П Р И К Л А Д Н А Я   Л О Г И К А
(Вычислительные системы)
1986 год   Выпуск 116

УДК 519.681

АКСИОМАТИЧЕСКАЯ СЕМАНТИКА ПРОГРАММ, НЕ ДОПУСКАЮЩАЯ ПОБОЧНЫХ ЭФФЕКТОВ

И.А. Ломазова

В [I] определена аксиоматическая семантика AX для цикличе - ских программ, которые задаются над спецификацией абстрактных типов данных. Семантика AX является расширением обычной операционной семантики программ над моделями спецификаций абстрактных типов данных. Логика Хоара полна относительно семантики AX, в то время как относительно операционной семантики логика Хоара полна только в случае выразительности базового логического языка [2]. Однако, в отличие от операционной семантики, семантика AX допускает побочные эффекты: в соответствии с семантикой AX в ходе вы - полнения программы могут меняться значения переменных, не входя - щих в программу, и изменение начального значения переменной, не входящей в программу, может изменить результат работы программы. Последнее свойство семантики AX свидетельствует о некоторой неадекватности ее содержательному представлению о том, как выполняется программа.

В настоящей работе строится и исследуется устойчивая аксио - матическая семантики STAX такая, что

- логика Хоара полна относительно STAX,
- семантика STAX не допускает побочных эффектов на классе всех моделей спецификаций абстрактных типов данных.

Тем самым дается ответ на сформулированный в [I] вопрос о существовании основанной на логике Хоара семантики программ, кото - рая не допускала бы побочных эффектов. В работе приводится также решение еще одного вопроса, сформулированного в [I], о побочных эффектах семантики AX на классе термально-порожденных моделей спецификаций абстрактных типов данных.

I. Основные определения

Пусть Σ – многосортная сигнатура, Var – множество переменных с приписанными им сортами из Σ. Через $L(\Sigma)$ обозначим язык логических формул первого порядка с равенством над сигнатурой Σ и множеством Var, через $WP(\Sigma)$ – язык программ с циклами в сигнатуре Σ:

$$WP(\Sigma) \ni S ::= x := e \mid S_1 ; S_2 \mid \underline{\text{if}}\ b\ \underline{\text{then}}\ S_1\ \underline{\text{else}}\ S_2\ \underline{\text{fi}} \mid \underline{\text{while}}\ b\ \underline{\text{do}}\ S\ \underline{\text{od}},$$

где $x \in Var$, $b \in L(\Sigma)$, e – терм в сигнатуре Σ со свободными переменными из Var. Для программы $S \in WP(\Sigma)$ через $Var(S)$ будем обозначать множество переменных, входящих в S.

Пусть T – множество замкнутых формул из $L(\Sigma)$. Пара (Σ, T) образует спецификацию абстрактных типов данных [3]. Семантика спецификации (Σ, T) определяется как множество $Mod(\Sigma, T)$ всех Σ-моделей, удовлетворяющих аксиомам T. Σ-модель A называется термально-порожденной, если для любого элемента a модели A существует такой Σ-терм t без свободных переменных, что значение t в A совпадает с a. Через $TMod(\Sigma, T)$ обозначим множество всех термально-порожденных Σ-моделей, удовлетворяющих T.

Пусть $A \in Mod(\Sigma, T)$. Состоянием в A назовем функцию, сопоставляющую каждой переменной из Var элемент модели A того же сорта, что и переменная. Множество всех состояний в модели A обозначим через $State(A)$.

Пусть K – класс Σ-моделей. Семантика M программы $S \in WP(\Sigma)$ на классе моделей K определяется как семейство $M = M[K] = \{M(A) : A \in K\}$, где $M(A)(S) \subset State(A) \times State(A)$. Программа $S \in WP(\Sigma)$ частично-корректна с предусловием $p \in L(\Sigma)$ и постусловием $q \in L(\Sigma)$ относительно семантики $M(A)$ (обозначается $M(A) \models p\{S\}q$), если для любой пары состояний $(\sigma, \tau) \in M(A)(S)$ из $A \models p(\sigma)$ следует, что $A \models q(\tau)$. Полагаем $M[K] \models p\{S\}q \leftrightarrow (\forall A \in K)\ M(A) \models p\{S\}q$.

Примером семантики языка $WP(\Sigma)$ на классе моделей $Mod(\Sigma, T)$ может служить операционная семантика OP, которая определяется обычным образом: $(\sigma, \tau) \in OP(A) \cdot (S)$, если выполнение программы $S \in WP(\Sigma)$ в соответствии со стандартным пониманием программных конструкций, начатое в состоянии $\sigma \in State(A)$, завершается в состоянии $\tau \in State(A)$.

Пусть A – Σ-модель. Будем говорить, что язык $L(\Sigma)$ выразителен для модели A (относительно операционной семантики), если для любой формулы $p \in L(\Sigma)$ и любой программы $S \in WP(\Sigma)$ в $L(\Sigma)$ выразимо сильнейшее постусловие $sp_A(p,S) = \{\sigma \in State(A) : (\exists \tau \in State(A)) \times OP(A)(S)(\tau,\sigma) \ \& \ A \models p(\tau)\}$.

Аксиоматическая семантика программ будет определена на основе логики Хоара [4] для вывода утверждений о частичной корректности программ. Пусть (Σ,T) – спецификация. Логика Хоара $HL(\Sigma,T)$ состоит из следующих аксиом и правил вывода.

Аксиомы:

1. p, где $p \in L(\Sigma)$ и $T \vdash p$;

2. $p[e/x]\{x := e\}\,p$, где $p[e/x]$ – результат замены в p всех свободных вхождений переменной x на терм e.

Правила вывода:

1. $\dfrac{p\{S_1\}r,\ r\{S_2\}q}{p\{S_1;\ S_2\}q}$;

2. $\dfrac{p \,\&\, b\{S_1\}q,\ p \,\&\, \neg b\{S_2\}q}{p\{\underline{\text{if}}\ b\ \underline{\text{then}}\ S_1\ \underline{\text{else}}\ S_2\ \underline{\text{fi}}\}q}$;

3. $\dfrac{p \,\&\, b\{S\}p}{p\{\underline{\text{while}}\ b\ \underline{\text{do}}\ S\ \underline{\text{od}}\}\,p \,\&\, \neg b}$;

4. $\dfrac{p \supset p_1,\ p_1\{S\}q_1,\ q_1 \supset q}{p\{S\}q}$.

Запись $HL(\Sigma,T) \vdash p\{S\}q$ будем использовать для обозначения того, что утверждение $p\{S\}q$ выводимо в логике $HL(\Sigma,T)$.

Будем говорить, что семантика $M = M[K]$ не допускает побочных эффектов первого рода, если для любых $A \in K$, $S \in WP(\Sigma)$, $(\sigma,\tau) \in M(A)(S)$; если $v \notin Var(S)$, то $\sigma(v) = \tau(v)$.

Будем говорить, что семантика $M = M[K]$ не допускает побочных эффектов второго рода, если для любых $A \in K$, $S \in WP(\Sigma)$, $(\sigma,\tau) \in M(A)(S)$; $\sigma' \in State(A)$, если состояние σ' совпадает с σ на $Var(S)$, то найдется состояние τ', совпадающее с τ на $Var(S)$, такое что $(\sigma',\tau') \in M(A)(S)$.

2. Устойчивая аксиоматическая семантика программ

Устойчивую аксиоматическую семантику $STAX$ определим следующим образом: пусть A – модель спецификации (Σ,T), $S \in WP(\Sigma)$.

Пусть $Var(S) = x = x_1 x_2 \ldots x_n$. Полагаем $STAX(\Sigma,T)(A)(S) = \{(\sigma,\tau) \in State(A) \times State(A) :$ для любых $v \in Var \setminus Var(S)$;

$p, q \in L(\Sigma)$, $\sigma(v) = \tau(v)$, и если $HL(\Sigma,T) \vdash p(x,z)\{S\}q(x,z)$, где $z = z_1 z_2 \ldots z_k$, причем $z_i \notin Var(S)$, то $A \models (\forall z)(p(x(\sigma),z) \supset q(x(\tau),z))\}$. Соответственно $STAX(\Sigma,T) = STAX(\Sigma,T)[Mod(\Sigma,T)]$.

Непосредственно из определения видно, что семантика STAX не допускает побочных эффектов как первого, так и второго рода.

В определении семантики STAX для модели A присутствует множество T аксиом спецификации этой модели. Можно показать, что на самом деле семантика STAX на конкретной модели не зависит от спецификации этой модели.

Пусть A – Σ-модель. Через $Th(A)$ обозначим множество всех формул из $L(\Sigma)$, истинных в A. Определим $STAX(A)(S) = STAX(\Sigma, Th(A))(S)$.

ТЕОРЕМА I (единственности). Пусть (Σ,T) – спецификация. Для любой программы $S \in WP(\Sigma)$ и любой модели $A \in Mod(\Sigma,T)$

$$STAX(\Sigma,T)(A)(S) = STAX(A)(S).$$

ДОКАЗАТЕЛЬСТВО. Включение $STAX(A)(S) \subseteq STAX(\Sigma,T)(A)(S)$ следует непосредственно из определения семантики. Докажем, что $STAX(\Sigma,T)(A)(S) \subseteq STAX(A)(S)$.

Пусть $(\sigma,\tau) \in STAX(\Sigma,T)(A)(S)$ и пусть $HL(\Sigma,Th(A)) \vdash p\{S\}q$. Нужно доказать, что $A \models (\forall z)(p(x(\sigma),z) \supset q(x(\tau),z))$. Поскольку $HL(\Sigma,Th(A)) \vdash p\{S\}q$, то, очевидно, найдется такая замкнутая формула $\varphi \in L(\Sigma)$, что $A \models \varphi$ и $HL(\Sigma,T \cup \{\varphi\}) \vdash p\{S\}q$. Тогда $HL(\Sigma T) \vdash (p \wedge \varphi)\{S\}q \wedge \varphi$; и, следовательно, $A \models (\forall z)(p(x(\sigma),z) \wedge \varphi \supset q(x(\tau),z))$. Далее получаем $A \models \varphi \supset (\forall z)(p(x(\sigma),z) \supset q(x(\tau),z))$, и поскольку $A \models \varphi$, то $A \models (\forall z)(p(x(\sigma),z) \supset q(x(\tau),z))$.

Для доказательства теоремы о полноте логики Хоара относительно семантики STAX нам потребуется

ЛЕММА I. Пусть $S \in WP(\Sigma)$; $Var(S) = x = x_1 x_2 \ldots x_n$; $p(x,z)$, $q(x,z) \in L(\Sigma)$. Расширим сигнатуру Σ, добавив к ней n констант $c = c_1 c_2 \ldots c_n$. Полагаем $\Sigma_c = \Sigma \cup \{c\}$. Тогда

$$HL(\Sigma,T) \vdash p\{S\}q \Leftrightarrow HL(\Sigma_c,T) \vdash (x=c)\{S\}(\forall z)(p(c,z) \supset q(x,z)),$$

где $x = c$ означает $(x_1 = c_1) \& (x_2 = c_2) \& \ldots \& (x_n = c_n)$.

ДОКАЗАТЕЛЬСТВО. а) Пусть $HL(\Sigma,T) \vdash p\{S\}q$. Применяя основные и допустимые правила логики Хоара, последовательно получаем:

$$HL(\Sigma_c,T) \vdash p(x,z) \vee \neg p(c,z)\{S\}q(x,z) \vee \neg p(c,z),$$

$$HL(\Sigma_c,T) \vdash (x=c) \& (p(x,z) \vee \neg p(c,z))\{S\}p(c,z) \supset q(x,z),$$

$$HL(\Sigma_c,T) \vdash (x=c)\{S\}p(c,z) \supset q(x,z),$$

$$HL(\Sigma_c,T) \vdash (x=c)\{S\}(\forall z)(p(c,z) \supset q(x,z)).$$

б) Пусть $HL(\Sigma_c,T) \vdash (x=c)\{S\}(\forall z)(p(c,z) \supset q(x,z))$. Получаем:

$$HL(\Sigma_c,T) \vdash (x=c) \& p(c,z)\{S\}(\forall z)(p(c,z) \supset q(x,z)) \& p(c,z),$$

$$HL(\Sigma_c,T) \vdash (x=c) \& p(c,z)\{S\}q(x,z).$$

По лемме о константе [I] отсюда следует, что

$$HL(\Sigma,T) \vdash (x=y) \& p(y,z)\{S\}q(x,z),$$

где $y = y_1, y_2, \ldots, y_n$ – переменные, отличные от x и z. Тогда

$$HL(\Sigma,T) \vdash (\exists y)(x=y \& p(y,z))\{S\}q(x,z),$$

и поскольку $T \vdash p(x,z) \supset (\exists y)(x=y \& p(y,z))$, то

$$HL(\Sigma,T) \vdash p(x,z)\{S\}q(x,z).$$

Лемма доказана.

ТЕОРЕМА 2 (о полноте).

$$HL(\Sigma,T) \vdash p\{S\}q \Leftrightarrow STAX(\Sigma,T) \models p\{S\}q.$$

ДОКАЗАТЕЛЬСТВО. Импликация $\Rightarrow$ следует непосредственно из определения семантики STAX.

Обратная импликация доказывается методом от противного. Пусть $HL(\Sigma,T) \nvdash p\{S\}q$. Нужно доказать, что $STAX(\Sigma,T) \not\models p\{S\}q$, т.е., что найдутся модель $A \in Mod(\Sigma,T)$ и состояния $(\sigma,\tau) \in STAX(\Sigma,T)(A)(S)$ такие, что $A \models p(\sigma)$, но $A \not\models q(\tau)$.

Пусть $Var(S) = x = x_1 x_2 \ldots x_n$, $c = c_1 c_2 \ldots c_n$ – новые константы и пусть $\{(r_i, s_i): i \in \omega\}$ – перечисление всех пар Σ_c-формул, таких что $HL(\Sigma_c,T) \vdash r_i\{S\}s_i$. Пусть $r_i = r_i(x,y_i)$, $s_i = s_i(x,y_i)$. Обозначим $\Phi_i(x) = (\forall y_i)(r_i(c,y_i) \supset s_i(x,y_i))$ и $\Psi(x) = (\forall z)(p(c,z) \supset q(x,z))$. Покажем, что для всех $n \in \omega$ $T \nvdash (\bigwedge_{i=1}^{n} \Phi_i(x)) \supset \Psi(x)$. В самом деле, пусть для некоторого m $T \vdash (\bigwedge_{i=1}^{m} \Phi_i(x)) \supset \Psi(x)$. В силу леммы I для всех $i \in \omega$ $HL(\Sigma_c,T) \vdash (x=c)\{S\}\Phi_i(x)$ и, следовательно, $HL(\Sigma_c,T) \vdash (x=c)\{S\}\bigwedge_{i=1}^{m}\Phi_i(x)$, но тогда $HL(\Sigma_c,T) \vdash (x=c)\{S\}\Psi(x)$.

Отсюда в силу леммы I получаем $HL(\Sigma,T) \vdash p\{S\}q$, что противоречит первоначальному предположению.

Тогда найдутся такая модель $B \in Mod(\Sigma_c,T)$ и такие элементы $b = b_1 b_2 \ldots b_n$ модели B, что для всех $i \in \omega$ $B \models \Phi_i(b)$, но $B \not\models \Psi(b)$, т.е., поскольку $\Psi(b) = (\forall z)(p(c,z) \supset q(b,z))$, найдутся такие элементы $a = a_1 a_2 \ldots a_k$ модели B, что $B \models p(c,a)$, но $B \not\models q(b,a)$.

Состояния $\sigma, \tau \in State(B)$ определим следующим образом: $\sigma(z_j) = \tau(z_j) = a_j$ $(j = \overline{1,k})$; $\sigma(x_j) = d_j$, где d_j – значение константы c_j в модели B $(j = \overline{1,n})$; $\tau(x_j) = b_j$ $(j = \overline{1,n})$ и $\sigma(v) = \tau(v)$ для остальных переменных $v \in Var$. Покажем, что $(\sigma,\tau) \in STAX(\Sigma_c,T)(B)(S)$. Имеем $\sigma = \tau$ вне $Var(S)$. Пусть $r,s \in L(\Sigma_c)$ и $HL(\Sigma_c,T) \vdash r\{S\}s$. Тогда $r = r_i$, $s = s_i$ для некоторого $i \in \omega$ и $B \models \Phi_i(b)$, т.е. $B \models (\forall y_i)(r(c,y_i) \supset s_i(b_i,y_i))$ или $B \models (\forall y_i)(r(x(\sigma),y_i) \supset s(x(\tau),y_i))$, что и требовалось доказать.

Искомую модель $A \in Mod(\Sigma,T)$ построим как сужение модели B на Σ. Заметим, что состояния в модели B являются также состояниями в модели A и что непосредственно из определения семантики следует: $STAX(\Sigma,T)(A)(S) \subseteq STAX(\Sigma_c,T)(B)(S)$ для любой программы $S \in WP(\Sigma)$. Поэтому $(\sigma,\tau) \in STAX(\Sigma,T)(A)(S)$. Имеем $A \models p(x,z)(\sigma)$, ибо $B \models p(c,a)$ и $A \not\models q(x,z)(\tau)$, ибо $B \not\models q(b,a)$. Таким образом, $STAX(\Sigma,T)(A) \not\models p\{S\}q$. Теорема доказана.

3. Семантика STAX и операционная семантика программ

Пусть $S \in WP(\Sigma)$ и $x = Var(S)$. Графиком программы S в модели $A \in Mod(\Sigma,T)$ назовем предикат $gr_A(S)(x,y)$ (где y – переменные, количество которых совпадает с количеством переменных x), такой что

$$(\sigma,\tau) \in OP(A)(S) \Leftrightarrow (\sigma = \tau \text{ вне } x) \,\&\, A \models gr_A(S)(x(\sigma),x(\tau)).$$

Это определение корректно, так как операционная семантика не допускает побочных эффектов.

Нетрудно заметить, что $gr_A(S)(x_0,x)$ является сильнейшим постусловием для программы S и предусловия $x=x_0$ в модели A относительно операционной семантики, и, следовательно, если язык $L(\Sigma)$ выразителен для модели A, то в $L(\Sigma)$ выразим $gr_A(S)(x_0,x)$.

Соотношение между аксиоматической семантикой STAX и операционной семантикой программ устанавливает следующая

ТЕОРЕМА 3. Пусть A - Σ-модель, $S \in WP(\Sigma)$. Тогда $OP(A)(S) \subseteq STAX(A)(S)$, и если язык $L(\Sigma)$ выразителен для модели A, то $OP(A)(S) = STAX(A)(S)$.

ДОКАЗАТЕЛЬСТВО. Включение $OP(A)(S) \subseteq STAX(A)(S)$ является прямым следствием корректности логики Хоара относительно операционной семантики.

Пусть язык $L(\Sigma)$ выразителен для модели A. По определению графика программы $OP(A) \models (x=x_0)\ \{S\} gr_A(S)(x_0,x)$. В силу выразительности логического языка $L(\Sigma)$ логика Хоара полна для операционной семантики, и, следовательно, $HL(\Sigma, Th(A)) \vdash (x=x_0) \times \{S\} gr_A(S)(x_0,x)$. Пусть $(\sigma,\tau) \in STAX(A)(S)$. Тогда по определению семантики STAX $\sigma = \tau$ вне $Var(S)$ и $A \models (\forall x_0)(\sigma(x) = x_0 \supset gr_A(S)(x_0,\tau(x)))$, т.е. $A \models gr_A(S)(x(\sigma),x(\tau))$ и, следовательно, $(\sigma,\tau) \in OP(A)(S)$. Теорема доказана.

Заметим, что в случае невыразительности базового логического языка операционная семантика и семантика STAX могут не совпадать В качестве примеров такого несовпадения можно использовать примеры неполноты логики Хоара [5].

4. Семантика AX и побочные эффекты

Аксиоматическая семантика AX была определена в [I] следующим образом. Пусть $A \in Mod(\Sigma,T)$, $S \in WP(\Sigma)$ и $AX(\Sigma,T)(A)(S) = \{(\sigma,\tau) \in State(A) \times State(A)$: для любых $p,q \in L(\Sigma)$, если $HL(\Sigma,T) \vdash p\{S\}q$, то из $A \models p(\sigma)$ следует, что $A \models q(\tau)\}$.

Логика Хоара полна относительно семантики AX, однако семантика AX допускает побочные эффекты на классе всех моделей $Mod(\Sigma,T)$.

В [I] было показано, что семантика AX не допускает побочных эффектов первого рода на классе термально-порожденных моделей $TMod(\Sigma,T)$, и был сформулирован вопрос относительно побочных эффектов второго рода для семантики AX на классе моделей $TMod(\Sigma,T)$. Ответ на этот вопрос дает

УТВЕРЖДЕНИЕ. Семантика AX не допускает побочных эффектов второго рода на классе термально-порожденных моделей $TMod(\Sigma,T)$.

Для доказательства нам потребуется

ЛЕММА 2. Пусть $S \in WP(\Sigma)$, $p,q \in L(\Sigma)$, $x \in Var \setminus Var(S)$, t – Σ-терм без свободных переменных. Тогда если $HL(\Sigma,T) \vdash p\{S\}q$, то $HL(\Sigma,T) \vdash p[t/x] \times \{S\}q[t/x]$, где $p[t/x], q[t/x]$ – формулы, полученные из p и q соответственно заменой всех свободных вхождений x на t.

Доказательство леммы проводится непосредственно индукцией по длине вывода утверждения $p\{S\}q$ в аксиоматике Хоара и сводится к очевидному разбору случаев и здесь приводиться не будет.

Перейдем к непосредственному доказательству утверждения. Пусть $A \in TMod(\Sigma,T)$, $(\sigma,\tau) \in AX(\Sigma,T)(A)(S)$ и состояние σ' совпадает с σ на $Var(S)$. Состояние τ' определим следующим образом: $\tau'(x) = \tau(x)$ для $x \in Var(S)$ и $\tau'(x) = \sigma'(x)$ для $x \notin Var(S)$. Докажем, что $(\sigma',\tau') \in AX(\Sigma,T)(A)(S)$.

Пусть для некоторых $p,q \in L(\Sigma)$ $HL(\Sigma,T) \vdash p\{S\}q$ и $A \models p(\sigma')$. Нужно доказать, что $A \models q(\tau')$. Пусть $x_1, x_2, \ldots, x_k$ – все переменные из p и q, не входящие в $Var(S)$. В силу термальной порожденности модели A найдутся такие Σ-термы $t_1, t_2, \ldots, t_k$, что $\sigma'(x_1) = t_1$, $\sigma'(x_2) = t_2, \ldots, \sigma'(x_k) = t_k$ в модели A. Полагаем $p' = p[t_1/x_1][t_2/x_2]\ldots[t_k/x_k]$, $q' = q[t_1/x_1][t_2/x_2] \ldots [t_k/x_k]$. Тогда, в силу леммы 2, $HL(\Sigma,T) \vdash p'\{S\}q'$. По построению p', имеем $A \models p'(\sigma')$, но p' не зависит от переменных, не принадлежащих $Var(S)$, и, следовательно, $A \models p'(\sigma)$. Тогда из того, что $(\sigma,\tau) \in AX(\Sigma,T)(A)(S)$, следует, что $A \models q'(\tau)$ и, поскольку все свободные переменные q' содержатся в $Var(S)$, $A \models q'(\tau')$. По построению q' отсюда следует, что $A \models q(\tau')$. Утверждение доказано.

Заметим, что согласно [I] семантика $AX(\Sigma,T)(A)$ совпадает с операционной семантикой $OP(A)$ только при выполнении условий выразительности базового логического языка $L(\Sigma)$ и термальной порожденности модели A. Доказанное выше утверждение показывает, что условие термальной порожденности обеспечивает отсутствие побочных эффектов для семантики AX. Семантика $STAX$, не допускающая побочных эффектов, совпадает с операционной семантикой при единственном условии выразительности базового логического языка.

Литература

1. BERGSTRA J.A.,TUCKER J.V. The axiomatic semanties of programs based on Hoare's logic. - Acta Inf.,1984,v.21,N 3,p.293-320,

2. COOK S.A. Soundness and completeness of an axiomatic system for program verification.-SIAM J.Comput,1978, v.7,p.70-90.

3. GOGUEN J.A.,THATCHER J.W.,WAGNER E.G. An initial algebra approach to the specification, correctness and implementation of abstract data types.- In: Current trends in programming methodology IV; Data structuring, R.T.Yeh (ed), Englewood Cliffs, New Jersey: Prentice-Hall,1978,p.80-149.

4. HOARE C.A.R. An axiomatic basis for computer programming. - Commun.ACM,1969,v.12,N 10,p.576-583.

5. ЛОМАЗОВА И.А. О сложности индуктивных условий для верификации арифметических программ. -В кн.: Материалы Всесоюз. студенческой конф. "Студент и научно-технический прогресс".-Новосибирск, 1978, с.85-94.

Поступила в ред.-изд.отд.
26 мая 1986 года