ВЕСТНИК УДМУРТСКОГО УНИВЕРСИТЕТА. МАТЕМАТИКА. МЕХАНИКА. КОМПЬЮТЕРНЫЕ НАУКИ
МАТЕМАТИКА 2015. Т. 26. Вып. 2

УДК 510.252

© *А. П. Бельтюков*

ИНТЕРАКТИВНЫЕ РЕАЛИЗАЦИИ ЛОГИЧЕСКИХ ФОРМУЛ

Рассматривается новое конструктивное понимание логических формул, согласованное с интуицией и с традиционными средствами конструктивного логического вывода. Новое понимание логически проще традиционной реализуемости (в смысле кванторной глубины), но является также естественным с точки зрения алгоритмического решения задач. Это понимание, кроме свидетельства (реализации, подтверждения) понимаемой формулы, привлекает понятия теста (противодействия, препятствия) этой реализации на данной формуле. Для понимания формулы A рассматриваются предложения вида $a : A : b$. Это предложение означает, что объект a (выдвигаемый в подтверждение формулы A) выигрывает у объекта b (который противодействует выполнению формулы A) формулу A в процессе осуществления специальной процедуры сопоставления этих объектов друг с другом и с данной формулой. Данная процедура может считаться некоторой процедурой арбитража для вынесения необходимого решения. Базис процедуры арбитража для атомарных формул задается интерпретацией языка. Процедура для сложных предложений задается специальными правилами определения смысла логических связок. При наиболее естественном определении процедура арбитража имеет полиномиальную временную сложность. Формула A считается истинной в новом смысле этого слова, если имеется подтверждение, выигрывающее ее у всех возможных противодействий. Рассматривается логический язык без отрицаний. Доказана теорема о корректности в новом смысле традиционных интуиционистских аксиом и правил вывода. При этом рассматривается секвенциальное логическое исчисление, ориентированное на обратный метод поиска вывода.

Ключевые слова: логические формулы, понимание, реализация, противодействие.

DOI: 10.20537/vm160204

Введение

Конструктивное понимание формул, при котором классическая истинность уступает место наличию у них реализации (свидетельства или подтверждения) [1–3] логически является более простым, чем классическое дескриптивное понимание: оно имеет дело только с конструктивными объектами, которые можно описать конечными цепочками символов. Тем не менее такое традиционное конструктивное понимание все еще далеко от понимания, выраженного через непосредственно проверяемые утверждения: утверждение о том, что какой-то объект является реализацией данной формулы, может обладать слишком большой логической сложностью, связанной с неограниченным вложением кванторов и импликаций [2]. Это остается так, даже если ограничиться вычислениями ограниченной сложности.

В настоящей работе мы строим вариант конструктивного понимания формул, согласованного с общепринятыми способами рассуждений, при котором используются непосредственно проверяемые утверждения, с префиксом вида «существуют такие ..., что для любых ... выполнено ...».

Оказывается, для этого достаточно рассмотреть реализации (подтверждения, свидетельства) формул в условиях противодействия. Под противодействием можно понимать тестирование (попытку опровержения) этих реализаций, испытание их на предмет выявления ошибок. Такой подход хорошо согласуется с требованиями практической реализации, когда конструктивная реализация формулы практически используется как решение алгоритмической задачи, выраженной этой формулой. Тогда правильная реализация должна выдержать проверку (тестирование) своим практическим использованием, не выявив при этом никаких ошибок. Более глубокое понимание истинности при таком практическом подходе не требуется.

Самая большая сложность здесь проявляется при тестировании реализации импликации $(B \Rightarrow C)$. Традиционная реализация формулы $(B \Rightarrow C)$ подразумевает алгоритм, перерабатывающий реализацию формулы B в реализацию формулы C. Тогда тестирование такой реализации должно состоять в предъявлении какой-то реализации формулы B и последующем тестировании получаемого результата переработки ее в реализацию формулы C.

Чтобы избежать «провала» при такой процедуре проверки, реализация формулы $(B \Rightarrow C)$ в новом предлагаемом нами смысле должна содержать дополнительную «диагностическую» компоненту, тестирующую предъявленную реализацию B. При этом, чтобы облегчить работу этой компоненте, мы предлагаем разрешить ей пользоваться тестом для получаемой реализации B (что, разумеется, для основной части реализации запрещено).

Такое тестирование можно считать «последующим тестированием» в том смысле, что оно выполняется после «провала» работы основной компоненты реализации, для того чтобы оценить «закономерность» этого провала: если реализация B неправильная, то ожидать правильности полученной из нее реализации C не следует.

Настоящая работа — продолжение работ автора [6–8].

§ 1. Синтаксис формул

Будем рассматривать логические формулы, вид которых определен традиционной грамматикой:

```
<formula> ::= <predicate> ( <term>,... <term>)
            | (<formula> <connective> <formula>)
            | <quantifier> <variable> <formula>

<term> ::= <variable> | <constant>
```

Здесь логические связки (connectives) — конъюнкция, дизъюнкция и импликация; кванторы (quantifiers) — всеобщность и существование; predicate, variable, constant — непересекающиеся множества имен предикатов, предметных переменных и констант соответственно. Конструкция с многоточием — очевидное традиционное сокращение.

Заметим, что в этом языке нет отрицаний. При желании отрицание формулы B можно выразить импликацией:

$$(B \Rightarrow Error),$$

где $Error$ — некоторая формула, которая должна быть ложной.

§ 2. Типы значений

Для интерпретации формул введем понятия значений и их типов. Исходные типы (множества) значений:

L — логический тип, состоящий из двух значений 0 и 1;

U — универсум (значения переменных в формулах);

$T(P)$ — тип реализаций (свидетельств, подтверждений) атомарных формул с предикатом P (то, что можно предъявить в качестве обоснования формул вида $P(c)$; например, утверждение об изоморфности графов подтверждается соответствием между их вершинами);

$T'(P)$ — тип противодействий (опровержений) атомарным формулам с предикатом P (то, что можно предъявить для отвержения предъявленной реализации; например в случае с утверждением об изоморфности можно указать на вершины, соответствие между которыми нарушает требования, предъявляемые к изоморфизму).

Производные типы значений определяются следующим образом. Если мы уже имеем типы значений B и C, то определим также следующие типы:

$$(B \,\&\, C) = \{(b, c) \,|\, b : B, c : C\}, \qquad (B \vee C) = \{(0, b) \,|\, b : B\} \cup \{(1, c) \,|\, c : C\}, \qquad (B \Rightarrow C)$$

— функции с областью определения B и множеством значений C. Типы аргументов многоместных функций при этом соединяются операцией конъюнкции. Например, тип двухместной функции с аргументами типов B и C и значениями типа D определяется выражением

$$((B \,\&\, C) \Rightarrow D).$$

Для трехместной функции тип будет записан, например (для определенности расстановки скобок), выражением

$$(((B \,\&\, C) \,\&\, D) \Rightarrow W).$$

Далее будем рассматривать только алгоритмически заданные функции. Более того, можно ограничиться алгоритмами из какого-либо субрекурсивного класса, например алгоритмами, время работы которых ограничено многочленом от длины записи исходных данных.

При рассмотрении алгоритмов, определенных на объектах, содержащих функции, можно считать, что эти алгоритмы задаются в виде алгоритмов с оракулами (см, например, [4]), в качестве которых используются данные функции. Если эти функции, в свою очередь, также имеют функциональные аргументы, то в качестве таковых можно этим оракулам также задавать алгоритмы, которые также могут иметь свои оракулы, и так далее (до глубины левой вложенности операции импликации в определении типа).

При этом нетрудно выдерживать сложностные ограничения рассматриваемого субрекурсивного класса, например указанную выше полиномиальную ограниченность времени работы (с оракулом, если это необходимо).

При таких условиях рассматриваемый класс функционалов обладает следующими «хорошими» свойствами замкнутости.

А. Замкнутость относительно подстановок.

Если функция f определяется формулой

$$f(x) = e,$$

где x — кортеж переменных, а e — выражение, составленное из констант, операции построения упорядоченной пары ((u, w)), применений имен функций нашего класса и переменных x, типы которых естественным образом согласованы с типами используемых функций, то f также принадлежит нашему классу и может иметь соответствующий тип. Простой пример — определение функции f суперпозицией функций g и h:

$$f(x) = g(h(x)).$$

Здесь в качестве x можно брать не только кортежи переменных, но и более сложные конструкции из переменных, операции пары и констант 0 и 1. В последнем случае необходимо задать не одно равенство $f(x) = e$, а несколько равенств, исчерпывающих все комбинации 0 и 1 для обеспечения полноты определения f. Все эти равенства должны быть согласованы по типам переменных и функций. Например, определение

$$f(0, x) = g(x), \qquad f(1, x) = h(x)$$

задает функцию f типа $((B \vee C) \Rightarrow D)$, если функция g имеет тип $(B \Rightarrow D)$, а функция h имеет тип $(C \Rightarrow D)$.

Б. Лямбда-замкнутость.

Если функция f принадлежит нашему классу, то нашему классу также принадлежит и функция g, определяемая равенствами

$$g(x) = h : h(y) = f(x, y),$$

где x, y — кортежи переменных. Тип функции g определяется так, чтобы в приведенных здесь выражениях типы аргументов функций были согласованы. Например, если f имеет тип $((B \,\&\, C) \Rightarrow D)$, то определение

$$g(x) = h : h(y) = f(x, y)$$

задает функцию g типа $(B \Rightarrow (C \Rightarrow D))$.

На самом деле для дальнейших рассмотрений достаточно потребовать только эти два свойства замкнутости, а не рассматривать все детали алгоритмических определений.

§ 3. Типы реализаций и противодействий формул

Для каждой формулы B определяются два типа сопровождающих ее объектов:
– $T(B)$ — тип возможных реализаций (свидетельств, подтверждений) формулы B;
– $T'(B)$— тип возможных противодействий (опровержений, тестов) формулы B.
Эти типы определяются приведенными ниже равенствами.

Типы реализации атомарной формулы и ее теста определяются интерпретацией. Записывать их будем следующим образом:

$$T(P(c)) = T(P), \qquad T'(P(c) = T'(P).$$

Конъюнкция реализуется и тестируется по частям:

$$T(B \,\&\, C) = (T(B) \,\&\, T(C)), \qquad T'(B \,\&\, C) = (T'(B) \,\&\, T'(C)).$$

В дизъюнкции реализуется одна часть, а тестируется любой вариант:

$$T(B \vee C) = (T(B) \vee T(C)), \qquad T'(B \vee C) = (T'(B) \,\&\, T'(C)).$$

В реализации импликации, кроме традиционного преобразования реализации посылки в реализацию заключения, присутствует сильное тестирование реализации посылки (сильное в том смысле, что ему известна и сама реализация посылки, и тест для заключения):

$$T(B \Rightarrow C) = \Big(\big(T(B) \Rightarrow T(C)\big) \,\&\, \Big(\big(T(B) \,\&\, T'(C)\big) \Rightarrow T'(B) \Big) \Big), \quad T'(B \Rightarrow C) = (T(B) \,\&\, T'(C)).$$

Это делается для облегчения выявления ошибок в тестах.

Для всеобщности реализация подкванторной формулы вычисляется по значению переменной, а тест содержит это значение:

$$T(\forall\, x\, B) = (U \Rightarrow T(B)), \qquad T'(\forall\, x\, B) = (U \,\&\, T'(B)).$$

Для существования реализация содержит значение подкванторной переменной, а тестируется подкванторная формула:

$$T(\exists\, x\, B) = (U \,\&\, T(B)), \qquad T'(\exists\, x\, B) = T'(B).$$

§ 4. Реализационное понимание формул с противодействием

Если B — формула, b — ее реализация, а b' — ее противодействие, то может быть выполнена особая процедура определения результата «борьбы» реализации b за формулу B против противодействия b'. Запись «$b : B : b'$» означает результат такой борьбы. Здесь мы рассматриваем только простейший бинарный случай, при котором результат может выражаться в «победе» реализации, тогда выражение $b : B : b'$ считается истинным, иначе реализация «терпит поражение» и выражение $b : B : b'$ считается ложным.

Понимание атомарной формулы выражается эквивалентностью

$$(p : P(c) : p') \Leftrightarrow I[P](p, c, p'),$$

где $I[P]$ — реализационно-интерактивная интерпретация предиката P, задаваемая для всех предикатов в качестве базиса, с которого начинается понимание всех формул языка; $I[P]$ —

одна из рассматриваемых нами допустимых функций соответствующего типа. Например, для одноместного предиката P она имеет тип

$$(((TP \,\&\, U) \,\&\, T'P) \Rightarrow L).$$

Далее понимание всех логических связок подобрано так, чтобы соответствовать типам объектов, сопровождающих формулы.

Понимание конъюнкции задается эквивалентностью

$$((b,c) : (B \,\&\, C) : (b',c')) \Leftrightarrow (b : B : b') \,\&\, (c : C : c').$$

Здесь и далее логические связки справа понимаются в традиционном, классическом смысле.

Понимание дизъюнкции задается эквивалентностями

$$((0,b) : (B \vee C) : (b',c')) \Leftrightarrow (b : B : b')$$

и

$$((1,C) : (B \vee C) : (b',c')) \Leftrightarrow (C : B : C').$$

Понимание импликации задается эквивалентностью

$$(i,i") : (B \Rightarrow C) : (b,c') \Leftrightarrow ((b : B : i"(b,c')) \Rightarrow (i(b) : C : c')).$$

Понимание всеобщности задается эквивалентностью

$$(i : \forall\, x\, B(x) : (c,b')) \Leftrightarrow (i(c) : B(c) : c').$$

Понимание существования задается эквивалентностью

$$((c,b) : \exists\, x\, B(x) : b') \Leftrightarrow (b : B(c) : b').$$

Запись $a : A : a'$ можно читать так: «реализация a формулы A устойчива к противодействию a'». Аналогом классически общезначимой формулы в таком понимании является формула, имеющая в любой из интерпретаций рассматриваемого вида реализацию, устойчивую ко всем противодействиям.

Такие формулы будем называть интерактивно общезначимыми для данного класса интерпретаций.

Непосредственная проверка интерактивной общезначимости невозможна, поскольку предполагала бы проверку в бесконечном множестве интерпретаций. Тем не менее в некоторых случаях можно осуществлять логический вывод интерактивно общезначимых формул, для чего можно использовать различные логические исчисления. Далее мы показываем, что исчисление, представляющее собой приводимую ниже систему постулатов, обычную для традиционных конструктивных рассуждений, годится и для вывода интерактивно общезначимых формул. Более того, это исчисление устроено так, что реализации выведенных формул нетрудно алгоритмически построить по соответствующим логическим выводам.

§ 5. Исчисление

Выводимыми единицами описываемого далее логического исчисления (как и в [5]) являются секвенции — формулы специального класса в особой записи, описываемой ниже. В нашем варианте любая секвенция имеет вид

$$xD \to B,$$

где x — кортеж переменных (возможно, пустой); D — цепочка формул (возможно, пустая), посылка секвенции; B — формула, заключение секвенции.

Здесь, в отличие от традиционной записи секвенций, по техническим причинам мы явно выписываем x — список переменных нашей секвенции. Подразумевается, что переменные x

связаны кванторами всеобщности, между формулами D подразумеваются конъюнкции, стрелка обозначает импликацию. Таким образом, например, секвенция

$$yzP(z)Q(y)R(y) \to Q(y)$$

равнозначна формуле

$$\forall\, y \forall\, z(((P(z) \,\&\, Q(y)) \,\&\, R(y)) \Rightarrow Q(y)).$$

В частных случаях традиционно имеем, что

— секвенция $D \to B$ эквивалентна формуле $(D \Rightarrow B)$,

— секвенция $x \to B(x)$ эквивалентна формуле $\forall\, x\, B(x)$,

— секвенция $\to B$ эквивалентна формуле B.

Доказать в исчислении формулу B и означает доказать секвенцию

$$\to B.$$

Потребуем, чтобы все свободные переменные из D и B секвенции

$$xD \to B$$

содержались в кортеже x.

Далее перечислим постулаты нашего исчисления.

Все логические аксиомы исчисления порождаются одной схемой аксиом:

$$xDBS \to B,$$

где x — кортеж переменных, D и S — цепочки формул, B — формула. Здесь мы рассматриваем чисто логическое исчисление, никаких предметных аксиом не предполагается.

Далее опишем правила вывода секвенций.

Описываемое исчисление ориентировано на алгоритмическую автоматизацию процедуры построения (поиска) логического вывода обратным методом: от цели к аксиомам (см., например, [5]). Для удобства чтения правила ниже также записываются в обратной форме:

$$(R) \leftarrow (P)$$

или

$$(R) \leftarrow (P)(Q),$$

где R — секвенция заключения (цели) правила, P, Q — секвенции посылок правила (к которым сводится цель).

Все приводимые правила преобразуют: либо заключение цели R (это правила порождения); либо посылки цели R (это правила использования).

Во всех правилах далее: B, C и W — формулы; D и S — цепочки формул; y, z — цепочки переменных; x, u — переменные.

Далее перечислим правила порождения.

Порождение конъюнкции:

$$(yD \to (B \,\&\, C)) \leftarrow (yD \to B)(yD \to C).$$

Порождение дизъюнкции:

$$(yD \to (B \vee C)) \leftarrow (yD \to B), \qquad (yD \to (B \vee C)) \leftarrow (yD \to C).$$

Порождение импликации:

$$(yD \to (B \Rightarrow C)) \leftarrow (yDB \to C).$$

Порождение всеобщности:

$$(yD(y) \to \forall\, xB(y,x)) \leftarrow (uyD(y) \to B(y,u)),$$

где u — новая переменная (не входящая в y), $B(y,u)$ — результат замены в $B(y,x)$ всех свободных вхождений переменной x на переменную u (здесь и далее подразумевается переименование связанных переменных в преобразуемой заменой формуле для исключения коллизий обозначений).

Порождение существования:

$$\big(yuzD(y,u,z) \to \exists\, x\, B(y,u,z,x)\big) \leftarrow \big(yuzD(y,u,z) \to B(y,u,z,u)\big),$$

где $B(y,u,z,u)$ — результат замены в $B(y,u,z,x)$ всех свободных вхождений x на u.

Далее перечислены правила использования.

Правило использования конъюнкции имеет вид

$$(xS(B\,\&\,C)D \to W) \leftarrow (xSBCD \to W).$$

Правило использования дизъюнкции имеет вид

$$(x(B \vee C)D \to W) \leftarrow (xBD \to W)(xCD \to W).$$

Правило использования импликации имеет вид

$$(xS(B \Rightarrow C)D \to W) \leftarrow (xS(B \Rightarrow C)D \to B)(xSCD \to W).$$

Правило использования всеобщности имеет вид

$$(yuzS(y,u,t)\,\forall\, x\, B(y,u,z,x)D(y,u,z) \to W(y,u,z)) \leftarrow$$

$$\leftarrow (yutB(y,u,z,u)\,\forall\, x\, B(y,u,z,x)D(y,u,z) \to W(y,u,z)),$$

где $B(y,u,z,u)$ — результат замены в $B(y,u,z,x)$ всех свободных вхождений переменной x на переменную u.

Правило использования существования имеет вид

$$(yS(y)\,\exists\, x\, B(y,x)D(y) \to W(y)) \leftarrow (uyS(y)B(y,u)D(y) \to W(y)),$$

где u — новая переменная (не входящая в y), $B(y,u)$ — результат замены в $B(y,x)$ всех свободных вхождений переменной x на переменную u.

Заметим, что поскольку формулы не содержат отрицаний, то стандартное утверждение о том, что «из лжи следует все, что угодно», здесь не нашло отражения. Если отрицание формулы B выражается импликацией

$$(B \Rightarrow Error)$$

для какой-то ложной формулы, выбранной в качестве стандартна лжи, то аналогов упомянутого утверждения достичь нетрудно, допустив, например, правило вида

$$(xD \to B) \leftarrow (xD \to Error).$$

Заметим, что ложность $Error$ означает, что эта формула не имеет реализаций, устойчивых к противодействиям (любая попытка реализовать ее приводит к ошибке для некоторого противодействия). Приведенное правило мы далее не рассматриваем, так как намеренно ограничиваемся позитивным языком (языком без отрицания). Это имеет практический смысл для задач алгоритмизации: нам часто требуется алгоритм, как-то работающий и в некорректных случаях, хотя бы выдающий сообщения об ошибках, а не приводящий к тотальному краху создаваемой системы. К тому же на практике часто приходится рассматривать и так называемые «паранепротиворечивые» системы, в которых из ложных утверждений следует далеко не «все, что угодно». Эти случаи также остаются в области нашего рассмотрения.

§ 6. Реализации и противодействия секвенций

Будем считать, что реализация секвенции

$$xD(x) \to B(x)$$

состоит из двух функций — f и $f"$, противодействие f' состоит из трех элементов ($f' = (c, d, b')$):

цепочки c соответственных значений переменных x;

цепочки d соответственных реализаций формул из D;

цепочки b' — противодействия для формулы B.

Функция f определена на кортежах вида (c, d) и имеет значением реализацию B. Функция $f"$ определена на кортежах вида f' и имеет значением кортеж соответственных противодействий формулам D. Результат борьбы этой реализации, составленной из f и $f"$, за секвенцию $xD(x) \to B(x)$ с противодействием f' записывается в виде

$$(f : f" : (xD(x) \to B(x)) : f')$$

и определяется формулой:

$$(f : f" : (xD(x) \to B(x)) : (c, d, b')) \Leftrightarrow ((d : D(c) : f"(f')) \Rightarrow (f(c, d) : B(c) : b')).$$

§ 7. Теорема о корректности вывода

Сформулируем теорему о корректности построенного исчисления.

Теорема 1. *Все выводимые секвенции во всех реализационно-интерактивных интерпретациях имеют реализации, устойчивые ко всем противодействиям.*

Далее приводится доказательство теоремы о корректности вывода.

Для упрощения записи далее приведены частные случаи правил с небольшим количеством посылок и без переименований переменных (обобщения очевидны).

Опишем реализации всех выводимых секвенций.

Реализация аксиомы. Рассмотрим аксиому вида $BS \to B$. Формула ее реализации выглядит так:

$$f : f" : (BS \to B) : f',$$

где $f' = (b, s, b')$. Определим функции f и $f"$ равенствами:

$$f(b, s) = b, \qquad f"(f') = (b', 0).$$

Тогда истинная формула

$$(b : b" : B : b') \,\&\, (s : s" : S : 0) \Rightarrow (b : b" : B : b')$$

означает, что $f : f" : (BS \to B) : f'$, где $f' = (b, s, b')$, что и доказывает корректность аксиомы.

Далее опишем реализации результатов применения правил порождения. Исходя из понимания конъюнкции, выраженного формулой

$$((b, c) : (B \,\&\, C) : (b', c')) \Leftrightarrow (b : B : b') \,\&\, (c : C : c'),$$

докажем корректность правила порождения конъюнкции:

$$(D \to B \,\&\, C) \leftarrow (D \to B)(D \to C).$$

С этой целью достаточно для любых g, $g"$, h, $h"$ и f' так построить g', h', f и $f"$, чтобы следующая запись могла быть прочитана как истинная формула:

$$f : f" : (D \to B \,\&\, C) : f' \leftarrow g : g" : (D \to B) : g', h : h" : (D \to C) : h',$$

т. е. из утверждений посылок правила

$$g : g" : (D \to B) : g' \qquad \text{и} \qquad h : h" : (D \to C) : h'$$

должно следовать утверждение

$$f : f" : (D \to B \,\&\, C) : f'.$$

Для этого используем следующие определения:

$$(d, (b', c')) = f', \qquad g' = (d, b'), \qquad b = g(d), \qquad d' = g"(g').$$

Здесь и далее значения переменных в левых частях равенств определяются через значение правой части. Тогда первая посылка

$$(g : g" : (D \to B) : g')$$

рассматриваемого правила означает, что

$$(d : D : d') \Rightarrow (b : B : b').$$

Введя определения

$$h' = (d, c'), \qquad c = h(b), \qquad (d_1, c") = h"(g'),$$

получим, что вторая посылка $(h : h" : (D \to C) : h')$ рассматриваемого правила означает, что

$$(d : D : d_1) \Rightarrow (c : C : c').$$

Введем также определения

$$f(d) = (b, c) = (g(d), h(d)), \qquad f"(f') = ([d : D :]d'|d_1).$$

Здесь и далее через

$$([b : B :]x|y)$$

обозначаем выражение, которое равно y, если $b : B : x$, иначе оно равно x.

Тогда в любом случае формула

$$(d : D : f"(f')) \Rightarrow (f(d) : (B \,\&\, C) : (b', c')),$$

выражающая заключение рассматриваемого правила, означает, что

$$(d : D : d') \,\&\, (d : D : d_1) \Rightarrow (b : B : b') \,\&\, (c : C : c'),$$

что непосредственно следует из преобразованных нами посылок рассматриваемого правила. Это и доказывает корректность правила.

Исходя из понимания дизъюнкции, выраженного следующими формулами:

$$((0, b) : (B \vee C) : (b', c')) \Leftrightarrow (b : B : b'), \qquad ((1,) : (B \vee C) : (b', c')) \Leftrightarrow (\, : \; : '),$$

докажем корректность правила порождения дизъюнкции:

$$(D \to B \vee C) \leftarrow (D \to B).$$

С этой целью достаточно для любых g, $g"$ и f' так построить g', f и $f"$, чтобы следующая запись могла быть прочитана как истинная формула:

$$f : f" : (D \to B \vee C) : f' \leftarrow g : g" : (D \to B) : g'.$$

Введем следующие определения:

$$(d,(b',c')) = f', \qquad g' = (d,b'), \qquad b = g(d), \qquad d' = g"(g').$$

Тогда посылка $g : g" : (D \to B) : g'$ рассматриваемого правила означает, что

$$(d : D : d') \Rightarrow (b : B : b').$$

Определим функции f и $f"$ равенствами

$$f(d) = (0,b) = (0,g(d)), \qquad f"(f') = d'.$$

Тогда из преобразованной посылки рассматриваемого правила следует, что

$$(d : D : f"(f')) \Rightarrow (f(d) : (B \vee C) : (b',c')),$$

что и доказывает корректность рассмотренного правила. Другое правило порождения дизъюнкции рассматривается аналогично.

Исходя из понимания импликации, выраженного следующей формулой:

$$(i,i") : (B \Rightarrow C) : g' = (b,c') \Leftrightarrow ((b : B : i"(g')) \Rightarrow (i(b) : C : c')),$$

докажем корректность правила порождения импликации:

$$(D \to (B \Rightarrow C)) \leftarrow (DB \to C).$$

С этой целью достаточно для любых g, $g"$ и f' так построить g', f и $f"$, чтобы следующая запись могла быть прочитана как истинная формула:

$$f : f" : (D \to (B \Rightarrow C)) : f' \leftarrow g : g" : (DB \to C) : g'.$$

Для этого введем следующие определения:

$$(d,(b,c')) = f', \qquad f(d) = (i,i"), \qquad \text{где} \qquad i(b) = g(d,b),$$

$$i"(i') = b', \qquad \text{где} \qquad i' = (b,c'), \qquad g' = (d,b,c'), \qquad (d',b') = g"(g'), \qquad c = i(b).$$

Тогда посылка $g : g" : (DB \to C) : g'$ рассматриваемого правила означает, что выполнено утверждение

$$(d : D : d') \,\&\, (b : B : b') \Rightarrow (c : C : c').$$

Из этого утверждения, определив $f"$ равенством $f"(f') = d'$, получим

$$(d : D : d') \Rightarrow ((b : B : b') \Rightarrow (c : C : c')),$$

т. е.

$$(d : D : d') \Rightarrow ((b : B : i"(i')) \Rightarrow (i(b) : C : c')),$$

т. е.

$$(d : D : d') \Rightarrow ((i,i") : (B \Rightarrow C) : (b,c')),$$

т. е.

$$(d : D : f"(f')) \Rightarrow ((i,i") : (B \Rightarrow C) : i'),$$

т. е. выполнено заключение рассматриваемого правила:

$$f : f" : (D \to (B \Rightarrow C)) : f',$$

что и доказывает корректность этого правила.

Исходя из понимания всеобщности, заданного следующей формулой:

$$(i : \forall\, x\, B(x) : (c, b')) \Leftrightarrow (i(c) : B(c) : c'),$$

докажем корректность правила порождения всеобщности:

$$(D \to \forall\, x\, B(x)) \leftarrow (xD \to B(x)).$$

С этой целью достаточно для любых g, g" и f' так построить g', f и f", чтобы следующая запись могла быть прочитана как истинная формула:

$$f : f\text{"} : (D \to \forall\, x\, B(x)) : f' \leftarrow g : g\text{"} : (xD \to B(x))g'.$$

Для этого зададим следующие определения:

$$(d, i') = f', \quad (c, b') = i', \quad g' = (c, d, b'); \quad b = f(d)(c) = g(c, d), \quad i = f(d), \quad b = i(c), \quad d' = g\text{"}(g').$$

Тогда посылка $g : g\text{"} : (xD \to B(x))g'$ рассматриваемого правила означает утверждение

$$(d : D : d') \Rightarrow (b : B(c) : b').$$

Тогда, определив f" равенством $f\text{"}(f') = d'$, получим

$$(d : D : f\text{"}(f')) \Rightarrow (i(c) : B(c) : b'),$$

т. е.

$$(d : D : f\text{"}(f')) \Rightarrow (i : \forall\ xB(x) : (c, b')),$$

т. е.

$$(d : D : f\text{"}(f')) \Rightarrow (f(d) : \forall\, xB(x) : i'),$$

т. е.

$$f : f\text{"} : (D \to \forall\, xB(x)) : (d, i'),$$

т. е. выполнено заключение рассматриваемого правила:

$$f : f\text{"} : (D \to \forall\, xB(x)) : f',$$

что и доказывает корректность этого правила.

Исходя из понимания существования, задаваемого следующей формулой:

$$((c, b) : \exists\, x\, B(x) : b') \Leftrightarrow (b : B(c) : b'),$$

докажем корректность правила порождения существования:

$$(yD \to \exists\, x\, B(x)) \leftarrow (yD \to B(y)).$$

С этой целью достаточно для любых g, g" и f' так построить g', f и f", чтобы следующая запись могла быть прочитана как истинная формула:

$$f : f\text{"} : (yD(y) \to \exists\, x\, B(y, x)) : f' \leftarrow g : g\text{"} : (yD(y) \to B(y, y)) : g'.$$

Для этого зададим определения

$$g' = (c, d, b') = f', \qquad (c, b) = f(c, d) = (c, g(c, d)), \qquad d' = g\text{"}(g').$$

Тогда формула $(g : g\text{"} : (yD(y) \to B(y, y)) : g')$ посылки рассматриваемого правила означает утверждение

$$(d : D(c) : d') \Rightarrow (b : B(c, c) : b').$$

Отсюда, определив функцию f" равенством $f"(f') = d'$, получим утверждение

$$(d : D(c) : f"(f')) \Rightarrow ((c, b) : \exists\, x\, B(c, x) : b'),$$

т. е.

$$(d : D(c) : f"(f')) \Rightarrow (f(c, d) : \exists\, x\, B(c, x) : b'),$$

т. е.

$$f : f" : (yD(y) \to \exists\, x\, B(y, x)) : (c, d, b'),$$

т. е. выполнена формула заключения рассматриваемого правила:

$$f : f" : (yD(y) \to \exists\, x\, B(y, x)) : f',$$

что и доказывает корректность этого правила.

Опишем теперь реализации результатов применения правил использования.

Исходя из понимания конъюнкции, заданного формулой

$$((b, c) : (B \,\&\, C) : (b', c')) \Leftrightarrow (b : B : b') \,\&\, (c : C : c'),$$

докажем корректность правила использования конъюнкции:

$$((B \,\&\, C)D \to W) \leftarrow (BCD \to W).$$

С этой целью для любых g, $g"$ и f' построим g', f и $f"$ так, чтобы следующая запись могла быть прочитана как истинная формула:

$$f : f" : ((B \,\&\, C)D \to W) : f' \leftarrow g : g" : (BCD \to W) : g'.$$

Введем следующие определения:

$$((b, c), d, w') = f', \qquad g' = (b, c, d, w'), \qquad w = f((b, c), d) = g(b, d, c), \qquad (b', c', d') = g"(g').$$

Тогда посылка рассматриваемого правила означает, что выполнено утверждение

$$(b : B : b') \,\&\, (c : C : c') \,\&\, (d : D : d') \Rightarrow (w : W : w'),$$

из которого, введя определение

$$f"(f') = ((b', c'), d'),$$

получаем

$$((b, c) : (B \,\&\, C) : (b', c')) \,\&\, (d : D : d') \Rightarrow (w : W : w'),$$

т. е.

$$((b, c), d) : ((B \,\&\, C)D) : f"(f') \Rightarrow (f((b, c), d) : W : w'),$$

т. е. выполнено заключение рассматриваемого правила:

$$f : f" : ((B \,\&\, C)D \to W) : f',$$

что и доказывает корректность этого правила.

Исходя из понимания дизъюнкции, заданного следующими формулами:

$$((0, b) : (B \vee C) : (b', c')) \Leftrightarrow (b : B : b'), \qquad ((1, c) : (B \vee C) : (b', c')) \Leftrightarrow (c : C : c'),$$

докажем корректность правила использования дизъюнкции:

$$((B \vee C)D \to W) \leftarrow (BD \to W)(CD \to W).$$

С этой целью для любых g, g", h, h" и f' построим g', h', f и f" так, чтобы следующая запись могла быть прочитана как истинная формула:

$$f : f" : ((B \vee C)D \to W) : f' \leftarrow g : g" : (BD \to W) : g', h : h" : (CD \to W) : h'.$$

Рассмотрим случай

$$((0, b), d, w') = f'.$$

Введем определения

$$g' = (b, d, w'), \qquad f((0, b), d) = g(b, d), \qquad w = f((0, b), d) = g(b, d), \qquad (b', d') = g"(g').$$

Тогда первая посылка рассматриваемого правила означает, что выполнено утверждение

$$(b : B : b') \,\&\, (d : D : d') \Rightarrow (w : W : w'),$$

из чего, введя определение:

$$f"(f') = ((0, b'), d'),$$

получим

$$((0, b) : (B \vee C) : (b', 0)) \,\&\, (d : D : d') \Rightarrow (f((0, b), d) : W : w'),$$

т. е.

$$(((0, b), d) : ((B \vee C)D) : ((b', 0), d')) \Rightarrow (f((0, b), d) : W : w'),$$

т. е. выполнено заключение рассматриваемого правила:

$$f : f" : ((B \vee C)D \to W) : f'.$$

Случай

$$((1, c), d, (1, c"), d\, w') = f'$$

рассматривается аналогично, что и завершает доказательство корректности этого правила.

Исходя из понимания импликации, заданного следующей формулой:

$$(i, i") : (B \Rightarrow C) : g' \Leftrightarrow ((b : B : i"(g')) \Rightarrow (i(b) : C : c')),$$

где $g' = (b, c)$, докажем корректность правила использования импликации:

$$((B \Rightarrow C)D \to W) \leftarrow ((B \Rightarrow C)D \to B)(CD \to W).$$

С этой целью для любых g, g" и f' построим g', f и f" так, чтобы было выполнено следующее утверждение:

$$f : f" : ((B \Rightarrow C)D \to W) : f' \leftarrow g : g" : ((B \Rightarrow C)D \to B) : g'h : h" : (CD \to W) : h'.$$

Введем определения:

$$((i, i"), d, w') = f', \qquad b = g((i, i"), d), \qquad c = i(b),$$

$$w = f(i, d) = h(i(g(i, d)), d) = h(c, d), \qquad h' = (c, d, w'), \qquad (c', d') = h"(h').$$

Тогда вторая посылка рассматриваемого правила означает утверждение

$$(c : C : c') \,\&\, (d : D : d') \Rightarrow (w : W : w').$$

Введем определения:

$$i_1 = (b, c'), \qquad b' = i"(i_1).$$

Тогда посылка заключения рассматриваемого правила $(B \Rightarrow C)$ может означать

$$(i, i") : (B \Rightarrow C) : i_1$$

т. е.

$$(b : B : b') \Rightarrow (c : C : c').$$

Положим

$$g' = (i, d, b'), \qquad (i_0, d') = g"(g').$$

Тогда первая посылка рассматриваемого правила означает утверждение

$$((i, i") : (B \Rightarrow C) : i_0) \,\&\, (d : D : g') \Rightarrow (b : B : b').$$

Положим

$$i' = [(i, i") : (B \Rightarrow C) : i_0]i_1, \qquad f"(f') = (i', d').$$

Тогда из преобразованных формул для посылок рассматриваемого правила и формулы посылки его заключения $((b : B : b') \Rightarrow (c : C : c'))$ следует, что выполнено утверждение

$$((i, i") : (B \Rightarrow C) : i') \,\&\, (d : D : d') \Rightarrow (w : W : w'),$$

т. е. выполнена формула заключения рассматриваемого правила:

$$f : f" : ((B \Rightarrow C)D \to W) : f',$$

что и доказывает корректность этого правила.

Исходя из понимания всеобщности, выраженного формулой

$$(i : \forall x\, B(x) : (c, b')) \Leftrightarrow (i(c) : B(c) : c'),$$

докажем корректность правила использования всеобщности:

$$(y \forall x\, B(y, x) D(y) \to W(y)) \leftarrow (yB(y, y) \forall x\, B(y, x) D(y) \to W(y)).$$

С этой целью для любых g, $g"$ и f' построим g', f и $f"$ так, чтобы следующая запись для этого правила читалась как истинная формула:

$$f : f" : (y \forall x\, B(y, x) D(y) \to W(y)) : f' \leftarrow g : g" : (yB(y, y) \forall x\, B(y, x) D(y) \to W(y)) : g'$$

Введем следующие определения:

$$(c, i, d, w') = f', \qquad b = i(c), \qquad g' = (c, b, i, d, w'),$$

$$w = f(c, i, d) = g(c, b, i, d), \qquad (b', i_1, d') = g"(g').$$

Тогда посылка рассматриваемого правила запишется в виде

$$(b : B(c, c) : b') \,\&\, (i : \forall x\, B(c, x) : i_1) \,\&\, (d : D(c) : d') \Rightarrow (w : W(c) : w'),$$

откуда при использовании определения $i_0 = (c, b')$ следует формула

$$(i : \forall x\, B(c, x) : i_0) \,\&\, (i : \forall x\, B(c, x) : i_1) \,\&\, (d : D(c) : d') \Rightarrow (w : W(c) : w'),$$

из которой при определении

$$i' = ([i : \forall x\, B(c, x) :]i_0 | i_1)$$

следует формула

$$(i : \forall x B(c, x) : i') \,\&\, (d : D(c) : d') \Rightarrow (w : W(c) : w'),$$

из которой при определении
$$f"(f') := (i', d')$$
следует формула заключения рассматриваемого правила:
$$f"f" : (y \forall x\, B(y,x)D(y) \to W(y)) : f',$$
что и доказывает корректность правила.

Исходя из понимания существования, задаваемого формулой
$$((c,b) : \exists x\, B(x) : b') \Leftrightarrow (b : B(c) : b'),$$
докажем корректность правила использования существования:
$$(\exists x\, B(x)D \to W) \leftarrow (yB(y)D \to W).$$
С этой целью для любых g, $g"$ и f' построим g', f и $f"$ так, чтобы следующая запись рассматриваемого правила читалась как истинная формула:
$$f : f" : (\exists x\, B(x)D \to W) : f' \leftarrow g : g" : (yB(y)D \to W) : g'.$$
Введя определения:
$$((c,b),d,w') = f', \qquad g' = (c,b,d,w'), \qquad w = f((c,b),d) = g(c,b,d), \qquad (b',d') = g"(g'),$$
получим, что посылка записанного так правила означает, что выполнено утверждение
$$(b : B(c) : b')\,\&\,(d : D : d') \Rightarrow (w : W : w'),$$
т. е.
$$((b,c) : \exists xB(x) : b')\,\&\,(d : D : d') \Rightarrow (f((c,b),d) : W : w').$$
Отсюда, введя определение
$$f"(f') = (b', d),$$
получим запись заключения рассматриваемого правила:
$$f : f" : (\exists x\, B(x)D \to W) : f',$$
что и доказывает корректность этого правила.

Таким образом, все случаи рассмотрены и доказательство корректности вывода в построенном исчислении завершено.

Заключение

В заключение надо отметить, что предложенное конструктивное понимание существенно изменяет традиционное понятие реализации: в реализации импликации, кроме обычного прямого преобразования реализации посылки в реализацию заключения, появляется диагностическая компонента. Этого можно избежать, если всю диагностику вынести в отдельные объекты. Тогда диагностика появится не только у реализации импликации, но и у реализаций вообще всех формул. В настоящей работе этого не сделано, так как слишком усложнило бы изложение для первой работы по исследуемой проблеме. Тем не менее при использовании подобного понимания формул для практического синтеза алгоритмов такое построение имеет смысл проводить.

СПИСОК ЛИТЕРАТУРЫ

1. Марков А.А. Попытка построения логики конструктивной математики // Исследования по теории алгорифмов и математической логике: Сб. статей. Т. 2. М.: ВЦ АН СССР, 1976. С. 3–31.
2. Шанин Н.А. Об иерархии способов понимания суждений в конструктивной математике // Труды Математического института АН СССР. 1973. Т. 129. С. 203–266.
3. Клини С.К. Введение в метаматематику. М.: Издательство иностранной литературы, 1957. 526 с.
4. Верещагин Н.К., Шень А. Лекции по математической логике и теории алгоритмов. Часть 3. Вычислимые функции. М.: МЦНМО, 2012. 160 с.
5. Маслов С.Ю. Обратный метод установления выводимости для логических исчислений // Труды Математического института АН СССР. Логические и логико-математические исчисления. I. 1968. Т. 98. С. 26–87.
6. Beltiukov A.P. Intuitionistic formal theories with realizability in subrecursive classes // Annals of Pure and Applied Logic. 1997. Vol. 89. P. 3–15.
7. Beltiukov A.P. A strong induction scheme that leads to polynomially computable realizations // Theoretical Computer Science. 2004. Vol. 322. 17–39.
8. Beltiukov A.P. A polinomial programming language // Mathematical Problems of Computer Science: Transactions of the Institute for Informatics and Automation Problems of the National Academy of Sciences of Armenia. 2006. Vol. 27. P. 11–19.

Поступила в редакцию 05.05.2016

Бельтюков Анатолий Петрович, д. ф.-м. н., профессор, кафедра теоретических основ информатики, Удмуртский государственный университет, 426034, Россия, г. Ижевск, ул. Университетская, 1.
E-mail: belt.udsu@mail.ru

A. P. Beltyukov
Interactive realizations of logical formulas

Keywords: logical formulas, understanding, realization, counteraction.

MSC: 03B20, 03B30

A new constructive understanding of logical formulas is considered. This understanding corresponds to intuition and traditional means of constructive logical inference. The new understanding is logically simpler than traditional realizability (in the sense of quantifier depth), but it also natural with respect to algorithmic solution of tasks. This understanding uses not only witness (realization) of the formula to understand but it also uses notion of test (counteraction) of this realization at the given formula. The main form of a sentence to understand a formula A is $a : A : b$, that means that "the witness a wins the obstacle b while trying to approve the formula A". This procedure can be regarded as a procedure of arbitration for making the necessary solution. The basis of the arbitration procedure for atomic formulas is defined by the interpretation of the language. The procedure for complex sentences is given by special rules determining the meaning of logical connectives. In the most natural definition of the arbitration procedure it has polynomial time complexity. A formula A is considered to be true in the new sense if there is a witness of the formula that wins all possible obstacles at the formula. A language without negation is considered. A theorem of correctness of traditional intuitionistic axioms and inference rules is proved. The system of logical inference is formulated in sequent form. It is oriented to the inverse method of logical inference search.

REFERENCES

1. Markov A.A. An attempt to construct the logic of constructive mathematics, *Issledovaniya po teorii algorifmov i matematicheskoi logike: Sbornik statei, Tom 2* (Investigations on the theory of algorithms and mathematical logic: Proceedings, vol. 2), Moscow, 1976, pp. 3–31 (in Russian).
2. Shanin N.A. A hierarchy of ways of understanding judgements in constructive mathematics, *Tr. Mat. Inst. Steklova*, 1973, vol. 129, pp. 203–266 (in Russian).

3. Kleene S.C. *Introduction to metamathematics*, North-Holland, 1951, 500 p. Translated under the title *Vvedenie v metamatematiku*, Moscow: Inostr. Lit., 1957, 526 p.
4. Vereshchagin N.K., Shen' A. *Lektsii po matematicheskoi logike i teorii algoritmov. Chast' 3. Vychislimye funktsii* (Lectures on mathematical logic and algorithm theory. Part 3. Computable functions), Moscow: Moscow Center for Continuous Mathematical Education, 2012, 160 p.
5. Maslov S.Yu. The inverse method for establishing deducibility for logical calculi, *Tr. Mat. Inst. Steklova*, 1968, vol. 98, pp. 26–87 (in Russian).
6. Beltiukov A.P. Intuitionistic formal theories with realizability in subrecursive classes, *Annals of Pure and Applied Logic*, 1997, vol. 89, pp. 3–15.
7. Beltiukov A.P. A strong induction scheme that leads to polynomially computable realizations, *Theoretical Computer Science*, 2004, vol. 322, pp. 17–39.
8. Beltiukov A.P. A polinomial programming language, *Mathematical Problems of Computer Science: Transactions of the Institute for Informatics and Automation Problems of the National Academy of Sciences of Armenia*, 2006, vol. 27, pp. 11–19.

Received 05.05.2016

Beltyukov Anatolii Petrovich, Doctor of Physics and Mathematics, Professor, Department of Theoretical Foundations of Computer Science, Udmurt State University, ul. Universitetskaya, 1, Izhevsk, 426034, Russia.
E-mail: belt.udsu@gmail.com