

ЧЕБЫШЕВСКИЙ СБОРНИК

Том 26. Выпуск 3.

УДК 519.682.1

DOI 10.22405/2226-8383-2025-26-3-113-124

Логика Хоара для императивного языка, учитывающего некоторые аппаратные ограничения

Д. Ю. Ковалев

Ковалев Даниил Юрьевич — аспирант, Национальный исследовательский университет «Высшая школа экономики» (г. Москва).

e-mail: dyukovalev@hse.ru

Аннотация

В статье определен императивный язык программирования, учитывающий аппаратные ограничения вычислителя с набором инструкций RV32I, заданы его синтаксис и аксиоматическая семантика в виде логики Хоара. Необходимость подобного языка определяется невозможностью напрямую применять формальные доказательства, проведенные для программ на языках, не учитывающих аппаратные ограничения, к транслированному коду, исполняющемуся на реальном аппаратном вычислителе. В то же время проведение доказательств для программ, написанных напрямую в кодах вычислителя, чрезвычайно трудоемко. Описанный язык учитывает такие аппаратные ограничения, как конечная ширина регистра, конечный объем памяти и использование арифметики по модулю вместо классической арифметики. В статье приведен пример программы вычисления НОД двух неотрицательных целых чисел, написанной на этом языке, а также доказательство ее корректности. Таким образом, продемонстрировано, что можно строить формальные доказательства для программ на языке, учитывающем некоторые аппаратные ограничения, и эти доказательства будут сопоставимы по сложности с теми, что построены для аналогичных программ на не учитывающих эти ограничения императивных языках. Одним из направлений развития работы является построение алгоритма трансляции из определенного в статье языка в коды вычислителя и доказательство корректности этого алгоритма.

Ключевые слова: логика Хоара, формальные методы, аксиоматическая семантика.

Библиография: 8 названий.

Для цитирования:

Ковалев, Д. Ю. Логика Хоара для императивного языка, учитывающего некоторые аппаратные ограничения // Чебышевский сборник, 2025, т. 26, вып. 3, с. 113–124.

CHEBYSHEVSKII SBORNIK

Vol. 26. No. 3.

UDC 519.682.1

DOI 10.22405/2226-8383-2025-26-3-113-124

Hoare logic for an imperative language considering some hardware limitations

D. Yu. Kovalev

Kovalev Daniil Yur'evich — postgraduate student, HSE University (Moscow).

e-mail: dyukovalev@hse.ru

Abstract

In this paper, an imperative programming language considering some hardware limitations of a computer based on the RV32I instruction set is defined, including its syntax and semantics in a form of Hoare logic. The need for such language comes from the fact that formal proofs conducted for programs in languages not considering hardware limitations cannot be directly applied to translated code running on real hardware. At the same time, conducting proofs for programs written directly in machine code is extremely laborious. The language defined in the paper takes into account such hardware limitations as finite register width, finite memory capacity and the usage of modulo arithmetic instead of regular arithmetic. The paper contains an example of a program computing GCD of two non-negative integers, which is written in the proposed language. A proof of the program correctness is also included. Thus, the paper demonstrates that formal proofs could be conducted for programs in the language considering some hardware limitations, and the complexity of the proofs would be comparable to ones conducted for equivalent programs written in regular imperative languages, which do not take hardware limitations into account. Directions for future work include composing an algorithm for translation of the proposed language to machine code and proving the algorithm's correctness.

Keywords: Hoare logic, formal methods, axiomatic semantics.

Bibliography: 8 titles.

For citation:

Kovalev, D. Yu. 2025, "Hoare logic for an imperative language considering some hardware limitations", *Chebyshevskii sbornik*, vol. 26, no. 3, pp. 113–124.

1. Введение

Применение формальных методов позволяет делать суждения о корректности программ. Языки программирования, к программам на которых чаще всего применяют формальные методы, как правило, не учитывают аппаратных особенностей реальных вычислителей, таких как ограниченный объем памяти и ширина регистров. В то же время, для исполнения программы требуется ее трансляция в код вычислителя. Индуцированные во время этой трансляции ограничения и, возможно, допущенные ошибки могут сделать проведенное формальное доказательство неприменимым для программы, транслированной в код вычислителя.

Проведение доказательств для программ непосредственно в кодах вычислителя не является оптимальным решением этой проблемы. Существующие в большинстве наборов инструкций команды переходов плохо ложатся на концепцию императивного программирования и затрудняют определение даже таких базовых понятий, как программа: действительно, код программы в таком случае уже нельзя рассматривать, как непрерывную последовательность исполняющихся друг за другом команд языка, т.к. инструкции перехода изменяют последовательность исполнения команд. В случае с инструкциями непрямого перехода, когда целевой адрес перехода является неконстантным, решение этой проблемы становится еще более затруднительным. Это хорошо продемонстрировано в работе [5]: авторы были вынуждены ввести громоздкое и усложняющее анализ понятие позиционирующих функций (positioning functions), чтобы иметь возможность делать суждения о программах в кодах вычислителя.

Чтобы избежать указанных трудностей, разумнее построить императивный язык, учитывающий аппаратные ограничения вычислителя. Такой подход позволяет и упростить формальные доказательства за счет применения существующих подходов, и сохранить применимость этих доказательств к транслированному в инструкции вычислителя коду.

Описанный в этой статье язык основан на языке IMP, определенном в книге [1]. В качестве «элементарных» команд используется подмножество набора инструкций RV32I [2]. Кроме того, описанный язык позволяет переиспользовать код за счет поддержки процедур.

Для описания синтаксиса и семантики будет использоваться форма Бэкуса – Наура (БНФ) [6]. Нетерминальные символы БНФ будем обозначать **шрифтом без засечек**. Порождаемые символами БНФ множества будем обозначать каллиграфическим шрифтом ($\mathcal{X}, \mathcal{C}$ и т.д.) Связанные (binded) переменные будем обозначать готической буквой $\mathfrak{x}$ с натуральным индексом ($\mathfrak{x}_1, \mathfrak{x}_2, \dots$). Ячейки памяти и регистры будем обозначать **моноширинным** шрифтом ($\mathbf{x}_2, \mathbf{m}_{230}, \dots$).

2. Синтаксис

Определим множества k -битных целых чисел для $k \in \{5, 12\}$:

$$\mathcal{N}^{(k)} \stackrel{\text{def}}{=} \left\{ n^{(k)} : n^{(k)} \in \mathbb{Z} \wedge 0 \leq n^{(k)} < 2^k \right\}, \text{ где } \mathbb{Z} \text{ — множество всех целых чисел.}$$

Множества регистров общего назначения $\mathcal{X}$ и всех регистров $\mathcal{X}_0$:

$$\mathcal{X} \stackrel{\text{def}}{=} \{\mathbf{x}_2, \mathbf{x}_3, \dots, \mathbf{x}_{31}\}, \quad \mathcal{X}_0 \stackrel{\text{def}}{=} \{\mathbf{x}_0\} \cup \mathcal{X}$$

Регистр $\mathbf{x}_1$ в определении $\mathcal{X}$ опущен намеренно — как правило, его используют в качестве link register, т.е. для хранения адреса возврата [3]. Для упрощения последующей трансляции из описываемого императивного языка в язык ассемблера использовать этот регистр не будем.

Для $\mathcal{X}, \mathcal{X}_0, \mathcal{N}^{(k)}$ ($k \in \{5, 12\}$) определим соответствующие нетерминальные символы:

$$X ::= \mathbf{x}_2 \mid \mathbf{x}_3 \mid \mathbf{x}_4 \mid \dots \mid \mathbf{x}_{31}, \quad X_0 ::= X \mid \mathbf{x}_0, \quad N^{(k)} ::= 0 \mid 1 \mid 2 \mid \dots \mid 2^k - 1$$

Обозначим за P нетерминальный символ, порождающий имена процедур вида π_q , где q — непустая последовательность букв латинского алфавита. Порождаемое P множество имен процедур обозначим $\mathcal{P}$.

Нетерминальный символ для команд языка:

$$\begin{aligned} C ::= & \text{add } X, X_0, X_0 \mid \text{sub } X, X_0, X_0 \mid \text{addi } X, X_0, N^{(12)} \mid \text{slli } X, X_0, N^{(5)} \mid \\ & \text{ld4 } X, [X_0, N^{(12)}] \mid \text{st4 } X, [X_0, N^{(12)}] \mid C; C \mid \text{if } X_0 < X_0 \text{ then } C \text{ else } C \text{ fi} \mid \\ & \text{while } X_0 < X_0 \text{ do } C \text{ done} \mid \text{call } P \mid \text{skip} \end{aligned}$$

Обозначим за $\mathcal{C}$ множество команд, порожденных символом C . Если для $c \in \mathcal{C}$ присвоено имя $\pi_q \in \mathcal{P}$, будем называть c процедурой с именем π_s и писать $\pi_s \equiv c$. Программа — конечное непустое множество процедур Π с заданной главной процедурой $\text{main}(\Pi) \in \Pi$.

3. Семантика и логика Хоара

3.1. Выражения и формулы

Определим аксиоматическую семантику, ключевыми элементами которой являются выражения, формулы и тройки Хоара [4]. Аксиоматическая семантика, в отличие от операционной, позволяет делать суждения о программах в виде пред- и постусловий.

Определим множество ячеек памяти и соответствующий нетерминальный символ БНФ. Ячейки с адресами менее 2^{30} не включены намеренно, т.к. при трансляции в язык ассемблера предполагается их использовать для хранения кода программы.

$$\mathcal{M} = \{\mathbf{m}_{230}, \mathbf{m}_{230+4}, \mathbf{m}_{230+8}, \dots, \mathbf{m}_{232-4}\}, \quad M ::= \mathbf{m}_{230} \mid \mathbf{m}_{230+4} \mid \mathbf{m}_{230+8} \mid \dots \mid \mathbf{m}_{232-4}$$

Определим множества $\mathcal{N}^{(5)}, \mathcal{N}^{(12)}, \mathcal{X}, \mathcal{X}_0$ и нетерминальные символы $N^{(5)}, N^{(12)}, X, X_0$ так же, как при определении синтаксиса. Множество 32-битных чисел $\mathcal{N}^{(32)}$ определим аналогично $\mathcal{N}^{(5)}$ и $\mathcal{N}^{(12)}$, соответствующий нетерминальный символ $N^{(32)}$ — аналогично $N^{(5)}$ и $N^{(12)}$.

Определим символ, порождающий все натуральные числа: $N ::= 1 \mid 2 \mid 3 \mid \dots$. Символы $\mathfrak{s}_N$ будем называть связанными переменными. Формируемое ими множество обозначим $\mathcal{S}$, соответствующий нетерминальный символ — S .

Выпишем нетерминальный символ для выражений. Сущности $\oplus, \ominus, \ll, \text{sext}$ будем называть функциональными символами.

$$E ::= X_0 \mid N^{(32)} \mid E \oplus E \mid E \ominus E \mid E \ll N^{(5)} \mid \text{sext} \left(N^{(12)} \right) \mid M \mid S$$

Множество выражений, порождаемых E , обозначим $\mathcal{E}$.

Выпишем нетерминальный символ для формул. Сущности $\neg, \wedge, \vee, \rightarrow, \text{less}, \text{eq}$ будем называть предикатными символами; $\forall$ и $\exists$ — кванторами всеобщности и существования соответственно; $\top$ и $\perp$ — литералами, обозначающие логическую истину и ложь соответственно.

$$F ::= \top \mid \perp \mid \neg F \mid F \wedge F \mid F \vee F \mid F \rightarrow F \mid \text{less} (E, E) \mid \text{eq} (E, E) \mid \forall S F \mid \exists S F$$

Множество формул, порожденных символом F , обозначим как $\mathcal{F}$.

3.2. Правила преобразования формул

Семантику используемых в выражениях и формулах кванторов, предикатных и функциональных символов зададим через описанные ниже правила преобразования.

Используемые обозначения: $n_1 \in \mathcal{N}^{(32)}, n_2 \in \mathcal{N}^{(32)}, n \in \mathcal{N}^{(32)}, n^{(5)} \in \mathcal{N}^{(5)}, n^{(12)} \in \mathcal{N}^{(12)}, n^{(12)} = \sum_{i=0}^{11} \beta_i \cdot 2^i, \beta_i \in \{0, 1\}, e_1 \in \mathcal{E}, e_2 \in \mathcal{E}, e \in \mathcal{E}, f \in \mathcal{F}$.

Запись $f[e_1/e_2]$, обозначает формулу, полученную из формулы f заменой всех вхождений выражения e_2 выражением e_1 . Аналогично, запись $f[f_1/f_2]$, обозначает формулу, полученную из формулы f заменой всех вхождений формулы f_2 формулой f_1 .

$$f = f[\hat{n}/n_1 \oplus n_2] \wedge \text{eq}(\hat{n}, n_1 \oplus n_2), \text{ где } \hat{n} = n_1 + n_2 \pmod{2^{32}} \quad (\text{S1})$$

$$f = f[\hat{n}/n_1 \ominus n_2] \wedge \text{eq}(\hat{n}, n_1 \ominus n_2), \text{ где } \hat{n} = n_1 - n_2 \pmod{2^{32}} \quad (\text{S2})$$

$$f = f[\hat{n}/n \ll n^{(5)}] \wedge \text{eq}(\hat{n}, n \ll n^{(5)}), \text{ где } \hat{n} = n \cdot 2^{n^{(5)}} \pmod{2^{32}} \quad (\text{S3})$$

$$f = f[\hat{n}/\text{sext}(n^{(12)})] \wedge \text{eq}(\hat{n}, \text{sext}(n^{(12)})), \text{ где } \hat{n} = n^{(12)} + \beta_{11} \cdot \sum_{j=12}^{31} 2^j \quad (\text{S4})$$

$$f = f[\text{eq}(e_2, e_1)/\text{eq}(e_1, e_2)]; f = f[\top/\text{eq}(e, e)]; f = f[\perp/\text{eq}(n_1, n_2)], \text{ где } n_1 \neq n_2 \quad (\text{S5})$$

$$f \wedge \text{eq}(e_1, e_2) = f[e_2/e_1] \wedge f \wedge \text{eq}(e_1, e_2) \quad (\text{S6})$$

$$f \wedge \text{eq}(e_1, e_2) \wedge \text{eq}(e_2, e_3) = f \wedge \text{eq}(e_1, e_3) \wedge \text{eq}(e_2, e_3) \quad (\text{S7})$$

$$f = f[\neg/\text{less}(n_1, n_2)], \text{ где } n_1 < n_2; \quad f = f[\perp/\text{less}(n_1, n_2)], \text{ где } n_1 \geq n_2 \quad (\text{S8})$$

$$f = f[\neg \text{less}(e_2, e_1) \wedge \neg \text{eq}(e_1, e_2)/\text{less}(e_1, e_2)] \quad (\text{S9})$$

$$f \wedge \text{less}(e_1, e_2) = f \wedge \text{less}(e_1, e_2) \wedge \neg \text{eq}(e_2, 0) \quad (\text{S10})$$

$$\forall \mathfrak{s}_i f = f[0/\mathfrak{s}_i] \wedge f[1/\mathfrak{s}_i] \wedge \dots \wedge f[2^{32} - 1/\mathfrak{s}_i] \quad (\text{S11})$$

$$\exists \mathfrak{s}_i f = f[0/\mathfrak{s}_i] \vee f[1/\mathfrak{s}_i] \vee \dots \vee f[2^{32} - 1/\mathfrak{s}_i] \quad (\text{S12})$$

$$f = f[\top/\text{eq}(x_0, 0)] \quad (\text{S13})$$

Правила преобразования для предикатных символов отрицания ($\neg$), конъюнкции ($\wedge$), дизъюнкции ($\vee$), импликации ($\rightarrow$), такие как $f \wedge \top = f$, $\neg(f_1 \wedge f_2) = \neg f_1 \vee \neg f_2$, $f \rightarrow f = \top$ и др. будем считать известными из логики высказываний [7].

Если после раскрытия всех кванторов по правилам S11 и S12 в формуле остаются связанные переменные, она называется некорректной. Далее будем говорить только о корректных формулах.

3.3. Тройки Хоара, аксиомы и правила вывода

Тройками Хоара будем называть сущности вида $\{f_1\} \ c \ \{f_2\}$, где $c \in \mathcal{C}$, $f_1 \in \mathcal{F}$, $f_2 \in \mathcal{F}$ [4]. Формула f_1 называется предусловием, f_2 — постусловием. Для такой тройки будем говорить: если верно предусловие f_1 , то после исполнения команды c будет верно постусловие f_2 (если исполнение c завершилось¹). Логика Хоара состоит из правил вывода, записываемых в виде горизонтальной черты, сверху которой — предпосылки, снизу — следствие, и аксиом, являющихся правилами без предпосылок. Тройка называется верной, если она может быть выведена из аксиом и других верных троек через правила вывода, а также при преобразовании предусловия и/или постусловия другой верной тройки через правила, определенные в подразделе 3.2

В определениях аксиом будем считать $\mathbf{x}_i \in \mathcal{X}$, $\mathbf{x}_j \in \mathcal{X}_0$, $\mathbf{x}_k \in \mathcal{X}_0$, $i \neq j$, $i \neq k$. Для аксиом с предусловием вида $\dots \wedge f$ обозначим его за f' во избежание дублирования.

$$\{f\} \text{ skip } \{f\} \quad (\text{A14})$$

$$\{\text{eq}(\mathbf{x}_j, n_1) \wedge \text{eq}(\mathbf{x}_k, n_2) \wedge f\} \text{ add } \mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k \ \{\text{eq}(\mathbf{x}_i, n_1 \oplus n_2) \wedge f'\} \quad (\text{A15})$$

$$\{\text{eq}(\mathbf{x}_j, n_1) \wedge \text{eq}(\mathbf{x}_k, n_2) \wedge f\} \text{ sub } \mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k \ \{\text{eq}(\mathbf{x}_i, n_1 \ominus n_2) \wedge f'\} \quad (\text{A16})$$

$$\{\text{eq}(\mathbf{x}_j, n_1) \wedge f\} \text{ addi } \mathbf{x}_i, \mathbf{x}_j, n_2^{(12)} \ \{\text{eq}(\mathbf{x}_i, n_1 \oplus \text{sext}(n_2^{(12)})) \wedge f'\} \quad (\text{A17})$$

$$\{\text{eq}(\mathbf{x}_j, n_1) \wedge f\} \ \{\text{eq}(\mathbf{x}_i, n_1 \ll n_2^{(5)}) \wedge f'\} \quad (\text{A18})$$

$$\begin{aligned} & \left\{ \text{eq}(\mathbf{x}_j, n_1) \wedge \text{eq}\left(n_3, n_1 \oplus \text{sext}\left(n_2^{(12)}\right)\right) \wedge \text{eq}(\mathbf{m}_{n_3}, n_4) \wedge f \right\} \\ & \text{ld4 } \mathbf{x}_i, \left[\mathbf{x}_j, n_2^{(12)} \right] \ \{ \mathbf{x}_i = n_4 \wedge f' \} \end{aligned} \quad (\text{A19})$$

$$\begin{aligned} & \left\{ \text{eq}(\mathbf{x}_i, n_1) \wedge \text{eq}(\mathbf{x}_j, n_2) \wedge \text{eq}\left(n_4, n_2 \oplus \text{sext}\left(n_3^{(12)}\right)\right) \wedge f \right\} \\ & \text{st4 } \mathbf{x}_i, \left[\mathbf{x}_j, n_3^{(12)} \right] \ \{ \mathbf{m}_{n_4} = n_1 \wedge f' \} \end{aligned} \quad (\text{A20})$$

Правила вывода:

$$\frac{\{f_1\} \ c \ \{f_2\}}{\{f_1\} \ \text{call } \pi_q \ \{f_2\}}, \text{ где } \pi_q \equiv c, \ \pi_q \in \Pi \quad (\text{R21})$$

$$\frac{f'_1 \rightarrow f_1, \ \{f_1\} \ c \ \{f_2\}, \ f_2 \rightarrow f'_2}{\{f'_1\} \ c \ \{f'_2\}} \quad (\text{R22})$$

$$\frac{\{f_1\} \ c_1 \ \{f_2\}, \ \{f_2\} \ c_2 \ \{f_3\}}{\{f_1\} \ c_1; c_2 \ \{f_3\}} \quad (\text{R23})$$

$$\frac{\{\text{less}(\mathbf{x}_i, \mathbf{x}_j) \wedge f_1\} \ c_1 \ \{f_2\}, \ \{\neg \text{less}(\mathbf{x}_i, \mathbf{x}_j) \wedge f_1\} \ c_2 \ \{f_2\}}{\{f_1\} \ \text{if } \mathbf{x}_i < \mathbf{x}_j \ \text{then } c_1 \ \text{else } c_2 \ \text{fi } \{f_2\}} \quad (\text{R24})$$

$$\frac{\{\text{less}(\mathbf{x}_i, \mathbf{x}_j) \wedge f_1\} \ c \ \{f_1\}}{\{f_1\} \ \text{while } \mathbf{x}_i < \mathbf{x}_j \ \text{do } c \ \text{done } \{\neg \text{less}(\mathbf{x}_i, \mathbf{x}_j) \wedge f_1\}} \quad (\text{R25})$$

Запись $\{f_1\} \ \Pi \ \{f_2\}$ будем считать эквивалентной $\{f_1\} \ \text{main}(\Pi) \ \{f_2\}$.

¹Понятие завершенности определяется через операционную семантику и здесь опущено.

Не к каждой синтаксически корректной команде можно применить соответствующую аксиому или правило вывода. Так, если в программе нет процедуры π_{name} , не получится построить верную тройку Хоара вида $\{f_1\} \text{ call } \pi_{\text{name}} \{f_2\}$.

Отметим, что в правиле R25 формулу f_1 часто называют инвариантом цикла. Его нахождение — наиболее нетривиальная из задач, стоящих при формальной верификации императивных программ. Остальные действия, как правила, достаточно хорошо автоматизируются.

Заданная логика обладает важным ограничением: она позволяет доказать лишь частичную корректность (partial correctness) программ, а именно, нет способа доказать их завершимость. Для доказательства полной корректности (total correctness) необходимо определить соответствующее правило вывода для циклов, учитывающее их завершимость [8]. Указанное ограничение не является принципиальным, но в рамках данной работы соответствующее правило и пример доказательства полной корректности опускаются для упрощения.

4. Пример

Приведем пример программы Π , вычисляющей $\text{НОД}(a, b)$, $a \in \mathcal{N}^{(32)}, b \in \mathcal{N}^{(32)}, a \geq b, a \neq 0$; a и b хранятся перед началом работы программы в ячейках памяти с адресами 2^{30} и $2^{30} + 4$ соответственно. После работы, программа сохраняет $\text{НОД}(a, b)$ в ячейку памяти с адресом $2^{30} + 8$.

Расширим понятие выражений для поддержки доменной области этой задачи. Добавим в определение $\mathbf{E}$ символы вида $\text{НОД}(\mathbf{E}, \mathbf{E})$ и $\mathbf{E} - \mathbf{E}$, где НОД и « $-$ » — функциональные символы.

Определим семантику этих функциональных символов, выразив в терминах правил преобразования формул известные математические свойства соответствующих операций:

$$f \wedge \neg \text{less}(e_1, e_2) = f[e_1 - e_2 / e_1 \ominus e_2] \wedge \neg \text{less}(e_1, e_2) \quad (\text{S26})$$

$$f = f[\text{НОД}(e_1, e_2) / \text{НОД}(e_1 - e_2, e_2)] \quad (\text{S27})$$

$$f = f[\text{НОД}(e_1, e_2) / \text{НОД}(e_2, e_1)] \quad (\text{S28})$$

$$f \wedge \neg \text{eq}(e, 0) = f[e / \text{НОД}(e, 0)] \wedge \neg \text{eq}(e, 0) \quad (\text{S29})$$

Формально требования к программе можно описать тройкой Хоара $\{f_1\} \Pi \{f_2\}$, где

$$\begin{aligned} f_1 &= \text{eq}(\mathbf{m}_{2^{30}}, a) \wedge \text{eq}(\mathbf{m}_{2^{30}+4}, b) \wedge \neg \text{less}(a, b) \wedge \neg \text{eq}(a, 0) \\ f_2 &= f_1 \wedge \text{eq}(\mathbf{m}_{2^{30}+8}, \text{НОД}(a, b)) \end{aligned}$$

Программа $\Pi = \{\pi_{\text{main}}, \pi_{\text{iter}}\}$, $\text{main}(\Pi) = \pi_{\text{main}}$:

$\pi_{\text{main}} \equiv \text{addi } \mathbf{x}_2, \mathbf{x}_0, 2^7; \text{slli } \mathbf{x}_3, \mathbf{x}_2, 23; \text{ld4 } \mathbf{x}_4, [\mathbf{x}_3, 0]; \text{ld4 } \mathbf{x}_5, [\mathbf{x}_3, 4];$

$\text{while } \mathbf{x}_0 < \mathbf{x}_5 \text{ do call } \pi_{\text{iter}} \text{ done; st4 } \mathbf{x}_4, [\mathbf{x}_3, 8]$

$\pi_{\text{iter}} \equiv \text{sub } \mathbf{x}_6, \mathbf{x}_4, \mathbf{x}_5; \text{if } \mathbf{x}_5 < \mathbf{x}_6 \text{ then addi } \mathbf{x}_4, \mathbf{x}_6, 0 \text{ else addi } \mathbf{x}_4, \mathbf{x}_5, 0; \text{addi } \mathbf{x}_5, \mathbf{x}_6, 0 \text{ fi}$

Докажем $\{f_1\} \Pi \{f_2\}$.

5. Доказательство

Доказательство будет разделено на несколько подсекций для удобства восприятия. Для краткости, ссылки на используемые правила иногда будем записывать в прямоугольниках по месту их применения, например: $\text{eq}(\mathbf{x}_2, 5 \oplus 7) = \boxed{\text{S1}} = \text{eq}(\mathbf{x}_2, 12)$.

5.1. Загрузка входных данных из памяти

1. **Предусловие:** $f_1 = f_1 \wedge \top = \boxed{\text{S13}} = f_1 \wedge \text{eq}(x_0, 0)$
Команда: $c_1 = \text{addi } x_2, x_0, 2^7$, **аксиома:** A17, **постусловие:**
 $f_3 = \text{eq}(x_2, 0 \oplus \text{sext}(2^7)) \wedge \text{eq}(x_0, 0) \wedge f_1 = \boxed{\text{S1, S4}} = \text{eq}(x_2, 2^7) \wedge \text{eq}(x_0, 0) \wedge f_1$
Вывод: $\{f_1\} \ c_1 \ \{f_3\}$
2. **Предусловие:** $f_3 = \text{eq}(x_2, 2^7) \wedge \text{eq}(x_0, 0) \wedge f_1$
Команда: $c_2 = \text{slli } x_3, x_2, 23$, **аксиома:** A18
Постусловие: $f_4 = \text{eq}(x_3, 2^7 \ll 23) \wedge f_3 = \boxed{\text{S3}} = \text{eq}(x_3, 2^{30}) \wedge f_3$, **вывод:** $\{f_3\} \ c_2 \ \{f_4\}$

3. Покажем, что $f_4 \rightarrow f_5$, где $f_5 = f_4 \wedge \text{eq}(2^{30}, 2^{30} \oplus \text{sext}(0))$:

$$\begin{aligned} f_4 &= f_4 \wedge \top = \boxed{\text{S5}} = f_4 \wedge \text{eq}(2^{30}, 2^{30}) = \boxed{\text{S1}} = \\ &= f_4 \wedge \text{eq}(2^{30}, 2^{30} \oplus 0) = \boxed{\text{S4}} = f_4 \wedge \text{eq}(2^{30}, 2^{30} \oplus \text{sext}(0)) = f_5 \end{aligned}$$

Значит, $f_4 \rightarrow f_5$.

4. **Предпосылки:** $f_3 \rightarrow f_3, \{f_3\} \ c_2 \ \{f_4\}, f_4 \rightarrow f_5$, **правило:** R22, **вывод:** $\{f_3\} \ c_2 \ \{f_5\}$
5. **Предусловие:** $f_5 = \text{eq}(x_3, 2^{30}) \wedge \text{eq}(m_{2^{30}}, a) \wedge \text{eq}(2^{30}, 2^{30} \oplus \text{sext}(0)) \wedge \dots$
Команда: $c_3 = \text{ld4 } x_4, [x_3, 0]$, **аксиома:** A19, **постусловие:** $f_6 = \text{eq}(x_4, a) \wedge f_5$
Вывод: $\{f_5\} \ c_3 \ \{f_6\}$
6. Покажем, что $f_6 \rightarrow f_7$, где $f_7 = f_6 \wedge \text{eq}(2^{30} + 4, 2^{30} \oplus \text{sext}(4))$:

$$\begin{aligned} f_6 &= f_6 \wedge \top = \boxed{\text{S5}} = f_6 \wedge \text{eq}(2^{30} + 4, 2^{30} + 4) = \boxed{\text{S1}} = \\ &= f_6 \wedge \text{eq}(2^{30} + 4, 2^{30} \oplus 4) = \boxed{\text{S4}} = f_6 \wedge \text{eq}(2^{30} + 4, 2^{30} \oplus \text{sext}(4)) = f_7 \end{aligned}$$

Значит, $f_6 \rightarrow f_7$.

7. **Предпосылки:** $f_5 \rightarrow f_5, \{f_5\} \ c_3 \ \{f_6\}, f_6 \rightarrow f_7$, **правило:** R22, **вывод:** $\{f_5\} \ c_3 \ \{f_7\}$
8. **Предусловие:** $f_7 = \text{eq}(x_3, 2^{30}) \wedge \text{eq}(m_{2^{30}+4}, b) \wedge \text{eq}(2^{30} + 4, 2^{30} \oplus \text{sext}(4)) \wedge \dots$
Команда: $c_4 = \text{ld4 } x_5, [x_3, 4]$, **аксиома:** A19, **постусловие:** $f_8 = \text{eq}(x_5, b) \wedge f_7$
Вывод: $\{f_7\} \ c_4 \ \{f_8\}$

5.2. Вывод инварианта цикла из предусловия цикла

9. Учитывая $f_8 = \text{eq}(x_4, a) \wedge \text{eq}(x_5, b) \wedge \neg \text{less}(a, b) \wedge \dots$ и правила S5, S6, S7, S12, получаем $f_8 \rightarrow f_9 \rightarrow f_{10} \rightarrow f_{11}$, где

$$\begin{aligned} f_9 &= f_8 \wedge \neg \text{less}(x_4, x_5) \wedge \neg \text{eq}(x_4, 0) \wedge \text{eq}(\text{НОД}(x_4, x_5), \text{НОД}(a, b)) \\ f_{10} &= f_4 \wedge \neg \text{less}(x_4, x_5) \wedge \neg \text{eq}(x_4, 0) \wedge \text{eq}(\text{НОД}(x_4, x_5), \text{НОД}(a, b)) \\ f_{11} &= f_4 \wedge \exists s_1 \exists s_2 : \left(\text{eq}(x_4, s_1) \wedge \text{eq}(x_5, s_2) \wedge \neg \text{less}(s_1, s_2) \wedge \right. \\ &\quad \left. \neg \text{eq}(s_1, 0) \wedge \text{eq}(\text{НОД}(s_1, s_2), \text{НОД}(a, b)) \right) \end{aligned}$$

10. **Предпосылки:** $f_7 \rightarrow f_7, \{f_7\} \ c_4 \ \{f_8\}, f_8 \rightarrow f_{11}$, **правило:** R22 **вывод:** $\{f_7\} \ c_4 \ \{f_{11}\}$

11. Введем обозначение

$$\begin{aligned} f_{12} &= \text{less}(x_0, x_5) \wedge f_{11} = \boxed{\text{S12}} = \\ &= f_4 \wedge \exists s_1 \exists s_2 : \left(\text{eq}(x_4, s_1) \wedge \text{eq}(x_5, s_2) \wedge \neg \text{less}(s_1, s_2) \wedge \neg \text{eq}(s_1, 0) \wedge \right. \\ &\quad \left. \wedge \text{eq}(\text{НОД}(s_1, s_2), \text{НОД}(a, b)) \wedge \text{less}(0, s_2) \right) \end{aligned}$$

f_{11} является инвариантом цикла, f_{12} — конъюнкцией инварианта цикла и его условия.

5.3. Доказательство сохранения инварианта цикла (начало)

12. **Предусловие:** $f_{12} = \exists s_1 \exists s_2 : \left(\text{eq}(x_4, s_1) \wedge \text{eq}(x_5, s_2) \wedge \dots \right) \wedge \dots$

Команда: $c_5 = \text{sub } x_6, x_4, x_5$, **аксиома:** A16, **правило:** S12

Постусловие: $f_{13} = \exists s_1 \exists s_2 : \left(\text{eq}(x_6, s_1 \ominus s_2) \wedge \text{eq}(x_4, s_1) \wedge \text{eq}(x_5, s_2) \wedge \dots \right) \wedge \dots$

Вывод: $\{f_{12}\} \ c_5 \ \{f_{13}\}$

13. Учитывая правила S26 и S27, получаем $f_{13} \rightarrow f_{14} \rightarrow f_{15}$, где

$$\begin{aligned} f_{14} &= f_4 \wedge \exists s_1 \exists s_2 : \left(\text{eq}(x_6, s_1 - s_2) \wedge \text{eq}(x_4, s_1) \wedge \text{eq}(x_5, s_2) \wedge \neg \text{less}(s_1, s_2) \wedge \right. \\ &\quad \left. \neg \text{eq}(s_1, 0) \wedge \text{eq}(\text{НОД}(s_1, s_2), \text{НОД}(a, b)) \wedge \text{less}(0, s_2) \right) \\ f_{15} &= f_4 \wedge \exists s_1 \exists s_2 : \left(\text{eq}(x_6, s_1 - s_2) \wedge \text{eq}(x_4, s_1) \wedge \text{eq}(x_5, s_2) \wedge \neg \text{less}(s_1, s_2) \wedge \right. \\ &\quad \left. \neg \text{eq}(s_1, 0) \wedge \text{eq}(\text{НОД}(s_1 - s_2, s_2), \text{НОД}(a, b)) \wedge \text{less}(0, s_2) \right) \end{aligned}$$

14. **Предпосылки:** $f_{12} \rightarrow f_{12}$, $\{f_{12}\} \ c_5 \ \{f_{13}\}$, $f_{13} \rightarrow f_{15}$ **правило:** R22

Вывод: $\{f_{12}\} \ c_5 \ \{f_{15}\}$

5.3.1. Доказательство для ветки if условного оператора

15. **Предусловие:**

$$\begin{aligned} f_{16} &= f_{15} \wedge \text{less}(x_5, x_6) = \boxed{\text{S12}} = \\ &= f_4 \wedge \exists s_1 \exists s_2 : \left(\text{eq}(x_6, s_1 - s_2) \wedge \text{eq}(x_4, s_1) \wedge \text{eq}(x_5, s_2) \wedge \neg \text{less}(s_1, s_2) \wedge \neg \text{eq}(s_1, 0) \wedge \right. \\ &\quad \left. \text{eq}(\text{НОД}(s_1 - s_2, s_2), \text{НОД}(a, b)) \wedge \text{less}(s_2, s_1 - s_2) \wedge \text{less}(0, s_2) \right) \end{aligned}$$

Команда: $c_6 = \text{addi } x_4, x_6, 0$, **аксиома:** A17, **правило:** S12

Постусловие:

$$\begin{aligned} f_{17} &= f_4 \wedge \exists s_1 \exists s_2 : \left(\text{eq}(x_6, s_1 - s_2) \wedge \text{eq}(x_4, s_1 - s_2) \wedge \text{eq}(x_5, s_2) \wedge \right. \\ &\quad \left. \neg \text{less}(s_1, s_2) \wedge \neg \text{eq}(s_1, 0) \wedge \text{less}(s_2, s_1 - s_2) \wedge \right. \\ &\quad \left. \text{eq}(\text{НОД}(s_1 - s_2, s_2), \text{НОД}(a, b)) \wedge \text{less}(0, s_2) \right) \end{aligned}$$

Вывод: $\{f_{16}\} \ c_6 \ \{f_{17}\}$

16. Согласно правилам S9, S10, S12, получаем $f_{17} \rightarrow f_{18}$, где

$$\begin{aligned} f_{18} = f_4 \wedge \exists s_1 \exists s_2 : & \left(\text{eq}(x_4, s_1 - s_2) \wedge \text{eq}(x_5, s_2) \wedge \neg \text{less}(s_1 - s_2, s_2) \wedge \right. \\ & \left. \neg \text{eq}(s_1 - s_2, 0) \wedge \text{eq}(\text{НОД}(s_1 - s_2, s_2), \text{НОД}(a, b)) \right) = \\ f_4 \wedge \exists s_1 \exists s_2 : & \left(\text{eq}(x_4, s_1) \wedge \text{eq}(x_5, s_2) \wedge \neg \text{less}(s_1, s_2) \wedge \right. \\ & \left. \neg \text{eq}(s_1, 0) \wedge \text{eq}(\text{НОД}(s_1, s_2), \text{НОД}(a, b)) \right) \end{aligned}$$

Получили, что f_{18} совпадает с f_{11} , следовательно, $f_{17} \rightarrow f_{11}$.

17. **Предпосылки:** $f_{16} \rightarrow f_{16}$, $\{f_{16}\} \vdash \{f_{17}\}$, $f_{17} \rightarrow f_{11}$, **правило:** R22

Вывод: $\{f_{16}\} \vdash \{f_{11}\}$

5.3.2. Доказательство для ветки else условного оператора

18. **Предусловие:**

$$\begin{aligned} f_{19} = f_{15} \wedge \neg \text{less}(x_5, x_6) = \boxed{\text{S12}} = \\ f_4 \wedge \exists s_1 \exists s_2 : & \left(\text{eq}(x_6, s_1 - s_2) \wedge \text{eq}(x_4, s_1) \wedge \text{eq}(x_5, s_2) \wedge \neg \text{less}(s_1, s_2) \wedge \neg \text{eq}(s_1, 0) \wedge \right. \\ & \left. \text{eq}(\text{НОД}(s_1 - s_2, s_2), \text{НОД}(a, b)) \wedge \neg \text{less}(s_2, s_1 - s_2) \wedge \text{less}(0, s_2) \right) \end{aligned}$$

Команда: $c_7 = \text{addi } x_4, x_5, 0$, **аксиома:** A17, **правила:** S1, S12

Постусловие:

$$\begin{aligned} f_{20} = f_4 \wedge \exists s_1 \exists s_2 : & \left(\text{eq}(x_6, s_1 - s_2) \wedge \text{eq}(x_4, s_2) \wedge \text{eq}(x_5, s_2) \wedge \neg \text{less}(s_1, s_2) \wedge \neg \text{eq}(s_1, 0) \wedge \right. \\ & \left. \text{eq}(\text{НОД}(s_1 - s_2, s_2), \text{НОД}(a, b)) \wedge \neg \text{less}(s_2, s_1 - s_2) \wedge \text{less}(0, s_2) \right) \end{aligned}$$

Вывод: $\{f_{19}\} \vdash \{f_{20}\}$

19. **Предусловие:** f_{20} , **команда:** $c_8 = \text{addi } x_5, x_6, 0$, **аксиома:** A17

Правила: S1, S12, **постусловие:**

$$\begin{aligned} f_{21} = f_4 \wedge \exists s_1 \exists s_2 : & \left(\text{eq}(x_6, s_1 - s_2) \wedge \text{eq}(x_4, s_2) \wedge \text{eq}(x_5, s_1 - s_2) \wedge \right. \\ & \neg \text{less}(s_1, s_2) \wedge \neg \text{eq}(s_1, 0) \wedge \neg \text{less}(s_2, s_1 - s_2) \wedge \\ & \left. \text{eq}(\text{НОД}(s_1 - s_2, s_2), \text{НОД}(a, b)) \wedge \text{less}(0, s_2) \right) \end{aligned}$$

Вывод: $\{f_{20}\} \vdash \{f_{21}\}$

20. Согласно правилам S10, S12, получаем $f_{21} \rightarrow f_{22} \rightarrow f_{23}$, где

$$\begin{aligned} f_{22} = f_4 \wedge \exists s_1 \exists s_2 : & \left(\text{eq}(x_4, s_2) \wedge \text{eq}(x_5, s_1 - s_2) \wedge \neg \text{less}(s_2, s_1 - s_2) \wedge \right. \\ & \left. \text{less}(0, s_2) \wedge \text{eq}(\text{НОД}(s_1 - s_2, s_2), \text{НОД}(a, b)) \right) \\ f_{23} = f_4 \wedge \exists s_1 \exists s_2 : & \left(\text{eq}(x_4, s_2) \wedge \text{eq}(x_5, s_1 - s_2) \wedge \neg \text{less}(s_2, s_1 - s_2) \wedge \right. \\ & \neg \text{eq}(s_2, 0) \wedge \text{eq}(\text{НОД}(s_1 - s_2, s_2), \text{НОД}(a, b)) \right) = \\ f_4 \wedge \exists s_1 \exists s_2 : & \left(\text{eq}(x_4, s_1) \wedge \text{eq}(x_5, s_2) \wedge \neg \text{less}(s_1, s_2) \wedge \right. \\ & \left. \neg \text{eq}(s_1, 0) \wedge \text{eq}(\text{НОД}(s_1, s_2), \text{НОД}(a, b)) \right) \end{aligned}$$

Получили, что f_{23} совпадает с f_{11} , следовательно, $f_{21} \rightarrow f_{11}$.

21. **Предпосылки:** $f_{20} \rightarrow f_{20}, \{f_{20}\} \ c_8 \ \{f_{21}\}, f_{21} \rightarrow f_{11}$, **правило:** R22
Вывод: $\{f_{20}\} \ c_8 \ \{f_{11}\}$

5.3.3. Доказательство сохранения инварианта цикла (продолжение)

22. **Предпосылки:** $\{f_{19}\} \ c_7 \ \{f_{20}\}, \{f_{20}\} \ c_8 \ \{f_{11}\}$, **правило:** R23, **вывод:** $\{f_{19}\} \ c_7; c_8 \ \{f_{11}\}$
 23. **Предпосылки:** $\{f_{16}\} \ c_6 \ \{f_{11}\}, \{f_{19}\} \ c_7; c_8 \ \{f_{11}\}$, где

$$f_{16} = f_{15} \wedge \text{less}(x_5, x_6), \quad f_{19} = f_{15} \wedge \neg \text{less}(x_5, x_6)$$

Правило: R24, **вывод:** $\{f_{15}\} \ c_9 \ \{f_{11}\}$, где $c_9 = \text{if less}(x_5, x_6) \text{ then } c_6 \text{ else } c_7; c_8 \text{ endif}$

24. **Предпосылки:** $\{f_{12}\} \ c_5 \ \{f_{15}\}, \{f_{15}\} \ c_9 \ \{f_{11}\}$, **правило:** R23, **вывод:** $\{f_{12}\} \ c_5; c_9 \ \{f_{11}\}$
 25. **Предпосылки:** $\{f_{12}\} \ c_5; c_9 \ \{f_{11}\}, \pi_{\text{iter}} = c_5; c_9$, **правило:** R21
Вывод: $\{f_{12}\} \ \text{call } \pi_{\text{iter}} \ \{f_{11}\}$
 26. **Предпосылки:** $\{f_{12}\} \ \text{call } \pi_{\text{iter}} \ \{f_{11}\}$, где $f_{12} = \text{less}(x_0, x_5) \wedge f_{11}$, **правило:** R25
Вывод: $\{f_{11}\} \ c_{10} \ \{f_{24}\}$, где

$$c_{10} = \text{while less}(x_0, x_5) \text{ do call } \pi_{\text{iter}} \text{ done}; \quad f_{24} = f_{11} \wedge \neg \text{less}(x_0, x_5)$$

Таким образом, f_{11} действительно является инвариантом цикла c_{10} .

5.4. Сохранение результата в память

27. **Предпосылки:** $\{f_1\} \ c_1 \ \{f_3\}, \{f_3\} \ c_2 \ \{f_5\}, \{f_5\} \ c_3 \ \{f_7\}, \{f_7\} \ c_4 \ \{f_{11}\}, \{f_{11}\} \ c_{10} \ \{f_{24}\}$
Правило: R23, **вывод:** $\{f_1\} \ c_{11} \ \{f_{24}\}$, где $c_{11} = c_1; c_2; c_3; c_4; c_{10}$
 28. Согласно правилам S5, S9, S12, S13, S29, получаем, что $f_{24} \rightarrow f_{25} \rightarrow f_{26} \rightarrow f_{27} \rightarrow f_{28}$, где

$$\begin{aligned} f_{25} &= f_4 \wedge \neg \text{less}(x_4, x_5) \wedge \neg \text{eq}(x_4, 0) \wedge \text{eq}(\text{НОД}(x_4, x_5), \text{НОД}(a, b)) \wedge \neg \text{less}(x_0, x_5) \\ f_{26} &= f_4 \wedge \text{eq}(\text{НОД}(x_4, x_5), \text{НОД}(a, b)) \wedge \text{eq}(x_0, x_5) \wedge \neg \text{eq}(x_4, 0) \\ f_{27} &= f_4 \wedge \text{eq}(\text{НОД}(x_4, 0), \text{НОД}(a, b)) \wedge \neg \text{eq}(x_4, 0) \\ f_{28} &= f_4 \wedge \text{eq}(x_4, \text{НОД}(a, b)) \end{aligned}$$

29. Покажем, что $f_{28} \rightarrow f_{29}$, где $f_{29} = f_{28} \wedge \text{eq}(2^{30} + 8, 2^{30} \oplus \text{sext}(8))$:

$$\begin{aligned} f_{28} &= f_{28} \wedge \top = \boxed{\text{S5}} = f_{28} \wedge \text{eq}(2^{30} + 8, 2^{30} + 8) = \boxed{\text{S1}} = \\ &= f_{28} \wedge \text{eq}(2^{30} + 8, 2^{30} \oplus 8) = \boxed{\text{S4}} = f_{28} \wedge \text{eq}(2^{30} + 8, 2^{30} \oplus \text{sext}(8)) = f_{29} \end{aligned}$$

Значит, $f_{28} \rightarrow f_{29}$.

30. **Предпосылки:** $f_1 \rightarrow f_1, \{f_1\} \ c_{11} \ \{f_{24}\}, f_{24} \rightarrow f_{28} \rightarrow f_{29}$, **правило:** R22
Вывод: $\{f_1\} \ c_{11} \ \{f_{29}\}$
 31. **Предусловие:** $f_{29} = \text{eq}(x_4, \text{НОД}(a, b)) \wedge \text{eq}(2^{30} + 8, 2^{30} \oplus \text{sext}(8)) \text{eq}(x_3, 2^{30}) \wedge \dots$
Команда: $c_{12} = \text{st4 } x_4, [x_3, 8]$, **аксиома:** A20
Постусловие: $f_{30} = f_{29} \wedge \text{eq}(m_{2^{30}+8}, \text{НОД}(a, b))$, **вывод:** $\{f_{29}\} \ c_{12} \ \{f_{30}\}$
 32. Получили, что f_{30} имеет вид $f_2 \wedge \dots$, следовательно, $f_{30} \rightarrow f_2$.
 33. **Предпосылки:** $f_{29} \rightarrow f_{29}, \{f_{29}\} \ c_{12} \ \{f_{30}\}, f_{30} \rightarrow f_2$, **правило:** R22
Вывод: $\{f_{29}\} \ c_{12} \ \{f_2\}$

34. **Предпосылки:** $\{f_1\} \text{ } c_{11} \{f_{29}\}, \{f_{29}\} \text{ } c_{12} \{f_2\}$, **правило:** R23, **вывод:** $\{f_1\} \text{ } c_{11}; c_{12} \{f_2\}$.

35. Т.к. $\pi_{\text{main}} = c_{11}; c_{12}$, получаем $\{f_1\} \text{ } \Pi \{f_2\}$. Искомая тройка доказана.

Отметим, что, несмотря на относительно высокую кажущуюся длину доказательства, практически все его шаги являются тривиальными и легко автоматизируются, т.к. они основаны на применении простых формально описанных аксиом и правил. Нетривиальным является лишь нахождение инварианта цикла.

6. Заключение

Предыдущие работы других авторов о формальной верификации с учетом аппаратных ограничений вычислителя концентрировались на описании логики Хоара для языка ассемблера или попытках по программе на ассемблере восстановить «высокоуровневую» логику работы программы. Такой подход позволяет проводить верификацию уже существующих программ, но является трудозатратным.

В этой работе описан «промежуточный» между IMP и ассемблером язык, позволяющий проводить формальные доказательства с учетом некоторых аппаратных ограничений и с трудозатратами, сопоставимыми доказательствам для программ на IMP. Описание языка включает в себя определение его синтаксиса и аксиоматической семантики (логики Хоара). Приведен пример программы вычисления НОД двух целых неотрицательных чисел, написанной на этом языке. Проведено формальное доказательство корректности этой программы (доказана верность искомой тройки Хоара, описывающей ожидаемый результат).

Ограничением использования описанной логики, очевидно, является ее применимость только для тех программ, что написаны на заданном в этой работе языке. Это ограничение — цена за упрощение доказательств в сравнении с доказательствами для программ в машинных кодах.

Направления для дальнейшей работы включают в себя описание алгоритма трансляции из предложенного языка в машинные коды и доказательство корректности этого алгоритма, а также расширение логики для возможности доказательства полной, а не только частичной корректности программ.

СПИСОК ЦИТИРОВАННОЙ ЛИТЕРАТУРЫ

1. Winskel G. The Formal Semantics of Programming Languages. The MIT Press, Massachusetts. P. 11–26. ISBN 0262231697.
2. RISC-V Foundation, The RISC-V Instruction Set Manual Volume I Unprivileged Architecture Version 20240411. 2024. URL: <https://github.com/riscv/riscv-isa-manual/releases/download/20240411/unpriv-isa-asciidoc.pdf> (дата обращения: 23 февраля 2025).
3. RISC-V Foundation, RISC-V ABIs Specification Version 1.0: Ratified. 2022. URL: <https://github.com/riscv-non-isa/riscv-elf-psabi-doc/releases/download/v1.0/riscv-abi.pdf> (дата обращения: 23 февраля 2025).
4. Hoare C. A. R. An axiomatic basis for computer programming // Communications of the ACM. 1969. vol. 12, issue 10. P. 576–580. doi: 10.1145/363235.363259. URL: <https://dl.acm.org/doi/pdf/10.1145/363235.363259> (дата обращения: 23 февраля 2025).
5. Myreen M. O., Gordon M. J. C. Hoare Logic for Realistically Modelled Machine Code // Proceedings of the 13th International Conference on Tools and Algorithms for the Construction and Analysis of Systems. 2007. Springer-Verlag, Berlin, Heidelberg. P.568–582. doi:

- 10.1007/978-3-540-71209-1_44. URL: https://link.springer.com/content/pdf/10.1007/978-3-540-71209-1_44.pdf (дата обращения: 23 февраля 2025).
6. McCracken D. D., Reilly E. D. Backus-Naur form (BNF) // In *Encyclopedia of Computer Science*. 2003. John Wiley and Sons Ltd. P. 129–131. ISBN 0470864125. URL: <https://dl.acm.org/doi/pdf/10.5555/1074100.1074155> (дата обращения: 23 февраля 2025).
7. Войшвилло Е. К., Дегтярев М. Г. Логика. 2001. ВЛАДОСС-ПРЕСС, Москва, С. 91–99. ISBN 5-305-00001-7.
8. Jacobs D., Gries D. General correctness: a unification of partial and total correctness // *Acta Inf.* 1985. vol. 22, num. 1, P. 67–83. doi: 10.1007/BF00290146.

REFERENCES

1. Winskel G 1993, *The Formal Semantics of Programming Languages*, The MIT Press, Massachusetts, pp. 11–26. ISBN 0262231697.
2. RISC-V Foundation, The RISC-V Instruction Set Manual Volume I Unprivileged Architecture Version 20240411 (2024) Available at: <https://github.com/riscv/riscv-isa-manual/releases/download/20240411/unpriv-isa-asciidoc.pdf> (accessed 23 February 2025).
3. RISC-V Foundation, RISC-V ABIs Specification Version 1.0: Ratified (2022), Available at: <https://github.com/riscv-non-isa/riscv-elf-psabi-doc/releases/download/v1.0/riscv-abi.pdf> (accessed 23 February 2025).
4. Hoare C A R 1969, “An axiomatic basis for computer programming”, *Communications of the ACM*, vol. 12, issue 10. pp. 576–580. doi: 10.1145/363235.363259. Available at: <https://dl.acm.org/doi/pdf/10.1145/363235.363259> (accessed 23 February 2025).
5. Myreen M O, Gordon M J C “Hoare Logic for Realistically Modelled Machine Code” *Proceedings of the 13th International Conference on Tools and Algorithms for the Construction and Analysis of Systems*. Springer-Verlag, Berlin, Heidelberg, 2007, pp.568–582. doi: 10.1007/978-3-540-71209-1_44. Available at: https://link.springer.com/content/pdf/10.1007/978-3-540-71209-1_44.pdf (accessed 23 February 2025).
6. McCracken D D, Reilly E D 2003, “Backus-Naur form (BNF)” *In Encyclopedia of Computer Science*. John Wiley and Sons Ltd. pp. 129–131. ISBN 0470864125. Available at: <https://dl.acm.org/doi/pdf/10.5555/1074100.1074155> (accessed 23 February 2025).
7. Voishvillo E K, Degtyarev M G 2001 *Logika* [Logic]. Vladoss-Press, Moscow, pp. 91–99. ISBN 5-305-00001-7.
8. Jacobs D, Gries D 1985, “General correctness: a unification of partial and total correctness” *Acta Inf.*, vol. 22, num. 1, pp. 67–83. doi: 10.1007/BF00290146.

Получено: 24.02.2025

Принято в печать: 27.08.2025