МОДЕЛИ КОГНИТИВНЫХ ПРОЦЕССОВ
(Вычислительные системы)
1997 год Выпуск 158

УДК 519:517.12

ПРОЕКТ РАЗРАБОТКИ ЯЗЫКОВ СПЕЦИФИКАЦИЙ ЗАДАЧ, ОРИЕНТИРОВАННЫХ НА ПОЛЬЗОВАТЕЛЯ [1]

Е.В.Казаков, А.А.Москвитин, К.Ф.Самохвалов

Одна из важнейших проблем информатики — принципы построения языков спецификаций задач. Цель статьи — подойти к этой проблеме с такой стороны, чтобы получить ответ на вопрос, как может (если вообще может) проектировщик программного обеспечения разумным образом учесть на этапе конструирования своей системы интеллектуальные ресурсы и интеллектуальные запросы данной категории пользователей создаваемой программной системы. Такого рода учет — залог умения управлять факторами (стоимостью и качеством разработки), повышающими спрос на рынке компьютерных услуг.

Статья опирается на работы [1,2].Изложение распадается на две части. Первая (§ 2,3) соответствует предположению, что пользователь в некотором (точно определенном) смысле идеален. Вторая часть (§ 4,5) является естественным компаньоном первой и соответствует предположению, что пользователь не идеален.

Доказательства результатов первой части содержатся в [2].

---

[1]Работа поддержана РФФИ, грант №96-06-80970.

§ 1. Неформальная постановка проблемы

Приведем неформальные соображения, на фоне которых должны быть ясны мотивировки вводимых далее понятий.

1.1. **Интеллектуальные ресурсы**. Когда пользователь обращается к компьютеру, то он либо знает, чего он хочет, либо не знает этого. В первом случае он, как минимум, *понимает* свою задачу, во втором — ему вообще не стоит иметь дело с вычислительной машиной. Стало быть, без всякого ущерба для ожидаемой эксплуатационной практики построение программного обеспечения всегда можно основывать на предположении, что подвергаться компьютерной обработке будут лишь те задачи, которые *понятны* заказчику. В этой связи приобретает актуальность вопрос: чем же определяется понятность для человека той или иной задачи? Самый общий ответ таков: данный человек *понимает* беспокоящую его задачу, если и только если любую вещь, предъявляемую его вниманию, он в состоянии распознать как утоление или как неутоление упомянутого беспокойства. Нелепо, например, говорить, что некто понял задачу, решил ее, но не знает (продолжает беспокоиться), решил ли он ее. Либо он решил чужую (понятную нам, но не ему) задачу, либо он не дорешал свою — ему осталось еще убедиться в том, что найденное решение действительно есть то, которое его удовлетворяет. Словом, *понятная* человеку задача не может иметь неубедительных для него решений. Неубедительные решения — вообще не решения.

Это заявление выглядит трюизмом, но на самом деле оно чревато серьезными последствиями. Обратимся, например, к так называемым "задачам на доказательсто". Это — важный класс задач, так как к ним сводятся, в конечном итоге, все проблемы, возникающие в точной научной деятельности и требующие выхода на вычислительные машины.[2]

[2]Напомним, что любые вычисления — разновидность доказательства.

Решением любой такой задачи является некоторое доказательное рассуждение. Однако, если длина рассуждения достаточно велика, то человек может не усмотреть в нем никакой доказательности (не поймет его) именно потому, что оно чрезмерно громоздко. Следовательно, всякое рассуждение, содержащее, например, 10^{100} и более символов, не может быть решением какой-либо *понятной* человеку задачи.

Оценка границ допустимых длин убедительных (понятных) доказательств колоссальным числом 10^{100} мало что дает разработчику программных средств; для него важно уметь находить подобные границы (для различных сообществ людей) как можно более точно, дабы иметь возможность удовлетворять спрос на свою продукцию как можно более дешево. А это означает, что нужно исследовать вопрос о влиянии длины математических выводов на степень убедительности их для человека.

Однако, в такого рода исследованиях немедленно наталкиваются на специфическое затруднение, известное как "парадокс кучи". В идеализированном варианте это затруднение выглядит так. Для данного человека, если язык, аксиомы и правила вывода математической системы, в рамках которой он рассуждает, выбраны разумно, то для некоторого фиксированного числа $n_0 \ll 10^{100}$:

а) доказательства длины не более n_0 (символов) убедительны;

б) для любого n, если доказательства длины не более n убедительны, то доказательства длины не более $n+1$ также убедительны;

в) найдется (можно указать) такое натуральное число $M > n_0$ (например, $M = 10^{100}$), что доказательства длины $\geq M$ заведомо не убедительны.

Если "а" рассматривать как базис математической индукции, а "б" — как индукционный шаг, то "а", "б" противоречат "в". Между тем все три пункта "а"-"в" выглядят интуитивно несомненными. В этом парадокс кучи.

Есть два пути попыток его преодоления. Во-первых, можно попытаться показать, что на самом деле не все три утверждения "а"–"в" верны, хотя кажутся на первый взгляд несомненно таковыми. Во-вторых, можно подвергнуть сомнению применимость обычных средств рассуждений (например, принципа математической индукции) к "существенно размытым" свойствам (например, к свойству доказательства "быть убедительным"). Некоторые авторы предпочитают второй путь, усматривая в "парадоксе кучи" мотив для радикальной перестройки привычного концептуального аппарата ("ультраинтуиционизм" А.Есенина-Вольпина [3], "полумножества" П.Вопенки [4], различные "размытые логики" и т.д.). Другие придерживаются более консервативных взглядов, развивая первый путь. Мы примыкаем к этим другим.

Начало перому пути положил М.Даммит [5], который опираясь на тщательный анализ различных вариантов "парадокса кучи" (Даммит называет его "парадоксом Вана"), показал, что индукционный шаг "б" не верен, и попутно объяснил, почему все же кажется, что это не так.

1. *Интеллектуальный ресурс пользователя*. Развивая аргументацию М.Даммита, можно указать экспериментальную процедуру установления зависимости степени убедительности (в некоторой шкале баллов) доказательств от их длины для испытуемого человека, когда он рассуждает в рамках фиксированной аксиоматической системы [1]. Аналогичные процедуры можно указать также для установления зависимости степени безошибочного распознавания испытуемым человеком формул и термов данной аксиоматической системы от их длины. Поэтому можно оценивать **интеллектуакльный ресурс данного человека** p **относительно данной аксиоматической системы** S тройкой $\mathrm{res}(p,S) = (m_1(p,S), m_2(p,S), m_3(p,S))$ натуральных чисел $m_1(p,S), m_2(p,S), m_3(p,S)$ таких, что:

- $m_1(p,S)$ — наибольшая длина доказательств в данной системе S, все еще имеющих достаточно высокую (заранее фиксированную) балльную оценку степени убедительности для данного человека p;

- $m_2(p,S)$ — наибольшая длина последовательностей символов алфавита языка данной системы S, все еще имеющих достаточно высокую (заранее фиксированную) балльную оценку степени безошибочной распознаваемости данным человеком p как формул (или не формул) языка системы;

- $m_3(p, S)$ — то же, что и $m_2(p, S)$, но применительно к термам.

2. *Интеллектуальный ресурс сообщества пользователей.* Пусть π — какое-то множество людей и пусть Σ — какая-то функция на π, ставящая в соответствие каждому p из π свою аксиоматическую систему $\Sigma(p)$. Если все $\Sigma(p)$, $p \in \pi$, — аксиоматические системы в одном и том же языке L, то π называется **сообществом пользователей относительно** Σ. **Интеллектуальный ресурс сообщества пользователей** π **относительно** Σ — это тройка $\mathrm{RES}(\pi, \Sigma) = (\mathrm{M}_1(\pi, \Sigma), \mathrm{M}_2(\pi, \Sigma), \mathrm{M}_3(\pi, \Sigma))$ натуральных чисел $M_1(\pi, \Sigma)$, $M_2(\pi, \Sigma)$, $M_3(\pi, \Sigma)$ таких, что:

- $M_1(\pi, \Sigma)$ — наименьшее число среди всех чисел $m_1(\pi, \Sigma(p))$, $p \in \pi$;

- $M_2(\pi, \Sigma)$ — наименьшее число среди всех чисел $m_2(p, \Sigma(p))$, $p \in \pi$;

- $M_3(\pi, \Sigma)$ — наименьшее число среди всех чисел $m_3(p, \Sigma(p)), p \in \pi$.

1.2. **Интеллектуальные запросы.** Под ними естественно понимать задачи, которые собираются решать пользователи, приступая к эксплуатации компьютера. Поэтому первый шаг в обеспечении возможности учета интеллектуальных запросов при создании программных средств — это выработка понятия "задача для пользователя". При этом мы выделяем два варианта: "задача для *идеального* пользователя" и "задача для *реального* пользователя". Идеальный пользователь — это тот, для которого убедительны доказательства любой длины, лишь бы они были правильно построены в рамках принятой им системы рассуждений. Реальный пользователь — это тот реальный человек, для которого убедительность доказательств в рамках принятой им системы рассуждений зависит от их длины. Прямой практической значимостью обладает лишь второй вариант, рассмотрение же первого полезно косвенным образом: некоторые особенности нашего подхода ощущаются более выпукло.

1. *Интеллектуальные запросы идеального пользователя.* Мы исходим из предположения, что каждый пользователь может быть полностью представлен (в результате специальной процедуры опроса) совокупностью тех сведений об интересующей его предметной области, которыми он исходно располагает или считает, что располагает. Без особых натяжек наше предположение означает, что идеальный пользователь — это какая-то первопорядковая теория T, одна из моделей которой (не важно, какая именно) — подразумеваемая предметная область. В этом случае "задача для пользователя" — это "задача для произвольной модели теории T," или, кратко, "T-задача".

Каждая такая задача отождествляется с произвольным алгоритмом разбиения множества всех доказательств в T на два подмножества: подмножество доказательств, объявляемых **решением** рассматриваемой T-задачи, и подмножество остальных доказательств. Если первое подмножество не пусто — задача имеет решение. Если пусто — то не имеет решения. Всякое описание любого алгоритма такого разбиения — **формулировка** T-задачи.

В частности, формулировкой какой-то T-задачи можно считать любую формулу φ в языке теории T, так как φ эффективно определяет разбиение множества всех доказательств в T на два подмножества: подмножество доказательств, оканчивающихся данной формулой φ, и подмножество всех остальных доказательств. О любом из доказательств из первого подмножества мы говорим, что оно **решает** рассматриваемую **формульную T-задачу** $\hat{\varphi}$. О любом доказательстве из второго подмножества мы говорим, что оно **не решает** $\hat{\varphi}$.

Для любой теории T множество формульных T-задач — малая часть всех T-задач.[3] Тем не менее, это множество достаточно велико, чтобы считать оправданным предположение: на практике идеального пользователя всегда интересуют только *формульные* T-задачи. Поэтому впредь принимаем, что "задача для идеального пользователя" — это некоторая формульная T-задача для некоторой теории T. Иными словами, ниже "задача" — это всегда "формульная задача".

На практике теории T могут быть самыми причудливыми — с причудливой сигнатурой, причудливыми аксиомами, причудливыми правилами вывода. Но для всякой такой теории T хотелось бы уметь корректно, конструктивно и удобным образом рассуждать о любых формульных T-задачах, имеющих или не имеющих решения в T.

Обеспечить такую возможность значит обеспечить учет интеллектуальных запросов идеального пользователя. Цель последующих двух параграфов — сделать это точным образом.

Но прежде, чем приступить к точным рассмотрениям, полезно иметь ввиду два следующих неформальных пояснения.

Во-первых, что значит "рассуждать удобным образом"? Мы уже сказали, что теория T может быть самой необычной. Но коль скоро она уже выбрана пользователем в качестве исходного описания интересующей его предметной области, она должна считаться удобной для него. Стало быть, есть резон предположить, что рассуждать удобным образом о T-задачах значит рассуждать о них в языке, который, по-крайней мере, как-то тесно связан с языком теории T, в частности, в языке, имеющем ту

---

[3]Если допустить к рассмотрению эффективно распознаваемые формулы бесконечной длины, то тогда множество формульных T-задач и множество всех T-задач могли бы (при соответствующих переопределениях) совпадать.

же сигнатуру и тот же синтаксис, что и язык теории T.

Разумеется, при этом семантика языка должна измениться, так как теперь мы говорим уже не о самой предметной области, как это делает теория T, а говорим о задачах, относящихся к ней. В связи с этим возможны отличия в дедуктивном аппарате наших новых рассуждений. Если эти дедуктивные отличия случайно окажутся малыми или даже вообще отсутствующими, то тем лучше. Такие случаи представляют особый интерес, и мы должны с самого начала предусмотреть их возможность.

Второе пояснение относится к слову "конструктивно". Обычные конструктивные истолкования первопорядковых языков следуют идее о том, что конструктивный смысл сложной формулы должен редуцироваться, как это имеет место и в классическом случае, к конструктивным смыслам ее конституентов. Отсюда — индуктивные определения классической и интуиционистской истинностей. Однако на этом пути возникают известные технические (если не принципиальные) трудности. Приходится говорить, например, о функционалах высших типов при попытках конструктивно истолковать сложную формулу с глубиной вхождения импликации больше единицы. Ниже мы убедимся, что ценою отказа от идеи редукционизма можно развить новый подход к конструктивным истолкованиям языков первого порядка, технически менее сложный, чем подходы с обычными интерпретациями. В частности, в этом новом подходе нет нужды оперировать функционалами высших типов при построении конструктивных семантик сложных формул.

2. *Интеллектуальные запросы реального пользователя.* Итак, **интеллектуальные запросы идеального пользователя** p — это множество всех формульных T_p-задач, где T_p — теория предметной области, которой (теорией) исходного располагает p. Подчеркнем, что это понимание запросов зиждется на предположении, что идеальный пользователь p полностью репрезентируется теорией T_p. Однако не следует забывать, что это — *упро-*

щающее предположение. Более реалистичным было бы предположить, что репрезентация пользователя — это не просто какая-то теория T_p, а пара $(T_p, \mathrm{res}(p, T_p))$, где $\mathrm{res}(p, T_p)$ — интеллектуальный ресурс пользователя p. Этому более реалистичному предположению отвечают термин **задача для реального пользователя** p как сокращение для "формульная T_p-задача доступная пониманию p" и термин **интеллектуальные запросы реального пользователя** p как сокращение для "множество всех таких задач".

§ 2. Языки и логики спецификаций задач для идеальных пользователей

1. **Языки**. Пусть L — язык первого порядка сигнатуры σ. Логические символы L : $\&, \vee, \to, \forall, \exists, \perp$, и, может быть, "равенство" $=$; $\perp$ — атомное высказывание "ложно". Пусть T — исчисление в языке L.[4] Исчисление T называется

- **формальным**, если множество $\mathrm{Thm}(T)$ теорем T эффективно перечислимо;

- **противоречивым**, если $\mathrm{Thm}(T) = F(L)$, где $F(L)$ — множество всех формул языка L;

- **непротиворечивым**, если $\mathrm{Thm}(T) \neq F(L)$;

- **сильно непротиворечивым**, если $\perp \notin \mathrm{Thm}(T)$.

Для любого формального T в L определим специальное множество $C(T)$, формул из L так, чтобы заготовить возможность говорить: $\varphi \in C(T)$ тогда и только тогда, когда φ есть символическая запись какого-то истинного высказывания о каких-то формульных задачах в T.

ОПРЕДЕЛЕНИЕ 1.

i) если φ — атомарная формула L, то $\varphi \in C(T)$ тогда и только тогда, когда $\varphi \in \mathrm{Thm}(T)$;

ii) если φ имеет вид $(\varphi_1 \ \& \ \varphi_2)$, то $\varphi \in C(T)$ тогда и только тогда, когда $\varphi_1 \in \mathrm{Thm}(T)$ и $\varphi_2 \in \mathrm{Thm}(T)$;

---

[4]Термин "исчисление" понимается так, как он определен Ершовым Ю.Л. и Палютиным Е.А. в кн. Ершов Ю.Л., Палютин Е.А. Математическая логика.-М., 1979.-320 с.

iii) если φ имеет вид $(\varphi_1 \vee \varphi_2)$, то $\varphi \in C(T)$ тогда и только тогда, когда $\varphi_1 \in \mathrm{Thm}(T)$ или $\varphi_2 \in \mathrm{Thm}(T)$;

iv) если φ имеет вид $(\varphi_1 \to \varphi_2)$, то $\varphi \in C(T)$ тогда и только тогда, когда существует эффективное (вообще говоря, частичное) отображение $A_{\varphi_1 \to \varphi_2}$ из множества всех в T доказательств в себя такое, что если x — доказательство в T формулы φ_1, то $A_{\varphi_1 \to \varphi_2}$ определено на x и $A_{\varphi_1 \to \varphi_2}(x)$ — доказательство в T формулы φ_2;

v) если φ имеет вид $\forall x \varphi_1(x)$, то $\varphi \in C(T)$ тогда и только тогда, когда существует эффективное тотальное отображение $A_{\forall x \varphi_1(x)}$ из множества всех замкнутых термов из L во множество всех доказательств в T такое, что если t — произвольный замкнутый терм в L, то $A_{\forall x \varphi_1(x)}(t)$ — доказательство в T формулы $\varphi_1(t)$;

vi) если φ имеет вид $\exists x \varphi_1(x)$, то $\varphi \in C(t)$ тогда и только тогда, когда $\varphi_1(t) \in \mathrm{Thm}(T)$ для некоторого замкнутого терма t в L.

ОПРЕДЕЛЕНИЕ 2. Пусть T — формальное исчисление в языке L. **Формула φ из L конструктивно истинна относительно T (T-истинна)**, если и только если $\varphi \in C(T)$. Если при этом исчисление T рассматривается как репрезентация идеального заказчика, то любая T-истинная формула φ называется также **задачно истинной относительно T**.

Определения 1 и 2 говорят о том, что мы рассматриваем конструктивно истинные относительно T высказывания в L как коды определенных суждений о системе формальных выводов в T и, следовательно, как коды суждений о T-задачах.

К сказанному нужно добавить, что и при обычных конструктивных интерпретациях высказывания в L, в конечном счете, высказываются о (относятся к) некой (неявно подразумеваемой или указанной явно, формализованной частично или формализованной с достаточной полнотой) системе средств убеждения ("канонические доказательства", "реализации" и т.д.). Самый общий аспект отличия предлагаемого подхода заключается не в самом факте относительности конструктивных смыслов формул, а в способе, каким эти конструктивные смыслы образуют суждения о (доказательствах в) T.

Наш способ, как это следует из определений 1 и 2, нарушает у конструктивных значений сложных формул их редукцию к конструктивным значениям их конституентов. Рассмотрим поясняющий пример.

Пусть PA — арифметика Пеано (записанная с учетом того, что $\neg\varphi$ определяется как $\varphi \to \bot$). Предположим, что PA — непротиворечивая теория (в нашей терминологии, сильно непротиворечивое исчисление). Тогда существует высказывание вида $\forall x\Psi(x)$ (например, $\mathrm{Consis}_{\mathrm{PA}}$) такое, что для любого натурального числа n $\mathrm{PA} \vdash \Psi(\bar{n})$, но не $\mathrm{PA} \vdash \forall x\Psi(x)$.

В этом случае формула $\forall x\Psi(x)$ — PA-истинное высказывание, а высказывание $(\forall x\Psi(x)\ \&\ \forall x\Psi(x))$ — не PA-истинно.

Все дело в том, что определение 1 не является индуктивным (хотя внешне и походит на него). Если бы оно было индуктивным, то, например, в "ii" вместо "$\varphi_1 \in \mathrm{Thm}(T)$ и $\varphi_2 \in \mathrm{Thm}(T)$" должно было бы стоять "$\varphi_1 \in C(T)$ и $\varphi_2 \in C(T)$". В результате, рассматриваемая как код суждения о PA-задачах, формула $\forall x\Psi(x)$ означает: все (для любого n) формульные PA-задачи вида $\widehat{\Psi(\bar{n})}$ имеют решения; и поскольку, по предположению, в PA действительно доказуемы все формулы $\widehat{\Psi(\bar{n})}$, то $\forall x\Psi(x)$ — задачно истинное относительно PA предложение языка арифметики. С другой стороны, предложение $(\forall x\Psi(x)\ \&\ \forall x\Psi(x))$, рассматриваемое как код суждения о PA-задачах, означает: PA-задача $\widehat{\forall x\Psi(x)}$ и (та же самая) PA-задача $\widehat{\forall x\Psi(x)}$ обе имеют решения, что не верно, ибо в PA не доказуема формула $\forall x\Psi(x)$. Следовательно, $(\forall x\Psi(x)\ \&\ \forall x\Psi(x))$ — не задачно истинное относительно PA предложение языка арифметики PA.

Если говорить коротко, то более конкретный аспект отличия предлагаемого здесь подхода к конструктивному истолкованию формул первопорядковой теории T состоит в том, что конструктивный смысл сложной формулы определяется здесь *только ее внешней связкой* (включая кванторы) *и структурой дедуктивного аппарата теории.*

ОПРЕДЕЛЕНИЕ 3. Пусть L — первопорядковый язык, T — формальное исчисление в L. Тогда пара $(L, C(T))$ называется **конструктивной семантикой языка** L **относительно** T, или, кратко, **Т-конструктивной семантикой (языка** L**).** Если при этом T считается репрезентацией идеального пользователя, то T-конструктивная семантика $(L, C(T))$ называется также **семантикой формульных задач для идеального пользователя** T, или, кратко, T**-задачной семантикой**. Если $(L, C(T))$ — T-задачная семантика, то L называется **языком спецификаций формульных задач для пользователя** T, или, кратко, **языком спецификаций** T**-задач.**

2. Логики. Имеет место следующая очевидная

ТЕОРЕМА 1. *Если формальное исчисление* T *в языке* L *противоречиво, то любая формула в* L T*-истинна.*

Пусть, далее, τ — произвольный (непустой) класс формальных исчислений в языке L.

ОПРЕДЕЛЕНИЕ 4. Множество $L(\tau)$ всех тех и только тех формул из L, каждая из которых конструктивно истинна относительно каждого исчисления T из τ, называется **конструктивной логикой класса** τ, или, кратко, τ**-логикой** (в L). Если при этом τ считается репрезентацией класса сообщества пользователей, то τ-логика $L(\tau)$ называется также **логикой спецификаций задач для сообщества пользователей** τ**.**

Из этого определения видно, что

$$L(\tau) = \bigcap_{T \in \tau} C(T). \tag{1}$$

Следовательно, имеет место

ТЕОРЕМА 2. *Если* $\tau \subseteq \tau'$, *то* $L(\tau') \subseteq L(\tau)$.

Кроме того, из теоремы 1 и (1) вытекает

ТЕОРЕМА 3. *Если* T *— противоречивое формальное исчисление в* L *и* $\tau' = \tau \cup \{T\}$, *то* $L(\tau) = L(\tau')$.

Ясно, что τ-логики $L(\tau)$ как множества (формул) частично упорядочены по включению $\subseteq$. Из теоремы 1 следует, что этот порядок имеет наибольший элемент, и им является множество $F(L)$, совпадающее в силу теоремы 3 с любой τ_{inc}-логикой, где τ_{inc} — произвольный класс, состоящий только из противоречивых исчислений.

Из теоремы 2 вытекает существование наименьшего элемента в рассматриваемом порядке. Этим наименьшим элементом является τ_{all}-логика, где τ_{all} — класс всех формальных исчислений в языке L.

Если обозначить через τ_{con} класс всех непротиворечивых формальных исчислений L, а через τ^*_{con} — класс всех сильно непротиворечивых формальных исчислений в L, то в силу теорем 2 и 3 имеем $L(\tau_{\text{all}}) = L(\tau_{\text{con}}) \subseteq L(\tau^*_{\text{con}})$. Таким образом, в семействе всех возможных τ-логик в фиксированном языке L, упорядоченных по включению $\subseteq$, наименьшим элементом является конструктивная логика класса τ_{con} (или τ_{all}), а наибольшим — множество $F(L)$ всех формул в L; τ^*_{con}-логика занимает промежуточное положение.

Исчерпывающую информацию на этот счет дают две нижеследующие теоремы.

ТЕОРЕМА 4. *Пусть $F_0(L)$ — множество, состоящее точно из всех формул L вида $(\varphi \to \varphi)$. Тогда $L(\tau_{\text{con}}) = F_0(L)$.*

ТЕОРЕМА 5. *Пусть $F_1(L)$ — множество, состоящее точно из всех формул L вида $(\varphi \to \varphi)$ и всех формул вида $(\perp \to \varphi)$. Тогда $L(\tau^*_{\text{con}}) = F_1(L)$.*

3. **τ_{sym}-логики.** Классы $F_0(L), F_1(L)$ выглядят слишком бедными, а класс $F(L)$ — чересчур богатым, поэтому естественно предположить, что τ-логики, представляющие хоть какой-либо практический интерес, все лежат строго между $F_1(L)$ и $F(L)$. Особо отметим один вид таких логик.

Пусть $|\sigma|$ — множество сигнатурных символов языка L. И пусть g — произвольная перестановка на $|\sigma|$ такая, что она каждый предикатный символ переводит в предикатный, а каждый функциональный — в функциональный, причем местности символов обоих видов сохраняются. Очевидно, что каждая такая перестановка g на $|\sigma|$ индуцирует перестановку g' на $F(L)$. Очевидно также, что множества всех перестановок g и всех перестановок g' образуют группы $G(\sigma)$ и $G'(L)$ соответственно.

Мы говорим, что перестановка g' на $F(L)$ переводит правило вывода

$$\frac{\varphi_0, \ldots, \varphi_n}{f(\varphi_0, \ldots, \varphi_n)}$$

в правило вывода

$$\frac{\varphi'_0, \ldots, \varphi'_n}{f'(\varphi'_0, \ldots, \varphi'_n)},$$

если для всех $\varphi_0, \ldots, \varphi_n, \varphi'_0, \ldots, \varphi'_n \in F(L)$ выполняется равенство $f'(\varphi'_0, \ldots, \varphi'_n) = g' f(\varphi_0, \ldots, \varphi_n)$ всякий раз, когда $\varphi'_0 = g'\varphi_0, \ldots, \varphi'_n = g'\varphi_n$.

ОПРЕДЕЛЕНИЕ 5. Пусть T — исчисление в L. Будем называть **группой (сигнатурных) симметрий T** подгруппу $H(T)$ группы $G(\sigma)$ такую, что индуцируемая подгруппа $H'(T)$ группы $G'(L)$ состоит в точности из всех перестановок g', переводящих аксиомы T в аксиомы T, а правила вывода T — в правила вывода T. Если $H(T)$ не единична, то исчисление T будем называть **(сигнатурно) симметричным**.

Пусть sym — какая-то эффективно перечислимая подгруппа группы $G(\sigma)$. Тогда через τ_{sym} будем обозначать класс всех сильно непротиворечивых формальных исчислений T в L таких, что $H(T) = \mathrm{sym}$. Очевидно, что для любой sym класс τ_{sym} не пуст. Ему, например, принадлежит исчисление $T^{\varphi}_{\mathrm{sym}}$, имеющее пустое множество правил вывода и эффективно перечислимое множество аксиом $\mathrm{sym}(\varphi) = \{g'\varphi | g \in \mathrm{sym}\}$, где φ — фиксированная произвольная формула L, отличная от $\perp$.

ТЕОРЕМА 6. *Для любой эффективно перечислимой группы* sym *группы* $G(\sigma)$ *логика* $L(\tau_{\mathrm{sym}})$ *состоит в точности из всех формул вида* $(\perp \to \varphi)$ *и всех формул вида* $(\varphi \to g\varphi)$, $g \in \mathrm{sym}$.

Из этой теоремы видно, что $F_1(L) \subseteq L(\tau_{sym}) \supset F(L)$, причем равенство $L(\tau_{sym}) = F_1(L)$ достигается всякий раз, когда sym — единичная подгруппа группы $G(\sigma)$.

У читателя может возникнуть вопрос: что оправдывает наш специальный интерес к τ_{sym}-логикам? В качестве намека на ответ заметим, что, например, физика часто имеет дело с вещами (левое — правое, частица — античастица и т.д.), характеристики которых симметрично входят в физические теории.

4. **Немонотонность некоторых τ-логик.** До сих пор мы исследовали конструктивные логики весьма широких классов исчислений (логики спецификаций задач для весьма широких классов пользователей). Перейдем теперь к изучению конструктивных логик наименьших возможных классов — единичных (логик спецификаций задач для одного пользователя).

ОПРЕДЕЛЕНИЕ 6. Пусть T_1 и T_2 — произвольные формальные исчисления в L. Будем говорить, что T_2 **больше** T_1 (писать $T_1 \leq T_2$), если и только если все аксиомы T_1 являются аксиомами T_2 и все правила вывода T_1 являются правилами вывода T_2.

Отношение "больше" не следует путать с отношением "логически (дедуктивно) сильнее". В самом деле, если $T_1 \leq T_2$, то $\mathrm{Thm}(T_1) \subseteq \mathrm{Thm}(T_2)$, но обратное, вообще говоря, не верно: легко указать примеры исчислений T_1 и T_2 таких, что $\mathrm{Thm}(T_1) \subseteq \mathrm{Thm}(T_2)$, но не $T_1 \leq T_2$.

Далее, это отношение — частичный порядок на τ_{all} (и следовательно, на τ_{con}). Поэтому правомерен вопрос о монотонности $\{T\}$-логик (логик единичных классов $\{T\}$ исчислений T) относительно $\leq$. Нетрудно показать, что $\{T\}$-логика, вообще говоря, не монотонна, а именно: справедлива

ТЕОРЕМА 7. *Для данного L, если $G(\sigma)$ имеет неединичную эффективно перечислимую подгруппу* sym, *то существуют такие T_1 и T_2, принадлежащие классу τ_{con} (в L), что $T_1 \leq T_2$, но не $L(\{T_1\}) \subseteq L(\{T_2\})$ и не $L(\{T_2\}) \subseteq L(\{T_1\})$.*

Эта теорема говорит, в частности, о том, что если пользователь T_2 знает больше истин о подразумеваемой предметной области, чем пользователь T_1, то это еще не обязытельно значит, что T_2 знает больше истин о формульных задачах, относящихся к подразумеваемой предметной области, чем T_1.

§ 3. Представления логик спецификаций задач и конструктивные системы

1. Представления τ–логик.

ОПРЕДЕЛЕНИЕ 7. Если $L(\tau)$ — конструктивная логика класса τ и $K \subseteq L(\tau)$, то K называется **фрагментом τ–логики** $L(\tau)$.

ОПРЕДЕЛЕНИЕ 8. Если формальное исчисление T в L таково, что множество его теорем является фрагментом конструктивной логики (является конструктивной логикой) класса τ, то T называется **(тотальным) формальным представлением τ–логики.**

Пусть X — произвольное множество формул языка L и пусть T_X означает исчисление в L, имеющее пустое множество правил вывода и множество аксиом X. Пусть $F(L), F_0(L), F_1(L)$ — ранее определенные множества формул (см. разделы 1 и 2 § 2). И пусть $F_{sym}(L)$ — множество всех формул вида $(\bot \supset \varphi)$ и всех формул вида $(\varphi \to g'\varphi)$, $g \in \mathrm{sym}$, где sym — эффективно перечислимая подгруппа группы $G(\sigma)$. Тогда из теорем 1,3 и 4 непосредственно следует, что $T_{F(L)}$ является тотальным формальным представлением любой τ_{inc}–логики, а $T_{F_0(L)}$ — тотальным формальным представлением как для τ_{all}–логики, так и для τ_{con}–логики. Теорема 5 говорит о том, что $T_{F_1(L)}$ — тотальное формальное представление τ^*_{con}–логики. Теорема 6 утверждает, что $T_{F_{sym}(L)}$ есть тотальное формальное представление для τ_{sym}–логики.

Таким образом, имеет место

ТЕОРЕМА 8. *Существуют тотальные формальные представления τ_{all}–, τ_{inc}–, τ_{con}–, τ^*_{con}–логик. Если sym — эффективно перечислимая подгруппа группы $G(\sigma)$, то существует тотальное формальное представление τ_{sym}–логики.*

2. Конструктивные системы.

ОПРЕДЕЛЕНИЕ 9. Если формальное исчисление T в L является формальным представлением $\{T\}$-логики, то T называется **автоконструктивной** (или просто **конструктивной**) **системой**. Если при этом T репрезентирует идеального пользователя, то T называется **автозадачной** (или просто **задачной**) **системой этого пользователя**. Конструктивная (задачная) система T **точна**, если T — тотальное формальное представление $\{T\}$-логики.

Из формулы (1) вытекает, что для любого формального исчисления T $\{T\}$-логика совпадает с $C(T)$. Поэтому определение 9 говорит, в частности, о том, что сведения пользователя об интересующей его предметной области могут одновременно рассматриваться как некоторые истинные утверждения о задачах для него, если эти сведения T являются задачной (конструктивной) системой. Эти сведения могут рассматриваться как все истинные утверждения об упомянутых задачах, если T — точная задачная (точная конструктивная) система.

Если формальное исчисление T не является (конструктивной) задачной системой, то T среди своих теорем содержит ложное утверждение о каких-то T-задачах.

Очевидно, например, что исчисление $T_{F(L)}$ есть точная задачная (точная конструктивная) система. В то же время классическое исчисление предикатов или, скажем, арифметика Пеано вообще не являются, как показывают две нижеследующие теоремы, конструктивными (задачными) системами.

ТЕОРЕМА 9. *Исчисление $CPC(L)$ (классическое исчисление предикатов в L) — не конструктивная система.*

ТЕОРЕМА 10. *Исчисление PA (первопорядковая арифметика Пеано) — не конструктивная система.*

Из теоремы 9 вытекает, что не все логические истины можно истолковать как истины о логических задачах.

Из теоремы 10 вытекает, что не все классические арифметические истины можно истолковать как истины об арифметических задачах.

ТЕОРЕМА 11. *Исчисление $HPC(L)$ (исчисление предикатов Гейтинга в языке L) — конструктивная система, и если $G(\sigma)$ — нееединичная группа, то $HPC(L)$ заведома не точна.*

ТЕОРЕМА 12.

i) *HA (Арифметика Гейтинга) — конструктивная (задачная) система;*

ii) *HA — не точна.*

Из теоремы 11 следует, что теоремы интуиционисткой логики — это одновременно и некоторые истины о задачах, соответствующих исходным сведениям (о предметной области), состоящим из интуционистски верных суждений, общих вообще всем областям. А из теоремы 12 вытекает, что интуиционистская арифметика — это одновременно совокупность истин о задачах интуиционистской арифметики, но не всех таких истин.

Если условиться, что аксиома — это правило вывода с пустым множеством посылок, и что допустимость в T правил вывода понимается обычным образом, то для точных конструктивных (задачных) систем справедлива следующая

ТЕОРЕМА 13. *Если T (в L) — точная конструктивная система, то в T допустимы все аксиомы и правила вывода минимального исчисления высказываний Иогансона. Если, дополнительно, T — сильно непротиворечивое исчисление, то в T допустимы все аксиомы и правила вывода интуиционистского исчисления высказываний Гейтинга.*

§4. Языки и логики спецификаций задач для рельных пользователей

1. **Res-языки**. Пусть, по-прежнему, L — язык первого порядка (сигнатуры σ), T — исчисление в L. Пусть res — тройка (m_1, m_2, m_3) натуральных чисел $m_1, m_2, m_3 > 0$. Всякую формулу языка L назовем **res–формулой** (этого языка), если ее запись в L содержит не более m_2 вхождений символов (т.е. если длина формулы не более m_2).

Всякий терм языка L назовем **res–термом** (этого языка), если его запись в L содержит не более m_3 вхождений символов (т.е. если его длина не более m_3). Всякое доказательство в T назовем **res–доказательством в T**, если его запись в L содержит не более m_1 вхождений символов, т.е. если его длина не более m_1.

Исчисление T в L называется:

- **res–формальным**, если множество $\mathrm{Thm}(T, \mathrm{res})$ res–теорем T (оконечных res–формул res–доказательств в T) эффективно перечислимо;

- **res–противоречивым**, если $\mathrm{Thm}(T, \mathrm{res}) = F(L, \mathrm{res})$, где $F(L, \mathrm{res})$ — множество всех res–формул языка L;

- **res–непротиворечивым**, если $\mathrm{Thm}(T, \mathrm{res}) \subset F(L, \mathrm{res})$;

- **сильно res–непротиворечивым**, если $\bot \notin \mathrm{Thm}(T, \mathrm{res})$.

Ясно, что если σ — конечная сигнатура, то любое формальное T в L — res–формальное исчисление; более того, в этом случае множества $F(L, \mathrm{res})$, $\mathrm{Thm}(T, \mathrm{res})$ и множество $\mathrm{Term}(L, \mathrm{res})$ всех res–термов в L всегда конечны.

Для любого res–формального T в L определим множество $C(T, \mathrm{res})$ res–формул из L так, чтобы заготовить возможность говорить: $\varphi \in C(T, \mathrm{res})$ тогда и только тогда, когда φ есть символическая запись какого-то истинного высказывания о каких-то формульных задачах в T, *доступных пониманию* пользователя, представленного парой (T, res).

ОПРЕДЕЛЕНИЕ 10.

i) Если φ — атомарная формула L, то $\varphi \in C(T, \mathrm{res})$ тогда и только тогда, когда $\varphi \in F(L, \mathrm{res})$ и $\varphi \in \mathrm{Thm}(T, \mathrm{res})$;

ii) если φ имеет вид $(\varphi_1 \ \& \ \varphi_2)$, то $\varphi \in C(T, \mathrm{res})$ тогда и только тогда, когда $\varphi \in F(L, \mathrm{res})$, $\varphi_1 \in \mathrm{Thm}(T, \mathrm{res})$ и $\varphi_2 \in \mathrm{Thm}(T, \mathrm{res})$;

iii) если φ имеет вид $(\varphi_1 \vee \varphi_2)$, то $\varphi \in C(T, \mathrm{res})$ тогда и только тогда, когда $\varphi \in F(L, \mathrm{res})$ и выполняется условие: $\varphi_1 \in \mathrm{Thm}(T, \mathrm{res})$ или $\varphi_2 \in \mathrm{Thm}(T, \mathrm{res})$;

iv) если φ имеет вид $(\varphi_1 \to \varphi_2)$, то $\varphi \in C(T, \mathrm{res})$ тогда и только тогда, когда $\varphi \in F(L, \mathrm{res})$ и существует эффективное (вообще говоря, частичное) отображение $A\varphi_1 \to \varphi_2$ из множества всех res–доказательств в T в себя такое, что если x — res-доказательство в T формулы φ_1, то $A\varphi_1 \to \varphi_2$ определено на x и $A\varphi_1 \to \varphi_2(x)$ — res-доказательство в T формулы φ_2;

v) если φ имеет вид $\forall x\varphi_1(x)$, то $\varphi \in C(T, \text{res})$ тогда и только тогда, когда $\varphi \in F(L, \text{res})$ и существует эффективное тотальное отображение $A_{\forall x\varphi_2(x)}$ из множества всех замкнутых res-термов в L во множество всех res-доказательств в T такое, что если t — произвольный замкнутый res-терм в L такой, что $\varphi_1(t) \in F(L, \text{res})$, то $A_{\forall x\varphi_2(x)}(t)$ — res-доказательство в T формулы $\varphi_1(t)$;

vi) если φ имеет вид $\exists x\varphi_1(x)$, то $\varphi \in C(T, \text{res})$ тогда и только тогда, когда $\varphi \in F(L, \text{res})$ и $\varphi_1(t) \in \text{Thm}(T, \text{res})$ для некоторого замкнутого res-терма t в L.

ОПРЕДЕЛЕНИЕ 11. Пусть T — res-формальное исчисление в языке L. Res-формула φ из L **конструктивно res-истинна относительно** T((T,**res**)-**истинна**), если и только если $\varphi \in C(T, \text{res})$. Если при этом пара ($T$, res) рассматривается как репрезентация реального пользователя, то любая (T, res)-истинная формула φ называется также **задачно истинной относительно** (T,**res**).

Подобно определениям 1 и 2 определения 10 и 11 говорят о том, что мы рассматриваем конструктивно res-истинные относительно T формулы в L как коды определенных суждений о системе формальных res-доказательств в T и, следовательно, как коды суждений о формальных T-задачах, доступных пониманию пользователя с интеллектуальными ресурсами res и интеллектуальными запросами, определяемыми T и res.

ОПРЕДЕЛЕНИЕ 12. Пусть L — первопорядковый язык, T — res-формальное исчисление в L. Тогда пара ($L, C(T$,res)) называется **конструктивной res-семантикой языка** L **относительно** (T,**res**), или, кратко, (T,**res**)-**конструктивной семантикой (языка** L). Если при этом (T,**res**) считается репрезентацией реального пользователя, то (T,res)-конструктивная семантика ($L, C(T$,**res**)) называется также **res-семантикой формульных задач для реального пользователя** (T,**res**), или, кратко, (T,**res**)-**задачной семан-**

тикой. Если $(L, C(T,\text{res}))$ — (T,res)-задачная семантика, то (L,res) называется **языком спецификаций формульных задач для любого (при любом T в L) реального пользователя** (T,res).

$(T$–res)-конструктивная семантика $(L, C(T, \text{res}))$ языка L называется **рекурсивно–перечислимой, рекурсивной (разрешимой), конечной,** если соответственно рекурсивно перечислимо, рекурсивно, или конечно множество $C(T, \text{res})$.

ТЕОРЕМА 14. *Если сигнатура σ языка L конечна, то для любого* res*-формального исчисления T в L (T, res)-конструктивная семантика $(L, C(T, \text{res}))$ конечна.*

Эта теорема очевидна. Более того, очевиден универсальный переборный алгоритм, который по любому res–формальному исчислению T в языке L конечной сигнатуры σ строит соответствующее множество $C(T, \text{res})$. Таким образом, имеется принципиальная возможность учесть интеллектуальные запросы реального пользователя во всех практически значимых случаях, пользуясь только определениями 10 и 11, чего нельзя было сказать применительно к запросам идеального пользователя. Последующие определения и теоремы, являясь аналогами результатов § 2 и 3, полезны как дополнительная информация, позволяющая упомянутый универсальный переборный алгоритм построения $C(T, \text{res})$ для конечных сигнатур преобразовать в тех или иных конкретных обстоятельствах в более экономный.

2. **Res–логики.** Пусть язык L и тройка res–чисел фиксированны. Аналогом теоремы 1 является следующая очевидная

ТЕОРЕМА 15. *Если* res*-формальное исчисление T в языке L* res*-противоречиво, то любая* res*-формула в L (T, res)-истинна.*

Обозначим через $\bar{\tau}$ произвольный (непустой) класс res–формальных исчислений в L.

ОПРЕДЕЛЕНИЕ 13. Множество $L(\bar{\tau}, \text{res})$ всех тех и только тех res-формул из L, каждая из которых конструктивно res–истинна относительно каждого T из $\bar{\tau}$, называется **res–конструктивной логикой класса** $\bar{\tau}$, или, кратко, ($\bar{\tau}$, **res)-логикой** (в L). Если при этом пара ($\bar{\tau}$, res) считается репрезентацией соответствующего сообщества π реальных пользователей относительно Σ (т.е. если $\bar{\tau} = \{\Sigma(p)/p \in \pi\}, \mathrm{res} = \mathrm{RES}(\pi, \Sigma)$), то ($\bar{\tau}$, res)-логика $L(\bar{\tau}, \mathrm{res})$ называется также **res-логикой спецификаций задач для сообщества π реальных пользователей (относительно Σ)**.

Из этого определения вытекает:

$$L(\bar{\tau}, \mathrm{res} = \bigcap_{T \in \bar{\tau}} C(T, \mathrm{res}).$$

Следовательно, имеет место аналог теоремы 2.

ТЕОРЕМА 16. *Если $\bar{\tau} \subseteq \bar{\tau}'$, то $L(\bar{\tau}', \mathrm{res}) \subseteq L(\bar{\tau}, \mathrm{res})$.*

Кроме того, имеет место аналог теоремы 3:

ТЕОРЕМА 17. *Если $\bar{\tau}$ — произвольный класс res-формальных исчислений в L, T — res-противоречивое res-формальное исчисление в L, и $\bar{\tau}' = \bar{\tau} \cup \{T\}$, то $L(\bar{\tau}, \mathrm{res}) = L(\bar{\tau}', \mathrm{res})$.*

Ясно, что (T, res) — логики $L(T, \mathrm{res})$ как множества (res-формул) частично упорядочны по включению. Из теоремы 14 следует, что этот порядок имеет наибольший элемент и им яляется множество $F(L, \mathrm{res})$, совпадающее, в силу теоремы 16, с любой ($\bar{\tau}_{\mathrm{inc}}$, res)-логикой, где $\bar{\tau}_{\mathrm{inc}}$ — произвольный класс, состоящий только из res-противоречивых исчислений.

Из теоремы 15 вытекает существование наименьшего элемента в рассматриваемом порядке. Этим наименьшим элементом является ($\bar{\tau}_{\mathrm{all}}$, res)-логика, где $\bar{\tau}_{\mathrm{all}}$ — класс всех res-формальных исчислений в языке L.

Если обозначить через $\bar{\tau}_{\mathrm{con}}$ класс всех res-непротиворечивых res-формальных исчислений в L, а через $\bar{\tau}^{*}_{\mathrm{con}}$ — класс всех сильно res-непротиворечивых res-формальных исчислений, то в силу теорем 16 и 17 имеем:

$$L(\bar{\tau}_{\mathrm{all}}, \mathrm{res}) = L(\bar{\tau}_{\mathrm{con}}, \mathrm{res}) \subseteq L(\bar{\tau}^{*}_{\mathrm{con}}, \mathrm{res}).$$

Таким образом, в семействе всех возможных ($\bar{\tau}$, res)-логик (для фиксированных L и res), упорядоченных по включению $\subseteq$, наименьшим элементом является res-конструктивная логика класса $\bar{\tau}_{\rm con}$ (или $\bar{\tau}_{\rm all}$), а наибольшим — множество $F(L, {\rm res})$ всех res-формул в L; $(\bar{\tau}^*_{\rm con}, {\rm res})$-логика занимает промежуточное положение. Аналог теоремы 4 — это

ТЕОРЕМА 18. *Пусть $F_0(L, {\rm res})$ — множество, состоящее точно из всех res-формул в L вида $(\varphi \to \varphi)$. Тогда $L(\bar{\tau}_{\rm con}, {\rm res}) = F_0(L, {\rm res})$.*

ДОКАЗАТЕЛЬСТВО. Включение $F_0(L, {\rm res}) \subseteq L(\bar{\tau}_{\rm con}, {\rm res})$ очевидно. Докажем обратное включение

$$L(\bar{\tau}_{\rm con}, {\rm res}) \subseteq F_0(L, {\rm res}). \tag{2}$$

Допустим противное. Тогда найдется res-формула φ в L такая, что

$$\varphi \in L(\bar{\tau}_{\rm con}, {\rm res}) \tag{3}$$

и

$$\varphi \notin F_0(L, {\rm res}). \tag{4}$$

Если выполняется (4), то имеет место один из следующих случаев:

а) $\varphi = (\varphi_1 \to \varphi_2)$, $\varphi_1 \neq \varphi_2$[5];
б) $\varphi = \forall x \varphi_1(x)$;
в) $\varphi = \exists x \varphi_1(x)$;
г) $\varphi = (\varphi_1 \,\&\, \varphi_2)$;
д) $\varphi = (\varphi_1 \vee \varphi_2)$;
е) φ — атомарная.

Случай "а". Так как φ, по предположению, — res-формула L, то φ_1 и φ_2 — также res-формулы L. Отсюда $F(L, {\rm res}) \supseteq \{\varphi_1, \varphi_2, \varphi_3\}$. Рассмотрим исчисление T_{φ_1} с пустым множеством правил вывода и единственной аксиомой φ_1. Ясно, что T_{φ_1} — res-формальное исчисление в L, и ${\rm Thm}(T_{\varphi_1}, {\rm res}) = \{\varphi_1\} \not\supseteq F(L, {\rm res})$. Отсюда $T_{\varphi_1} \in \bar{\tau}_{\rm con}$. Но относительно T_{φ_1} res-формула $\varphi_1 \to \varphi_2$ не конструктивно res-истинна (т.е. φ не является $(T_{\varphi_1}, {\rm res})$-истинной). Следовательно, $\varphi \notin L(\bar{\tau}_{\rm con}, {\rm res})$, что противоречит (3). Следовательно "а" невозможен.

[5] Запись $\varphi_1 = \varphi_2$ $(\varphi_1 \neq \varphi_2)$ — сокращение для "φ_1 есть (не есть) формула φ_2".

Случай "б". Так как φ, по предположению, — res-формула L, то φ_1 — также res-формула L. Рассмотрим любое исчисление T_{φ_2}, имеющее пустое множество правил вывода и единственную res-аксиому φ_2 такую, что $\varphi_2 \neq \varphi_1$. Такая res-аксиома существует, ибо в качестве нее можно взять саму res-формулу φ. Снова $T_{\varphi_2} \in \bar{r}_{\mathrm{con}}$, но относительно T_{φ_2}, формула φ вида "б" не конструктивно res-истина. Следовательно $\varphi \notin L(\bar{r}_{\mathrm{con}})$, что ведет к противоречию с (3). Следовательно, случай "б" также невозможен.

Аналогичным образом доказывается, что все остальные случаи "в" — "е" невозможны. Поэтому неверно, что выполняется (4). Противоречие. □

Аналогом теоремы 5 является

ТЕОРЕМА 19. *Пусть $F_1(L, \mathrm{res})$ — множество, состоящее точно из всех res-формул вида $(\varphi \to \varphi)$ и всех res-формул вида $(\perp \to \varphi)$ языка L. Тогда $L(\bar{r}^*_{\mathrm{con}}, \mathrm{res}) = F_1(L, \mathrm{res})$.*

Мы не приводим доказательства этой теоремы, так как оно получается из предыдущего почти дословным повторением.

3. **($\bar{r}_{\mathrm{sym}}$,res)-логики**. Пусть, как и прежде, $|\sigma|$ — множество сигнатурных символов языка L. И пусть g — произвольная перестановка на $|\sigma|$ такая, что она каждый предикатный символ переводит в предикатный, а каждый функциональный — в функциональный, причем местности символов обеих видов сохраняются. Очевидно, что каждая перестановка g на $|\sigma|$ индуцирует перестановку g' на $F(L, \mathrm{res})$. Очевидно также, что множества всех таких перестановок g и g' образуют группы $G(\sigma)$ и $G'(L, \mathrm{res})$ соответственно.

Мы говорим, что перестановка g' на $F(L, \mathrm{res})$ **res-переводит** правило вывода

$$\frac{\varphi_0, \ldots, \varphi_n}{f(\varphi_0, \ldots, \varphi_n)}$$

в правило вывода

$$\frac{\varphi'_0,\ldots,\varphi'_n}{f'(\varphi'_0,\ldots,\varphi'_n)},$$

если для всех $\varphi_0,\ldots,\varphi_n$, $\varphi'_0,\ldots,\varphi'_n \in F(L,\mathrm{res})$ выполняется равенство $f'(\varphi'_0,\ldots,\varphi'_n) = g'f(\varphi_0,\ldots,\varphi_n)$ всякий раз, когда $\varphi'_0 = g'\varphi_0,\ldots,\varphi_n = g'\varphi_n$.

ОПРЕДЕЛЕНИЕ 14. Пусть T — исчисление в L. Будем называть **res–группой (сигнатурных) симметрий** T подгруппу $H(T,\mathrm{res})$ группы $G(\sigma)$ такую, что индуцированная подгруппа $H'(T,\mathrm{res})$ группы $G'(L,\mathrm{res})$ состоит в точности из всех перестановок g', res–переводящих правила вывода T в правила вывода T, а аксиомы T — в аксиомы T (аксиомы рассматриваются при этом как правила вывода с пустыми множествами посылок). Если $H(T,\mathrm{res})$ не единична, то исчисление T называется **(сигнатурно) res–симметричным**.

Пусть sym — произвольная эффективно перечислимая подгруппа группы $G(\sigma)$. Тогда через $\bar{\tau}_{\mathrm{sym}}$ будем обозначать класс всех сильно res–непротиворечивых res–формальных исчислений T в L таких, что $H(T,\mathrm{res}) = \mathrm{sym}$. Очевидно, что для любой пары (L,res) такой, что $F(L,\mathrm{res}) \supseteq \{\perp\}$ и $F(L,\mathrm{res}) \neq \{\perp\}$, и любой указанной sym класс $\bar{\tau}_{\mathrm{sym}}$ не пуст. Ему, например, принадлежит исчисление $T^{\varphi}_{\mathrm{sym}}$, имеющее пустое множество правил вывода и эффективно перечислимое множество аксиом $\mathrm{sym}(\varphi,\mathrm{res}) = \{g'\varphi | g \in \mathrm{Sym}\}$, где φ — произвольная фиксированная res-формула языка L, отличная от $\perp$.

Аналогом теоремы 6 служит

ТЕОРЕМА 20. *Для любой эффективно перечислимой подгруппы* sym *группы* $G(\sigma)$ *логика* $L(\bar{\tau}_{\mathrm{sym}},\mathrm{res})$ *состоит в точности из всех* res–*формул вида* $(\perp\to\varphi)$ *и всех* res–*формул вида* $(\varphi\to g'\varphi)$, $g \in \mathrm{sym}$.

ДОКАЗАТЕЛЬСТВО. Очевидно, что все res–формулы указанных видов принадлежат $L(\bar{\tau}_{\mathrm{sym}},\mathrm{res})$. Покажем, что $L(\bar{\tau}_{\mathrm{sym}},\mathrm{res})$ принадлежат только они. Допустим противное.

Тогда найдется res-формула Ψ в L такая, что:

$$\Psi \in L(\bar{r}_{\mathrm{sym}}, \mathrm{res}); \tag{5}$$

$$\Psi \notin \{(\perp \to \varphi) \mid \varphi \in F(L, \mathrm{res})\}; \tag{6}$$

$$\Psi \notin \{(\varphi_1 \to \varphi_2) \mid \varphi_1 \in F(L, \mathrm{res}),\ \varphi_2 \in \mathrm{sym}(\varphi_1, \mathrm{res})\}. \tag{7}$$

Из (6) и (7) следует, что: либо

а) $\Psi = (\varphi_1 \to \varphi_2)$, $\varphi_1 \neq \perp$, $\varphi_2 \notin \mathrm{sym}(\varphi_1, \mathrm{res})$, $\Psi, \varphi_1, \varphi_2 \in$ $\in F(L, \mathrm{res})$, либо

б) $\Psi = \forall x\ \varphi(x)$, φ, $\Psi \in F(L, \mathrm{res})$, либо

в) $\Psi = \exists x\ \varphi(x)$, φ, $\Psi \in F(L, \mathrm{res})$, либо

г) $\Psi = (\varphi_1\ \&\ \varphi_2)$, φ_1, φ_2, $\Psi \in F(L, \mathrm{res})$, либо

д) $\Psi = (\varphi_1 \vee \varphi_2)$, φ_1, φ_2, $\Psi \in F(L, \mathrm{res})$, либо

е) Ψ — атомарная, $\Psi \in F(L, \mathrm{res})$.

Случай "а". Рассмотрим исчисление $T_{\mathrm{sym}}^{\varphi_1}$. С одной стороны, $T_{\mathrm{sym}}^{\varphi_1} \in \bar{r}_{\mathrm{sym}}$, с другой, res-формула Ψ вида "а" не $(T_{\mathrm{sym}}^{\varphi_1}, \mathrm{res})$-истинна. Следовательно, в рассматриваемом случае $\Psi \notin L(\bar{r}_{\mathrm{sym}}, \mathrm{res})$. Это противоречит (5). Следовательно, случай "а" не имеет места.

Случай "б". Рассмотрим исчисление $T_{\mathrm{sym}}^{\varphi_1}$, где $\varphi_1 \neq$ $\neq \varphi$, $\varphi_1 \neq \perp$ (например, $\varphi_1 = \forall x\ \varphi(x)$). Снова $T_{\mathrm{sym}}^{\varphi_1} \in \bar{r}_{\mathrm{sym}}$, но формула Ψ вида "б" не $(T_{\mathrm{sym}}^{\varphi_1}, \mathrm{res})$-истинна. Следовательно, $\Psi \notin L(\bar{r}_{\mathrm{sym}}, \mathrm{res})$, что противоречит (5). Следовательно, случай "б" не может иметь места.

Аналогичным образом доказывается, что все остальные случаи "в"-"е" также невозможны. Следовательно, неверно, что выполняются (6) и (7). Противоречие с допущением. □

Из этой теоремы видно, что $F_1(L, \mathrm{res}) \subsetneq L(\bar{r}_{\mathrm{sym}}, \mathrm{res}) \subset$ $\subset F(L, \mathrm{res})$, причем равенство $L(\bar{r}_{\mathrm{sym}}, \mathrm{res}) = F_1(L, \mathrm{res})$ достигается всякий раз, когда sym — единичная подгруппа группы $G(\sigma)$.

4. **Немонотонность некоторых $(\bar{r}$,res)-логик.** Пока мы исследовали res-конструктивные логики весьма широких классов res-формальных исчислений (логики спецификаций задач для весьма широких сообществ реальных пользователей). Как и в п.4 § 2, перейдем теперь к изучению

res-конструктивных логик наименьших возможных классов — единичных (логик спецификаций задач для одного реального пользователя).

ОПРЕДЕЛЕНИЕ 15. Пусть T_1 и T_2 — произвольные res-формальные исчисления в L. Будем говорить, что T_2 **res-больше** T_1 (писать $T_1 \overset{\text{res}}{\leq} T_2$), если и только если все res-аксиомы T_1 являются res-аксиомами T_2 и все правила вывода T_1 являются правилами вывода T_2.

Отношение "res-больше" — частичный порядок на $\bar{\tau}_{\text{all}}$ (и, следовательно, на $\bar{\tau}_{\text{con}}$). Поэтому правомерен вопрос о монотонности $(\{T\}, \text{res})$-логик (res-конструктивных логик единичных классов $\{T\}$ res-формальных исчислений T) относительно $\overset{\text{res}}{\leq}$. Имеет место следующий аналог теоремы 7.

ТЕОРЕМА 21. *Пусть L и $\text{res} = (m_1, m_2, m_3)$ таковы, что $G(\sigma)$ имеет неединичную эффективно перечислимую подгруппу* sym, *и $m_2 \geq 4n + 9$, где n — длина некоторой формулы φ_1 языка L, причем $\varphi_1 \neq g_0'\varphi_1$ для некоторой перестановки $g_0 \in$* sym. *Тогда существуют такие T_1 и T_2, принадлежащие $\bar{\tau}_{\text{con}}$ (в L), что $T_1 \overset{\text{res}}{\leq} T_2$, но не $L(\{T_1\}, \text{res}) \subseteq L(\{T_2\}, \text{res})$ и не $L(\{T_2\}, \text{res}) \subseteq$ $\subseteq L(\{T_1\}, \text{res})$.*

ДОКАЗАТЕЛЬСТВО. Возьмем в качестве T_1 исчисление $T_{\text{sym}}^{\varphi_1}$, имеющее пустое множество правил вывода и множество аксиом

$$\text{sym}(\varphi_1, \text{res}) = \{g'\varphi_1 | g \in \text{sym}\}, \tag{8}$$

где φ_1 — указанная в условии формула L длины n, удовлетворяющая условию

$$\varphi_1 \neq g_0'\varphi_1 = \varphi_2. \tag{9}$$

Очевидно $T_{\text{sym}}^{\varphi_1}$ — res-симметрично, res-формально и принадлежит $\bar{\tau}_{\text{con}}$. Пусть

$$\varphi_3 = (\varphi_1 \ \& \ \varphi_2). \tag{10}$$

По условию теоремы φ_3 имеет длину $2n+3 < m_2$, ибо φ_2 получена перестановкой g_0' из φ_1 и, следовательно, имеет длину, как и φ_1, равную n, кроме того, в записи φ_3 нужно еще учесть две скобки и символ конъюнкции. Таким образом, φ_3 — res-формула L.

Учитывая (8)-(10), имеем

$$\varphi_3 \notin \mathrm{sym}(\varphi_1, \mathrm{res}) = \mathrm{Thm}(T_1, \mathrm{res}); \tag{11}$$

$$g_0'\varphi_3 \notin \mathrm{sym}(\varphi_1, \mathrm{res}); \tag{12}$$

$$g_0'\varphi_3 \neq \varphi_3; \tag{13}$$

$$\varphi_3 \in F(L, \mathrm{res}). \tag{14}$$

Выберем в качестве T_2 исчисление $T_{\mathrm{sym}}^{\varphi_1} + {}^{\varphi_3}$, имеющее пустое множество правил вывода и множество аксиом

$$\mathrm{sym}(\varphi_1, \mathrm{res}) + \varphi_3 = \mathrm{sym}(\varphi_1, \mathrm{res}) \cup \{\varphi_3\}. \tag{15}$$

Ясно, что T_2 res-формально и также принадлежит $\bar{r}_{\mathrm{con}}$.

Рассмотрим формулу $\varphi_4 = (\varphi_3 \,\&\, \varphi_1)$. Ее длина равна $3n + 5 < m_2$ и поэтому φ_4 — res-формула L. В силу (11) очевидно, что φ_4 не (T_1, res)-истинна, следовательно,

$$\varphi_4 \notin L(\{T_1\}, \mathrm{res}). \tag{16}$$

С другой стороны, φ_4 конструктивно res-истинна относительно T_2, так как φ_3 и φ_1 обе принадлежат $\mathrm{Thm}(T_2, \mathrm{res})$ в силу (15). Поэтому

$$\varphi_4 \in L(\{T_2\}, \mathrm{res}). \tag{17}$$

Из (16) и (17) следует, что

$$L(\{T_2\}, \mathrm{res}) \not\subseteq L(\{T_1\}, \mathrm{res}). \tag{18}$$

Рассмотрим формулу $\varphi_5 = (\varphi_3 \rightarrow g_0'\varphi_3)$. Ее длина равна $4n + 9 \leq m_2$, отсюда φ_5 — res-формула L. Из (11) и (12) следует, что φ_5 конструктивно res-истинна относительно T_1. Поэтому

$$\varphi_5 \in L(\{T_1\}, \mathrm{res}). \tag{19}$$

В силу (13), (14) res-формула $g'_0\varphi_3$ не принадлежит Thm(T_2,res), а в силу (15) res-формула φ_3 принадлежит Thm(T_2, res). Отсюда

$$\varphi_5 \notin L(\{T_2\}, \text{res}). \tag{20}$$

Из (19) и (20) вытекает

$$L(\{T_1\}, \text{res}) \not\subseteq L(\{T_2\}, \text{res}). \tag{21}$$

Наконец, прямо из определений T_1 и T_2 следует, что

$$T_1 \overset{\text{res}}{\leq} T_2 \tag{22}$$

В силу (18),(21),(22) теорема доказана. □

§5. Представления логик спецификаций для реальных пользователей

1. **Представления ($\bar{\tau}$,res)-логик.**

ОПРЕДЕЛЕНИЕ 16. Если $L(\bar{\tau}, \text{res})$ — res-конструктивная логика класса $\bar{\tau}$, и $K \subseteq L(\bar{\tau}, \text{res})$, то K называется **фрагментом ($\bar{\tau}$,res)-логики $L(\bar{\tau}$,res)**.

ОПРЕДЕЛЕНИЕ 17. Если res-формальное исчисление T в L таково, что множество его res-теорем является фрагментом res-конструктивной логики (является res-конструктивной логикой) класса $\bar{\tau}$, то T называется **(тотальным) res-формальным представлением ($\bar{\tau}$,res)-логики**.

Пусть X — произвольное множество res-формул языка L, и пусть T_X означает ичисление в L, имеющее пустое множество правил вывода и множество аксиом X. Пусть $F(L, \text{res}), F_0(L, \text{res}), F_1(L, \text{res}), \text{sym}(\varphi, \text{res})$ — ранее определенные множества res-формул. Тогда из теорем 15,17 и 18 непосредственно следует, что $T_{F(L, \text{res})}$ является тотальным res-формальным представлением любой ($\bar{\tau}_{\text{nc}}$, res)-логики, а $T_{F_0(L, \text{res})}$ — тотальным res-формальным представлением как для ($\bar{\tau}_{\text{all}}$, res)-логики, так и для ($\bar{\tau}_{\text{con}}$, res)-логики. Теорема 19 говорит о том, что $T_{F_1}(L,\mathrm{res})$ — тотальное res-формальное представление $(\bar{r}^*_{\mathrm{con}},\mathrm{res})$-логики. Теорема 20 утверждает, что $T_{\mathrm{sym}(\varphi,\mathrm{res})}$ есть тотальное res-формальное представление для $(\bar{r}_{\mathrm{sym}},\mathrm{res})$-логики.

Таким образом, имеет место (аналог теоремы 8)

ТЕОРЕМА 22. *Существуют тотальные* res-*формальные представления* $(\bar{r}_{\mathrm{all}},\mathrm{res})$-, $(\bar{r}_{\mathrm{inc}},\mathrm{res})$-, $(\bar{r}_{\mathrm{con}},\mathrm{res})$-, $(\bar{r}^*_{\mathrm{con}},\mathrm{res})$-*логик. Если* sym — *эффективно-перечислимая подгруппа группы* $G(\sigma)$, *то существует тотальное* res-*формальное представление* $(\bar{r}_{\mathrm{sym}},\mathrm{res})$-*логики.*

2. **Res–конструктивные системы.**

ОПРЕДЕЛЕНИЕ 18. Если res-формальное исчисление T в L является res-формальным представлением $(\{T\},\mathrm{res})$-логики, то T называется **res–автоконструктивной** (или просто **res–конструктивной**) **системой**. Если при этом (T,res) репрезентирует реального пользователя, то T называется **res–автозадачной** (или просто **res–задачной**) **системой этого пользователя**. Res-конструктивная (res–задачная) система T **res–точна**, если T — тотальное res-формальное представление $(\{T\},\mathrm{res})$-логики.

Аналогично тому, как это было в первой части работы, из формулы $L(\bar{r},\mathrm{res})=\bigcap\limits_{T\in\bar{r}} C(T,\mathrm{res})$ вытекает, что для любого res-формального исчисления T $(\{T\},\mathrm{res})$-логика совпадает с $C(T,\mathrm{res})$. Поэтому определение 18 говорит, в частности, о том, что понятные реальному пользователю сведения об интересующей его предметной области могут одновременно рассматриваться как некоторые истинные утверждения о понятных для него задачах, относящихся к подразумеваемой предметной области, *если эти сведения T являются res-задачной (res-конструктивной) системой.* Эти сведения могут рассматриваться как все истинные утверждения об упомянутых понятных пользователю задачах, *если T — res-точная res-задачная (res-точная res-конструктивная) система.*

Если res-формальное исчисление T не является (res-конструктивной) res-задачной системой, то T

среди своих res–теорем содержит ложное утверждение о каких-то (T, res)–задачах.

Очевидно, например, что исчисление $T_{F(L,\mathrm{res})}$ есть res–точная res–задачная (res–точная res–конструктивная) система. В то же время, при выполнении некоторых слабых условий, классическое исчисление предикатов или, скажем, арифметика Пеано не являются, как показывают две нижеследующие теоремы, res–конструктивными (res–задачными) системами.

ТЕОРЕМА 23. *Пусть L и* res *таковы, что формула φ вида $(\varphi_1 \vee (\varphi_1 \to \perp))$ — res–теорема (где φ_1 — атомарная формула L, отличная от $\perp$ и аксиом равенства) исчисления* $\mathrm{CPC}(L)$ *(классического исчисления предикатов в L). Тогда* $\mathrm{CPC}(L)$ — *не* res–*конструктивная система.*

ДОКАЗАТЕЛЬСТВО. Имеем: $\mathrm{CPC}(L) \nvdash \varphi_1$, $\mathrm{CPC}(L) \nvdash (\varphi_1 \to \perp)$. Следовательно, res–формула φ не конструктивно истинна относительно $\mathrm{CPC}(L)$. Тем более, она не конструктивно res–истинна относительно $\mathrm{CPC}(L)$. С другой стороны, по условию, $\varphi \in \mathrm{Thm}(\mathrm{CPC}(L), \mathrm{res})$. □

ТЕОРЕМА 24. *Исчисление PA (первопорядковая арифметика Пеано) — не* res–*конструктивная система, если язык исчисления PA и* res *таковы, что формула φ вида $(\varphi_1 \vee (\varphi_1 \to \perp))$, — res–теорема T, где φ_1 — какая-то неразрешимая в PA формула.*

ДОКАЗАТЕЛЬСТВО этой теоремы аналогично предыдущему.

З а к л ю ч е н и е

Теоремы 23 и 24 аналогичны теоремам 8 и 9 соответственно. Таким образом, в § 4 и 5 установлены аналоги для теорем 1–9. Однако на данной стадии проекта не ясно, имеются ли и, если имеются, то как следует сформулировать, подходящие "res–аналоги" для теорем 10–13. В какой степени это обстоятельство снижает практическую значимость материалов § 4 и 5 — вопрос, на который невозможно ответить априори, до проведения экспериментальных исследований по установлению интеллектуальных ресурсов реальных людей. Быть может, фактически окажется, что эти ресурсы так малы, что удастся во многих практически важных случаях опереться непосредственно на теорему 14 и на соответствующую малым значениям m_1, m_2, m_3 ($\mathrm{res} = (m_1, m_2, m_3)$) ограниченную версию, упомянутую в связи с теоремой 14, универсального алгоритма построения (T, res)-задачных семантик.

Этот вопрос определяет направление ближайших шагов разработки проекта.

Литература

1. САМОХВАЛОВ К.Ф. Экспериментальная процедура для установления зависимости степени убедительности доказательств от их длины //Методологические и технологические проблемы информационно-логических систем. — Новосибирск, 1988. — Вып. 125: Вычислительные системы. — С. 162–169.

2. САМОХВАЛОВ К.Ф. Относительно конструктивные системы //Языки спецификаций и логическое программирование. — Новосибирск, 1988. — Вып. 124: Вычислительные системы. — С. 99–113.

3. ESENIN-VOLPIN A. On the ultra — intuitionistic foundations of mathematics //Infinitistic methods. — Oxford: Pergamon Press, 1961. — P. 201–223.

4. ВОПЕНКО П. Математика в альтернативной теории множеств. — М., 1983. — 152 с.

5. DUMMETT M. Wang's paradox //Synthese. — 1975. — Vol. 30. — P. 301–324.

Поступила в редакцию
25 декабря 1996 года