

И.Б.Микиргумов

СТРУКТУРА ЗНАЧЕНИЯ И КОМПЕТЕНТНОСТЬ СУБЪЕКТА В ЛОГИКЕ СМЫСЛА И ДЕНОТАТА *

Abstract. *The main issue of the paper is the realisation of Alternative (0) within the logic of sense and denotation (LSD). The structural analysis of meaning and the notion of logical (semantical) competence of subject are suggested as the approaches for solution, that provide epistemic character for the interpretation of LSD.*

Проблема Альтернативы (0)

Созданная Алонзо Чёрчем в начале 50-х годов логика смысла и денотата (ЛСД)¹, ставшая основанием общей интенциональной логики, включала три содержательные альтернативы: Альтернатива (0) – смыслы двух синтаксически различных выражений всегда различны; Альтернатива (1) – λ -конверсия не затрагивает смысл; Альтернатива (2) – смыслы двух синтаксически различных выражений могут совпадать. Работы по ЛСД самого Чёрча затрагивают главным образом варианты Альтернатив (1) и (2). Версию Альтернативы (0), так называемую логику синонимического изоморфизма, предложил ученик Чёрча Энтони Эндерсон [2]. В ней ключевая для Альтернативы (0) проблема различения смысла выражений вида $A(x // B)$ и $(\lambda x A x)B$ решается за счёт сопоставления им в качестве смыслов классов синонимического изоморфизма, т. е. классов выражений, которые могут быть получены друг из друга, посредством переименования связанных переменных. Поскольку синтаксически $A(x // B)$ и $(\lambda x A x)B$ различны, различны и соответствующие классы.

Чёрч сформулировал Альтернативу (0), исходя из первоначального замысла ЛСД [3], который был шире идеи применения ЛСД для анализа интенциональных контекстов. А именно предполагалось, что ЛСД будет равным образом применима как к естественному, так и к формализованному языку и в конечном счете окажется системой оснований математики, базирующейся на формализованной теории значения. Еще раньше на параллельном пути

* Работа выполнена при поддержке ZEIT-Stiftung Ebelin und Gerd Bucerius.

¹ Подробное описание языка и свойств ЛСД см. в статьях Чёрча [4], [5], [6] и Эндерсона [2].

потерпело неудачу, оказавшись противоречивым, созданное Чёрчем бестиповое λ -исчисление. Но сочетание λ -конверсии с простой теорией типов [7], как доказал позднее Генкин [8]), дает непротиворечивую и относительно полную теорию. Добавление к этой теории интенциональных типов образует ЛСД, в которой концепт истинностного значения всегда имеет концепт, что ведет к появлению иерархии, подобной иерархии предикатов истинности и выполнимости. Предчувствуя неминуемые сложности, и, возможно, сомневаясь в реализуемости своего плана, Чёрч уже в первой развернутой формулировке ЛСД оставляет в стороне наиболее радикальную Альтернативу (0), предлагая для ее интерпретации синтаксически ориентированную теорию синонимического изоморфизма, и исследует Альтернативы (1) и (2), которые позднее легли в основу интенциональных систем Каплана, Монтегю и других авторов. В итоге, кроме теории Эндерсона, Альтернатива (0) не имеет иной содержательно прозрачной интерпретации. На наш взгляд, из-за чрезмерной строгости полная реализация Альтернативы (0) вряд ли может быть полезна, но ее ключевая идея, а именно различение смысла выражений до и после λ -конверсии заслуживает внимания и будет обсуждаться ниже.

Смысл (концепт) как процедура

Языком ЛСД является язык простой теории типов порядка ω , который расслоён на интенциональные уровни. Интенциональному уровню 0 соответствуют сущности языка простой теории типов, интенциональный уровень 1 содержит смыслы сущностей уровня 0, интенциональный уровень 2 содержит смыслы сущностей уровня 1 и т. д. Определены также все функции через уровни, что, правда, порождает ряд проблем, которые мы не будем здесь рассматривать.

Основанием интерпретации ЛСД является трактовка понятия “смысл” (“концепт”). Под смыслом мы будем понимать внелингвистическую процедуру (операцию) сопоставления выражению денотата. Такие процедуры в языке ЛСД репрезентированы на ненулевых интенциональных уровнях как функции от выражений некоторого языка-объекта в область денотатов. В ЛСД каждая лингвистическая единица имеет смысл. Так, если A - имя, а $[A]$ имя смысла A , то $[[A]]$ - имя смысла имени смысла A и т. д. Лингвистическое оправдание такой свободы порождения интенциональных сущностей состоит в том, что мы легко образуем имена вида

“смысл выражения A ”,

«СМЫСЛ выраж ения “СМЫСЛ выраж ения А” »,
 "СМЫСЛ выраж ения «СМЫСЛ выраж ения
 “СМЫСЛ выраж ения А”»"

и т. д. и всегда можем адекватно их понять. Согласно известному определению Чёрча, знать смысл суждения (предложения) означает знать условия, при которых оно истинно. Другая формулировка может звучать так: смысл суждения – это процедура проверки его истинности. Это определение мы модифицируем, следуя известной идее Фреге: смысл суждения – это процедура проверки суждения, выраженная его утверждением. Легко заметить, что введение понятия “утверждение” делает анализ смысла анализом специфических пропозициональных установок.

Смысл именного выражения – это внелингвистическая процедура, отделяющая одни объекты от других. Очевидно, что может быть сколько угодно разных способов задания одного и того же денотата для имени, т. е. для того, чтобы указывать на некоторый объект, имя может получить любой из бесконечного числа смыслов. Смысл сложного выражения в соответствии с принципом композициональности есть функция от смыслов его компонент, поэтому смысл предложения, в котором компоненты могут иметь каждый раз новый смысл, также будет в каждом случае новым.

Структура и двойственность значения

Попытаемся различить смысл выражений $(\lambda y. f y)(x)$ и $f x$, где f – примитивная константа, исходя из способа установления значения выражений, как он представлен в теоретико-типовой семантике ЛСД. Рассмотрим выражения $(\lambda x y. x + y)(2)(3)$ и $(2 + 3)$. Денотат первого определяется так:

$$\begin{aligned} \text{val}_v((\lambda x y. x + y)(2)(3)) &= \text{val}_v(\lambda x y. x + y)(\text{val}_v 2)(\text{val}_v 3) \\ &= g a b \end{aligned}$$

где функция g , такая, что для любых a и b , $(g a) b = \text{val}_{v^*}(x + y)$, где $v^* = v(x/a, y/b)$. Точно так же,

$$\begin{aligned} \text{val}_v(2 + 3) &= \text{val}_v(+)(\text{val}_v 2)(\text{val}_v 3) \\ &= g a b, \end{aligned}$$

где относительно g мы не можем сказать, чем она является, пока мы не знаем, обозначает ли выражение “+” примитивную константу или является λ -выражением. В любом случае, раскрывая определение функции “+” до тех пор, пока мы не придём к примитивным константам и переменным, мы сделаем одинаковое число шагов, начав с $\text{val}_{v^*}(x + y)$ и начав с $\text{val}_v(+)$. Это значит, что процедура установления значения первого выражения, помимо необхо-

димости выполнять условие для g , на шаг длиннее. В самом деле, пусть

$$(A + B) =_{\text{df}} \text{plus}(A, B)$$

где plus есть примитивная константа. Тогда

$$(2 + 3) \text{ есть сокращение для } \text{plus}(2)(3)$$

а

$$(\lambda x y. x + y)(2)(3) \text{ есть сокращение для } (\lambda x \lambda y. \text{plus}(x, y))(2)(3)$$

и для того, чтобы в последнем случае дойти до значения константы plus , требуется сделать лишний шаг и ввести дополнительную функцию. Экстенционально, функциональные константы plus и $(\lambda x \lambda y. \text{plus}(x, y))$ тождественны, но процессы анализа их значения различаются.

Рассматривая семантические определения как описания структуры, т. е. ряда процедур (программы) установлении значения, мы для таких процедур, соответствующих A и B , где $A \text{ conv } B$, обнаруживаем, во-первых, что они содержат различное число шагов, и, во-вторых, что в одном случае мы можем иметь дело непосредственно с примитивной константой, а не с λ -выражением. Можно ли отсюда получить основание для различия смысла? Да, и этот метод не является новым. Впервые он был использован Дэвидом Льюисом [9], который отождествил смысл с деревом семантических операций, определяющих денотат. При таком анализе можно различить смысл даже для выражений A и $\neg\neg A$, хотя такое различие и будет слишком сильным. Но чем тогда окажутся сами такие деревья или описания семантических процедур? Ниже мы увидим, что они станут элементами специфических пропозициональных установок субъекта.

Здесь неожиданно возникает проблема дифференциации смысла внутри- и внелогических единиц языка. Определение истинности для отрицания и обычный смысл слова “лошадь” далеко отстоят друг от друга, а главное, по-разному осознаются и используются носителями языка. В семантике формализованного языка мы получаем две независимые группы семантических процедур, одна из которых описывает, в частности, λ -конверсию, а другая имеет дело с нелогическими константами. Отождествление смыслов выражений оказывается при этом зависящим от того, какая группа семантических процедур задействована. Если обычная семантика λ -исчисления сопоставляет формулам в качестве значений классы эквивалентности по метаязыковому отношению λ -конверсии conv или классы λ -эквивалентности, то в семантике ЛСД каждому типу сопоставлен домен сущностей, который в общем случае и, например, для Альтернативы (0) не является

классом λ -эквивалентности. Простое совмещение двух семантик невозможно, и нам потребуется сохранить эту двойственность, превратив процедуры одной семантики в объекты языка, интерпретируемые другой семантикой. Здесь нам на помощь приходит анализ естественного языка, в котором наличие параллельных способов интерпретации обычно. Итак, чему может соответствовать λ -абстракция в естественном языке?

λ -абстракция и естественный язык. Компетентный субъект

Запись $(\lambda xPx)u$ есть “приложение функции к аргументу”, а запись Pu есть “результат корректной подстановки u в P вместо всех свободных вхождений x ”. Пусть для некоторых интерпретации и означивания P соответствует предложению

(1) *Сократ бел*

На роль предложений, соответствующих $(\lambda xPx)u$, могут претендовать два выражения естественного языка:

(2) ***Значение функции “белый” для аргумента “Сократ”***

и

(2') ***Приписывание Сократу свойств в быт в белым***

(2) будем называть экстенциональной переформулировкой (1) в метаязыке или экстенциональной λ -абстракцией, а (2') будем называть интенциональной переформулировкой и, соответственно, интенциональной λ -абстракцией. Жирным шрифтом выделены существенные для данного фрагмента метаязыка термины. В (2) это “значение”, “функция”, “аргумент”, а в (2') - это “свойство”, “приписывать”. Какая из формулировок более подходит на роль аналога применения функции к аргументу в естественном языке? Ответ на этот вопрос зависит от того, какой уровень компетентности мы будем предполагать для субъекта, понимающего высказывание (1).

Поместим (1), (2) и (2') в косвенный контекст:

(3) *Саша полагает, что Сократ бел*

(4) *Саша полагает, что о **значении функции “белый” для аргумента “Сократ”** есть в “И”*

(4') *Саша полагает, что Сократ у **правильно будет приписать в свойств во быт в белым***

В (4) слова “есть “И”” и “правильно” в (4') появляются в связи с тем, что пропозициональная установка “полагает” вводит утверждение. В какой степени выражения (3), (4) и (4') имеют сходный смысл и будет ли замена (1) на (2) или (2'), например, в (3) допустима? Как известно, во всех случаях, когда мы сталкиваемся с

пропозициональными установками, возможность замены тождественных имен либо вовсе исключается, либо, в особых случаях, ограничена теми тождествами, которые также являются объектом пропозициональной установки. Например,

- (5) *Алиса полагает, что о Лондон - ст олицы Париж а*
(6) *Алиса полагает, что о Париж - ст олицы Рима*

(7) *Алиса полагает, что о Лондон ст олицы ст олицы Рима*

Переход от (5) и (6) к (7) не является безусловным, если не предполагается достаточная языковая и логическая компетентность Алисы. В самом деле, Алиса, полагая (5) и (6), может, например, впопыхах, не успеть сделать вывод (7). Легко воспроизвести подобную ситуацию уже со многими посылками, когда Алиса, например, знает условия какой-то логической задачи, но еще не вывела следствия. Поэтому естественно устранить или по крайней мере ослабить пропозициональную установку, разрешив, например, переход от (5) и (6) к

- (7') *Алиса не ст анет от рицат ь, что о Лондон - ст олицы ст олицы Рима,*

предполагая, тем самым, что Алиса столь же компетентна в оценке корректности этого перехода, как и любой другой субъект. “Не станет отрицать” – это одна из возможностей. Другие таковы: “готова (должна) признать”, “признает”, “может (должна) понимать”, “понимает”, “поймет” и т. п.

Интересен ли для анализа субъект, лишенный любой семантической или логической компетентности? Видимо, нет. Во-первых, способность оценивать некоторые финитные рассуждения как правильные, не является логическим всеведением, т. е. минимальная логическая или семантическая компетентность субъекта пропозициональной установки – это очень слабое требование. Во-вторых, если считать субъект свободным от любых логических обязательств, то он сможет отрицать истинность, например, предъявленного ему суждения

Лондон - эт о Лондон.

Мир мнения такого субъекта будет “слишком” невозможным. В-третьих, отсутствие логической компетентности ограничит сферу знания субъекта некоторым случайным набором пропозиций, хотя он, будучи “по определению” способным к познанию, может рассмотреть в связи с известными ему пропозициями любую пропозицию. Слабость требования минимальной логической компетентности субъекта проявляется также и в том, что приняв его, мы сохраняем полную свободу при выборе метода описания мира мнения и пропозициональных установок.

Вернемся к примерам (3), (4) и (4'). При каких условиях (3) истинно, а, например, (4') – ложно, т. е. при каких условиях Саша будет полагать, что Сократ бел и не будет полагать, что Сократу правильно будет приписать свойство быть белым? Если Саша обладает минимальной логической компетентностью, т. е. понимает смысл фраз (1) и (2'), но не считает их взаимозаменяемыми, то, очевидно, он не знает принципа

$$A \Leftrightarrow T(A)$$

где T – предикат истинности, входящий в значение слова “правильно”. Кроме того, он, возможно, не знает, что значит “быть свойством” или что значит “приписать свойство”. Иными словами, Саша не оценит как корректную операцию λ -абстракции, позволяющую перейти от (1) к (2'). Если же мы возьмем утверждение (2) как подходящее на роль применения функции к аргументу, то признание корректным перехода от (1) к (2) потребует от Саши знания того, что такое “функция”, “аргумент”, “значение функции для аргумента” и что значит быть “истинным” (“И”). И такого рода знание является более специальным, чем знание понятий, содержащихся в интенциональной формулировке.

Понятно, что смыслы (1), (2) и (2'), понимаемые как процедуры установления субъектом значения этих выражений, не являются тождественными. В первом случае осуществляется проверка того, бел ли Сократ (без временной оценки). Во втором сначала производится интерпретация того, что значит “значение функции для аргумента” и “истина”, а лишь затем осуществляется приложение функции к аргументу и проверка. Точно так же для (2') сначала требуется выяснить, что такое “свойство”, “приписывать”, “правильность”, “объект”. Приемлемость для субъекта перехода по правилам λ -конверсии, т. е. перехода от (1) к (2) или родственного перехода от (1) к (2') объясняется тем, что к его логической компетентности относится знание ряда понятий, имеющих отношение к синтаксису и семантике языка. Сами эти понятия и правила, в частности, понятия о функции, значении функции для аргумента, “истине”, объекте, свойстве, приписывании свойства объекту являются необходимыми как для осуществления переформулировок выражения, сохраняющих его истинность, так и для формулирования утверждений о смысле выражения, коль скоро субъект употребляет его осмысленно и понимает, что это значит. Следовательно, можно предположить, что принятие субъектом корректности переходов по правилам λ -конверсии связано с его знанием того, что такое объект, свойство и предикация.

Принятие описанных выше переходов, которые мы сопоставили λ -конверсии в качестве ее аналога в естественном языке, не

зависит от содержания предложений. Например, от утверждения того, что

(8) *Бряка курдявит ся*

мы можем перейти к утверждениям

(9) **Значение функции “курдявит ься” от аргумент а “бряка” ест ь “И”**

(9') **Свойст во курдявит ься правильно приписывает ся объект у, являющемуся брякой**

без знания смысла имен “бряка” и “курдявиться”. Выделенные слова играют здесь, так сказать, служебную роль, но их понимание субъектом обязательно и, хотя они, вообще говоря, не являются синкатегорематическими терминами, в данном контексте они используются именно в таком качестве. Это означает, что в языке, к которому принадлежат выражения, выступающие в роли аргумента и функции, т. е. “бряка” и “курдявиться”, значения выделенных в (9) и (9') слов “функция”, “значение”, “аргумент”, “И”, “свойство”, “правильно”, “приписывают”, “объект” заведомо иные, чем в метаязыке.

Такое одновременное присутствие двух пластов значений хорошо иллюстрируют следующие пары предложений, в которых нижнее получено из верхнего по неформальным правилам λ -конверсии как в случае экстенционального, так и в случае интенционального прочтения (кванторы не рассматриваем):

Функция имеет в качестве денот ат а операцию

Значение функции “имет ь в качестве денот ат а операцию” для аргумент а “функция” ест ь “И”

Свойст во - эт о т о, чт о приписывает ся объект у

Свойст во быт ь т ем, чт о приписывает ся объект у правильно приписывает ся объект у свойст во

“Ист ина” - эт о возмож ный денот ат предлож ения

Значение функции “возмож ный денот ат предлож ения” для аргумент а «”ист ина”» ест ь “И”

Правильност ь ест ь характ ерист ика приписывания свойст ва

Значение функции “характ ерист ика приписывания свойст ва” для аргумент а “правильност ь” ест ь “И”

Правильност ь ест ь характ ерист ика приписывания свойст ва

Свойст во быт ь характ ерист икой приписывания свойст ва правильно приписывает ся объект у правильност ь.

Приписыват ь свойст во объект у мож но и неправильно

Возмож ност ь быт ь неправильным ест ь свойст во, кот орое

правильно приписывает ся объект у приписывание свойст ва объект у

Объект ы - эт о т акие сущност и, кот орым мож но приписыват ь свойст ва

Возмож ност ь быт ь т ем, чему приписывает ся свойст во ест ь свойст во, кот орое правильно приписыват ь объект у объект

В этих примерах слова “значение функции”, “аргумент”, “И”, “свойство”, “объект”, “приписывать”, “правильно” употребляются сразу в двух ролях. Будучи выделенными, они играют служебную роль, в то время как невыделенные и не стоящие в контексте цитирования вхождения тех же слов употреблены прямо референциально. Очевидно, что генерировать вложенность одних употреблений этих выражений в другие можно сколько угодно. Как интерпретировать такого рода переходы?

В логической теории свойств, пропозиций и истины [11, Р. 69-72] понятие “свойство” определяется как такая семантическая функция, значение которой для любого аргумента образует пропозицию. Не будем здесь вдаваться в подробности, касающиеся соответствующего определения пропозиции [1], скажем только, что это понятие определено так, что оказываются исключенными семантические парадоксы. Истина или “правильность приписывания” рассматривается как предикат на пропозициях, а “приписыванию” соответствует функционал *App*. Это значит, что термины “функция”, “значение”, “аргумент”, “И”, “свойство” (или “операция”), “объект”, “приписывать” (или “применять функцию к аргументу”), “правильно” (или “значение”) всегда относятся к метаязыку и их использование прямо референциально возможно только в метаязыке для репрезентации тех или иных семантических функций и операций. Поэтому в каждом из приведенных примеров переходы от верхнего предложения к нижнему оказываются переходами от предложения некоторого языка-объекта к предложению в метаязыке, хотя эти языки и не различаются синтаксически. То есть там, где фигурируют выделенные слова, мы имеем дело по меньшей мере с *мет* аметаязыком. Но если интенциональная или экстенциональная переформулировки, подобные описанным выше, рассматриваются как аналоги приложения функции к аргументу, то запись $(\lambda xAx)B$, не отличаясь в своей денотации от $A(x // B)$, предполагает существенно иное прочтение в метаязыке. Понятно поэтому, что семантика $(\lambda xAx)B$, раскрывающая смысл этого выражения, должна осуществлять интер-

претацию метаязыковой формулировки, описывающей значение $(\lambda xAx)B$.

Впрочем, реализовав такую интерпретацию, мы столкнемся с необходимостью уточнения понятия концепта функции в ЛСД и его приложения в качестве функции к концептам аргументов. Возьмём пример не со свойством, а с операцией. Если

$$(2 + 3) \text{ conv}((\lambda x\lambda y. x + y)2)3,$$

то в метаязыке

$$\text{val}(+)(\text{val}(2), \text{val}(3)) \text{ conv}^* \text{ значение применения функции val}(+) \text{ к аргументам val}(2) \text{ и val}(3),$$

где conv^* обозначает метаязыковую конверсию. Здесь хорошо видно, что при осуществлении конверсии в метаязыке мы используем сущности, невыразимые в языке-объекте. Выражение справа от conv^* есть описание способа задания денотата для формулы $(2 + 3)$, которое после интерпретации констант предстаёт в следующем виде:

Результат операции “сложение” для чисел 2 и 3.

Здесь “сложение” есть имя константной функции и вся фраза описывает процедуру, соответствующую смыслу выражения $(2 + 3)$. В ЛСД смысл $(2 + 3)$ будет значением смысла имени “+” от смыслов имён “2” и “3”. В нашей интерпретации смысл имени “+” – это процедура сопоставления этому имени операции сложения и поэтому “+” есть константное имя, которое не является λ -выражением, а смыслы имен “2” и “3” – это процедуры сопоставления этим именам чисел 2 и 3. Поскольку процедура задания денотата “+” не является функцией на других процедурах, воспроизвести конструкцию Чёрча можно, добавив, например, комбинатор композиции K , задача которого будет состоять в преобразовании нескольких процедур в новую процедуру. В рассматриваемом случае,

$$K([\lambda xy. x + y][2])[3] = K([\lambda y. 2 + y][3]) = [2 + 3]$$

Работа комбинатора K в этом случае сходна с функциями аналогичного комбинатора в системе ЛСД Давида Каплана².

Семантика метаязыковых переформулировок λ -выражений

Итак, переформулировки выражений естественного языка, которые можно уподобить λ -конверсии, могут быть интерпретированы только как выражения метаязыка, поскольку аналог λ -

² См. статью Чарльза Парсонса [10].

выражения предполагает использование ряда терминов, которые являются синкатегорематическими в языке-объекте, а в метаязыке получают интерпретацию. Тем самым для реализации идеи Альтернативы (0) семантика λ -выражений должна стать семантикой их метаязыковых переформулировок. Семантика метаязыка становится тогда общим пространством интерпретации, в котором получают значения как выражения языка-объекта, так и самого метаязыка. Может ли введение такого рода комбинированной семантики помочь дифференцировать смыслы выражений $(\lambda xAx)B$ и $A(x // B)$? Или иначе, существует ли интерпретируемая такой семантикой функция f , для которой, если $A \text{ conv } B$, то $f[A] \neq f[B]$? Рассмотрим пример:

- (10) *Боря полагает, что небо синее*
 (11) *Боря полагает, что о значении функции “синий” для аргумента “небо” есть “И”.*
 (11') *Боря полагает, что о свойстве быть в синем правильно приписывает ся объект у небо*

Функция “полагает”, выражающая пропозициональную установку субъекта, очевидно, годится для различения соответствующих пропозиций. В самом деле, стоит нам предположить, что субъект Боря не обладает достаточной компетентностью для того, чтобы считать утверждение предложения

- (12) *Небо синее*

равносильным утверждению предложения

- (13) *Значение функции “синий” для аргумента “небо” есть “И”.*

как мы получаем различие смыслов выражений $A(x // B)$ и $(\lambda xAx)B$. И это выглядит естественно, ведь переход от (12) к (13) совсем не очевиден и даже менее очевиден, чем переход от (12) к утверждению

- (13') *Свойство быть в синем правильно приписывает ся объект у небо*

Иными словами, мы должны иметь дело по меньшей мере с тремя разными уровнями логической (семантической) компетентности, один из которых, более продвинутый, обеспечивает корректность перехода от (10) к (11) и (11'), другой – менее продвинутый, только от (10) к (11') и, наконец, минимальный – не принимающий ни одного из этих переходов, но позволяющий субъекту оценить истинность (12).

Для того чтобы переход от (10) к (11) или к (11') стал невозможен, должно быть неверно одно из утверждений:

- (14) *Боря полагает, что о (12) = (13)*

или

(14') *Боря полагает, что* $(12) = (13')$

В первом случае это будет означать, что субъект Боря обладает самое большее минимальной логической компетентностью, а во втором, что он не обладает некоторым продвинутым ее вариантом. Выше уже говорилось, что субъект, не обладающей логической компетентностью, достаточной для верификации предложения

Лондон - эт о Лондон,

не представляет интереса как ответственный носитель знаний и коммуникатор. Но такого рода субъект существует (в логическом смысле) и его пропозициональные установки должны фигурировать в ЛСД, с тем чтобы иметь возможность дифференцировать ответственного и неответственного носителей информации. Тем более интересно рассмотреть пропозициональные установки такого Бори, для которого не определены значения утверждений (14) и (14'), поскольку для их описания в реляционной семантике понадобятся "невозможные" возможные миры, относительно которых значения выражений, содержащих формулы вида $(\lambda xAx)B$, также не определены, и применение λ -абстракции в любом таком мире немедленно выводит за его пределы. Можно предложить гораздо более детальную и разнообразную иерархию видов логической компетентности субъекта и соответствующих эпистемических структур, но мы оставим этот вопрос в стороне.

Вернемся к проблеме дифференциации смысла $A(x // B)$ и $(\lambda xAx)B$. Оказалось, что функция "полагать" есть такая метаязыковая функция на смыслах выражений, что для некомпетентного в той или иной мере субъекта она не отождествляет пропозиции, выраженные (12), (13) и (13').

Опишем теперь набросок комбинированной семантики языка-объекта и фрагмента метаязыка, в которой существование таких функций обеспечивается для языка-объекта. Обозначим язык ЛСД как L и будем считать L языком-объектом или "внутренним" языком. Язык L^+ , являющийся фрагментом метаязыка для L , задан функцией-переводчиком φ относительно языка L и мы будем считать L^+ "внешним" языком. В язык L^+ входят примитивные функциональные константы, определенные на переводах выражений языка L :

Funk – оператор "функция";

App – оператор "приложение функции к аргументу"³.

Определим функцию-переводчик φ формул языка L на язык L^+ :

³ Эти операторы происходят из теории свойств и семантики λ -исчисления. См. книгу Раймонда Тёрнера [11].

- если A – переменная или примитивная константа, то $\varphi(A) = A$;
- если A имеет вид $\lambda x_\beta B_\alpha$, то $\varphi(A) = \text{Funk}(\varphi(B_\alpha(x_\beta)))$;
- если A имеет вид $B_{\alpha\beta} C_\beta$ и $B_{\alpha\beta}$ – λ -выражение, то

$$\varphi(A) = \text{App}(\varphi(B_{\alpha\beta}), \varphi(C_\beta));$$
- если A имеет вид $B_{\alpha\beta} C_\beta$ и $B_{\alpha\beta}$ не является λ -выражением, то

$$\varphi(A) = \varphi(B_{\alpha\beta})\varphi(C_\beta).$$

Значение формул L^+ задается функцией V относительно означиваний и относительно функции val_v , задающей стандартные значения выражениям языка L , не содержащим имена смыслов λ -выражений. Определим сначала значения примитивных констант L^+ :

- (a) $V_v(\text{Funk})$ есть функция $g: \text{Expr}_\alpha(x_\beta) \rightarrow D_{\alpha\beta}$, где $\text{Expr}_\alpha(x_\beta)$ есть множество выражений типа α языка L , содержащих единственную свободную переменную типа β .
- (b) $V_v(\text{App}) = h$, где h – функция, определенная на парах выражений языка L^+ и для пары $\langle e, d \rangle$, $h(e, d) = V_v(e) V_v(d)$.
- (1) если X_α – переменная или примитивная константа языка L , то $V_v(X) \in D_\alpha$;
- (2) если X – формула языка L^+ вида $\text{Funk}(a(b))$, то $V_v(X)$ есть единственная f , такая, что для любого b , $f(b) = V_{v^*} a$, где $v^* = \lambda(x \leftarrow b)$.
- (3) если X – формула языка L^+ вида $\text{App}(a, b)$, для $n \geq 0$, то $V_v(X) = (V_v \text{App})(a, b)$.

Назовем *семантической программой* для формулы X последовательность выражений метаязыка, которую можно получить редукцией записи $V_v(X)$ по определению значения формул языка L^+ . Переход из одной строки в другую осуществляется заменой по определению значения самой внутренней подформулы, к которой редукция может быть применена. В качестве денотата формулы $[X]$ языка L будем рассматривать семантическую программу для X , считая такие программы сущностями, образующими домены интенциональных типов. (Здесь мы временно не рассматриваем смысл сложного выражения как функцию смыслов его составляющих.)

Легко увидеть, что в приведенном определении, выражения экстенциональных типов языка L получают обычную теоретико-типическую интерпретацию. В то же время, поскольку денотатами интенциональных сущностей, т. е. смыслов являются семантические программы, смыслы выражений $A(x // B)$ и $(\lambda x A)x B$ окажутся, очевидно, различными. В самом деле, значения формул, в которых фигурируют подформулы вида $\lambda x A$, потребует использования метаязыковых функций App и Funk , в то время как $A(x // B)$ может

и не содержать подформулы такого вида и, следовательно, привлечение этих функций не требуется. Предполагая различные степени семантической компетентности, мы можем считать, что смысл функций *App* и *Funk* остается неизвестным для некомпетентного субъекта и поэтому он не в состоянии осуществить процедуру задания денотата для $(\lambda xAx)B$, т. е. для него это выражение остается бессмысленным.

Рассмотрим пример, содержащий примитивные константы естественного языка (отклоняясь от ЛСД, равенство также будем считать примитивной константой):

$$[(\lambda x. \text{синий}(x))(\text{небо})] = [\text{синий}(\text{небо})]$$

Денотатом левой части этого выражения является следующая семантическая программа

$$\begin{aligned} &V_v(\text{App}(\text{Funk}(\text{синий}(x)), \text{небо})) \\ &V_v\text{App}(V_v(\text{Funk}(\text{синий}(x))), V_v\text{небо}) \\ &V_v(\text{Funk}(\text{синий}(x)) V_v\text{небо}) \\ &f(V_v\text{небо}), \end{aligned}$$

где f – единственная функция, такая, что для любого b , $f(b) = V_{v^*}(\text{синий}(x))$, где $v^* = \nu(x \leftarrow b)$. Правой части равенства соответствует семантическая программа

$$\begin{aligned} &V_v\text{синий}(\text{небо}) \\ &V_v\text{синий} V_v\text{небо} \\ &g(V_v\text{небо}), \end{aligned}$$

где g – функция, сопоставленная константе “синий” функцией интерпретации и совпадающая с f . Очевидно, что эти семантические программы нетождественны.

Существенным отличием полученных семантических программ оказывается уже не количество шагов и даже не структура, а наличие в первой программе функций *App* и *Funk*, интерпретация которых может быть не известна некомпетентному субъекту. Пусть теперь функция f_{ool} ⁴ имеет следующую интерпретацию: “то, с чем согласен не вполне компетентный субъект Боря”. Тогда, оставляя Боре минимум компетентности, мы получим

$$f_{ool}[\text{синий}(\text{небо})] \vee f_{ool}[\neg . \text{синий}(\text{небо})]$$

но неверно, что

$$f_{ool}[(\lambda x. \text{синий}(x))(\text{небо})] \vee f_{ool}[\neg . (\lambda x. \text{синий}(x))(\text{небо})],$$

поскольку в последнем случае, обе пропозиции остаются для Бори неопределенными по смыслу и значению.

Конечно, градации логической компетентности субъекта относительно семантических понятий, формулируемых в расширении языка-объекта, гораздо многообразнее, чем градация по

⁴ Ее тип – из пропозиций в истинностные значения.

принципу осведомленности в тех или иных способах рассуждения. Например, мы можем считать, что субъект понимает, что такое истинность формулы языка-объекта, но не понимает, что такое истинность выражения метаязыка. Для такого субъекта оказываются неразрешимыми семантические парадоксы. С другой стороны, можно предположить и максимальную степень компетентности, проявляющуюся в знании принципа построения иерархии метаязыков и ветвления предиката истинности. Соответствующие теории могут быть развиты и в эпистемической логике.

Некоторые итоги

На наш взгляд, введение понятия логической или семантической компетентности субъекта может оказаться полезным для достижения интуитивно приемлемой реализации основной идеи Альтернативы (0) ЛСД, т. е. для различения смысла выражений вида $(\lambda x_\beta A_\alpha)(B_\beta)$ и $A_\alpha(x_\beta // B_\beta)$. Для этого необходимо иметь дело с метаязыком ЛСД, в котором фигурируют эпистемические функции, характеризующие логическую компетентность субъекта. Стандартные семантические процедуры или программы репрезентируют в метаязыке смысл выражений языка-объекта и становятся объектами специфических пропозициональных установок, в рамках которых λ -конверсия оказывается некоторым преобразованием таких объектов. Развитие ЛСД в предлагаемом нами направлении превращает ее в одну из разновидностей эпистемической логики, что кажется неизбежным, по крайней мере в связи с проблемой Альтернативы (0).

ЛИТЕРАТУРА

1. *Aczel P.* Frege Structures and the Notions of Proposition, Truth and Set // The Kleene Symposium / Eds. J.Barwise, H.J.Heisler and K.Kunen. North-Holland P.C. 1980. P. 31-59.
2. *Anderson C.A.* General Intensional Logic // Handbook of Philosophical Logic / Eds. D.Gabbay, F.Guenther. D.Reidel P.C. 1984. Vol. II. P. 355-385.
3. *Church A.* A Formulation of the Logic of Sense and Denotation (Abstract) // Journal of Symbolic Logic. V. 11. 1946. P. 31.
4. *Church A.* A Formulation of the Logic of Sense and Denotation // Structure, Method and Meaning. Essays in honor of H.M.Sheffer. New York: The Liberal Arts Press. 1951. P. 3-24.
5. *Church A.* A Revised Formulation of the Logic of Sense and Denotation. Alternative (1) // NOUS. V. 27. 1993. P. 141-157.

6. *Church A*. Outline of a Revised Formulation of the Logic of Sense and Denotation (Part I) // NOUS. V. 7. 1973. P. 24-33; (Part II) // NOUS. V. 8. 1974. P. 135-156.
7. *Church A*. A Formulation of the simply Theory of Types // Journal of Symbolic Logic. V. 5. 1940. P. 56-67.
8. *Henkin L*. Completeness in the Theory of Types // Journal of Symbolic Logic. V. 15. 1950. P. 81-91.
9. *Lewis D*. General semantics // Semantics of natural language / Eds. D.Davidson, G.Harman. D.Reidel P.C. North-Holland. 1977. P. 169-218.
10. *Parsons C*. Intensional logic in extensional language // Journal of Symbolic Logic. V. 47. 1982. P. 289-328.
11. *Turner R*. Truth and modality for knowledge representation. MIT Press. Cambridge, Mass., 1991.