператор a , и это свойство задается тождеством $ab = b$. Нас интересуют полугруппы операторов, определяемые системами тождеств коммутативности и поглощения.

3. Двухленточные детерминированные машины. Для исследования вопросов алгоритмической разрешимости и сложности проблемы эквивалентности программ в полугрупповых моделях программ необходимо предложить такой способ описания полугрупп, задающих семантику программных операторов, который позволял эффективно проверять равенство элементов полугрупп. Как известно, отношения на множестве конечных слов можно задавать при помощи многоленточных автоматов. В частности, некоторые отношения эквивалентности на множестве конечных слов можно задавать при помощи двухленточных автоматов. Поэтому для описания отношения равенства элементов полугрупп, представленных *A-цепочками*, можно воспользоваться двухленточными детерминированными автоматами (машинами).

Двухленточной детерминированной машиной (2-DM) называется система $D = \langle A, Q_1, Q_2, q_0, Q_{аксерт}, \varphi \rangle$, состоящая из алфавита A , двух непересекающихся множеств состояний Q_1 и Q_2 , начального состояния $q_0, q_0 \in Q_1 \cup Q_2$, множества допускающих состояний $Q_{аксерт}, Q_{аксерт} \subseteq Q_1 \cup Q_2$, и функции переходов



$\varphi : (Q_1 \cup Q_2) \times A \rightarrow Q_1 \cup Q_2$. Машина D прочитывает пару A -цепочек w_1 и w_2 , записанных на лентах 1 и 2. Когда машина D пребывает во внутреннем состоянии $q, q \in Q_\sigma, \sigma \in \{1, 2\}$, она прочитывает очередной оператор x (если таковой есть) цепочки w_σ , помещенной на ленте σ , и переходит в состояние $q' = \varphi(q, x)$. Пара A -цепочек $\langle w_1, w_2 \rangle$ допускается машиной D , если она прочитывает w_1 и w_2 и оказывается по прочтении этих слов в допускающем состоянии. Более формально поведение машины D определяется следующим образом.

Прогоном 2-DM D называется конечная или бесконечная последовательность пар $\alpha = (q_1, x_1), (q_2, x_2), \dots, (q_i, x_i), \dots$, удовлетворяющая соотношениям $q_i \in Q_1 \cup Q_2, x \in A$, и $q_{i+1} = \varphi(q_i, x)$ для всех $i, i \geq 1$. Если эта последовательность оканчивается парой (q_n, x_n) , то говорят, что прогон α начинается в состоянии q_1 и достигает состояния $q_{n+1} = \varphi(q_n, x)$. Для $\sigma \in \{1, 2\}$ рассмотрим подпоследовательность $\alpha_\sigma = (q_{i_1}, x_{i_1}), (q_{i_2}, x_{i_2}), \dots$, состоящую из всех тех пар (q_{i_j}, x_{i_j}) последовательности α , которые удовлетворяют условию $q_{i_j} \in Q_\sigma$. Тогда σ -проекцией прогона α будем называть слово $\alpha[\sigma] = x_{i_1} x_{i_2} \dots$. Функцию переходов 2-DM D можно распространить на множество пар слов $A^* \times A^*$ следующим образом: $\varphi^*(q, w_1, w_2) = q'$, если существует прогон α машины D , который начинается в состоянии q , достигает состояния q' , и при этом $\alpha[1] = w_1, \alpha[2] = w_2$; в противном случае значение $\varphi^*(q, w_1, w_2)$ неопределено. Каждая 2-DM $D = \langle A, Q_1, Q_2, q_0, Q_{\text{accept}}, \varphi \rangle$ порождает бинарное отношение $E_D = \{ \langle w_1, w_2 \rangle : \varphi^*(q_0, w_1, w_2) \in Q_{\text{accept}} \}$. Будем говорить, что 2-DM D описывает полугруппу F , если для любой пары A^* -цепочек h_1 и h_2 имеет место соотношение $[h_1]_F = [h_2]_F \Leftrightarrow \langle h_1, h_2 \rangle \in E_D$.

Теорема 1. Полугруппа F может быть описана некоторой 2-DM тогда и только тогда, когда F – упорядоченная полугруппа.

Доказательство. (\Rightarrow) Для любой полугруппы F отношение достижимости \prec_F рефлексивно и транзитивно. Если бы оно не было антисимметричным, то нашлась бы такая пара A -цепочек h', h'' , что $h'' \neq \lambda, [h'h'']_F = [h']_F$. Поскольку полугруппу F описывает 2-DM D , в этом случае нашлась бы пара допускающих состояний $q' = \varphi^*(q_0, h', h')$ и $q'' = \varphi^*(q_0, h', h'h'')$. Но это означало бы, что $q' \in Q_2$ и, следовательно, значение $\varphi^*(q_1, h'h'', h')$ неопределено, вопреки тому, что $[h'h'']_F = [h']_F$.

(\Leftarrow) Для упорядоченной полугруппы $F = \langle S, \lambda, \circ \rangle$ рассмотрим 2-DM $D = \langle A, Q_1, Q_2, q_0, Q_{\text{accept}}, \varphi \rangle$, где $Q_1 = \{ (s', s'') : s', s'' \in S, s' \prec_F s'' \}$, $Q_2 = (S \times S) \setminus Q_1$, $q_0 = (s_0, s_0)$, $Q_{\text{accept}} = \{ (s, s) : s \in S \}$ и функция переходов определена соотношениями $\varphi((s', s''), a) = (s' \circ a, s'')$, если $(s', s'') \in Q_1$, и $\varphi((s', s''), a) = (s', s'' \circ a)$, если $(s', s'') \in Q_2$. Какова бы ни была пара A -цепочек h_1, h_2 , либо есть такой префикс h'_1 цепочки h_1 , что $\varphi^*(q_0, h'_1, h_2)$ определено, либо есть такой префикс h'_2 цепочки h_2 , что $\varphi^*(q_0, h_1, h'_2)$ определено. Предположим, что $[h_1]_F = [h_2]_F, h_1 = h'_1 a_1 a_2 \dots a_k$, и $\varphi^*(q_0, h'_1, h_2) = q'$. Отношение достижимости \prec_F является частичным порядком, и



поэтому $[h'_1 a_1 a_2 \dots a_i]_F \prec_F [h_2]_F$ выполняется для всех $i, 1 \leq i \leq k$. Индукцией по i , можно убедиться, что $\varphi^*(q_0, h'_1 a_1 a_2 \dots a_i, h_2) = ([h'_1 a_1 a_2 \dots a_i]_F, [h_2]_F) \in Q_1$ для всех $i, 1 \leq i \leq k$. Поэтому значение $\varphi^*(q_0, h_1, h_2)$ определено и равно $([h_1]_F, [h_2]_F)$. Как видно из описания D , достигнутое состояние $\varphi^*(q_0, h_1, h_2)$ является допускающим. Теорема доказана.

4. Полу группы коммутативных и поглощаемых операторов. Пусть на множестве операторов A заданы два бинарных отношения $C_0 \subseteq A \times A, R_0 \subseteq A \times A$. Будем рассматривать полу группы $F(C_0, R_0)$ определяемые следующими двумя множествами тождеств: $\{ab = ba : \langle a, b \rangle \in C_0\}$ и $\{ab = b : \langle a, b \rangle \in R_0\}$. Обозначим символом C рефлексивно-симметричное замыкание отношения C_0 , а символом R – транзитивное замыкание отношения R_0 . Непосредственно из приведенных определений следует

Утверждение 1. $[h_1]_{F(C_0, R_0)} = [h_2]_{F(C_0, R_0)} \Leftrightarrow [h_1]_{F(C, R)} = [h_2]_{F(C, R)}$.

В дальнейшем для краткости запись $[h]$ будет обозначать элемент полу группы $F(C, R)$, порожденный A -цепочкой h . Введем ряд вспомогательных конструкций. Для A -цепочки h и оператора b , запись $N(b, h)$ будет обозначать число вхождений оператора b в цепочку h . Воспользуемся записью $h_1 \xrightarrow{(a, b, i)}_C h_2$ для обозначения отношения между A -цепочками, которое выполняется, если $h_1 = h'abh'', h_2 = h'bah'', |h'| = i - 2$, т.е. h_2 получается из h_1 применением тождества коммутативности $ab = ba$. Аналогично, запись $h_1 \xrightarrow{(a, b, i)}_R h_2$ в случае $h_1 = h'abh'', h_2 = h'bh'', |h'| = i - 2$, будет обозначать отношение применения тождества поглощения $ab = b$. Выводом равенства $[H'] = [H'']$ назовем такую конечную последовательность A -цепочек h_0, h_1, \dots, h_n , в которой $h_0 = H', h_n = H''$, и при этом для любого $k, 1 \leq k \leq n$, выполняется одно из трех отношений:

- $h_{k-1} \xrightarrow{(a, b, i)}_C h_k$ для некоторого $i, 2 \leq i \leq |h_{k-1}|$, и пары $\langle a, b \rangle \in C$,
- $h_{k-1} \xrightarrow{(a, b, i)}_R h_k$ для некоторого $i, 2 \leq i \leq |h_{k-1}|$, и пары $\langle a, b \rangle \in R$,
- $h_k \xrightarrow{(a, b, i)}_C h_{k-1}$ для некоторого $i, 2 \leq i \leq |h_k|$, и пары $\langle a, b \rangle \in R$.

Очевидно, каждое равенство $[H'] = [H'']$ имеет хотя бы один вывод. Нетрудно убедиться, что функция $N(b, h)$ обладает следующими свойствами:

1. $N(b, h'h'') = N(b, h') + N(b, h'')$;
2. если $h' \xrightarrow{(x, y, i)}_C h''$, то $N(b, h') = N(b, h'')$;
3. если $h' \xrightarrow{(x, y, i)}_R h''$, то $N(b, h') = N(b, h'') + 1$ в случае $b = x$, и $N(b, h') = N(b, h'')$ иначе.

Основным результатом настоящей статьи является

Теорема 2. Полу группа $F(C_0, R_0)$ упорядочена тогда и только тогда, когда $C \cap R = \emptyset$.

Доказательство. (\Rightarrow) Если $\langle a, b \rangle \in C \cap R$, то последовательность b, ab, ba – это вывод равенства $[b] = [ba]$, свидетельствующего о неупорядоченности полу группы.



(\Leftarrow) Предположим, что для некоторой пары A -цепочек H', H'' имеет место равенство $[HH''] = [H']$. Покажем, что для любого оператора b верно равенство $N(b, H'') = 0$. Отсюда будет следовать, что H'' является пустой цепочкой, а полугруппа $F(C_0, R_0)$ является упорядоченной. Для обоснования указанного равенства нам потребуется вспомогательная лемма, устанавливающая специальный инвариант всякого вывода в полугруппе $F(C, R)$. Введем следующие обозначения. Рассмотрим произвольную A -цепочку $h = x_1 x_2 \dots x_m$ и оператор b . Тогда будем использовать:

- запись $right(b, h)$ для обозначения самой правой позиции j в цепочке h , в которой располагается оператор, поглощающий b , т.е. $right(b, h) = \max\{j : \langle b, x_j \rangle \in R\}$,
- запись $succ(b, h)$ для обозначения числа вхождений оператора b в цепочку $x_{right(b, h)+1} \dots x_m$, т.е. $succ(b, h) = N(b, h|_{\geq right(b, h)+1})$.

Лемма 1. Если $C \cap R = \emptyset$, $[h_0] = [h_n]$ и значение $right(b, h_0)$ определено, то $succ(b, h_0) = succ(b, h_n)$.

Доказательство леммы. Рассмотрим произвольный вывод h_0, h_1, \dots, h_n равенства $[h_0] = [h_n]$ и покажем, что на любом шаге этого вывода значения $succ(b, h_i)$ и $succ(b, h_{i+1})$ одинаковы.

Предположим, что значение $succ(b, h_i)$ определено и $h_i \xrightarrow{(a, c, j)}_C h_{i+1}$. Значит, $h_i = x_1 \dots x_{j-2} a c x_{j+1} \dots x_m$ и $h_{i+1} = x_1 \dots x_{j-2} c a x_{j+1} \dots x_m$. Очевидно, что если $j < right(b, h_i)$ или $right(b, h_i) < j - 1$, то расположение операторов в цепочке h_{i+1} относительно позиции $right(b, h_i)$ не изменилось, и поэтому $succ(b, h_i) = succ(b, h_{i+1})$. Если $right(b, h_i) = j - 1$, то оператор a , коммутирующий с оператором c на основании тождества $ac = ca$ является самым правым оператором цепочки h_i , поглощающим рассматриваемый оператор b . Вследствие равенства $C \cap R = \emptyset$, оператор b не может совпадать с c . Поэтому число $succ(b, h_{i+1})$ операторов b , расположенных в цепочке h_{i+1} справа от оператора a , остается равным $succ(b, h_i)$. Если $right(b, h_i) = j$, то оператор c оказывается самым правым оператором цепочки h_i , поглощающим оператор b . Вследствие равенства $C \cap R = \emptyset$, оператор b не может совпадать с a . В том случае, если оператор a не поглощает b , верно $right(b, h_{i+1}) = j - 1$, а иначе выполняется $right(b, h_{i+1}) = j$. Но и в том и в другом случае число $succ(b, h_{i+1})$ остается равным $succ(b, h_i)$.

Предположим, что значение $succ(b, h_i)$ определено и $h_i \xrightarrow{(a, c, j)}_R h_{i+1}$ (случай $h_{i+1} \xrightarrow{(a, c, j)}_R h_i$ рассматривается аналогично). Это означает, что $h_i = x_1 \dots x_{j-2} a c x_{j+1} \dots x_m$ и $h_{i+1} = x_1 \dots x_{j-2} c x_{j+1} \dots x_m$. Если $j \leq right(b, h_i)$, то применение тождества поглощения $ac = c$ не затрагивает операторы цепочки h_i , располагающиеся в позиции $right(b, h_i)$ и правее. Поэтому значения $succ(b, h_i)$ и $succ(b, h_{i+1})$ остаются равными. Случай $j = right(b, h_i) + 1$, когда оператор a является самым правым оператором цепочки h_i , поглощающим рассматриваемый оператор b , невозможен, поскольку отношение поглощения R транзитивно замкнуто, и оператор c в этом случае



также должен поглощать b . Наконец, рассмотрим случай $right(b, h_i) + 1 < j$, когда самый правый оператор цепочки h_i , поглощающий оператор b , расположен слева от оператора a . Тогда оператор c не может поглощать b , и поэтому оператор b не может совпадать с a . Так как цепочка h_{i+1} получается из h_i удалением одного лишь оператора a , величины $succ(b, h_i)$ и $succ(b, h_{i+1})$ будут равны. Лемма доказана.

Так как отношение C рефлексивно, а отношение R транзитивно, пустота пересечения $C \cap R$ означает, что R является отношением строгого частичного порядка на конечном множестве операторов A . Тогда множество \hat{A} всех операторов, содержащихся в A -цепочке $H'H''$, разбивается на конечное число слоев L_0, L_1, \dots, L_m следующим образом:

L_0 – подмножество всех максимальных элементов множества \hat{A} относительно порядка R ;

для каждого $k, 1 \leq k \leq m, L_k$ – подмножество всех максимальных элементов множества $A \setminus \bigcup_{i=0}^{k-1} L_i$ относительно порядка R .

Используя лемму 1, завершим доказательство теоремы 2, показав индукцией по номеру слоя k , что для любого оператора $b, b \in L_k$, верно равенство $N(b, H'') = 0$.

Базис индукции ($k = 0$). Из перечисленных выше свойств функции $N(b, h)$ видно, что, каков бы ни был оператор b , значение этой функции изменяется после применения некоторого тождества к цепочке h только тогда, когда в h содержится хотя бы один оператор, поглощающий b . Поэтому для любого оператора $b, b \in L_0$, и для произвольного вывода h_0, h_1, \dots, h_n равенства $[H'H''] = [H']$ будут выполняться равенства $N(b, h_0) = N(b, h_1) = \dots = N(b, h_n)$.

Значит, $N(b, H'H'') = N(b, H') + N(b, H'') = N(b, H')$ и, следовательно, $N(b, H'') = 0$.

Индуктивный переход ($k \rightarrow k + 1$). Предположим, что для любого оператора $c, c \in \bigcup_{i=0}^k L_i$, верно равенство $N(c, H'') = 0$. Рассмотрим произвольный оператор $b, b \in L_{k+1}$. Так как $1 \leq k$, в A -цепочке $H'H''$ есть операторы, поглощающие b , и поэтому значение $right(b, H'H'')$ определено. Однако на основании индуктивного предположения ни один из этих операторов не содержится в цепочке H'' , и поэтому $right(b, H'H'') \leq |H'|$. Пусть $right(b, H'H'') = l$ и $H = H' \downarrow_{l+1}$. Тогда, согласно лемме 1 справедливы равенства $N(b, HH'') = succ(b, HH'') = succ(b, H') = N(b, H)$, из которых следует $N(b, H'') = 0$.

Этим завершается обоснование индуктивного перехода и доказательство теоремы 2.

5. Заключение. Из теорем 1 и 2 вытекает следующее утверждение.

Теорема 3. Полугруппа $F(C_0, R_0)$, порожденная системой тождеств коммутативности $\{ab = ba : \langle a, b \rangle \in C_0\}$ и поглощения $\{ab = b : \langle a, b \rangle \in R_0\}$, описывается двухленточной детерминированной машиной в том и только том случае, когда рефлекс-



сивно-симметричное замыкание C отношения коммутативности операторов C_0 и транзитивное замыкание R отношения поглощения операторов R_0 не пересекаются.

Если полугруппа операторов $F(C_0, R_0)$ оказывается упорядоченной, то, как показано в работах [3,8], проблема эквивалентности программ, семантика операторов которых определяется тождествами этой полугруппы, является алгоритмически разрешимой [3] и принадлежит классу сложности NP. Более того, для некоторых классов моделей программ, операторы которых подчиняются законам коммутативности и подавления, удалось построить алгоритмы полиномиальной сложности, разрешающие проблему эквивалентности программ в этих моделях (см. [1,7–9]). Заслуживает внимания гипотеза о том, что для любой системы тождеств коммутативности и перестановочности, удовлетворяющих соотношению $C \cap R = \emptyset$, проблема эквивалентности программ, семантика операторов которых определяется полугруппой $F(C_0, R_0)$, проблема эквивалентности программ в этой модели разрешима за время, полиномиальное относительно размеров программ. Для разрешения этой гипотезы требуется провести исследование устройства двухленточных машин, описывающих упорядоченные полугруппы операторов $F(C_0, R_0)$. Результаты этих исследований будут опубликованы в последующих статьях.

Проанализируем теперь, насколько задача проверки эквивалентности программ усложняется в том случае, когда $C \cap R \neq \emptyset$. Предположим, что для некоторой пары различных операторов a и b выполняется включение $\langle a, b \rangle \in C \cap R$. Это означает, что операторы a и b являются перестановочными, и при этом b поглощает оператор a . Но тогда множества переменных, определяемых этими операторами, обязаны удовлетворять двум соотношениям $Def(a) \cap Def(b) = \emptyset$ и $Def(a) \subseteq Def(b)$. Это возможно лишь в том случае, когда $Def(a) = \emptyset$. Последнее означает, что оператор не совершает никаких изменений данных, т.е. он является бесполезным оператором. Бесполезные операторы такого рода легко распознаются и не затрудняют анализ программ.

Пересечение $C \cap R$ может оказаться непустым также и потому, что для некоторого оператора a выполняется соотношение $Def(a) \cap Use(a) = \emptyset$, вследствие которого $\langle a, a \rangle \in C \cap R$. Наличие в программах такого оператора может существенно затруднить их анализ в том случае, если найдется еще один оператор b , который подавляется оператором a . Как было установлено в работе [10], проверка эквивалентности программ, операторы a и b которых удовлетворяют двум тождествам $ba = a$ и $aa = a$, является PSPACE-трудной задачей.

Литература

1. Захаров В.А. Быстрые алгоритмы разрешения эквивалентности операторных программ на уравновешенных шкалах // Математические вопросы кибернетики, вып. 7. – М.: Наука. – 1998. – с. 303–324.
2. Harel D. Dynamic logics // Handbook of Philosophical Logics, D.Gabbay and F.Guenther (eds.). – 1984. – p. 497–604.
3. Глушков В.М., Летичевский А.А. Теория дискретных преобразователей // Избранные вопросы алгебры и логики: сб. статей. – Новосибирск: Наука, 1973. – с. 5–39.
4. Подловченко Р.И. Полугрупповые модели программ // Программирование. – 1981. – N4. – с. 3–13.
5. Захаров В.А. Проверка эквивалентности программ при помощи двухленточных автоматов. I // Кибернетика и системный анализ. – 2010. – N 4.
6. Zakharov V.A. An efficient and unified approach to the decidability of equivalence of propositional programs // Lecture Notes in Computer Science. – 1998. – v. 1443. – p. 247–259.



7. Захаров В.А. Быстрые алгоритмы разрешения эквивалентности пропозициональных операторных программ на упорядоченных полугрупповых шкалах // Вестник Московского университета, сер. 15, Вычислительная математика и кибернетика. – 1999, N 3. – с. 29-35.

8. Захаров В.А. Проверка эквивалентности программ при помощи двухленточных автоматов. II // Кибернетика и системный анализ. – 2010. – N 5.

9. Щербина В.Л., Захаров В.А. Эффективные алгоритмы проверки эквивалентности программ в моделях, связанных с обработкой прерываний // Вестник Московского университета, сер. 15, Вычислительная математика и кибернетика. – 2008, N 1.

10. Podlovchenko R., Rusakov D., Zakharov V. On the equivalence problem for programs with mode switching // Lecture Notes in Computer Science. – 2006. – v. 3845. – p. 351-352.

ON A SEMIGROUP MODEL OF SEQUENTIAL PROGRAMS SPECIFIED BY MEANS OF TWO-TAPE AUTOMATA

V. V. PODYMOV

V. A. ZAKHAROV

*Lomonosov Moscow
State University*

e-mail: zakh@cs.msu.su

In this paper the authors study the semigroups of program statements subjected to the laws of commutativity and cancellation. The semigroups of such kind appear in the verification problems for propositional models of sequential programs. These models of programs can be easily constructed based on the outcome of a light-weight data-flow analysis of computer programs. The main results of the paper are necessary and sufficient conditions for the semigroups under consideration to be specified by two-tape one-way deterministic automata. Relying on this theorem the authors estimated the complexity of equivalence checking problem for sequential programs.

Key words: program, equivalence-checking, semigroup of program statements, commutative statements, absorbed statements, two-tape automata.