

УДК 1.168

ВЕЩИ, СВОЙСТВА, ОТНОШЕНИЯ В ТЕОРЕТИКО-ТИПОВОЙ ОНТОЛОГИИ

О. А. Доманов

Институт философии и права СО РАН (г. Новосибирск)

domanov@philosophy.nsc.ru

Аннотация. Вещь, свойство и отношение относятся к базовым онтологическим категориям с древности вплоть до настоящего времени. Начиная с Фреге, логика описывает свойства и отношения единообразно как n -местные предикаты, и теория типов часто следует этому. Вместе с тем их традиционное понимание не только приводит к трудностям и парадоксам, но и противоречит онтологическим интуициям самой теории типов. В статье описано традиционное представление свойств и отношений в теории типов и альтернативная формализация, исключающая появление гипотетической «вещи без свойств». Теория типов рассматривается как не только логическая, но и как онтологическая система. Показано, что распространенное представление свойств как типов, зависимых от объектов, следует модели описания отношений и становится неадекватным при применении к самим вещам. Приведены примеры простых формализаций на языке Lean. Показано, в каком смысле данная формализация лучше соответствует онтологическим основаниям теории типов.

Ключевые слова: теория типов, свойство, отношение, онтология теории типов.

Для цитирования: Доманов, О. А. (2025). Вещи, свойства, отношения в теоретико-типовую онтологию.

Respublica Literaria. Т. 6. № 4. С. 83-92. DOI: 10.47850/RL.2025.6.4.83-92

THINGS, PROPERTIES, RELATIONS IN TYPE-THEORETICAL ONTOLOGY

O. A. Domanov

Institute of Philosophy and Law SB RAS (Novosibirsk)

domanov@philosophy.nsc.ru

Abstract. Thing, property and relation from ancient to present times belong among fundamental ontological categories. Starting from Frege logic describes properties and relations uniformly as n -placed predicates and type theory often follows this course. However, their traditional understanding not only leads to problems and paradoxes but also goes against ontological intuitions of type theory itself. The article examines the traditional presentation of properties and relations in type theory and their alternative formalization which prevents the hypothetical emergence of any «thing without properties». Type theory is considered as not merely logical but also ontological system. It is argued that the common presentation of properties as types dependent on objects follows the description model of relations and becomes inadequate when applied to the things as such. Simple examples of formalizations in the language Lean are provided. The sense in which this formalization fits better the ontological foundations of type theory is described.

Keywords: type theory, property, relation, type theory ontology.

For citation: Domanov, O. A. (2025). Things, Properties, Relations in Type-Theoretical Ontology. *Respublica Literaria*. Vol. 6. No. 4. Pp. 83-92. DOI: 10.47850/RL.2025.6.4.83-92

Вещь, свойство и отношение относятся к базовым онтологическим категориям. Вещи и свойства различаются с самых первых шагов построения онтологии как дисциплины. Уже Аристотель в своей первой философии разделяет базовую категорию усию (то, о чем говорится) и остальные категории (то, что говорится). Смысл усии разделяется затем на субстанцию (*substantia*) и сущность (*essentia*), где первая обозначает саму отдельную вещь, а вторая – форму или чистоту этой вещи, то что эта вещь есть. Уильям Оккам в XIV в., классифицируя онтологические понятия и стремясь оставить из них лишь необходимый минимум, ограничивается только субстанциями и качествами (а также отношениями для некоторых специальных случаев, таких как Божественная Троица). Следы онтологии «субстанция-атрибут» мы находим во многих учениях прошлого. Понятие материи как чего-то бесформенного, обретающего форму, уже Платонуказалось необходимым, хотя и подозрительным. Парадоксы материи в классической онтологии так и не были разрешены, несмотря на активное обсуждение в Средние века. Остается также не вполне ясным различие между существенными свойствами вещей (атрибутами субстанции) и их несущественными свойствами. Старый спор о том, можно ли считать вещь множеством ее атрибутов, также коренится в этой онтологии. Критерий тождества неразличимых Лейбница – «вещи равны, если обладают тем же набором свойств» – исключает наличие двух вещей с одними и теми же свойствами. Это позволяет определить вещь как совокупность свойств. Это, однако, приводит ко многим трудностям и парадоксам, таким как «Корабль Тесея». Может ли объект потерять все свои свойства, оставшись тем же самым? Остается ли той же самой гусеница, превращающаяся в куколку и затем в бабочку? Кантовская вещь-сама-по-себе (ноумен) также подчиняется этой схеме. Она представляет собой неявленную вещь, которой наша познавательная способность прикладывает «форму явленности» или форму опыта. У Канта же мы найдем и прямое использование понятия материи, которое у него присутствует в виде неопределенного «многообразного», которое оформляется чувственностью и рассудком.

Что касается отношений, то трудности их адекватного описания преследовали онтологию вплоть до Фреге, который, совершая, по словам Рассела, крупнейший шаг в логике со времен Аристотеля, использует функцию как общую форму, позволяющую записывать предикаты с любым количеством аргументов. Тем не менее, и Фреге оставляет неизменной описанную выше схему: его онтология состоит из вещей и функций, которые формализуют как свойства, так и отношения. Вплоть до настоящего времени тройка понятий «вещь», «свойство», «отношение» выступает как набор базовых онтологических категорий. В своей книге, специально им посвященной, А. Уемов формулирует это следующим образом: «Можно изучать главным образом вещи, преимущественно отдельные свойства или отношения, но нельзя изучать что-либо иное, кроме вещей, свойств и отношений» [Уемов, 1963, с. 3]. Современные онтологии, используемые для компьютерного представления знаний, такие как OWL или UML, следуют этому пониманию. В них, как правило, онтология представлена как набор объектов, обладающих атрибутами и находящихся в отношениях, причем атрибуты не разделяются на сущностные и акцидентальные. Будучи привлекательным из-за простоты, это представление не лишено проблем. Одна из основных состоит в том, что оно допускает лишь ограниченное использование математических методов доказательства и проверки корректности. В то же время множество математических систем, показавших в последнее время свою эффективность как в формализации математики, так и в расширении возможностей генеративного искусственного интеллекта, таких как Rocq, Agda, Lean и др., основаны не на логиках,

восходящих к Фреге, а на вариантах теории типов. Последняя не только является альтернативной логической системой (интуиционистской и, шире, конструктивистской), но и опирается на иные философские – в частности, онтологические – идеи. Рассмотрим, какие трансформации это влечет для понимания вещей (объектов), свойств (атрибутов, качеств) и отношений.

Онтология теории типов

Теория типов является интерпретированной формальной системой [Martin-Löf, 1993, р. 1]. В этом смысле, она всегда уже оказывается моделью, т. е. описывает некоторую онтологию. Традиционное для формальной семантики различие между самостоятельно определенным формальным языком и его дальнейшей интерпретацией отсутствует в теории типов. По этой причине теорию типов можно рассматривать не только как логический, но и как онтологический язык.¹ Нашей первой задачей будет выделение этой предполагаемой теорией типов онтологии. Мы будем рассматривать язык теории типов не просто как формальный, а как схватывающий онтологическую структуру формализуемых объектов. Формализация понимается здесь не как перевод в формализованный язык, а как, прежде всего, усмотрение формы и лишь затем – ее запись в таком формализованном языке.

Начнем с рассмотрения понятий вещи и свойства (атрибута).

Термин «атрибут» происходит от латинского *attributio*, «приписывание», и означает качество или свойство, которое приписывается вещи (или субстанции в классической онтологии). Он связан с субъектно-предикатной логикой, в которой основной формой предложения является форма «*S* есть *P*». При обычной формализации [напр.: Chen, 1976, р. 12] атрибут понимается как функция из множества объектов в множество кортежей значений атрибутов:

$$f : E_i \rightarrow V_{i_1} \times \cdots \times V_{i_n}.$$

Например, если страна рассматривается как атрибут городов, которые в ней расположены, то атрибут формализуется как функция *Город* → *Страна*, где *Город* – множество городов, а *Страна* – множество стран. В теории типов соответствующая конструкция формализуется с помощью зависимого типа *Город(Страна)*, где *Страна* – это тип стран, а *Город* – семейство типов (в данном случае, множеств) городов, индексированное множеством стран. Иначе говоря, город представляет собой функцию, значением которой служит тип: *Город* : *Страна* → *Type* – т. е. тип, зависящий от другого типа. Эта конструкция изображена на рис. 1. На нем каждой стране $x_i \in \text{Страна}$ соответствует множество городов *Город*(x_i). Если мы объединим все эти множества городов в одно, то можем построить функцию из этого объединенного множества в множество стран, аналогичное той, что мы имеем в традиционном подходе. Обратно, если у нас имеется функция, из множества всех городов в множество стран, то первое из них разбивается на классы эквивалентности по равенству значений этой функции (рис. 2), и мы получаем семейство, аналогичное тому, что мы видим в теории типов. Таким образом, традиционный и теоретико-типовой подходы структурно эквивалентны и могут быть переведены друг в друга (в частном случае типов, понимаемых как множество).

¹ Что, в частности, выражено в идее так называемого вычислительного тринитаризма: Harper, R. (2011). The Holy Trinity [Online]. *Existential Type*. Available at: <https://existentialtype.wordpress.com/2011/03/27/the-holy-trinity/> (Accessed: 27 March 2011).

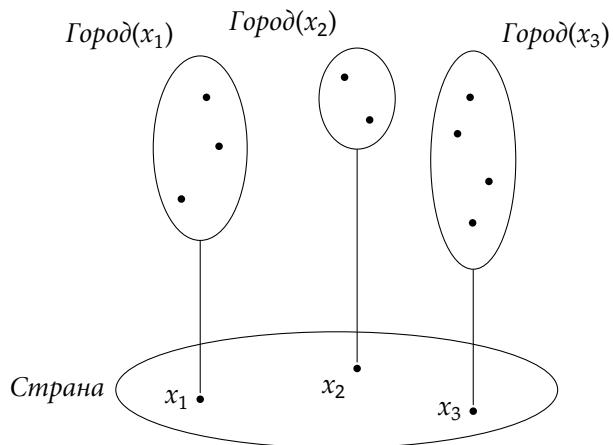


Рис. 1: Семейство типов городов.

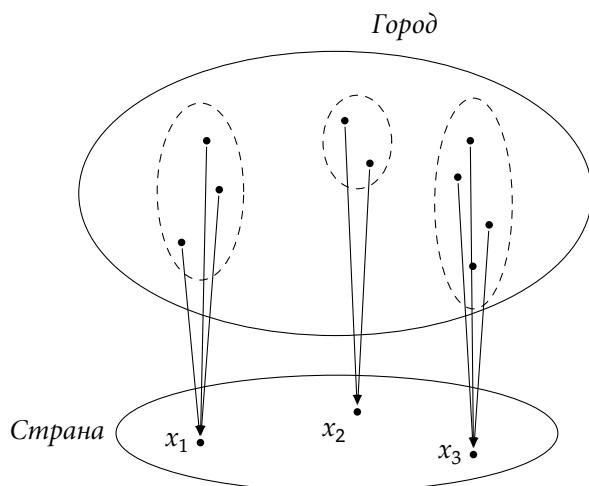


Рис. 2: Функция приписывания атрибута.

Онтологически, однако, мы имеем существенную разницу. Объекты в теории типов всегда относятся к какому-то типу. Этот тип должен быть указан при появлении любого объекта в теории и в дальнейшем не может быть изменен. С эпистемологической точки зрения, объект всегда «какой-то», не существует «никаких» объектов: при встрече с объектом мы опознаем его как нечто определенное, т. е. относящееся к тому или иному типу, пусть даже к типу «нечто смутное и неясное». Знание объекта всегда сопровождается знанием его типа (или формы, или чёткости, как это называлось в средневековой онтологии). В терминах атрибутов это означает, что объекты всегда появляются в теории (т. е. в нашем знании) как уже обладающие атрибутами. Не имеется никакого предполагаемого присвоения атрибута объекту, им еще не обладающему. В теории типов ни в каком виде не предполагается абстракция «вещи без свойств» или «вещи без формы». В то же время в семантике Фреге она присутствует в виде множеств, на которых интерпретируется его «исчисление понятий» (в этом смысле, хотя Фреге определенно совершает крупнейший шаг в логике со времен Аристотеля, у него остается онтологическая схема, предполагающая или допускающая гипотетический акт присваивания атрибута или оформления бесформенного).

Объекты, свойства и отношения мыслятся отдельно друг от друга.

В теории типов тип в определенном смысле предшествует объекту (терму типа). Появление объекта становится возможным через его узнавание, т. е. опознание как того, что определено и отлично от других объектов, т. е. как относящегося к типу. Тип в этом смысле задает априорную структуру, позволяющую объекту быть в качестве того, что он есть. Его функция аналогична трансцендентальным структурам у Канта, определяющим форму возможного мира. Это не случайно, поскольку теория типов заимствует идею трансцендентального у Канта через Гуссерля. В то же время аристотелевская онтология существенным образом опирается на идею онтологической независимости субстанции: она существует сама по себе, тогда как другие сущие зависят от нее в своем существовании.² Здесь есть определенная неясность, поскольку Аристотель одновременно предполагает наличие у вещей сущностных свойств – тех, отсутствие которых не позволяет вещи быть тем, что она есть. В этом отношении вещь, напротив, зависит от свойств и не существует без них. Эта двусмысленность приводит к неясностям в понимании вещей, свойств и отношений в онтологии и логике, включая их теоретико-типовыe варианты. Рассмотрим подробнее, как это происходит в теории типов.

Отношения и атрибуты в теории типов

Начнем с понятия отношения. Оно представляется как зависимый тип, зависящий от типов, находящихся в отношении. Например, для отношения $R \subseteq A \times B$ мы имеем:³

```
inductive R : A → B → Type where
| r1 : R a b
| r2 : R c d
```

Это функция с двумя аргументами, значением которой является тип. В силу принципа «пропозиция-как-тип» (соответствие Карри–Ховарда) мы можем рассматривать результирующие типы $R x y$ как пропозиции; тогда $r1, r2$ являются доказательствами пропозиций $R a b$ и $R c d$, соответственно. Типы также возможно рассматривать как ситуации [Cooper, Ginzburg, 2015]. Тогда отношение понимается как зависящая от объектов ситуация – в которой эти объекты находятся в данном отношении. Тогда $r1, r2$ – инстанциации или экземпляры соответствующих ситуаций. Интерпретируя онтологически, это можно рассматривать как «внешнее» наложение отношения в том смысле, что объекты предполагаются данными вне отношения и предшествуют ему.

Перейдем теперь к представлению атрибутов. Мы видели, что в традиционной формализации они записываются как функции из множества объектов в множество значений или кортежей значений атрибутов. В конструктивных теориях, к которым относится теория типов, построение этой функции должно опираться на некоторые доказательства. Рассмотрим случай одного свойства. Свойства соответствуют предикатам, которые в теории типов представляются как зависимые типы, т. е. как функции из множеств в типы [см., напр.: Ranta, 1994]. Понятый как ситуация, предикат при таком подходе представляет собой ситуацию, зависящую от объектов.

² Эта проблематика существенным образом зависит от двусмысленности глагола «быть», который может пониматься либо экзистенциально, в смысле существования, либо в смысле чайности в контексте ответа на вопрос «что значит быть чем-то». Мы, однако, оставляем это за рамками рассмотрения.

³ Для примеров я использую теоретико-типовую язык Lean: <https://lean-lang.org>.

Например, в случае цвета ситуацией является «объект 0 – красный», а его инстанциациями – доказательства того, что 0 действительно красный. Мы видим ту же схему, что и в случае отношений, но для функций от одного аргумента. Это означает, что онтологически это соответствует «внешнему» присоединению атрибута к заранее данному объекту. Фактически, теоретико-типовая формализация не отличается здесь от используемой Фреге и классической онтологией вообще. В этом можно видеть противоречие этой онтологии: не имея адекватных средств для представления отношений, она, тем не менее, моделирует свойства как отношения, т. е. как внешнее атрибутирование, полагая одновременно в некоторых случаях свойство «внутренним», относящимся к сущности.

Другим недостатком такого представления является то, что этот способ не учитывает абстрагирование. Например, в случае цвета в качестве атрибута приписываются значения (красный, зеленый и пр.), которым соответствуют, вообще говоря, разные атрибуты. Этот недостаток, однако, легко корректируется. Посмотрим, как это можно сделать на примере простой модели с цветами.

Пусть имеется тип объектов и предикатов на них для двух цветов, красного и зеленого:

```
inductive Obj where | o1 | o2 | o3 | o4
open Obj
```

```
inductive redP : Obj → Type where
| r1 : redP o1
| r2 : redP o2
inductive greenP : Obj → Type where
| g3 : greenP o3
| g4 : greenP o4
```

Для моделирования абстрагирования введем имена цветов вместе с их отображением в соответствующие предикаты:

```
inductive Colour where | red | green
open Colour
```

```
def ColPred : Colour → (Obj → Type)
| red => redP
| green => greenP
```

Это, в свою очередь, позволяет построить предикат «объект *o* имеет цвет *c*»:

```
def ColObj : Obj → Colour → Type := λ o c => ColPred c o
```

Поскольку, вообще говоря, наше знание цветов объектов может быть ограниченным, функция приписывания атрибутов будет в общем случае частичной, зависящей от существования доказательства для ColPred:

```
def ColAttrs (o : Obj) (p : Σ' c : Colour, ColPred c o) : Colour := p.fst
```

В нашем случае, однако, каждый объект имеет цвет, поэтому мы в состоянии построить полную функцию:

```
def cols : ∀ o, Σ' c, ColPred c o
| o1 => (red,.r1)
| o2 => (red,.r2)
| o3 => (green,.g3)
| o4 => (green,.g4)
def ColAttrs' (o : Obj) : Colour := ColAttrs o (cols o)
```

В таком виде она служит функцией присвоения атрибутов объектам. т. е. реализует традиционную формализацию.

Рассмотрим теперь альтернативное представление, не предполагающее объектов без атрибутов. Имея в виду дальнейшее изложение, расширим наш пример, введя еще один атрибут – размер:

```
inductive Size where | small | medium | large
open Size
```

Пусть тип физических объектов это тип объектов, имеющих цвет и размер. Тогда его можно определить следующим образом:

```
inductive PhysObj : Colour → Size → Type where
| o1 : PhysObj red small
| o2 : PhysObj red large
| o3 : PhysObj green small
| o4 : PhysObj green medium
open PhysObj
```

Это семейство типов и o1,... o4 – объекты различных членов этого семейства. Они изначально обладают атрибутами, которые мы можем вычислить. Например, для функции, вычисляющей цвет объекта, имеем:

```
def col {c s} : PhysObj c s → Colour := λ _ => c
```

Здесь видно, что для вычисления используется не сам объект, а аргументы его типа, представленные неявно.

Теперь предикаты цвета можно определить в общем виде как

```
def colP {co so} (c : Colour) : PhysObj co so → Prop := λ o => col o = c
```

или, для конкретных цветов:

```
def redP {c s} : PhysObj c s → Prop := colP red
def greenP {c s} : PhysObj c s → Prop := colP green
```

В нашем определении объекта каждый конкретный объект является доказательством того, что «имеется объект данной формы» (в нашем случае, объект данного цвета и размера). Тогда предикаты redP и greenP соответствуют утверждениям о том, что объект обладает данной формой. В качестве обобщения мы можем ввести понятие формы (физического объекта), объединяющего цвет и размер:

```
structure FormPhys where
    colour : Colour
    size   : Size
```

Тогда объекты и предикаты, определенные выше, примут вид:

```
inductive PhysObj : FormPhys → Type where
| o1 : PhysObj ⟨ red, small ⟩
| o2 : PhysObj ⟨ red, large ⟩
| o3 : PhysObj ⟨ green, small ⟩
| o4 : PhysObj ⟨ green, medium ⟩
open PhysObj

def col {form} : PhysObj form → Colour := λ _ => form.colour
def colP {form} (c : Colour) : PhysObj form → Prop := λ o => col o = c
def redP {form} : PhysObj form → Prop := colP red
def greenP {form} : PhysObj form → Prop := colP green
```

Объекты, таким образом, определяются как типы, зависящие от формы. Эта форма может быть более сложной, чем простое множество свойств. Например, для объекта, состоящего из частей, она может выглядеть следующим образом:

```
structure FormPart where
    typ : Type
    rel : typ → typ → Type
    wp : isWholePart rel
```

где `wp` – доказательство того, что отношение `rel` является отношением «часть-целое». Соответственно, тип объектов, состоящих из частей, можно определить как:

```
inductive PartObj : FormPart → Type where
| o1 : PartObj ⟨ A, PA, isPA ⟩
| o2 : PartObj ⟨ B, PB, isPB ⟩
```

где `A`, `B` – типы составляющих, `PA`, `PB` – отношения на них, а `isPA`, `isPB` – доказательства того, что эти отношения являются отношениями «часть-целое». Каждый из объектов `o1`, `o2` появляется изначально как состоящий из частей, т. е. как обладающий определенной формой.

Мы видели выше, что тип отношения можно рассматривать как тип ситуаций, зависящих от объектов. Эти ситуации, таким образом, соединяют объекты внешним образом. Соответствующие конструкторы можно понимать как конструкторы связей, которые устанавливают отношение между заранее данными объектами. Но сами объекты не всегда можно понимать таким же образом. В некоторых случаях, действительно, на обладание атрибутом можно смотреть как на результат его присвоения предсуществующему объекту. Но в других случаях нам требуются определения объектов, изначально обладающих определенной формой. Это различие, проведенное здесь в теоретико-типовых терминах, соответствует различию между сущностными

и случайными или акцидентальными свойствами в классической онтологии. В теории типов оно возникает естественным образом, поскольку объекты в ней всегда появляются определенным образом оформленными, относящимися к определенному типу.

Заключение

Мы видим, что онтология вещей, свойств и отношений оказывает свое влияние на теоретико-типовую формализацию. Вместе с тем, понятие зависимых типов или семейств типов позволяет построить представление, свободное от определенных проблем этой онтологии. Формирование семейств типов является неформальным процессом. Мы можем предположить, что люди сначала научаются, например, выделять объекты отдельных цветов, т. е. формировать типы для каждого отдельного цвета. Затем они объединяют их в семейство типов цвета, а затем – формируют типы различных объектов, обладающих цветами. В любом случае формирование семейств происходит в результате типизации объектов, которые никогда не даны отдельно от своих свойств и форм. Описанная теоретико-типовая формализация отражает этот процесс, в то время как традиционное представление его через приписывание атрибута напротив затемняет, что и приводит к трудностям и парадоксам.

Список литературы / References

Уемов, А. И. (1963). *Вещи, свойства и отношения*. М.: АН СССР. 183 с.

Ouemov, A. I. (1963). *Things, Properties, and Relations*. M. 183 p.

Chen, P. P.-S. (1976). The Entity-Relationship Model—Toward a Unified View of Data. *ACM Trans. Database Syst.* Vol. 1. No. 1. Pp. 9-36. DOI: 10.1145/320434.320440.

Cooper, R., Ginzburg, J. (2015). Type Theory with Records for Natural Language Semantics. In Lappin, S., Fox, C. (eds.). *The Handbook of Contemporary Semantic Theory*. Wiley. Pp. 376-407. DOI: 10.1002/9781118882139.ch12.

Martin-Löf, P. (1993). *Philosophical Aspects of Intuitionistic Type Theory. Lectures given at the Faculteit der Wijsbegeerte, Rijksuniversiteit Leiden, 23 September – 16 December 1993*. [Online]. 194 p. Available at: <https://pml.flu.cas.cz/uploads/PML-LeidenLectures93.pdf> (Accessed: 10 October 2025).

Ranta, A. (1994). *Type-Theoretical Grammar*. Clarendon Press. 226 p.

Сведения об авторе / Information about the author

Доманов Олег Анатольевич — кандидат философских наук, старший научный сотрудник Института философии и права СО РАН, г. Новосибирск, ул. Николаева, 8, e-mail: domanov@philosophy.nsc.ru, <https://orcid.org/0000-0003-0057-3901>.

Получена: 15.10.2025

После редактирования: 10.11.2025

Принята: 17.11.2025

Oleg Domanov — Candidate of Philosophical Sciences, Senior Researcher of the Institute of Philosophy and Law of the Siberian Branch of the Russian Academy of Sciences, Novosibirsk, Nikolaeva Str, 8, e-mail: domanov@philosophy.nsc.ru, <https://orcid.org/0000-0003-0057-3901>.

The paper was submitted 15.10.2025

Received after reworking 10.11.2025

Accepted for publication 17.11.2025