„Analiza i Egzystencja" 63 (2023), 27–54
ISSN (print): 1734-9923
ISSN (online): 2300-7621
DOI: 10.18276/aie.2023.63-02

ARTYKUŁY

JAKUB GOMUŁKA
ORCID: 0000-0002-9100-0334
AGH University of Krakow
email: jgomulka@agh.edu.pl

Towards a Computational Ontology for the Philosophy of Wittgenstein: Representing Aspects of the *Tractarian* Philosophy of Mathematics

Keywords: Ludwig Wittgenstein, Semantic Web, Knowledge Representation, *Tractatus Logico-Philosophicus*, Philosophy of Mathematics

Słowa kluczowe: Ludwig Wittgenstein, sieć semantyczna, reprezentacja wiedzy, *Tractatus logico-philosophicus*, filozofia matematyki

Abstract

The present paper concerns the Wittgenstein ontology project: an attempt to create a Semantic Web representation of Ludwig Wittgenstein's philosophy. The project has been in development since 2006, and its current state enables users to search for information about Wittgenstein-related documents and the documents themselves. However, the developers have much more ambitious goals: they attempt to provide a philosophical subject matter knowledge base that would comprise the claims and concepts formulated by the philosopher. The current knowledge representation technology is not well-suited for this task, and a non-standard approach is required. The creators of the Wittgenstein ontology project are aware of this fact; recently, they have been discussing conceptual devices adjusting the technology to their needs.

The main goal of this paper is to present examples of a representation of philosophical content that make use of both the devices already proposed and some new

inventions. The represented content comes from the *Tractatus Logico-Philosophicus*; more specifically, its theses concerning the problems in philosophy of mathematics.

Introduction

The Wittgenstein ontology project is a unique interdisciplinary research initiative that links philosophy and information science. The goal of the project is to create a Wittgenstein-related knowledge base that would be accessible to relatively simple algorithmic systems and easily searchable by users. The project has been run by the Wittgenstein Archives at the University of Bergen, Norway (WAB) since 2006. So far, the WAB team has been able to complete the part of the base that covers information about Wittgenstein's published and unpublished texts as well as their internal structure. They have also published a couple of theoretical papers regarding the possibility of representing philosophical content.

The present paper discusses an attempt at putting their ideas into practice. More specifically, it focuses on the segment of the *Tractatus Logico-Philosophicus* devoted to the philosophy of mathematics. There are several reasons why this is a much more difficult task than developing a knowledge base concerning just documents. The proposed solutions are sometimes quite complicated. Nevertheless, I believe that they demonstrate that the task is generally feasible.

The paper begins with a very short introduction to the paradigm being used in the development of the Wittgenstein ontology project: the Semantic Web technology. Next, we proceed to discuss the WAB's practical and theoretical achievements: both the current state of the project and its visions and proposals. However, the main goal of this paper is to propel the Wittgenstein ontology project forward and demonstrate how we can actually create representations of philosophical knowledge using the approach of the WAB researchers.

Our case study will be a fragment of the *Tractatus Logico-Philosophicus*; more precisely, theses 6–6.031 and 6.2–6.241. However, a full representation of this part of the philosophy of the early Wittgenstein, due to limitations of space and the relatively large scale and complexity of the task, will not be presented here; we shall only discuss several interesting sentences. I believe that although the reader is acquainted with only a limited number of examples, they will be capable of imagining the further development of the Wittgenstein ontology project.

Basic information on the technology of knowledge representation

The Semantic Web technology (SW) is a paradigm within Knowledge Representation & Reasoning (KR^2), a field in artificial intelligence research. SW enables us to represent and share information that can be easily searched, retrieved, and processed by both human users and automatic agents. Basic units of SW information are uniquely identifiable resources that can be assigned various primitive alphanumeric values and linked together through binary relations, thus forming complex knowledge graphs. Generally, there are three types of such resources: entities, object properties, and data properties. Entities can only occur as nodes in a graph; they should be considered objects that can be predicated with various values. They can be grouped into so-called classes (one entity can belong to more than one class). Object properties represent binary relations between entities; more precisely, an object property can be attributed to an entity and assign to that entity a different resource as a value. In turn, data properties are also attributable to entities, but their values are alphanumeric strings rather than resources. Both object and data properties can have certain meta-properties, like domain, range, and being functional (attributable only once to a single entity). Additionally, object properties can have such meta-properties as symmetry, transitivity, and reflexivity.

Resource Description Framework (RDF) is a method or language developed by the World Wide Web Consortium (W3C) that codifies the most general principles of representing knowledge according to the SW paradigm. Together with RDF Schema and Web Ontology Language (OWL), which allow for the creation of hierarchies of classes and properties, and SPARQL, a query language used to retrieve and manipulate information stored in SW knowledge bases, it provides the foundations for the technology under discussion. Within an SW knowledge base, the information is chunked into so-called RDF triples: each triple is articulated in the subject, property, and object. The subject can be a resource (an entity or a data/object property); the middle member of a triple can be data or object property; and the object can be a resource or an alphanumeric value. The structures that comprise the

RDF triples are called named graphs. They can be presented in graphic form or stored in text files coded in one of several serialization formats (N-Triples, N-Quads, Turtle, JSON-DL, RDF/XML, and so on). The "name" of a graph is its unique identifier; typically, its URL address.

Although the subject-property-object triple structure seems too limited to express a variety of propositional forms, a couple of RDF syntax additional features broaden the expressiveness of name graphs. One of them is the so-called blank node: a pseudo-entity that can occur in the place of a subject or an object within a triple without being a resource, namely, without having any identifier. Though it can have a name that identifies it locally within a given named graph, it can occur unnamed as well. Blank nodes are useful when we try to represent such propositions as: "John goes to Kirchberg by car." We would try to represent this example with two triples: "John"—"goes to"—"Kirchberg" and "John"—"goes by"—"car." However, such a representation would not inform us that this particular John's trip to Kirchberg is made by car. Therefore, it would be better to use a blank node and produce three triples: "John"—"travels"—[blank node X], [blank node X]—"to"—"Kirchberg," and [blank node X]—"by"—"car." Here, the blank node clearly represents the travel itself that is not explicitly mentioned in the sentence. The situation is presented graphically in Diagram 1.

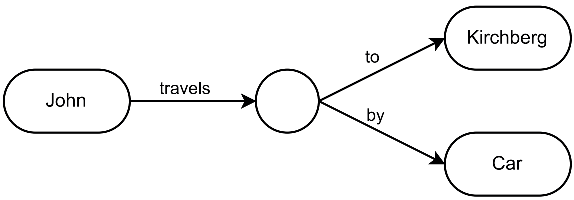

Diagram 1. The named ovals symbolize the entities (nodes); the arrows mark the properties (they point to their values); and the empty circle represents the blank node

Another useful RDF feature is reification, which allows for the representation of a sentence (in fact, an RDF triple) as an object. The object in question (which can be either a resource or a blank node) belongs to the special predefined RDF class "Statement." A "Statement" object has three special object properties: "Subject," "Predicate," and "Object" that have, as their values, the respective members of a triple that is to be reified. The reification mechanism allows us to create representations of intensional contexts such as "Mary believes that John is in Kirchberg": "Mary"—"believes"—[blank node R], [blank node R]—"type"—"Statement," [blank node R]—"Subject"—"John," [blank node R]—"Predicate"—"is in," [blank node R]—"Object"—"Kirchberg." Certainly, there is no direct assertion attributed to the reified part: only the triples can assert anything within an RDF knowledge base. As we can see, the reification syntax is quite complicated (the example discussed above is presented in Diagram 2), but it is inevitable for various purposes that are pertinent to the goal of this paper.

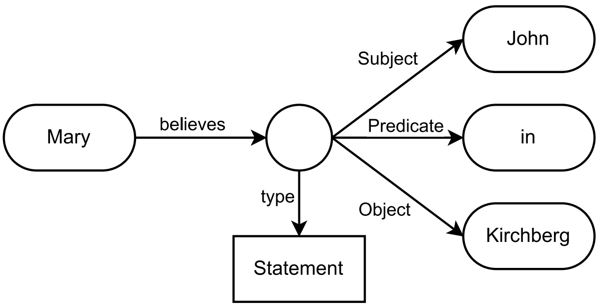

Diagram 2. The rectangular "Statement" node is a class: a group of entities of the same type; the reader should note that the property "in" occurs as a node (a value of the object property "Predicate")

WAB's Solutions to the Challenges of Representing Philosophical Knowledge

The present state of the SW paradigm facilitates its application to a specific kind of knowledge: well-established, clear-cut, non-debatable. Similar to information about a collection of documents, we can represent volumes, single pages, and even paragraphs; we can link them to people, topics, dates, and other documents.[1] We always know to which category a given entity

[1] Other suitable examples can be uncontroversial areas of science such as anatomy and genomics (Rosse & Mejino, 2003; Smith, Köhler & Kumar, 2004; see also Pichler et al., 2021, p. 62).

should be assigned; we know all the relationships between a single entity and others. An example of such an SW knowledge base is the working part of the Wittgenstein ontology project under development by the WAB. The interface of its explorer is available at the following web address: http://wab.uib.no/sfb. In its current form, the project is successful as an automatic index that covers the entire Wittgenstein *Nachlass* and all his published works. It should be underlined that the basic entities that the knowledge base consists of are not separate pages but single remarks that Wittgenstein sometimes numbered and often dated; there are over sixty thousand of them.

The KR^2-friendly knowledge, as was noticed by the group of authors related to the WAB in their 2021 paper "Crisscross Ontology" (Pichler et al., 2021), is characterized by stability, precision, and coherence. The task of representing such knowledge can be compared to arranging jigsaw puzzles: it consumes a significant amount of time, but eventually all the pieces find their right places. Obviously, not all our knowledge can be described in this way. Quite often, especially in science, our knowledge is very far from being precise and stable; there are systematically vague concepts, dynamisms, tensions, and conflicting interpretations. The situation in philosophy is even worse: as has been observed by Ludwig Wittgenstein, the very nature of philosophical investigation prevents us from proceeding along a single track of reasoning and "compels us to travel crisscross in every direction over a wide field of thought" (Wittgenstein, 2009, p. 3). Therefore, the WAB scholars use the term "crisscross" when referring to knowledge that is characterized by vagueness, instability, and multiperspectivism (Pichler et al., 2021, pp. 59–60). It seems that, like in David Chalmers' vision of the philosophy of mind, there are easy and hard problems within the KR^2 domain as well: the task of representing philosophical knowledge appears to be significantly more challenging than creating a knowledge base for a collection of some writings. Nevertheless, this is precisely the main goal of the Wittgenstein ontology project.

The WAB team is fully aware of the difficulty of the task they have undertaken. After trying out some misleading methods based on the apparent structural similarities of formal computational ontologies and philosophical ontologies such as that apparently presented by the *Tractatus Logico-Philosophicus* (Zöllner-Weber & Pichler, 2007), they have become convinced that a new model of knowledge representation is needed (Mácha, Falch, & Pichler, 2013; Pichler et al., 2021, p. 71). So far, their research has resulted in proposing a flat conceptual structure whose central elements are the three classes: “Perspective,” “Claim” (or “Point”), and “Concept” (or “Issue”). The alternative names for the latter two classes given in parentheses occur in the most recent version of the Wittgenstein ontology file (Pichler, Gjesdal & Ruwehy, 2007-, available for download from the following web address: http://wab.uib.no/cost-a32_philospace/wittgenstein.owl), but the WAB researchers have tended to replace them with “Claim” and “Concept” in their recent written (cf. Pichler et al., 2021, p. 62) and spoken statements. However, so far, the proposals the WAB team offers are rather inchoate and theoretical. Although the 2021 paper attempts to outline the relationships between possible members of the three classes, it ends with the caveat that the solutions are not yet definite and sufficiently detailed (Pichler et al., 2021, p. 71). They do not provide any real examples, either.

The Wittgenstein Ontology Class Structure

Before we proceed to the representation of the *Tractatus*’ philosophical content, we need to discuss some issues related to the project’s class hierarchy. Its current version includes the two main branches: the first of them, rooted in the “Source” main class, gathers classes responsible for representing Wittgenstein’s remarks, documents, volumes, and publications. The second, meanwhile, subordinated to the “Subject” main class, is designed for dealing with the philosophical subject matter. While the latter contains the three aforementioned classes responsible for perspectives, claims, and concepts, the former includes classes that are the backbone of the working part of the project. Among them, the two most important are “Nachlass Bemerkung” and “Part”: these are groups of entities that represent particular paragraphs, numbered theses, or remarks (the former is responsible for the unpublished *Nachlass* content, while the latter deals with the published texts). Wider-scope “Source” classes are: “Chapter,” “Werk,” “MS,” and “TS”; these consist of entities representing chapters, published books, *Nachlass* manuscripts, and typescripts, respectively. There is also one narrower-scope class; namely, “Sentence,” which represents single sentences that comprise Wittgenstein’s texts. It should be mentioned that the "Sentence" class is currently empty; the WAB has not yet filled it with entities.[2]

Although my general approach to the current Wittgenstein ontology class hierarchy is conservative, I propose a slight extension to it, which would be particularly important for the focal point of the present paper; that is, the Tractarian philosophy of mathematics. In addition to the existing classes, I shall introduce the "Symbol" class that groups representations of logical and mathematical formulas. The function of the "Symbol" nodes will be closely linked to yet another proposed extension of the class hierarchy: the "Ruleset" class that would be responsible for non-standard content-related automated reasoning within the SW knowledge base. Despite providing very interesting possibilities, the "Ruleset" class will not be discussed any further in the present paper. The third additional class—"Clause"—represents conceptual parts of claims that have no assertion. Typically, "Clause" members are the premises and conclusions of conditional expressions. There will be some examples of the "Clause" class application in the course of the paper.

It seems uncontroversial that the place of "Clause" and "Ruleset" is among "Subject" descendants. In turn, one can dispute the place of the "Symbol" class within the hierarchy. On the one hand, it fits in the "Subject" family because its members are involved in relationships with concepts, clauses, claims, and rulesets. On the other, "Symbol" nodes would represent literal parts of the source material and for this reason the class should rather

[2] Adding "Sentence" nodes would significantly expand the Wittgenstein ontology knowledge base, whose current size is slightly more than 520,000 RDF triples. Assuming that each separate paragraph or remark consists of, on average, three sentences, Wittgenstein's published and unpublished texts comprise roughly two hundred thousand sentences. Each sentence node should be attributed with at least three properties: "type" with the value of the "Sentence" class; "is part of" with the value of a particular "Part" or "Nachlass Bemerkung" entity that includes a given sentence; and a data property with the value being the actual text of a given sentence. (For now let us put aside the fact that a large part of Wittgenstein's texts have translations; therefore, the latter property should be attributed more than once to a number of sentence nodes with values in different languages.) This yields at least six hundred thousand new RDF triples; the knowledge base would therefore double its size. Fortunately, having both the *Nachlass* transcriptions and published texts already divided into remarks, we can easily automate generating these triples. It is worth mentioning that the WAB has already suggested a naming convention for sentence nodes in the comment to the "Sentence" class (Pichler, Gjesdal & Ruwehy, 2007).

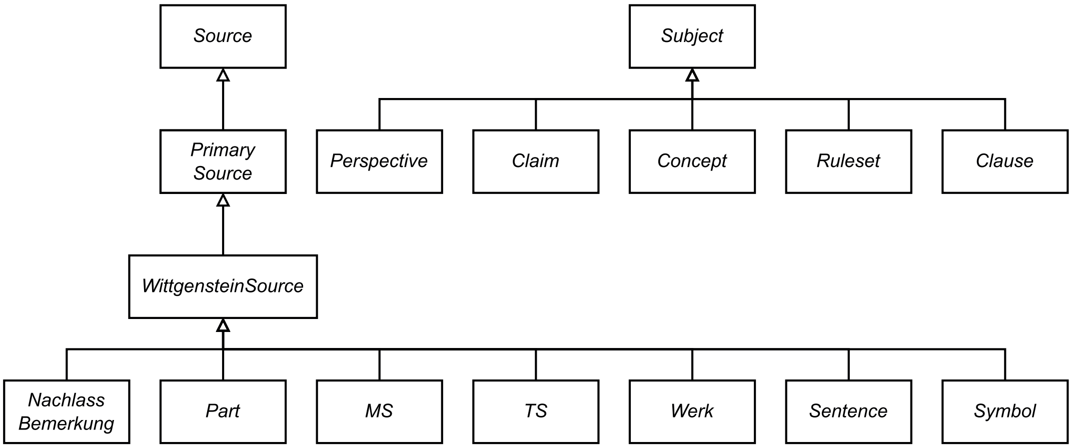

Diagram 3. Fragments of the Wittgenstein ontology class hierarchy (arrows point to parent classes). The “Source” branch is particularly ramified: Both “Primary Source” and “WittgensteinSource” have their siblings (not depicted in the diagram)

be a sibling of "Sentence" and "Nachlass Bemerkung." Tentatively, I shall take the latter option[3] without closing the case. One can also point to another issue related to the "Symbol" class: its name. The early Wittgenstein's philosophy of logic and language makes use of the concept of symbol and juxtaposes it with the concept of sign. Moreover, it appears that it is the latter part of the juxtaposition that should be applied to logical and mathematical formulas as abstracted from any non-formal contexts of use (cf. e.g., Potter, 2008, p. 210n). Nevertheless, we should refrain from mixing up our conceptual ontology with the categorization that is on the subject matter level. An abridged hierarchy of the Wittgenstein ontology supplemented with the three proposed classes is depicted in Diagram 3.

The SW Technology in Practice

The examples of SW representation of the part of the *Tractatus* presented below are made of instances of the classes: "Sentence," "Symbol," "Perspective," "Claim," "Concept," and "Clause." The graphs are composed according to the following general principles:

1) each claim is attributed to a certain perspective; it is also related to both its sentential counterpart (a member of the "Source" branch) and its structure;
2) a structure of a claim is a "Statement" instance having all the properties required for the reification mechanism: a claim's components are values of the properties;
3) a structure of a claim can be nested: Both the subject and object of a reified triple can be "Statement" instances themselves; they can also be "Clause," "Perspective," or "Claim" instances;
4) there are structural meta-properties that are responsible for linking perspectives, sentences, claims, symbols, clauses, rulesets, and statements together; there are also subject matter properties that are components of particular claims.

[3] This choice was strongly recommended by Alois Pichler when we discussed the "Symbol" class proposal.

Let us take a look at the representation of the first sentence of the *Tractatus* 6: “The general form of a truth-function is [p, ξ, N(ξ)].”[4] First of all, we need a perspective to anchor the claim; therefore, we introduce the node “Perspective 1” as a subject in the triple: “Perspective 1”—“type”—“Perspective.” Now, we can introduce another node, “Claim TLP 6 [1],” to represent the philosophical claim expressed with the sentence in question. The node appears in four triples: “Claim TLP 6 [1]”—“type”—“Claim,” “Perspective 1”—“claim”—“Claim TLP 6 [1],” “Claim TLP 6 [1]”—“source of claim”—“Tractatus logico-philosophicus (TLP, 1921/1922) 6 [1],” and “Claim TLP 6 [1]”—“structure of claim”—“Statement TLP 6 [1].” As we can see, the meta-properties responsible for linking a perspective with a claim, a claim with a sentence, and a claim with a statement are called “claim,” “source of claim,” and “structure of claim,” respectively.[5] There are two other nodes that occur in the above triples: the “Tractatus logico-philosophicus (TLP, 1921/1922) 6 [1]” node is of a type “Sentence” and is linked to the “Tractatus logico-philosophicus (TLP, 1921/1922) 6” node of a type “Part” (that node is currently present in the knowledge base). It is attributed with a data meta-property “content” that brings its actual content quoted above (as we shall see, the same meta-property is used to attribute the actual formula to “Symbol” instances). In turn, the “Statement TLP 6 [1]” node is of a type “Statement” and is attributed with the three standard reification properties thus occurring in the three following triples: “Statement TLP 6 [1]”—“Subject”—“General form of a truth-function,” “Statement TLP 6 [1]”—“Predicate”—“expressed with a sign,” and “Statement TLP 6 [1]”—“Object”—“Symbol TLP 6 [1].” The “General form of a truth-function” node is a “Concept,” while the “Symbol TLP 6 [1]” node is an instance of the “Symbol” class.

How should we ascribe the members of reified triples to their classes? There are two possible ways: either we could make a context-free ascription, or we could ascribe an entity to a class within a certain perspective as yet another reified triple. Tentatively, let us take the former option; it will make the graphs much less complicated. However, we should be aware that sticking

[4] All the quotations from the *Tractatus Logico-Philosophicus* come from Pears & McGuinness’s translation (Wittgenstein, 1965).

[5] By convention, the names of properties are written in lowercase; the names of the three reification properties are the exceptions to this rule.

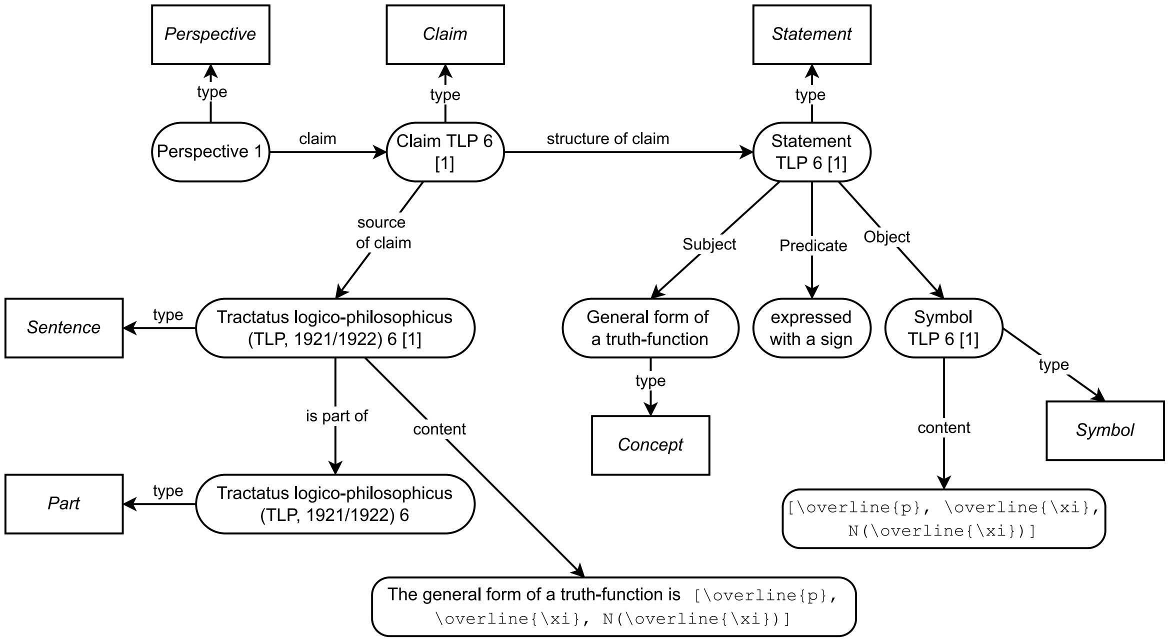

Diagram 4. Rectangular nodes are classes; oval nodes are instances of classes; and rounded rectangles are strings (the values of data properties). The last of these include TEX notation to represent the formula "[$\overline{p}$, $\overline{\xi}$, N($\overline{\xi}$)]"; the "expressed with a sign" node is not assigned to any class: it is an object property that occurs within a reified triple

to this practice can cause us problems in the future: there can be a disagreement between two perspectives about treating a particular part of a given Wittgenstein's sentence as a concept. In such situations, we should instead follow the latter option and represent an ascription as a reified triple.

The graph for the sentence under discussion is shown in Diagram 4. It is a rather complicated structure compared to the syntactic simplicity of the sentence. After all, we have a complex concept to the left, a complex symbol to the right, and a copula that links them together. Meanwhile, the graph is made of twelve triples (plus four triples about the "Sentence" class member). However, in order to make our presentation more concise, we can transform it to a much simpler structure by "resurfacing" our reified triples; that is, by presenting them as if they had not been subject to reification. Diagram 5 demonstrates the first sentence of the *Tractatus* 6 "resurfaced."

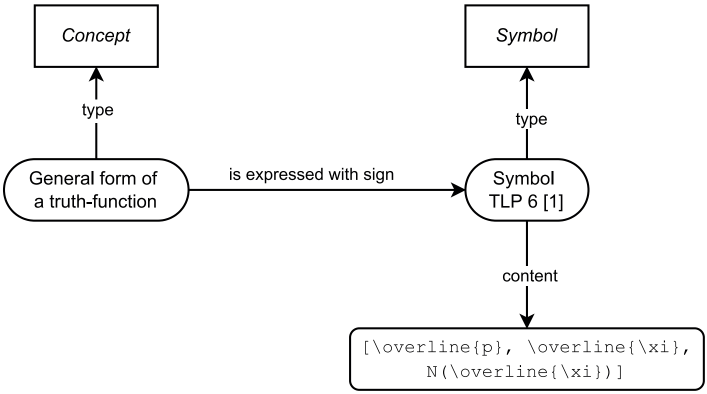

Diagram 5. The triple that represents the first sentence of the *Tractatus* 6

Finally, we come to a true philosophical issue: the meaning of the copula "is." There are two reasons why an SW representation of philosophical content cannot be a simple dissolution of its linguistic form. Firstly, what we call "philosophical content" is a conceptual structure that finds its expression in utterances but does not boil down to them. Sometimes, two separate words are used to express the same conceptual content; likewise, the same word can be used in a couple of different meanings. Secondly, a usable SW knowledge base must have a controlled vocabulary. In other words, the number of resources must be limited if we want to be capable of making effective searches. This is particularly true for object properties.

Our object property that stands for the linguistic “is” is called “is expressed with sign.” This is just a name for an RDF resource and can be replaced by anything; for example, “OP-TLP-6-1.” However, the crucial matter is in what triple the same property is used again and in what triple it is not. For instance, it does not occur in the representation of the second sentence of the *Tractatus* 6, despite the fact that the latter says, “This *is* the general form of a proposition” (italics mine). The structure of the relevant triple (“resurfaced”) is shown in Diagram 6.

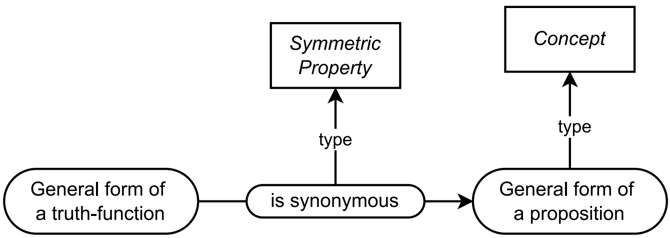

Diagram 6. The triple that represents the second sentence of the *Tractatus* 6. The main object property “is synonymous” is marked by an oval because the graph also represents its being a member of a certain class of properties

Here we have “is synonymous” in place of the copula. Again, it is just a name that can be replaced by, for instance, “OP-TLP-6-2.” However, the point is that this is a different property than the property “is expressed with sign” and, unlike the latter, it is symmetric. I have also “unwrapped” the anaphoric reference to the previous sentence by putting the right concept instead of the pronoun. The reason why I read “A general form of a truth-function is general form of a proposition” in this way is that the *Tractatus* demonstrates that a proposition is a truth-function.[6]

[6] One can notice that, according to the *Tractatus*, the concept of proposition is rather explicated by the concept of truth-function than synonymous with it (cf. Wittgenstein, 1965, p. 5). However, I believe that, in general, a relationship between complex concepts can be different from a relationship between their crucial differentiating components. Moreover, the concept of general form of a truth-function can be replaceable with the concept of general form of a proposition exactly for the reason that the

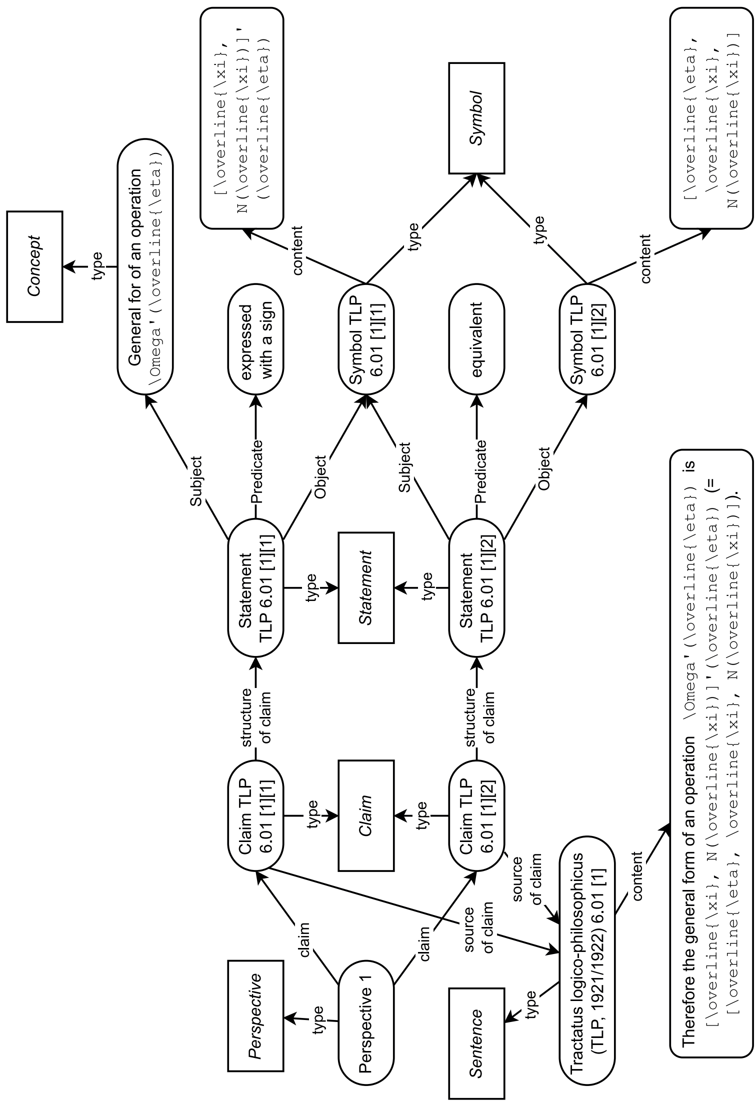

Diagram 7. Two claims, two statements, and, consequently, two reified triples.
The symbol "Symbol TLP 6.01 [1][1]" is both an object of the first triple and a subject of the second triple

There are two other places in the *Tractatus* 6–6.031 and 6.2–6.241 where we make use of the "expressed with a sign" property: the first sentence of 6.01 and the first (and only) sentence of 6.03. Both ascribe symbols to concepts, and both are translations of the copula "is." In fact, both are structurally very similar to the first sentence of the *Tractatus* 6. Let us take a closer look at the 6.01 example that is shown in Diagram 7.

The *Tractatus* sentence represented in the graph above is an example of a situation in which a single sentence corresponds to more than one claim. The additional content is placed in parentheses and states that the symbol "$[\overline{\xi}, N(\overline{\xi})]'(\overline{\eta})$" is equivalent to the symbol "$[\overline{\eta}, \overline{\xi}, N(\overline{\xi})]$." As we can see, the graph is rather complicated, consisting of twenty four triples.

There can also be a situation in which two or more reification statements correspond to a single claim. An example of such a situation is the *Tractatus* 6.001: "What this says is just that every proposition is a result of successive applications to elementary propositions of the operation $N(\overline{\xi})$." This thesis comprises only one sentence, and it makes only one claim. However, the structure of the claim is too complicated to be represented by just one triple. Diagram 8 shows the graph for that part of the *Tractatus*.

The graph includes yet another resolution of an anaphoric reference: in the beginning phrase ("What *this* says..."), the pronoun clearly points to the symbol "$[\overline{p}, \overline{\xi}, N(\overline{\xi})]$," but we only know that from the context of the rest of the sentence, not from its grammatical structure. It happens that the anaphor is ambiguous, and the way we resolve it sometimes depends on our reading of the text. Fortunately, this is not the case here, but ambiguous pronouns do occur in the text of the *Tractatus*.

The second claim of the *Tractatus* 6.001 also includes the copula "is." Here, like in a couple of other places—6.01 and 6.021—it is represented by the property "explicated as." So far, this is the third reading of the copula that has been discussed, but we should also point to yet another property used to render it: "is a kind of." The latter appears in 6.2 and 6.235 ("Mathematics"—"is a kind of"—"Logical method"),[7] as well as in 6.24 ("Method by which

concept of proposition is explicated by the concept of truth-function: the explication that occurs on a lower level results in synonymity on a higher level.

[7] It should be pointed out that the first sentence of the *Tractatus* 6.2 and the only sentence of the *Tractatus* 6.235 are represented by the identical reified triple because the differences between the two sentences are only superficial. However, instead of linking the same claim with two source sentences, we create two separate claims and two

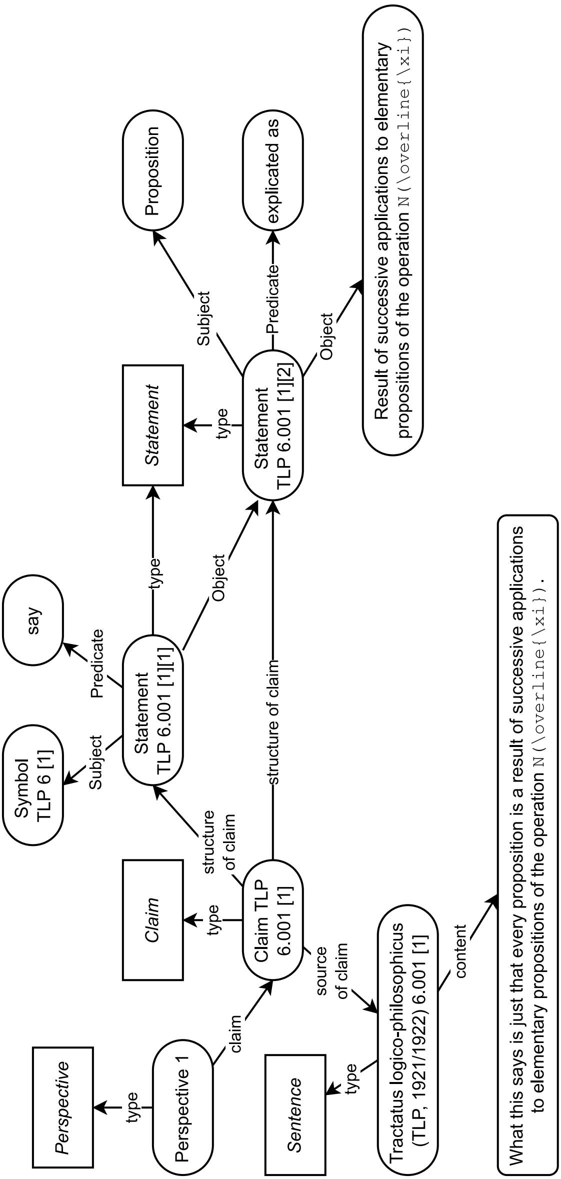

Diagram 8. One claim, two statements, and two triples. The reader should note that despite being an object of the first triple, “Statement TLP 6.001 [1][2]” belongs to the same claim and the same perspective as “Statement TLP 6.001 [1][1]”

mathematics arrives at its equations"—"is a kind of"—"Method of substitution"). Thus, we have four different representations of "is" in our knowledge base: the first of them, "is expressed with sign," signifies the relation of a given concept being expressed by a given symbol; the second, "is synonymous," means that two concepts have the same meaning and are interchangeable; the third, "explicated as," shows that an object-concept brings more information about the meaning of a given subject-concept. Finally, the fourth, "is a kind of," informs that a subject-concept is a specific type of some broader concept given as a value.

Let us move to yet another interesting cluster of examples. In the first sentence of thesis 6.232, Wittgenstein makes a critical comment about a certain alleged view of Gottlob Frege: "Frege says, that the two expressions have the same meaning but different senses." This sentence includes an anaphoric reference that is not immediately visible: the phrase "the two expressions" refers to one of the previous theses; namely, the first sentence of thesis 6.23 dealing with two expressions connected with an equals sign: "If two expressions are combined by means of the sign of equality, that means that they can be substituted for one another."[8]

Let us begin with the latter sentence, which is an example of a conditional statement. First of all, we should resist the temptation to represent it as a FOL rule: this would be as equally futile as trying to construe a philosophical ontology as a computational ontology. Instead, we make use of an object property "entail" that should also be used for similar examples for other in-text inferences.[9] The property links two "Clause" instances that represent the premise and conclusion. Neither of the two are easy to represent. After resolving the anaphoric reference, which is a non-trivial step, we can obtain the following two clauses: "The expression x is combined with the expression y by the sign of equality" and "The expression x can be substituted by the expression y, and vice versa." We are forced to use variables because both clauses deal with the same pair of expressions. Hence, we need some formal device to introduce variables to our representation. We can do this

statements. This enables some possible future cross-perspectival links with standpoints that interpret the two sentences differently.

[8] The reason why I believe that the reference points to thesis 6.23 rather than to 6.231, which is the direct predecessor of thesis 6.232, is that the basic structure of the *Tractatus* is a tree (cf. Stern, 2019).

[9] Later, such examples will be handled by a mechanism related to the "Ruleset" class.

by means of yet another meta-property that can be called "variable of"; a concept variable will be attributed this property with a value of a concept being the range of the variable. For simplicity's sake, we shall not introduce any mechanism of limiting the scope of variables; therefore, all the variables within a given knowledge base will be global. We should keep this in mind and use naming conventions to make an order. Therefore, the concept variables in 6.23 will be named "Expression (TLP 6.23 [1]) variable x" and "Expression (TLP 6.23 [1]) variable y." The subsumption of variables to their concepts can happen outside of the mechanism of reification.

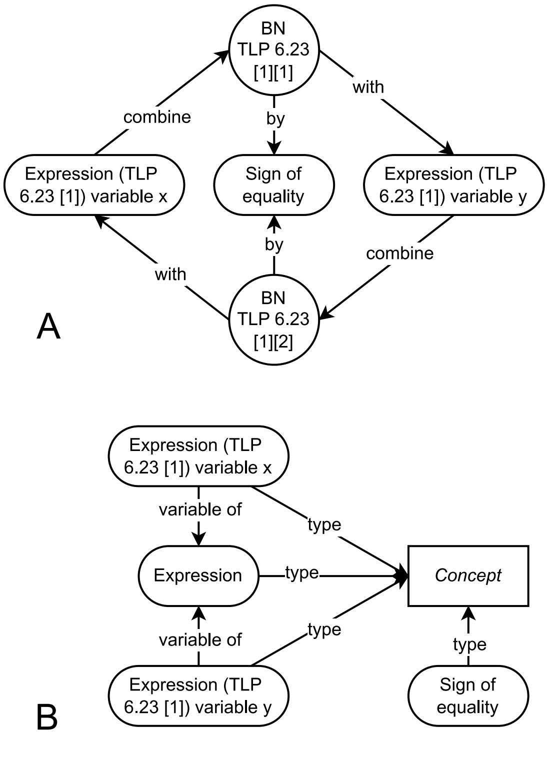

Diagram 9. Figure A is a "resurfaced" part of the graph that will be represented as reified; figure B represents the triples that can occur outside of the reification syntax. Blank nodes (named) are represented as circles

The premise poses another question: how to represent its structure? A less complicated solution is to make a simple triple: “Expression (TLP 6.23 [1]) variable x”—“combined by equals sign”—“Expression (TLP 6.23 [1]) variable y.” There are two shortcomings of such an approach, however: firstly, a very specific object property is added to the vocabulary; secondly, the sign of equality does not figure among the concepts that occur in the claim. Instead, we can take a different approach: use a blank node as a value of the simpler property “combine” together with the two auxiliary properties “by” and “with.” One weakness of this version is that we cannot simply make “combine” symmetric; to represent the symmetricity of the relationship between the two variables, we have to double the graph and swap them. Diagram 9 depicts the complete solution of the premise.

Now, we need to subsume the whole graph to one “Clause” entity that would represent the premise. We cannot subsume it to the claim directly because a premise in a conditional statement is not asserted. All six triples that link both expression variables, two blank nodes, and the “Sign of equality” concept are reified, and the “Statement” entities are attached to the “Clause” member called “Premise TLP 6.23 [1]” by the object property “structure of clause.” The whole situation is shown in Diagram 10.

The graph consists of thirty-one triples. It is rather complicated, and yet it is not connected to any perspective. However, since we assume that the “Clause” elements lack assertion, there is no need to anchor them directly in any “Perspective” instance: they will be free-floating “global” objects within our knowledge base. Similarly, the conclusion of the first sentence of thesis 6.23 is such a “global” object; however, it is much less complicated, as can be seen in Diagram 11.

Having the premise and conclusion, we are ready to represent the sentence in question as a reified triple “Premise TLP 6.23 [1]”—“entail”—“Conclusion TLP 6.23 [1].” Diagram 12 depicts the relevant graph.

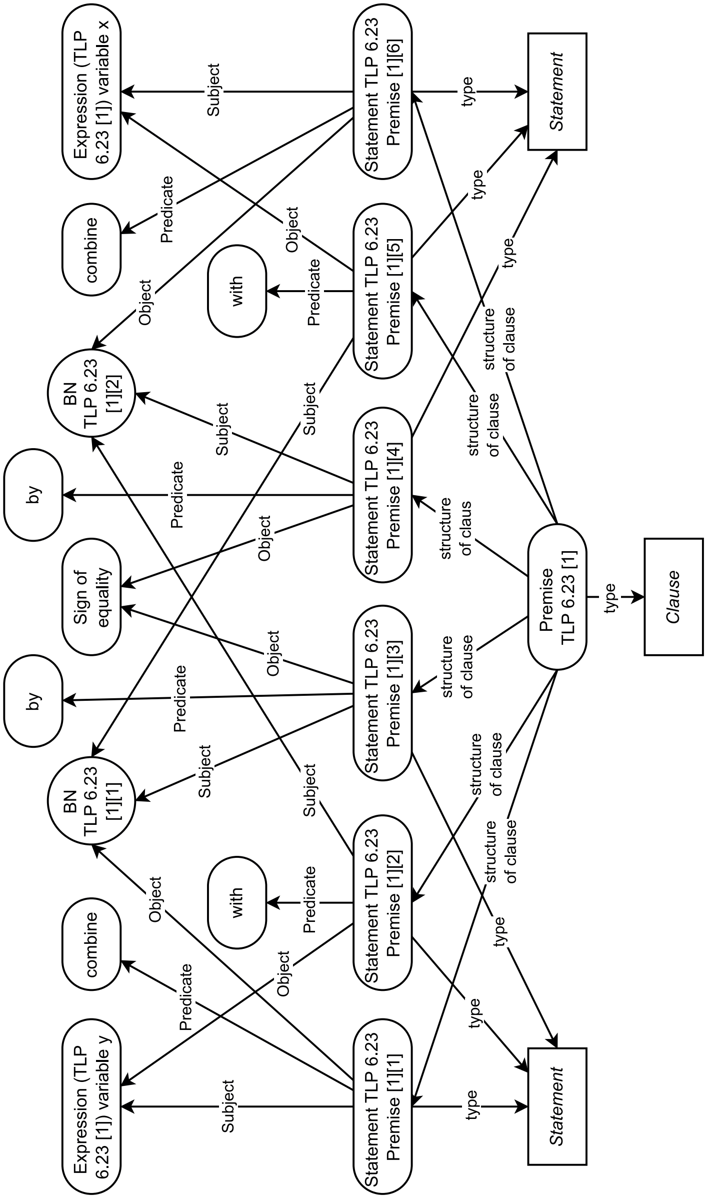

Diagram 10. The premise of the first sentence of the *Tractatus* 6.23. The nodes "combine," "with," "by" (properties), and "Statement" (class) are doubled to make the graph easier to read

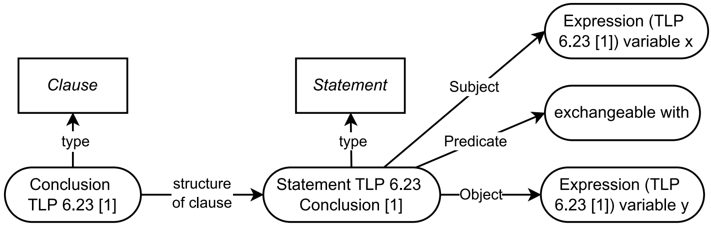

Diagram 11. The conclusion of the first sentence of the *Tractatus* 6.23

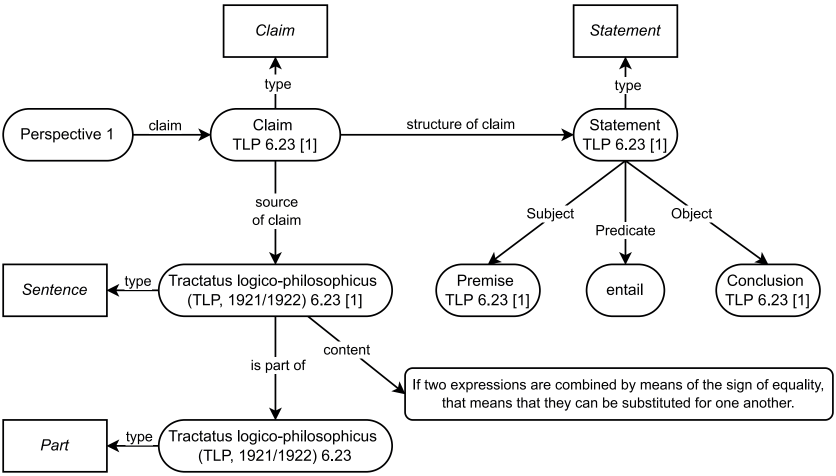

Diagram 12. The general structure related to the first sentence of the *Tractatus* 6.23

Now we can come back to the first sentence of thesis 6.232: "Frege says that the two expressions have the same meaning but different senses." We know now that the expressions are in fact the two expression variables we have dealt with; that is, "Expression (TLP 6.23 [1]) variable x" and "Expression (TLP 6.23 [1]) variable y"; Wittgenstein says that Frege says that they have the same meaning but different senses. In other words, according to the author of the *Tractatus*, Frege believes that the two expressions that are combined together with the sign of equality have the same meaning but not the same sense. The assumed Fregean claim can be presented as a conditional statement: if A, then B; that is, if the two expressions are combined, their meanings and senses are accordingly related. The reader should note that we have already defined clause A: it is the “Premise 6.23 [1]” entity that is a free-floating element of our knowledge base. Therefore, our task is reduced to reconstructing clause B. At the general level, it is a conjunction of the two separate clauses. In fact, we can try to represent them as follows: “Expression (TLP 6.23 [1]) variable x”—“same meaning as”—“Expression (TLP 6.23 [1]) variable y” and “Expression (TLP 6.23 [1]) variable x”—“different sense from”—“Expression (TLP 6.23 [1]) variable y.” The shortcoming of this approach is that we add the two very specific object properties to our knowledge base. If we want to avoid this, we should instead take a different approach. First, we create four additional concept variables: two for “Meaning” and the other two for “Sense” concepts. Subsequently, we declare that the “Meaning” variables are actually the same entity, while the “Sense” variables are distinct. Analogously to the case of the “Expression” variables, we can establish relationships between the four variables outside of the context of any perspective. Thus, the final structure of clause B can be slightly simpler than the structure of the premise. This is shown in Diagram 13, together with graphs representing the relationships between the four additional variables.

Finally, we are ready to represent the first sentence of thesis 6.232. This is an example of a nested perspective: our familiar “Perspective 1” includes the claim that states that “Fregean TLP 6.232 perspective”—yet another instance of the “Perspective” class—includes a claim being the conditional statement whose premise is “Premise TLP 6.23 [1]” and conclusion is “Conclusion TLP 6.232 [1].” The latter claim, unlike all the previously discussed instances of the “Claim” class, does not have “source of claim” property because it is a part of Wittgenstein’s reconstruction of Frege’s view. The whole situation is depicted in Diagram 14.

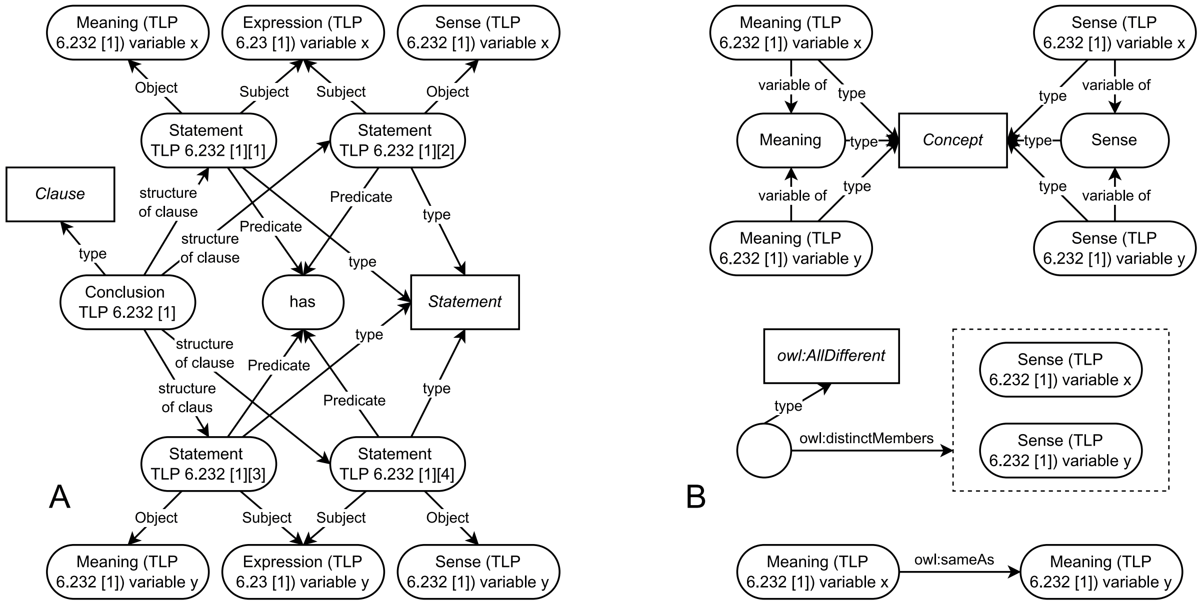

Diagram 13. Figure A is the structure of the reconstructed Fregean conclusion; figure B represents the triples that define the relationships between the four new variables. The dashed rectangle represents an RDF "Collection" entity (collections store lists of elements); the "owl:AllDifferent" class together with the "owl:distinctMembers" property make the OWL syntax that allows for the declaration of a group of individuals as distinct resources. The property "owl:sameAs" means that the related entities are actually the same resource (cf. Dean et al. 2009)

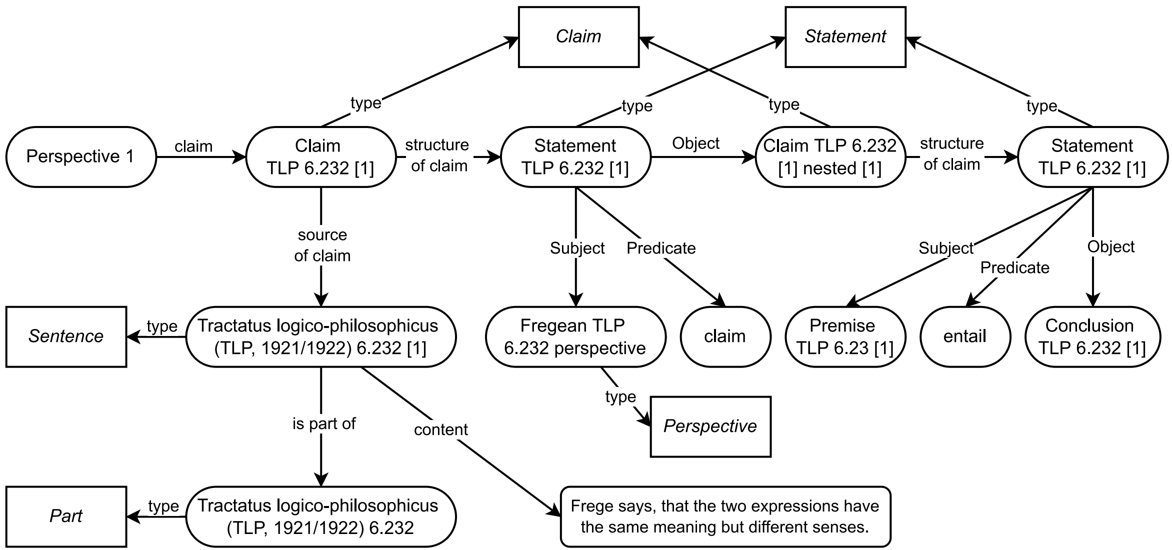

Diagram 14. The graph that represents the first sentence of thesis 6.232. The nodes "Perspective 1," "Premise TLP 6.23 [1]," and "Conclusion TLP 6.232 [1]" are assumed to have been defined earlier

Concluding Remarks

Developing a representation of philosophical content according to the SW paradigm is a strange form of translation: we try to render a philosopher's natural language into a controlled vocabulary; that vocabulary is a result of our own choices. Therefore, it is a game we play against ourselves: we try to limit our means of expression without reducing the meaning that is conveyed. In the course of this game, we have to make a number of philosophically weighty decisions. Some of the interpretations I offer can prove controversial, such as a particular reading of the copula or an introduction of a conditional statement that is not visible in the source material. One of the reasons for these moves is that my attempt concerns the conceptual structure of the philosophical doctrines presented in Wittgenstein's published and unpublished works, and that conceptual structure can be, from time to time, hidden under the surface of its linguistic expression.[10] The other is that the SW technology is generally based on binary relations, and any more complicated syntax must be represented as a structure made of them. The third reason is that I think of the *Tractatus* representation as a device that should perform a certain function once completed: it should be possible to create a user-friendly web interface that would enable scholars or students to search through the knowledge base and find useful information about Wittgenstein's conceptions. The working part of the Wittgenstein ontology project is already such a device with a comprehensive web interface. Although it does not do much more than linking dates and persons to Wittgenstein's particular remarks, its functionality enables one to check all the places in the *Nachlass* where Sigmund Freud, for example, is mentioned. The simplicity of its interface results from the fact that the Wittgenstein ontology in its current shape makes use of just a handful of properties. Unfortunately, I will not avoid a large number of object properties when I complete the "Subject" branch in a way suggested by my examples. However, I can try to reduce that number to make the project's interface design less challenging.

The SW technology was not created to represent philosophical ideas; if we try to do this anyway, we should expect our representations to be complex. Fortunately, the complexity develops according to certain patterns; therefore,

[10] Such a claim does not need to presuppose a doctrine of the inner language of thought, although such a doctrine is actually proposed by the *Tractatus* itself.

they can be easily automated with the help of a piece of uncomplicated custom-made software.

References

Bechhofer, S., Stein, L. A., Patel-Schneider, P. F., McGuinness, D. L., Horrocks, I., Hendler, J., & van Harmelen, F. (2009). Owl Web Ontology Language Reference. https://www.w3.org/TR/owl-ref.

Mácha, J., Falch, R., Pichler, A. (2013). Overlapping and competing ontologies in digital humanities. In F. Tomasi, & F. Vitali (Eds.), *DH-CASE '13: Proceedings of the 1st International Workshop on Collaborative Annotations in Shared Environment: Metadata, Vocabularies and Techniques in the Digital Humanities (10 September 2013, Florence, Italy)*. ACM.

Pichler, A., Gjesdal, Ø. L., & Ruwehy, H. A. (2007– work in progress). *Wittgenstein ontology*. WAB. http://wab.uib.no/cost-a32_philospace/wittgenstein.owl.

Pichler, A., Fielding, J. M., Gangopadhyay, N., & Opdahl, A. L. (2021). Crisscross ontology: Mapping concept dynamics, competing argument and multiperspectival knowledge in philosophy. *Quaderni di "Filosofia"*, *2*, 59–73.

Potter, M. (2008). *Wittgenstein's notes on logic*. Oxford University Press.

Rosse, C., & Mejino, J. L. V. Jr. (2003). A Reference ontology for bioinformatics: The foundational model of anatomy. *Journal of Biomedical Informatics*, *36*, 478–500.

Smith, B., Köhler, J., & Kumar, A. (2004). On the application of formal principles to life science data: A case study in the gene ontology. In R. Erhard (Ed.), *Data Integration in the Life Sciences. DILS 2004. Leipzig. Lecture Notes in Computer Science* (pp. 79–94). Springer.

Stern, D. (2019). The structure of *Tractatus* and the *Tractatus* numbering system. In C. Limbeck-Lilienau, & F. Stadler (Eds.), *The Philosophy of Perception: Proceedings of the 40th International Ludwig Wittgenstein Symposium* (pp. 377–398). De Gruyter.

Wittgenstein, L. (1965). *Tractatus logico-philosophicus* (D.F. Pears, & B.F. McGuinness, Trans.). Routledge & Kegan Paul.

Zöllner-Weber, A., & Pichler, A. (2007). Utilizing OWL for Wittgenstein's *Tractatus*. In H. Hrachovec, A. Pichler, & J. Wang (Eds.), *Papers of the 30th International Ludwig Wittgenstein Symposium (5–11 August 2007, Kirchberg am Wechsel)* (pp. 249–251). ALWS.

Author Note

Jakub Gomułka, PhD (hab), Prof. AGH at the Faculty of Humanities of AGH University of Krakow, is interested in the philosophy of Ludwig Wittgenstein and Stanisław Lem, as well as the application of AI technology in the humanities. His recent publications include the following papers: *Tractatus 6 Reconsidered: An Algorithmic Alternative to Wittgenstein's Trade-Off* (2023, co-authored with Adam Roman), *Wittgenstein's "On certainty" and Rational Argumentation* (2021), *Cantor's Paradise from the Perspective of Non-revisionist Wittgensteinianism* (2020), and *Meaning Before Subjectivity: The Primäre Sprache of the Tractatus* (2019).

Address for correspondence: Faculty of Humanities, AGH University of Krakow, Czarnowiejska 36, 30-054 Kraków, Poland.

Citation

Gomułka, J. (2023). Towards a Computational Ontology for the Philosophy of Wittgenstein: Representing Aspects of the *Tractarian* Philosophy of Mathematics. *Analiza i Egzystencja*, *63* (3), 27–54. DOI: 10.18276/aie.2023.63-02.