This document contains a condensed and simplified version of my Bachelor Thesis, retaining most of the introductory section alongside selected parts of the mathematical and philosophical components. While the introduction has been largely preserved, the conclusion has been significantly shortened, and several critical sections requiring technical explanations and definitions have been omitted to save space. Portions of the introduction that closely follow traditional treatments in the literature have also been removed. Importantly, every definition quoted in this document has a proper set-theoretic formulation in the original thesis. For instance, notions like "Language*" are formally defined in the original document and can be located by searching for the respective terms at the link provided.

I have chosen to include one of the thesis's key results: the construction of a translation into the language of set theory, starting from the simpler notion of encoding. However, the scope of the thesis is much broader and connects to several other works I am currently pursuing. First, it relates to the Research Project on Categorical Foundations to Structuralism, where translation theory provides the building blocks for defining structuralism in a non-categorical way. This serves as a competing foundation to the main categorical approach developed by Halvorson and explored in the rest of the project.

Additionally, my Master Thesis will incorporate van Benthem's literature on the same topic. Unfortunately, during the writing of my Bachelor Thesis, I was unaware of van Benthem's work on this precise topic. Now that I am familiar with it, I plan to base my Master Thesis on this literature, collaborating with the research group on Theories of Truth at the ILLC, with whom I will write the thesis.

As a final note, I decided to submit this more technical writing as a sample because I firmly believe that a strong technical background is essential for a successful career in Mathematical Philosophy, Philosophy of Mathematics, and Logic. While many of my previous and current writings are more philosophical in nature and have received greater recognition within the humanities, I believe that the aim of the writing sample is to demonstrate my competence in my field of research. In this regard, I believe that showcasing a technical work in Mathematical Philosophy will best reflect my abilities and expertise.

Finally I give some data on the following file:

Number of Words:	≤ 10.000
Number of Pages:	19
Date of Writing:	July 2024

A Formal Approach to Syntactic Structures

Language Theory as a Field in Mathematical Logic

Simone Testino

August 2024

Abstract

After thoroughly explaining the methodology of Mathematical Logic, I propose a fully formalised approach to defining formal languages. This involves presenting the primitive notions, establishing all required definitions, and illustrating the principal relationships between formal languages. More precisely, I give the definition and construction methods of formal translations between different languages, in particular those that can be algorithmically defined from an encoding into the language of set theory.

> Siquidem pene totum humanum genus ad opus iniquitatis coierat [...] cum celitus tanta confusione percussi sunt, [...] ab opere, multis diversificati loquelis, desinerent, et nunquam ad idem commertium convenirent.
>
> Dante, *De Vulgari Eloquentia*, I.vii.6

Motivation

The translation of natural languages presents a formidable challenge due to the inherent imprecision and cultural nuances embedded within them. Translators must convey the semantic meaning and maintain the intended tone and syntactical structure while navigating the transformation of symbols, sounds, and cultural contexts. This complexity has historically attracted intellectuals, driven by both practical necessity and the intellectual puzzle it poses.

In contrast, formal languages serve as the foundational bedrock across technical disciplines such as sciences, mathematics and computer science. They encompass various mathematical languages rooted in distinct foundations or specialized fields of application.

In my research, I am consolidating existing technical results from mathematical logic, the philosophy of logic, and formal linguistics. My primary aim is to formalize this body of work rigorously, ensuring precision comparable to that found in other fields of pure mathematics. This involves striking a balance between traditional approaches found in existing literature and introducing novel definitions that enhance precision and generality.

By focusing on formalization, I aim to elucidate the theoretical underpinnings and practical applications of formal languages. This approach not only enhances clarity but also fosters deeper insights into the relationships between different formal systems, thereby advancing the understanding and development of these foundational languages across disciplines.

Another major focus of this work is the syntactical approach. To achieve the desired level of formality, I will first concentrate on the most foundational level, which is that of syntax. This study will solely examine syntactical objects, their structures, and their relationships, remaining entirely separate from semantics or any denotation to metaphysical entities. This approach allows for broader application of the study to various semantical or metaphysical interpretations.

Translation of the epigraph: "Almost the whole of the human race had collaborated in this work of evil [...] until they were all struck by a great blow from heaven [...] now they were forced to leave off their labours, never to return to the same occupation, because they had been split up into groups speaking different languages." On the myth of the construction of the Tower of Babel, for a broader context consult the Princeton Dante Project.

1 Introduction: Syntactic Proof Theory

In this introductory section, I provide the necessary definitions to proceed and clarify various commonly used terms in model theory and formal linguistics. Specifically, I will explain the differences between three distinct levels of language syntax. This will equip the reader with a clearer structure, facilitating a better understanding of the more complex concepts discussed later in the sections. Additionally, I will introduce the three different (meta)languages required to conduct any formal proof in mathematical logic. This section will also contain references to [Shapiro &Kissel 2024], which represents the most traditional approach to the definition of classical logic. Maintaining a comparison with it will demonstrate to the reader why my formalism, in its general structure, is not original but follows traditional approaches.

The second section will explore the structural outcomes for the languages and objects defined in the first section. I will specifically focus on the concept of Translation*, which acts analogously to homomorphism defined on syntactic structures. This section discusses the implications of languages being homomorphic, addressing both weaker and stronger relational aspects. To ensure clarity while introducing new concepts, each set of results begins with a motivating example followed by proof methods that provide clear procedures for tackling similar cases.

I will also use the Appendix to make comparisons to the literature and provide a more in-depth explanation of the deviations I take from it. I chose to present these arguments only in a brief form in the main text, since I want to focus on the constructions and its relative independence from the existing literature.

Introduction to Syntactic Proof Theory From a purely formal perspective, I do math. This is to say that each theorem I prove can be formulated in the language of set theory (denoted as $\mathcal{L}_\epsilon$) and demonstrated from the axioms of ZFC[1]. Conceptually, these proofs differ significantly from those typically seen in mathematical practice, as they involve languages, well-formed formulae, symbols, and other syntactic entities that are not extensively discussed in most mathematical fields.

Furthermore, it is essential to acknowledge that I am using mathematics to establish theorems concerning, among others, mathematical syntax. Any concerns regarding this approach are addressed in Section 1.2, *Three Metalevels*.

Because of this distinction, it is necessary to understand the difference between a symbol within a well-formed formula and a symbol taken by itself. For instance, consider the formula $0 + x = 0$, which is a well-formed formula in the language of groups $\mathcal{L}_{gr}$ with one free variable x. This is radically different from referring directly to the symbol "+". In such a syntactic environment, one is often required to construct functions from symbols to symbols. To facilitate this, I denote symbols taken by themselves with a dot on top. This notational distinction is presented again after the coming section which give the necessary tools to fully understand it, see Justification to the Dot 1.1.

1.1 Three Levels of Syntax

To begin this chapter, I introduce several syntactic elements: Terms 1.1, Relations on them 1.2, and Relations on Formulae 1.3. With these elements, my aim is to first define the notion of Language* and then the one of Logic*. To maintain a clear structure for all these definitions, it is helpful to distinguish three Levels of Syntax. These levels represent different degrees of *meaning* conveyed by symbols. For instance, the lowest level deals strictly with symbols, the next organises them through grammatical rules, and the highest incorporates Inference Rules* that govern relationships between such formulae.

Each definition will clearly belong to one of these three levels depending on how it is denoted. Indeed, each level has its own specific notation. This distinction of levels, while formally definable, serves a purely didactic purpose here. I differentiate terms solely through notation, which therefore has no concrete or necessary impact on the proofs. However, it is crucially important for the reader to understand the structure of these linguistic objects.

I. Pure Syntax The most foundational level is that of pure syntax. Objects at this level are denoted by capital Greek letters (such as $\Sigma^{\Phi}_{\mathcal{L}}$, see 1.3). Elements in these sets are simply symbols or sets of

[1]This is detailed more precisely in Primitive Notions 1.2 and Appendix IV: On the Naturality of Primitive Terms*

symbols, like $\{\dot{+}, \{\}\}$ and $\dot{R}_n$, and they do not need to be well-formed or adhere to any grammatical structure. In a formal sense, any element can be conceived as part of this level, since ultimately, every element will either be a symbol or a set of symbols (or pure sets as in standard set theory).

However, from a notational perspective only, symbols appearing in formulae like $x = y$ will be considered together with their grammatical roles, similar to how 0 is not merely "just a set" but serves as the neutral element in groups.

II. Grammatical Syntax Sets at this level are denoted by capital calligraphic letters and are all subsets of $\mathcal{L}$, the set of all well-formed formulae. This means that each element of these sets must be a formula that adheres to grammatical rules, also known as *well-formed formula*. Apart from grammar, no additional rules such as Inference Rules* are applied.

For example, theories, which are any set of formulae[2], are represented by $\mathcal{T}$ and fall into this category.[3]

III. Inferential Syntax Sets at this level are denoted by Frankfurt letters (such as Logic*, $\mathfrak{L}$). These sets are enriched through Syntactic Consequence* derived by the Inference Rules*, representing the most complex state of syntax; any additional step would fall into the semantic domain. Models and structures also belong to this level[4], but these are beyond the scope of this introductory part.

Justification to the Dot

Among the notational conventions outlined here, the most important is the use of a dot above symbols and sets. Essentially, any object discussed at the metalevel (introduced in the following section) will be either a symbol or a set of symbols, and may be written with a dot above it, such as $\dot{\varphi}$[5]. However, there are instances where $\dot{\varphi}$ is not merely considered as a set of symbols, but as a well-formed formula, i.e., a set of symbols adhering to grammatical rules; in these cases, I will simply write φ.

It can be useful to compare this with other analogous situations in mathematical praxis. Consider the structures $(\mathbb{Q}, <)$ as a total order, $(\mathbb{Q}, +)$ as an abelian group, and finally $(\mathbb{Q}, +, \cdot)$ as a field. Now, if we represent all these structures in set theory, we can certainly claim that 3 is 3 in all structures, i.e., it is represented by the same set, even though it satisfies different formulae. Typically, in mathematical practice, this distinction is not particularly relevant, and thus making distinctions like $3^{<}$, 3^{+}, and $3^{+,\cdot}$ (such that $3^{<} = 3^{+} = 3^{+,\cdot}$) would not be very helpful. However, it is not difficult to imagine circumstances, such as the ones I will soon explore, where such a notational distinction becomes crucial.

There are relations that I only apply to strings (formulae with a dot) and not to plain formulae. For instance, consider the "$\in$" symbol. I can write $\varphi \in \mathcal{L}$ as well as $\dot{\varphi} \in \mathcal{L}$, but I will not write $+ \in \varphi$ as it would certainly create confusion[6]. Instead, I will use $\dot{+} \in \dot{\varphi}$, which holds if and only if $\dot{+}$ is a symbol in the string $\dot{\varphi}$, an ordered set of symbols that allows for repetitions. Furthermore, I will define Encoding Functions 3.1 & 3.6 for symbols, as in Example*. Specifically, I will write $\varepsilon(\dot{+})$ but not $\varepsilon(+)$. It is important to note that these distinctions are purely notational; therefore, readers are free to ignore the dot and understand it simply as indicating that the symbol belongs to the object level that I present in the next section.

---

[2]For this discussion, I define a theory simply as a subset of any language; it may be consistent or inferentially closed, though these properties are not assumed for an arbitrary theory. Thus, the actual Syntactically Closed Theory* will depend on the Logic*, i.e., on the Inference Rules*.

[3]This distinction is also highlighted in belief theory in [Cherniak 1986], where he distinguishes between theories of assent, i.e., those where "no inferences, however obvious and useful, need be made from the beliefs, and the belief set can include any and all inconsistencies." (p. 164), criticising, in my terms, belief sets for being in grammatical syntax instead of inferential syntax.

[4]While they can properly be analysed syntactically, one may well argue that an analysis of theories through their models is a Semantical one, see Structure of Scientific Theories. The diversity of possible correspondences of modes, of such a semantical taste, are presented in Structure on Models

[5]Strings will be ordered sets that allow repetitions, clearly there are many ways to write such sets down starting with simple sets of symbols, an example is: $\dot{\varphi} = \{\dot{x}, \{\dot{+}\}, \{\{\dot{y}\}\}, \{\{\{\dot{=}\}\}\}, \{\{\{\{\dot{x}\}\}\}\}\}$ for the proposition $\varphi = x + y = x$.

[6]One might argue that such confusion arises not from the inability to distinguish symbols used purely as symbols from those adhering to grammatical rules, but rather from the difficulty in distinguishing different metalevels. I will address these concerns in the next section. Readers may also consider that symbols marked with a dot must always belong to the object level (as I explain in the following section), and I will clarify the distinction of metalevels using other notational tools when I find it helpful.

1.2 Three Metalevels

The three distinct Levels of Syntax 1.1 I have presented provide the framework for the tools required to conduct any mathematical proof on formal languages. It is crucial to understand clearly how these languages are interconnected. The reader will notice that there are three languages involved in the forthcoming definitions and proofs. I introduce each language and provide the notation necessary to differentiate them. This distinction is particularly important because all these languages could potentially refer to the language of set theory but on different metalevels.[7]

Famously, the necessity of a metalanguage is argued by Tarski in [Tarski 1933]. In this case the usual object languages are the mathematical objects I consider, the usual metalanguage is my object language and the further meta level is my metalanguage, whose necessity famously explained in [Tarski 1933].

I. Object Languages

The object language is the one we are most used to in mathematical praxis, languages of these sorts are those I want first to define and whose relations I want to examine. Hence languages of this level will merely be objects of my mathematical theory. For instance, one may consider language of set theory and then state in it all *Mathematical Theorems*, i.e. those formulae of language of set theory that can be proved from ZFC. For instance consider the middle value theorem, it can be stated in $\mathcal{L}_\epsilon$ and has a proof in ZFC, hence write $ZFC \vdash_c MVT$. My theory clearly does not solely allows the definition of $\mathcal{L}_\epsilon$ but instead but instead of any other formal language one can think of, like classical logic with no predicate or terms $\mathcal{L}_c$, or language of groups $\mathcal{L}_{gr}$. Since often I use symbols of language of set theory, I denote those as: $\langle \bar{\exists}, \bar{\wedge}, \bar{\neg}, \bar{\in} \rangle$ when they are meant to be on this first level; in many cases, i.e. when it is explicitly obvious from the contexts or if they are followed or preceded by a $\vdash$ sign, I will use the standard notation (that is without the " $\bar{}$ ").

II. Meta Language

The study of mathematical structures and their languages is carried out in *Mathematical Logic* and, more broadly, in metamathematics. For instance the proposition $ZFC \vdash_c MVT$ is an atomic formula of the metalanguage, another example is the Löwenheim Skolem Theorem. At this secondary level, I will employ the conventional language of set theory, typically denoted as $\langle \exists, \wedge, \neg, \in \rangle$. To construct languages, however, it is useful to have some primitive notions; these serve as the fundamental components of the object languages, namely the symbols.

Primitive Notions The concept of a primitive notion was famously introduced by Tarski in [Tarski 1946]:

> When we set out to construct a given discipline, we distinguish, first of all, a certain small group of expressions of this discipline that seem to us to be immediately understandable; the expressions in this group we call *primitive terms* or *undefined terms*, and we employ them without explaining their meanings. At the same time we adopt the principle: not to employ any of the other expressions of the discipline under consideration, unless its meaning has first been determined with the help of primitive terms and of such expressions of the discipline whose meanings have been explained previously.

In a more formal tone, I introduce to the language of set theory some new notions that do not require further definition. These notions will be treated merely as symbols and there is very little information I need to assume about them. First, I distinguish these notions into three groups: Terms 1.1, Terms Relations 1.2, and Formulae Relations 1.3. Secondly, I need to assume some grammatical rules for these, i.e., how I am allowed to write these symbols. Those rules will be merely written in natural language, since they, as Tarski pointed out rely on being "*immediately understandable*".

Furthermore, I need to clarify how the relations present in the language of set theory behave with respect to symbols. For a set x and symbols s, s', the relations $x \in s$ or $s \in s'$ are not defined, i.e. grammatically incorrect. However, the relation $s \in x$ is true if and only if s is an element of x; indeed, one can construct sets of symbols as one would expect. Also, $x = s$ is trivially false, while $s = s'$ is true if and only if s is the same symbol as s'. One shall take this paragraphs as the usual Tarskian

[7] I discuss a related foundational issue on the role of the metalanguage in Metalinguistic Foundational Issue in Mathematics*

(see [Tarski 1933]) to define truth of relations in the II. meta language thanks to expressions in the III. further meta level.

A further point that needs clarification is how one should make the correct choice for primitive notions. It will be clear from the next section that there are many different sets of primitive notions that one could pick, which would still define equivalent languages (the notion of equivalent languages will then be formally clarified in the second section; for now, it can be pictured as languages that can express the same concepts). Therefore, from a very practical perspective, guidelines should be given on which primitive notions to choose in order to make the most efficient choice.

The first general priority in this respect is to use the minimal amount of symbols possible. It will be clear in fact (see Complete Description 1.7 and the Remark on Complete Description of Relations 1.9) that many symbols can be defined from within the language and it is therefore not necessary to take them as primitives.

There are circumstances, see Appendix II. Identity as a Logical Relation*, in which one has reasons to admit symbols that could be defined from within the language. For instance, a reason is to allow a logic to be a Classical Frame 1.5, as many theorems and definitions can apply only to those (see Problem on the Necessity of Classical Frame 1.10). Anyway, it must remain clear that these criteria are pretheoretic and hence informal, and under a purely formal perspective, completely irrelevant (as I mentioned before, the reader may come back to this paragraph after reading the second section and formally state the claim).

In the Literature. In [Shapiro &Kissel 2024], the author recognises the same necessity to introduce primitive symbols, which he refers to as *Building Blocks*. He offers a categorisation similar to the one I present below, and I will maintain a comparison with this paper throughout the following passages. Informally, he claims these building blocks to have some denoting relation to objects or truths in the world and have some counterparts in natural language: "Some aspects of the formal languages correspond to, or have counterparts in, natural languages like English. Technically, this "counterpart relation" is not part of the formal development" (2. Language) this informal details are of no interest for the scope of my discussion and are therefore fully ignored.

A slightly different approach in mathematical logic is to use purely $\mathcal{L}_\epsilon$ without any primitive notions. Instead of defining the set of terms to contain only primitive symbols, this approach substitutes them with other sets. From a formal perspective, this approach is entirely equivalent to the previous one (such equivalence can be proven using the tools presented in the second section). From an informal perspective, the second approach has the advantage of using only the language of set theory, without introducing any primitive notions. However, it requires taking some sets that will not function as ordinary sets but will serve as Encodings (see Definitions 3.1 & 3.6) of the symbols considered as primitives in the first approach. Because of this, I believe that the first approach maintains clarity even though it employs a more "expensive" language, i.e., one that assumes primitive notions. Nonetheless, it is important to note that everything discussed here can be easily Translated (see Definition*) into the language of set theory. A more precise comparison between these two choice is in the Appendix IV. On the Naturality of Primitive Terms*.

Primitive Notion 1.1 (Set of Terms)**.** Let $\Lambda = \{\Lambda_{\mathcal{L}} : \Lambda_{\mathcal{L}} \text{ a set of terms}\}$

Example & Remark. The set of terms always comprises the set of primitive constants and the set of variables within a language. First, I focus on the set of variables. Given the essence of what variables are, the set of variables is determined solely by its cardinality, which must be chosen for each language. Hence, each language $\mathcal{L}$ has a constant $\check{\lambda}^1_{\mathcal{L}}$ that specifies the number of allowed variables, which also influences the maximal arity of terms (see Relations 1.2).

Other than variables, one can find primitive constants in $\Lambda_{\mathcal{L}}$, those are terms that one requires to have as primitive in a language in order to get all wished formulae. An example of this is the 0, i.e. the neutral element, in the language of groups; in order to get all usual definitions in language of groups, if one has only the relation $+$, one needs to add a symbol for the neutral element $\dot{0}$. Since Λ is primitive in this metalinguistic level, it is free to contain any sort of symbols, as the reader knows from Primitive Notions 1.2, there are choices of Λ that can be more or less efficient.

It is also important to note that the set Λ comprises all sets of terms for each specific language being considered at the first level. Consequently, the set Λ varies significantly depending on this choice. Generally, it can be taken as the vocabulary of most languages used in mathematics. For the purposes

of this text, I consider the terms typically used in classical logic $\Lambda_c := \{\dot{x}_1, ..., \dot{x}_{\check{\lambda}_c^1}\}$, in the language of groups $\Lambda_{gr} := \{\dot{0}, \dot{x}_1, ..., \dot{x}_{\check{\lambda}_{gr}^1}\}$, and in the language of set theory $\Lambda_\epsilon := \{\dot{x}_1, ..., \dot{x}_{\check{\lambda}_\epsilon^1}\}$. For the scope of this discussion one can therefore instantiate in the example $\Lambda = \{\Lambda_c, \Lambda_\epsilon, \Lambda_{gr}\}$, though these definitions allow for general approaches.

To enable an efficient selection of the set of terms, it is best to avoid confusing primitive constants with those that can be defined later. A complete understanding of this distinction will be possible after the crucial definition of Complete Description 1.7 and will also become clear in the distinction between Atomic Formulae 1.12 and Atomic Term Formulae 1.4. An example of an inefficient selection would be claiming $\{\dot{\emptyset}, \dot{\aleph}_0, \dot{x}_1, ..., \dot{x}_{\check{\lambda}_\epsilon^1}\}$ as Λ_ϵ, since $\emptyset$ and $\aleph_0$ can be defined once the language is established. Therefore, these sets do not need to be considered primitive constants.

In the Literature. In [Shapiro &Kissel 2024], the author introduces among the first *Building Blocks* of a language the *singular terms*, divided into *individual constants* and *individual variables*. The main difference with my approach is that what I mean by classical logic in this text is only provided with variables and no individual constants. I do this in order to have a single language that I call classical logic, which does not prevent me from discussing other formal languages within a Classical Frame 1.5. Therefore, what I will define as classical logic $\mathfrak{L}_c$ is a special case of what Shapiro denotes as classical logic.

Importantly I make no distinction between variables and singular terms. I avoid this because, for the scope of this discussion, it is not necessary for me to keep track of how many free variables are present in a formula. This will be clearer once I get to the Primitive Notion of Formulae Relations 1.3 and I discuss this matter more in depth in the Appendix I. Bounded and Free Variables*.

I turn now to the terms relation, those symbols that will make a true or false formula once applied to Terms 1.1.

Primitive Notion 1.2 (Terms Relations)**.** Let $\Sigma^\Lambda = \{\Sigma^\Lambda_\mathcal{L} : \Sigma^\Lambda_\mathcal{L}$ a set of relations between terms, whose arity is determined by a function $\alpha\}$

Example. In classical logic there is only one such relation, namely "=", in language of set theory there also is "$\in$" and in language of groups there only are "=" and "+". Hence, for the scope of this discussion, we can take $\Sigma^\Lambda = \{\{\dot{=}\}, \{\dot{=}, \dot{\in}\}, \{\dot{=}, \dot{+}\}\}$, though the definitions apply for more general results.
Such relations are defined also thanks to a so-called Arity Function, i.e. a function α that takes these symbols to the ordinal number of places they have, for instance $\alpha(\dot{=}) = 2$.

In the Literature. The author of [Shapiro &Kissel 2024] similarly defines the n-place *predicate letters*, where each is given with the arity in a superscript. In this respect, the only difference is that I denote arity through the function α instead of writing it directly on the predicate. Shapiro also defines the *non-logical terminology* as "consisting of its individual constants and predicate letters", as these are not necessarily part of the language but depend on the application or intent for classical logic in a given context. For this reason, as I explained in the previous note of this kind, I leave such symbols out of the pure definition of classical logic $\mathfrak{L}_c$. There is one exception to this rule, namely the symbol "=", which Shapiro calls non-logical but which I accept in the definition of classical logic. I discuss this further in Appendix II. Identity as a Logical Relation*.

The last category of these objects are the formulae relations, i.e. those that we are used to be applied on formulae.

Primitive Notion 1.3 (Formulae Relations)**.** Let $\Sigma^\Lambda = \{\Sigma^\Phi_\mathcal{L} : \Sigma^\Phi_\mathcal{L}$ a set of relations between formulae, whose arity is determined by a function $\alpha\}$

Example & Remark. In classical logic, such relations include "$\wedge$", "$\neg$", and a quantifier like "$\exists$". It is evident that $\Sigma^\Phi_{\mathcal{L}_\epsilon} = \Sigma^\Phi_{\mathcal{L}_c} = \Sigma^\Phi_{\mathcal{L}_{gr}}$. Therefore, for the scope of this discussion, one can take $\Sigma^\Phi = \{\{\dot{\vdash}\}, \{\dot{\wedge}, \dot{\neg}, \dot{\exists}\}\}$.

Since these sets are purely at the syntax level, no inferences are considered yet; thus, the observation I just made is relatively weak. Indeed, if one were to consider another language with the same number

of primitive relations between formulae with the same arity, it would also be equal to these (apart from the use of different symbols). A clearer understanding of how these symbols acquire meaning is provided by the introduction of Inference Rules* and the Syntactic Consequence Relation*. One shall also notice that the only constant symbol I take among the Formulae Relations is $\dot{\vdash}$, this will be relevant for the definition of Inference Rules*.

Characterising these symbols through α, as I did for Term Relations 1.2, is no longer as straightforward. This is because the grammatical role of these relations also changes depending on whether they fill a free variable slot or not; those that do so are called quantifiers. In addition to defining the same α function for them, I also introduce α^{Φ}, which indicates the number of slots filled by the relation. For instance, $\alpha(\dot{\exists}) = 1$, $\alpha(\dot{\neg}) = 1$, $\alpha^{\Phi}(\dot{\exists}) = 1$, and $\alpha^{\Phi}(\neg) = 0$. I will though not practically use α^{Φ} because of the reasons mentioned in Appendix I. Bounded and Free Variables*.

In the Literature. Formulae relations as I present them are introduced in [Shapiro &Kissel 2024] in section 2.3, after the atomic formulas. The only difference between those that I list here for classical logic and Shapiro's is the use of brackets. Shapiro considers "(" and ")" as additional symbols in the lexicon. I avoid this because it introduces unnecessary complications. In the Appendix III. Exclusion of Brackets from Formulae Relations*, I provide a more complete justification for this approach.

Now that these three categories of primitive elements have been exposed, I can begin stating the proper definitions and, in particular, give the construction of a formal language.

Definition 1.4 (Atomic Term Formulae)**.** $\mathcal{A}^{\Lambda}_{\mathcal{L}} := \{\circ(\lambda_1, ..., \lambda_{\alpha(\dot{\circ})}) : \dot{\circ} \in \Sigma^{\Lambda}_{\mathcal{L}} \wedge \dot{\lambda}_1, ..., \dot{\lambda}_n \in \dot{\Lambda}\}$

Remark. In order to define atomic formulae, it is sufficient to take each combination of Terms Relations 1.2 with Terms 1.1. In this set we also find atomic formulae with free variables since variables are in the Set of Terms 1.1; those can be excluded or separately defined without much effort (see Appendix I. Bounded and Free Variables*). Note that those are only those formulae made by Terms 1.1 and not all Atomic Formulae 1.12, for instance $(x \in \emptyset) \in \mathcal{A}_{\epsilon,ZFC}$ (the Atomic Formulae 1.12) but $(x \in \emptyset) \notin \mathcal{A}^{\Lambda}_{\epsilon}$ since $\emptyset \notin \Lambda_{\epsilon}$, this is because $\emptyset$ is an object that can be defined through a Complete Description 1.7 but is not necessary to define $\mathcal{L}_{\epsilon}$. This difference will become clearer after the definition of Atomic Formulae 1.12.

Note that, according to the rules of notation, the dot above p can be omitted since it is now understood to be a well-formed formula. However, under a purely formal conception, $p = \dot{p}$, as every element in the metalanguage can only be sets along with primitive elements, i.e., symbols. Nevertheless, notation should enable the reader to distinguish when p is considered as a string or when it is given a grammatical or even inferential role; see Section 1.1, *Three Levels of Syntax* and in particular Justification to the Dot 1.1.

In the Literature. The definition of atomic formulas given in [Shapiro &Kissel 2024] is not different, I just avoid making the distinction between open and closed formulae. I avoid making this distinction because I will not need it in the definition of Syntactic Consequence* and that is where the distinction is meant to most be useful, closed formulae are either true or false in structures but open formulae cannot be determinate. This choice of not distinguishing between open and closed formulae is more deeply exposed in the Appendix I. Bounded and Free Variables*.

I omit the standard inductive definition of a language, which is typically constructed from the atomic term formulas and the primitive elements of the language, in order to focus on some more novel concepts which will be central for the results of the second section.

Definition 1.5 (Classical Frame)**.** $\mathfrak{C} := (\mathcal{L}, \mathfrak{I})$ s.t. $\mathcal{L}_c \subseteq \mathcal{L} \wedge \mathfrak{I}_c \subseteq \mathfrak{I}$

Examples. As I exposed in the examples of Inference Rules*, clearly $\mathfrak{L}_{\epsilon}$ and $\mathfrak{L}_{gr}$ are classical frames (hence I could write them as $\mathfrak{C}_{\epsilon}$, $\mathfrak{C}_{gr}$ too).

Remark 1.6 (Non-classical Frames)*.* One should notice that not all inference rules and formulae of classical logic are always necessary. Currently, I leave open for future work the question of precisely which inference rules and formulae are required.

The reader interested in this remark may check how the following definitions and results involving classical frames change for intuitionistic frames.

In the Literature. The concept of classical frames closely resembles classical logic as presented by Shapiro [Shapiro &Kissel 2024]. However, while my definition of Logic* is stricter compared to Shapiro's classical logic, my definition of Classical Frame is slightly more permissive. Specifically, it allows for changes in the Formulae Relations 1.3, which is not permitted in Shapiro's classical logic. Shapiro restricts additions to predicates and singular terms only, whereas the relations on formulae are meant to remain fixed as presented in section 3. *Deduction*. Additionally, I allow for the inclusion of more inferential rules, whereas Shapiro already provides an explicit set of rules.

This notion allows for the following crucial definition.

Definition 1.7 (Complete Description)**.** $\varphi^k_{\mathfrak{C},\mathcal{T}} \in \mathcal{L}$ is a description in $\mathfrak{C}$ with respect to $\mathcal{T} \subseteq \mathcal{L}$ if $\mathcal{T} \vdash_{\mathfrak{C}} \exists_x(\varphi^k_{\mathfrak{C},\mathcal{T}}(x) \wedge \forall_y \varphi^k_{\mathfrak{C},\mathcal{T}}(y) \rightarrow y = x)$.

Example: A simple example of such a Complete Description is that of the empty set. Consider, for instance, $\varphi^{\emptyset}_{\epsilon,ZFC}(x) := \forall_y \neg y \in x$. Also note that "being a Complete Description" is a property defined on formulae of the language. The role of k or $\emptyset$ in this example is solely notational (thus not a parameter). A more formal explanation of the role of k is provided in the Definition of Defined Object as Terms 1.11 and in subsequent remarks (I get then deeper into the matter in the Definition of Nominalistic Logical Universe 3.2 and mostly in the Remark on the Nominalistic and Antinominalistic Perspectives on the Universe3.3). In general terms, when referring to an object with a standard name, like the symbol $\dot{\emptyset}$ for the empty set, I denote it with a superscript in each of its complete descriptions.

It is also important to note that there are many Complete Descriptions for each such k. By definition, they are all equivalent (equivalence between formulae is determined, as usual, by $\leftrightarrow$). For example, $\varphi^{\emptyset,*}_{\epsilon,ZFC}(x) := \forall_y \neg y \in x$. Although they are not unique, this lack of uniqueness does not pose a notational problem, as their equivalence ensures that typically one shall consider only one Complete Formula at a time (i.e., they are unique up to $\leftrightarrow$).

Remark 1.8 (Dependence of Definability on a Theory). It is important to notice that φ depends not only on $\mathfrak{L}$ but also on a given theory $\mathcal{T}$. This is because the unique satisfaction requirement, i.e that $\forall_y \varphi^k_{\mathfrak{C},\mathcal{T}}(y) \rightarrow y = x$ may be true or false depending on the theory. For instance, begin by considering $\varphi^{\aleph_0}_{\epsilon,ZFC}(x) := x = |\omega|$ for the usual definition of ω (as a Complete Description) and of the cardinality function $|\cdot|$ (as a Complete Description of Relations 1.9). Then consider the description of $\aleph_1$ in ZFC as the smallest cardinal strictly bigger than $\aleph_0$, in symbol: $\varphi^{\aleph_1}_{\epsilon,ZFC} := \forall_{k>\aleph_0} x \leq k$ and the description of $2^{\aleph_0}$, intuitively $\varphi^{2^{\aleph_0}}_{\epsilon,ZFC} := x = 2^{\aleph_0}$. Note that, if one assumes the Continuum Hypothesis in the form $CH := 2^{\aleph_0} = \aleph_1$, then $\varphi^{\aleph_1}_{\epsilon,ZFC+CH} := \varphi^{\aleph_1}_{\epsilon,ZFC} \wedge \varphi^{2^{\aleph_0}}_{\epsilon,ZFC}$ since, as assumed in the Continuum Hypothesis, $2^{\aleph_0} = \aleph_1$ this formula is uniquely satisfied but $\varphi^{\aleph_1}_{\epsilon,ZFC+CH}$ is no description in ZFC because provably satisfied by no object. One can create similar tricks using Forcing (with the Collapse technique in particular) to have a definition that in some other theory is satisfied by more than one object.

Remark 1.9 (Complete Description of Relations). In the definition of Complete Description 1.7 I give an account of how one is to define an object in the language. On the other hand, one would rightfully claim, that one should be able to do the same for relations, both Terms Relations 1.2 and Formulae Relations 1.3. On the other hand, the definition of relations does not require a comparable complexity as the one for objects. A proper account on how to define an object requires the definition of Complete Description 1.7, of Defined Object as a Term 1.11. On the other hand, a relation can be defined by checking material equivalence, for instance a relation $\sim$ can be defined as a shorthand for any other sequence of relations with the same arity (on Terms 1.2 or on Formulae 1.3 or, potentially, some hybrid form). I discuss this for the case of identity in $\mathcal{L}_\epsilon$ in the Appendix II. Identity as a Logical Relation*.

Problem 1.10 (Necessity of Classical Frame). The notation $\mathcal{T} \vdash_{\mathfrak{L}} \varphi^k_{\mathfrak{L},\mathcal{T}}(k) \wedge \forall_y \varphi^k_{\mathfrak{L},\mathcal{T}}(y) \rightarrow y = k$ requires $\mathfrak{L}$ to be a Classical Frame, denoted as $\mathfrak{C}$, in accordance with 1.5. This is because the formula "$\varphi^k_{\mathfrak{L},\mathcal{T}}(k) \wedge \forall_y \varphi^k_{\mathfrak{L},\mathcal{T}}(y) \rightarrow y = k$" must be in the language of the logic $\mathcal{L}$ and it uses all $\neg, \wedge$ and $\exists$. Furthermore, for the formula to be satisfied in the correct cases, *most* (see the Remark on Non-classical Frames 1.6) of the Inference Rules* in $\mathfrak{I}_c$ are necessary. One might discuss exactly which inference rules are required for this definition to properly make sense. In fact, one can envision a similar scenario for intuitionistic logic. As mentioned earlier, I will not explore this excursion in further detail.

A very important lemma on Complete Descriptions 1.7 which involves the use of Translations*, is the Nominalistic Universe Correspondence*.

The following definition allows the formulae to be stated in a manner familiar to every mathematician. For instance, a formula like $x_1 \notin \{\emptyset\}$ is not properly within the language of set theory if interpreted as a relation between terms since $\{\emptyset\}$ is not a term. However, one can define it as the only y s.t. $\forall_x(\forall_y\, y \notin x) \rightarrow x_1 \in \{x\}$, then $x_1 \notin \{\emptyset\}$ clearly is a formula in $\mathcal{L}_\epsilon$.

Definition 1.11 (Defined Objects as Terms)**.** For $\varphi^k_{\mathfrak{C},\mathcal{T}}$ a Complete Description 1.7 and $\psi \in \mathcal{L}$, $\mathcal{T} \vdash_{\mathfrak{C}} \psi(k) := \mathcal{T} \vdash_{\mathfrak{C}} \forall_x \varphi^k_{\mathfrak{C},\mathcal{T}}(x) \rightarrow \psi(x)$

Remark. Note that one can write $\mathcal{T} \vdash_{\mathfrak{C}} \psi(k)$ only once a certain Complete Description 1.7, $\varphi^k_{\mathfrak{C},\mathcal{T}}$, is given. This avoids the need to pick a single Complete Description among all equivalent ones. Furthermore, one cannot naturally range over defined objects. One can define a set of defined objects and range over those, but the set of all definable objects can be defined only metalinguistically (i.e., at the level I am currently working at, the II. level, see Section 1.2 Three Metalevels), and I have not given its definition yet (I will give a possible definition in Nominalistic Logical Universe 3.2). I explore this issue in more depth in the Remark on Nominalistic and Antinominalistic Perspectives on the Universe 3.3.

Remark. Also note that one cannot define the formula $\psi(k)$ independently from $\mathcal{T}$ and the inference rules in $\mathfrak{L}$. This is because the property of being a Complete Description 1.7 depends on these, and therefore writing $\psi(k)$ without an explicit reference to the theory and the inference rules would have no unique meaning.

It is now finally possible to reconstruct the familiar concept of atomic formulae, I define it such that $\emptyset \in \{\emptyset\}$, for instance, is an atomic formula in the language of set theory with respect to ZFC.

Definition 1.12 (Atomic Formulae)**.** For $\varphi^{k_1}_{\mathfrak{C},\mathcal{T}}, ..., \varphi^{k_n}_{\mathfrak{C},\mathcal{T}}$ Complete Descriptions, define $\mathcal{A}_{\mathfrak{C},\mathcal{T}} := \{p \in \mathcal{L} : \exists_{\dot\circ \in \Sigma^\Lambda_\mathcal{L}} \exists_{\dot\lambda_1,...,\dot\lambda_{\alpha(\dot\circ)} \in \Lambda_\mathcal{L} \cup \{k_1,...,k_n\}} \dot p = \dot\circ(\dot\lambda_1, ..., \dot\lambda_{\alpha(\dot\circ)})\}$.

Remark 1.13 (Confusing Appearance of Atomic Formulae). The common intuition of an atomic formula is fairly simple: one looks at a formula, and if it involves only one relation on terms, then it is considered atomic. For instance, when asked whether $\emptyset \in \{\emptyset\}$ is atomic, most people would immediately say yes. However, when presented with $\forall_x \forall_{x'}(\forall_z(\forall_y y \notin z) \rightarrow (\forall_y y \in x \rightarrow y = z)) \wedge (\forall_y y \notin x') \rightarrow x' \in x$[8], few would think of it as atomic. However, by the Notation 1.11, which allows the use of defined objects as terms, I defined $\emptyset \in \{\emptyset\}$ using the second long formula. Thus, it shall be considered an atomic formula.

Remark 1.14 (The Role of the Nominalistic Logical Universe 3.2 in the Definition of Atomic Formulae 1.12). The definition of atomic formulae of a language cannot really be done independently from not only a theory (see the Remark on the Dependence of Definability on a Theory 1.8) but also of a previous set of Complete Descriptions $\varphi^{k_1}_{\mathfrak{C},\mathcal{T}}, \ldots, \varphi^{k_n}_{\mathfrak{C},\mathcal{T}}$. At a practical level, this is not an issue since, when one is asked whether a formula is or is not atomic, one already gives the definition of the symbols in it; for instance, one would never ask whether $\emptyset \in \{\emptyset\}$ is atomic without the previous definition of the empty set and the singleton. On the other hand, if given a language one wishes to define the set of atomic formulae without such dependence on a given set of Complete Descriptions, one shall proceed differently by defining a Nominalistic Logical Universe 3.2, i.e., a set of all definable objects (or names of those, see Nominalistic and Antinominalistic Perspectives on the Universe 3.3). Therefore for $\mathcal{U}_{\mathfrak{L},\mathcal{T}}$, such a definition would look like this:

$$\mathcal{A}^*_{\mathfrak{C},\mathcal{T}} := \{p \in \mathcal{L} : \exists_{\dot\circ \in \Sigma^\Lambda_\mathcal{L}} \exists_{\dot\lambda_1,...,\dot\lambda_{\alpha(\dot\circ)} \in \Lambda_\mathcal{L} \cup \mathcal{U}_{\mathfrak{L},\mathcal{T}}} \dot p = \dot\circ(\dot\lambda_1, \ldots, \dot\lambda_{\alpha(\dot\circ)})\}$$

Since the definition of Nominalistic Logical Universe 3.2 comes with a few philosophical and technical complications, I prefer to leave it for the following section.

[8]Consider the following Complete Descriptions 1.7: (i) $\varphi^{\emptyset}_{\epsilon,ZFC}(x) := \forall_y y \notin x$, (ii) $\varphi^{\{z\}}_{\epsilon,ZFC}(x) := \forall_y y \in x \rightarrow y = z$, (iii) $\varphi^{\{\emptyset\}}_{\epsilon,ZFC}(x) := \forall_z \varphi^{\emptyset}_{\epsilon,ZFC}(z) \rightarrow \varphi^{\{z\}}_{\epsilon,ZFC}(x) = \forall_z(\forall_y y \notin z) \rightarrow (\forall_y y \in x \rightarrow y = z)$. Hence the formula can be restated as: $\forall_x \forall_{x'} \varphi^{\{\emptyset\}}_{\epsilon,ZFC}(x) \wedge \varphi^{\emptyset}_{\epsilon,ZFC}(x') \rightarrow x' \in x$.

Remark. It may surprise the reader to realise that the definition of the set of Atomic Formulae $\mathcal{A}_{\mathfrak{L},\mathcal{T}}$ actually depends on a theory. This is because a Complete Description 1.7 also depends on a theory, as noted in Remark 1.8 regarding the Dependence of Definability on a Theory.

One shall here also note that writing "$\exists_{\dot\lambda_1,...,\dot\lambda_{\alpha(\dot\circ)}\in\Lambda_{\mathcal{L}}\cup\{k_1,...,k_n\}}\dot p = \dot\circ(\dot\lambda_1,...,\dot\lambda_{\alpha(\dot\circ)})$" uses the Defined Object as a Term 1.11, since it is an open formula with only the ks free, whose Complete Descriptions 1.7 have been given. This is more evident when one notices that "$\exists_{\dot\lambda_1,...,\dot\lambda_{\alpha(\dot\circ)}\in\Lambda_{\mathcal{L}}\cup\{k_1,...,k_n\}}\dot p = \dot\circ(\dot\lambda_1,...,\dot\lambda_{\alpha(\dot\circ)})$" formally stands for "$\exists_{\dot\lambda_1,...,\dot\lambda_{\alpha(\dot\circ)}}\dot\lambda_1,...,\dot\lambda_{\alpha(\dot\circ)} \in \Lambda_{\mathcal{L}}\cup\{k_1,...,k_n\} \wedge \dot p = \dot\circ(\dot\lambda_1,...,\dot\lambda_{\alpha(\dot\circ)})$".

With this last definition for this level, I gave the formal context necessary to perform all of the following proofs and understand the definitions yet to come.

III. Further Meta Level

Since the II. level constitutes my object language, [Tarski 1933] points out the necessity of an additional metalinguistic level for establishing a notion of truth. Moreover, numerous texts in this work are expressed in this language, which I define as being natural language supplemented by precise symbols detailed below.

The Syntactic Overwrite There is an important symbol used at this level that I use when stating definitions, namely ":=". This symbol is used as a syntactical substitution of the term on the left with the term on the right. Therefore, in a definition like 1.12, the symbol "$\mathcal{A}_{\mathfrak{C},\mathcal{T}}$" shall always be understood as its definition, i.e., the formula on the II. level on the right-hand side. Sometimes, the left-hand side of ":=" is a symbol already defined on the II. level (or the usual object level when one is doing mathematics instead of metamathematics); in such cases, its function remains unchanged — it substitutes the term on the left with the term on the right. An example of this occurs in usual mathematics when defining for groups $\mathcal{G} := (G, +_{\mathcal{G}})$ and $\mathcal{H} := (H, +_{\mathcal{H}})$ and their subset relation $\mathcal{G} \subseteq \mathcal{H} := G \subseteq H$; here, $\mathcal{G} \subseteq \mathcal{H}$ is already a well-formed formula that is overwritten.

The Syntactic Metaconsequence Since metamathematics uses the usual metalanguage as its object language, there is a need to define truth at this III. level. Following [Tarski 1933], this is established by the syntactic metaconsequence relation, denoted as "$\Vdash$". It is defined similarly to "$\vdash$", but it asserts the existence of a proof in the III. level rather than the II. level. Therefore, when a theory of languages is given as a set of axioms belonging to the II. level, the symbol "$\Vdash$" signifies the existence of a proof from those axioms to some formula φ also at the II. level. Even though I will rarely mention this symbol in the upcoming text, it is still important to be aware of its existence and linguistic role.

2 On Translations

I discuss here only one particular notion of translation, even though I have defined and compared several different types in the original work. This is the one that is most widely applicable among those I examined, and I choose it now to move rapidly to the results.

Definition 2.1 (Partial Strong Translation)**.** $\hat{\underline{\tau}}_x : \mathfrak{L}_1 \to \mathfrak{L}_2$ is a partial translation with respect to $\mathcal{T}_x$ if $\forall_{\varphi\in\mathcal{L}_1}\mathcal{T}_x \vdash_1 \varphi \leftrightarrow \hat{\underline{\tau}}_x(\mathcal{T}_x) \vdash_2 \hat{\underline{\tau}}_x(\varphi)$

Remark 2.2 (Use-cases of Partial Translations)*.* There are circumstances in which one can claim the existence of a Translation* or a Partial Translation*, and others where the only *meaningful* requirement is a Partial Translation*. For instance, consider the claim of a Strong Partial Translation 2.1 $\underline{\tau}_{gr} : \mathfrak{L}_{gr} \to \mathfrak{L}_\epsilon$, as required in Example*. We know that certain symbols in $\mathcal{L}_{gr}$, such as $\dot+$ and $\dot 0$, have an inferential role that is completely undetermined by $\mathfrak{I}_{gr}$ (since $\mathfrak{I}_{gr} = \mathfrak{I}_c$). Therefore, given that the 2.1 aims to preserve the inferential structure, the only meaningful result must involve the theory that provides an inferential structure to those symbols, which is $\mathcal{T}_{gr}$.

In general, this applies to all Logics* that do not determine the meaning of their symbols. Specifically, when there are symbols that do not appear at all in the Inference Rules*, one should only consider

the Partial Translations* that take the relevant theory into account. There are formal methods to define when a Logic* completely determines its symbols through the Inference Rules*, specifically when each symbol is assigned both an Introduction and an Elimination Rule 1. However, these notions are beyond the scope of this section and will be discussed in more detail later.

2.1 Tarskian Translations & Basic Interpretability Logic

The different notions of translations discussed above resemble a straightforward intuition: that a theory can be *interpreted* into another formal language without significant difficulties, provided the right conditions. In a manner akin to my definition of Partial Strong Translation 2.1, Tarski offered an account of this very concept:

> In [Tarski, Mostovski, &Robinson 1953] the notion of "relative" interpretability between theories was introduced. Intuitively, "T interprets S" means that the language of S can be translated into the language of T in such a way that T proves the translation of every theorem of S. [Dzhaparidze 1993]

The core aspect of a translation function seems, then, to be exactly this sort of "*linearity*" on formulae of the form $T \vdash \varphi$. In fact, one can say that such mappings, given the linearity, preserve the structure under $\vdash$. Therefore, giving much importance to translations defined through such linearity is giving importance to the structure that $\vdash$ creates on theories and formulae. In fact, it turns out to be particularly clarificatory to compare such a structure with more familiar structures in mathematics, which also have analogous morphisms.

2.1.1 Translation as a Morphism

In group theory, one of the most basic concepts introduced is that of a homomorphism, i.e., for groups $(G, +_g)$ and $(H, +_h)$, a function $f : G \to H$, such that $\forall_{g_1,g_2 \in G} f(g_1 +_g g_2) = f(g_1) +_h f(g_2)$. A valid question, probably also the one that Tarski posed himself before designing his final notion of translation, is which of the four versions of morphisms described above this corresponds to. Given the commutativity of the identity, one might argue that this notion of morphism should be *strong*, i.e., involving a double conditional in both directions.

This comparison, despite being didactically explanatory, is not of much help, since clearly there can be very different notions of morphisms between those in group theory and those in structures defined by syntactical inference.

2.1.2 The Tarskian Choice

Among the four possibilities presented above, Tarski must have made a choice when defining his notion of translation and, as I will present in later sections, there are specific criteria for choosing one of these four definitions.

First, one must decide whether reference to a single theory is necessary or not, i.e., distinguish between *partial* and *non-partial* translations. Since, for the vast majority of the languages used in mathematical practice, the inferential rules are not sufficient to define the objects on which the language is supposed to operate, it is more practical to add a reference to a specific theory. For instance, consider a language like $\mathfrak{L}_\epsilon$; since "$\in$" does not appear in any inference rule, it does not make sense to consider every theory expressible in that language. Instead, one should restrict the research to ZFC and, perhaps, its extensions. For some other logics, though, like $\mathfrak{L}_c$, every symbol has an introduction and an elimination rule among the inference rules, and one can therefore make sense of considering every theory expressible within that language.

Secondly, one must decide whether the linearity should be bidirectional or unidirectional. Specifically, this involves choosing between having the formula "$\tau(T \vdash_1 \varphi) \to \tau(T) \vdash_2 \tau(\varphi)$" or "$\tau(T \vdash_1 \varphi) \leftrightarrow \tau(T) \vdash_2 \tau(\varphi)$" in the definition of the translation. As previously discussed with respect to homomorphisms on groups, one might prefer the latter, as it resembles the commutativity associated with "$=$". On the other hand, to maintain the highest possible degree of generality, it is reasonable to initially define the translation in its most general form, which is "$\to$".

These reasons should drive the reader to the conclusion that the Partial Strong Translation 2.1 is the definition we should take most into consideration, and it should then not surprise that this is exactly the definition that Tarski takes into consideration for an *Interpretation of a theory into another*.

2.1.3 Basic Interpretability Logic

After Tarski, the literature took a precise path that soon led to the development of the so-called Basic Interpretability Logic, along with many other more or less complex versions. These developments took a turn in two particular directions that lost part of the generality initially intended. To begin with, before focusing on Basic Interpretability Logic, which concerns interpretations from and into different formal languages, I will consider the logic regarding only the translation into the language of Peano Arithmetic, namely ILM. Consider, for instance:

> [...] ILM is the complete logic of interpretability over PA (Peano arithmetic). That is, ILM yields exactly the schemata of PA-provable arithmetical sentences, if we understand $A \rhd B$ as a formalization of the assertion "$PA + A$ interprets $PA + B$". [Dzhaparidze 1993]

Papers in this area mostly focused on the analysis of interpretability not in its most general form, but solely concerning Peano Arithmetic, with the aim of understanding all fields that could be in some way reduced to arithmetic. In order not to narrow the scope of this work, I decided to keep a more general approach and continue considering translations between general formal languages. In doing so, I encountered a collection of papers, closely related to the former, which gives rise to the Basic Interpretability Logic. As stated in [Visser 1988]:

> Relative interpretability is a notion which is much studied, so it seems natural to ask about the "Modal Propositional Principles" valid for this notion. A characterization of these principles will tell us precisely what one can do with self-referential arguments formulated in this simple language. A spin-off of such a characterization will be that we have a powerful machinery to construct arithmetical examples. A further point is this: one could consider the modal language involving interpretability simply as a more refined and richer way to discuss formal provability: surely the choices of what are going to be the natural numbers of a language etc. are rather arbitrary, isn't it more natural to ask e.g. whether a sentence expressing the consistency of T is interpretable in T than whether such a sentence is provable in T? In the language we consider, the second incompleteness theorem can be expressed in its "interpretability" form (this is what we call: Feferman's Principle). Moreover, a wider range of metamathematical experience can be represented: e.g., a form of the Model Existence Lemma can be formulated.

This branch of studies, here presented by Visser in its applications and goals, is admittedly different from the one originally designed by Tarski. In fact, it studies the modal logic representing results on the structure of a multitude of translations (or interpretations) from one extension of a theory to another and does not focus to the most general but *single* translation function between two given formal languages. The scope of my work is in fact, given two formal languages, to study the translations between them, not in their structure but by themselves, trying to provide concrete examples of such functions or proofs of their non-existence.

3 Translations in Foundational Praxis

I now introduce a practical example that demonstrates how to take an encoding, in one of the two variants presented below, and algorithmically define a translation function from it. This is a generally applicable process for finding a translation between formal languages. This method serves as one of the foundational building blocks in the theory of translations, as it is the first approach to consider when defining a translation. More complex alternatives are then explored in the original work and in the current projects I am working on.

Definition 3.1 (Relational Encoding Function)**.** $\varepsilon_{x,y} : \Lambda_1 \cup \Sigma_1^\Lambda \to \Lambda_2 \cup \Sigma_2^\Lambda$ is an encoding function of $\mathcal{T}_x \subseteq \mathcal{L}_1$ to $\mathcal{T}_y \subseteq \mathcal{L}_2$ if (i) $\varepsilon_{x,y}|_{\Lambda_1} \subseteq \Lambda_2$, (ii) $\forall_{\circ_n \in \Sigma_1^\Lambda} \forall_{\lambda_1,...,\lambda_n \in \Lambda_1} (\mathcal{T}_x \vdash_1 \circ_n(\lambda_1, ..., \lambda_n) \leftrightarrow \mathcal{T}_y \vdash_2 \varepsilon_{x,y}(\circ_n)(\varepsilon_{x,y}(\lambda_1), ..., \varepsilon_{x,y}(\lambda_n))$

Remark. While the general notion of encoding may cover many different sort of functions, those captured in this definition map Terms 1.1 to Terms and Terms Relations 1.2 to Terms Relations (hence the name Relational Encoding Function); this means that the encoding function creates a correspondence from $\Lambda_1 \cup \Sigma_1^\Lambda$ to $\Lambda_2 \cup \Sigma_2^\Lambda$.

It is important to note that the function $\varepsilon_{x,y}$ depends on two theories, which is necessary to state the second requirement: $\forall_{\circ_n \in \Sigma_1^\Lambda} \forall_{\lambda_1,...,\lambda_n \in \Lambda_1} (\mathcal{T}_x \vdash_1 \circ_n(\lambda_1,...,\lambda_n) \leftrightarrow \mathcal{T}_y \vdash_2 \varepsilon_{x,y}(\circ_n)(\varepsilon_{x,y}(\lambda_1),...,\varepsilon_{x,y}(\lambda_n))$. In a practical context, however, one usually begins with a theory $\mathcal{T}_x$ for which one wishes to construct the Partial Translation and does not generally have such a $\mathcal{T}_y$ already. It should be noted that, if one follows the steps presented in the proof method, $\mathcal{T}_y$ will become the translation of $\mathcal{T}_x$. In such cases, one should define $\varepsilon_{x,y}$ for some $\mathcal{T}_y$ (the proof of its existence is trivial in most cases) such that $\varepsilon_{x,y}$ is a Relational Encoding Function 3.1; then, a specific $\mathcal{T}_y$ will be constructed in the later stages of the proof.

One can also structure encoding functions such that Terms Relations 1.2 in $\Lambda_1 \cup \Sigma_1^\Lambda$ are mapped only to objects defined in $\mathcal{L}_2$ that can retain the information of how the relation behaves on terms. Given that, providing a fully general definition of this second version of encoding is complex and not necessary for the scope of this discussion, I define it only for cases where the target language is $\mathfrak{L}_\epsilon$ equipped with ZFC.

Before discussing this though, I need to remark that such a definition would require a mapping to *all objects defined in* $\mathcal{L}_2$, which is not a set I defined yet. In fact, I need a function of the form $\varepsilon_x : \Lambda_1 \cup \Sigma_1^\Lambda \to \Lambda_\epsilon \cup \mathcal{U}_{\epsilon,ZFC}$, where $\mathcal{U}_{\epsilon,ZFC}$ is the set of all objects (sets) definable (see Complete Description 1.7) in $\mathfrak{L}_\epsilon$ equipped with ZFC. In the first section, I avoided such a definition by only stating how to handle Defined Objects as Terms 1.11, now though, it is necessary to introduce it.

Here I introduce the Nominalistic Logical Universe as the set of equivalence classes on $\leftrightarrow$ of Complete Descriptions 1.7. Each of these equivalence classes contains all Complete Descriptions 1.7 that define a particular object k.

Definition 3.2 (Nominalistic Logical Universe)**.** For $\mathfrak{C} = (\mathcal{L}, \mathfrak{I})$ a Classical Frame 1.5, $\mathcal{T} \subseteq \mathcal{L}$ and the equivalence class $[\varphi]^{\leftrightarrow}_{\mathfrak{C},\mathcal{T}} := \{\psi : \mathcal{T} \vdash_{\mathfrak{C}} \forall_y \psi(y) \leftrightarrow \varphi(y)\}$, define $\mathcal{U}_{\mathfrak{C},\mathcal{T}} := \{[\varphi^k_{\mathfrak{C},\mathcal{T}}]^{\leftrightarrow}_{\mathfrak{C},\mathcal{T}} : \mathcal{T} \vdash_{\mathfrak{C}} \exists_x(\varphi^k_{\mathfrak{C},\mathcal{T}}(x) \wedge \forall_y \varphi^k_{\mathfrak{C},\mathcal{T}}(y) \to y = x)\}$.

Remark 3.3 (Nominalistic and Antinominalistic Perspectives on the Universe). There are two perspectives one may have on the intuition of a logical universe: on one hand, one may have a platonic conception of it and picture each of the definable objects as entities in the language so that they can denote something, akin to the primitive notions in the Set of Terms 1.1. For instance, $\dot{0}$ in $\mathcal{L}_{gr}$ may informally be taken as a name denoting some object in the world, similar to how $\emptyset$ can be viewed in the same manner. However, this interpretation is not well represented by the construction I just presented in the Definition of Nominalistic Logical Universe 3.2; this is because each element in $\mathcal{U}_{\mathfrak{C},\mathcal{T}}$ is not a new element whose denotation is yet to be defined, but rather a collection of equivalent *names* (Complete Descriptions 1.7) that are constructed from within the language. To define a Language* whose elements in the Nominalistic Logical Universe 3.2 can denote new objects, one shall instead canonically define a new Language* $\mathcal{L}^*_{\mathfrak{C},\mathcal{T}}$ whose Set of Terms 1.1 now includes all those defined objects collected in the logical universe. This involves defining a Defining Quadruple* as $(\Lambda_{\mathcal{L}} \cup \mathcal{U}_{\mathfrak{C},\mathcal{T}}, \Sigma^\Lambda_{\mathcal{L}}, \Sigma_{\mathcal{L}^\Phi, \check{\lambda}^2_{\mathcal{L}}})$. In this way, $\mathcal{L}^*_{\mathfrak{C},\mathcal{T}}$ recognises elements in $\mathcal{U}_{\mathfrak{C},\mathcal{T}}$ as primitive notions whose denotation is yet to be defined. I discuss this issue formally in the Formal Appendix of Hypotheses IV. Antinominalistic Universe 1

There are two simple remarks necessary to fully comprehend this construction: first, that $\mathcal{L}^*_{\mathfrak{C},\mathcal{T}} = (\mathcal{L}^*_{\mathfrak{C},\mathcal{T}})^*_{\mathfrak{C},\mathcal{T}}$, meaning once the new defined objects have been added to the Set of Terms 1.1, there will be no further object that could not have been defined before but can be defined after this process; I give a formal statement of this in the Formal Appendix in the Hypothesis Exhaustion of the Antinominalistic Universe 1. Secondly, I remark that formally, this Antinominalistic approach or the Nominalistic one I propose in Nominalistic Logical Universe 3.2 are completely equivalent, in particular after the upcoming Definition of Universe Elements as Terms 3.4; this is also formally stated in the Formal Appendix, in Hypothesis Nominalistic and Antinominalistic Universe Equivalence 1.

Similarly to the definition of Complete Description 1.7, the Nominalistic Logical Universe 3.2 gives names to the defined object k but does not determine how it should be integrated in the well formed formulae of the respective language. I solved this issue for Complete Descriptions 1.7 by giving the Definition of Defined Objects as Terms 1.11, here I present its equivalent for the Nominalistic Logical Universe 3.2.

Definition 3.4 (Universe Elements as Terms)**.** For $\mathfrak{C} = (\mathcal{L}, \mathfrak{I})$ a Classical Frame, $\mathcal{T} \subseteq \mathcal{L}$, $[\varphi^k_{\mathfrak{C},\mathcal{T}}]^{\leftrightarrow}_{\mathfrak{C},\mathcal{T}} \in \mathcal{U}_{\mathfrak{C},\mathcal{T}}$ and $\psi(x) \in \mathcal{L}$, define $\mathcal{T} \vdash_{\mathfrak{C}} \psi([\varphi^k_{\mathfrak{C},\mathcal{T}}]^{\leftrightarrow}_{\mathfrak{C},\mathcal{T}}) := \mathcal{T} \vdash_{\mathfrak{C}} \psi(k)$

For this recall that "$\mathcal{T} \vdash_{\mathfrak{C}} \psi(k)$" has already been defined in the Definition of Defined Objects as Terms 1.11 as "$\mathcal{T} \vdash_{\mathfrak{C}} \forall_x \varphi^k_{\mathfrak{C},\mathcal{T}}(x) \to \psi(x)$".

Remark 3.5 (The Use of Metalinguistic Names). The attentive reader may note a potential difficulty in the given definition: the set $[\varphi^k_{\mathfrak{C},\mathcal{T}}]^{\leftrightarrow}_{\mathfrak{C},\mathcal{T}}$ does not belong to the I. level of the Object Language; instead, it is clearly defined to be at the II. level of the Metalanguage. Despite this, it is inserted in formulae of the form $\mathcal{T} \vdash_{\mathfrak{C}} \psi([\varphi^k_{\mathfrak{C},\mathcal{T}}]^{\leftrightarrow}_{\mathfrak{C},\mathcal{T}})$, which suggests its inclusion in the Object language.

Two remarks are necessary to satisfactorily address this concern: formally, this is not a problem; indeed, the Syntactic Overwrite 1.2 substitutes each occurrence of "$\mathcal{T} \vdash_{\mathfrak{C}} \psi([\varphi^k_{\mathfrak{C},\mathcal{T}}]^{\leftrightarrow}_{\mathfrak{C},\mathcal{T}})$" with the well-defined "$\mathcal{T} \vdash_{\mathfrak{C}} \psi(k)$" and then again, by Defined Objects as Terms 1.11, with "$\mathcal{T} \vdash_{\mathfrak{C}} \forall_x \varphi^k_{\mathfrak{C},\mathcal{T}}(x) \to \psi(x)$", which is a well defined formula.

Informally, the intuition behind the appearance of $[\varphi^k_{\mathfrak{C},\mathcal{T}}]^{\leftrightarrow}_{\mathfrak{C},\mathcal{T}}$ is analogous to that of k in the Definition of Defined Objects as Terms 1.11; these serve as *names* which may indeed originate from the metalanguage. What is crucial is that they are handled in a manner that formally respects the distinction between the metalevels.

Now that I have provided a construction for the Nominalistic Logical Universe 3.2, the Set Theoretic Encoding Function can formally map into all defined objects in the set-theoretic language with respect to ZFC, making formal sense of it thanks to the Definition of Universe Elements as Terms 3.4.

Definition 3.6 (Set Theoretic Encoding Function)**.** $\varepsilon_x : \Lambda_1 \cup \Sigma^{\Lambda}_1 \to \Lambda_\epsilon \cup \mathcal{U}_{\epsilon,ZFC}$ is a set theoretic encoding function of $\mathcal{T}_x$ if it holds that $\forall_{\dot{\circ}_n \in \Sigma^{\Lambda}_1} \forall_{\lambda_1,...,\lambda_n \in \Lambda_1} \mathcal{T}_x \vdash_1 \circ_n(\lambda_1, ..., \lambda_n) \leftrightarrow ZFC \vdash_\epsilon \{\varepsilon_x(\lambda_1), \{\varepsilon_x(\lambda_n)\}, ..., \{\{\{...\{\varepsilon_x(\lambda_n)\}...\}\}\} \bar{\in} \varepsilon_x(\dot{\circ}_n)$

Remark. A detail that may need clarification of this definition is that I construct the set of the form $\{\varepsilon_x(\lambda_1), \{\varepsilon_x(\lambda_n)\}, ..., \{\{\{...\{\varepsilon_x(\lambda_n)\}...\}\}\}$ in order to have an ordered set that allows for repetitions. This is necessary since the relation is sensitive to the order of the terms and permits their repetitions. Clearly there are many alternative and equivalent ways to define it, for practical reasons, I decided to pick one.

The ultimate goal of constructions like the Set Theoretic Encoding Function 3.6 is to construct a proper Translation*. To achieve this, I define a *canonical* process to encode such a function. By "canonical" I mean that while there are multiple equivalent approaches (for example, those generated from different methods of ordering a set), this is the version I focus on. First, I present a simpler version defined solely on the Atomic Term Formulae 1.4.

Definition 3.7 (Canonical Term-Atomic Partial Strong Translation from a Set Theoretic Encoding)**.** For $\mathcal{L}_1$ a Language*, $\varepsilon_x : \Lambda_1 \cup \Sigma^{\Lambda}_1 \to \Lambda_\epsilon \cup \mathcal{U}_{\epsilon,ZFC}$ a Set Theoretic Encoding 3.6, define $\tau^a_{\varepsilon_x} : \mathcal{A}^{\Lambda}_1 \to \mathcal{A}_{\epsilon,ZFC}, \circ_n(\lambda_1, ..., \lambda_n) \mapsto \{\varepsilon_x(\lambda_1), \{\varepsilon_x(\lambda_n)\}, ..., \{\{\{...\{\varepsilon_x(\lambda_n)\}...\}\}\} \bar{\in} \varepsilon_x(\circ_n)$ as a Canonical Term-Atomic Partial Strong Translation from ε_x.

Remark. It is here important to understand the reason why the domain of ε_x is the set of Atomic Term Formulae 1.4 in $\mathcal{L}_1$, i.e. $\mathcal{A}^{\Lambda}_1$, while its codomain consists of Atomic Formulae 1.12 in $\mathcal{L}_\epsilon$, denoted as $\mathcal{A}_{\epsilon,ZFC}$. In order to share this intuition, one shall first remember the difference between Atomic Term Formulae 1.4 and Atomic Formulae 1.12, as discussed in the Remark on the Confusing Appearance of Atomic Formulae 1.13 and the Remark on the Role of the Nominalistic Logical Universe in the Definition of Atomic Formulae 1.14.

Secondly, it is crucial to understand the rationale behind the domain and codomain of the Set Theoretic Encoding Function 3.6. This function precisely maps the symbols required to define Atomic Term Formulae 1.4 (i.e., the Set of Terms 1.1 and the Terms Relations 1.2) to sets and variables (i.e. the Nominalistic Logical Universe 3.2 of the language of set theory equipped with ZFC and the Set of Terms 1.1). The Canonical Term-Atomic Partial Strong Translation from a Set Theoretic Encoding 3.7 must therefore take as input exactly those symbols used to initially generate the Atomic Term Formulae 1.4. These symbols are also crucial for initiating a recursion on the complexity of formulae, as defined in the n-Language*. Consequently, the domain must consist of the Atomic Term Formulae of the given language, but this requirement does not extend to the codomain of the Set Theoretic Encoding Function 3.6, since the Term Relations 1.2 and the Terms can be encoded in any set.

Similarly, in defining a Canonical Term-Atomic Partial Strong Translation from a Set Theoretic Encoding 3.7, the aim is to map these symbols to formulae in language of set theory of the form

$\{\varepsilon_x(\lambda_1), \{\varepsilon_x(\lambda_n)\}, ..., \{\{\{...\{\varepsilon_x(\lambda_n)\}...\}\}\}\bar{\in}\varepsilon_x(\circ_n)$, which constitute a subset of $\mathcal{A}_{\epsilon,ZFC}$.

In order to arrive to the Definition of the Canonical Translation from Set Theoretic Encoding 3.11, I must first present the induction process that takes the Canonical Term-Atomic Partial Strong Translation from a Set Theoretic Encoding 3.7 as its 0th case. Then, I am able to extend this to the whole language by recursion on the complexity of formulae for any successor and limit ordinal (so that this recursion has the same generality as the one in the Definition of n-Language*). This process is familiar and should not surprise the reader, as it has already been employed in the definition of a Language* through the n-Language*.

The Definition of a n-Canonical Partial Strong Translation from Set Theoretic Encoding 3.10 that enables transfinite induction, requires first a mapping between the Formulae Relations 1.3 of the two languages. In fact in the definition of Set Theoretic Encoding Function 3.6 I exposed how the Terms 1.1 and the Terms Relations 1.2 are mapped, but I did not yet define any mapping on Formulae Relations 1.3, which is clearly necessary in order to perform the recursion on complexity of formulae that defines the n-Canonical Partial Strong Translation from Set Theoretic Encoding 3.10.

In order to give a formal definition of such a mapping between Formulae Relations 1.3, i.e. the Formulae Relations Encoding Function 3.9, it is necessary to provide a definition of String Substitution so that I can precisely formulate the structural preservation requirement for the named function.

I proceed then by first introducing the String Substitution 3.8 that allows for the Definition of the Formulae Relation Encoding Function 3.9 and finally present the n-Canonical Partial Strong Translation from Set Theoretic Encoding 3.10

Definition 3.8 (String Substitution)**.** For X^Λ and X^Φ sets, $f^\Lambda : \Lambda_1 \cup \Sigma_1^\Lambda \to X^\Lambda$ and $f^\Phi : \Sigma_1^\Phi \to X^\Phi$ functions, n an ordinal and finally $\varphi \in \mathcal{L}_1$, then $\exists_{\dot\sigma_1,...,\dot\sigma_n \in \Lambda_1 \cup \Sigma_1^\Lambda \cup \Sigma_1^\Phi} \dot\varphi = \{\dot\sigma_1, \{\dot\sigma_2\}, ..., \{\{\{...\{\dot\sigma_n\}...\}\}\}\}$.

(a) Then first define the set $\{\dot\sigma_i^* : i \in [1,n] \wedge (\dot\sigma_i^* = f^\Lambda(\dot\sigma_i) \vee \dot\sigma_i^* = f^\Phi(\dot\sigma_i))\}$, hence define $(\dot\varphi)_{f^\Lambda, f^\Phi} := \{\dot\sigma_1^*, \{\dot\sigma_2^*\}, ..., \{\{\{...\{\dot\sigma_n^*\}...\}\}\}\}$.

(b) Also for $(\dot{\mathcal{T}} \dot\vdash \dot\varphi) \in \mathfrak{I}_1$ and $\dot\vdash^* := \dot\vdash$, note that $(\dot{\mathcal{T}} \dot\vdash \dot\varphi)_{f^\Lambda, f^\Phi}$ is defined too.

Finally, for X^a a set, $f^a : \mathcal{A}_1^\Lambda \to X^a$ a function,

(c) Then define $(\dot\varphi)_{f^a, f^\Phi}$ as the unique set s.t. (i) $\forall_{\circ(p_1,...,p_{\alpha(\dot\circ)}) \subseteq \dot\varphi} p_1, ..., p_{\alpha(\dot\circ)} \in \mathcal{A}_1^\Lambda \to f^\Phi(\circ)(f^a(p_1), ..., f^a(p_{\alpha(\dot\circ)})) \in (\dot\varphi)_{f^a, f^\Phi}$, (ii) $\forall_{n \leq \check\lambda_1^2} \exists_{n_- < n} \forall_{\circ(\psi_1,...,\psi_{\alpha(\dot\circ)}) \subseteq \dot\varphi} \psi_1, ..., \psi_{\alpha(\dot\circ)} \in \mathcal{L}_1^{n_-} \to f^\Phi(\circ)((\psi_1)_{f^a, f^\Phi}, ..., (\psi_{\alpha(\dot\circ)})_{f^a, f^\Phi}) \in (\dot\varphi)_{f^a, f^\Phi}$. Similarly for $(\dot{\mathcal{T}} \dot\vdash \dot\varphi)_{f^a, f^\Phi}$.

I proceed by clarifying two potential causes of confusion in this definition, the first is on (a), while the second on (c).

Remark. A detail of this definition that may worry the reader is "$\dot\sigma_i^* = f^\Lambda(\dot\sigma_i) \vee \dot\sigma_i^* = f^\Phi(\dot\sigma_i)$" since, by the traditional notational rule for functions in set theory, either $f^\Lambda(\dot\sigma_i)$ or $f^\Phi(\dot\sigma_i)$ is defined (and therefore, one could claim, the named disjunction is never defined). In mathematical praxis, in fact, one usually accepts as a formal notation only functions on elements in the domain. Though, despite this notational convention, formally, functions are anything but relations with some properties, and relations shall be understood as sets of ordered pairs. That is to say, that a function $f : X \to Y$ is a set f s.t. $f \subseteq \{\{x, \{y\}\} : x \in X \wedge y \in Y\}$[9]; hence the formulation[10] $f(x) = y := \{x, \{y\}\} \in f$.

Therefore, one can easily note that $\sigma_i \mapsto \sigma_i^*$ is indeed a bijection and therefore the definition I gave is well-defined.

[9]Traditionally, a function $f : X \to Y$ is a set s.t. $\{X, Y, \{\{x, \{y\}\} : x \in X \wedge y \in Y\}\}$, here for simplicity I do not write the domain and codomain in the set representation of the function; even though I am aware them to be essential for the formal definition of the function. In fact the properties that distinguish a function from a relation necessarily use that more accurate representation.

[10]Here too, one might instead consider the definition "$f(x) := y$ if $\{x, \{y\}\} \in f$". For the purpose of this discussion though, I consider the definition I write in the text, so that it remains defined formula on every object. Note that clearly, as long as $x \in X$, the two definitions are equivalent, but when $x \notin X$ then the formulation given in the main text is trivially false, while the one given in the footnote is not defined. In the Definition 3.10 I will adopt the definition given in this footnote, this will be clarified by another footnote.

Remark. The reader might rightfully be concerned regarding the uniqueness claim in (c). One may resolve these concerns by considering that a formula is uniquely determined by the set of all its subformulae (i.e., all formulae of lower complexity contained within it). The definition, in fact, resembles a procedure analogous to recursion on the complexity of formulae: first, (i) I require $(\dot{\varphi})_{f^a,f^\Phi}$ to contain the mapping of all atomic formulae (both formulae of complexity 0 and complexity 1), and then (ii) I require $(\dot{\varphi})_{f^a,f^\Phi}$ to have the same subformulae for any complexity n_- inferior to n (n can be taken to be the complexity of the formula φ, I quantify on it, since φ could have any complexity).

Once the definition of $(\dot{\varphi})_{f^a,f^\Phi}$ is given for any formula, it is easy to see how one can proceed by extending the definition to Inference Rules*; this will be used extensively in the proof of Syntactic Enrichment 3.12.

Now that I defined the notion of String Substitution 3.8, I can define the Formulae Relation Encoding Function that enables me to define the notion of n-Canonical Partial Strong Translation from Set Theoretic Encoding 3.10.

This definition is meant to provide a mapping preserving the most out of the Formulae Relations 1.3 of one language while mapping those to the target language (that is language of set theory).

Definition 3.9 (Formulae Relations 1.3 Encoding Function)**.** For $\tau^a_{\varepsilon_x}$ a Canonical Term-Atomic Partial Strong Translation from a Set Theoretic Encoding 3.7, $\varepsilon^\Phi_{\varepsilon_x} : \Sigma^\Phi_1 \to \Sigma^\Phi_\epsilon$ is a formulae relations encoding function if (i) $\forall_{\dot{\circ}\in\Sigma^\Phi_1}\alpha(\dot{\circ}) = \alpha(\varepsilon^\Phi_{\varepsilon_x}(\dot{\circ}))$, (ii) $\forall_{(\dot{\mathcal{T}}_x \dot{\vdash} \dot{\varphi})\in\mathfrak{I}^c_1}(\dot{\mathcal{T}}_x \dot{\vdash} \dot{\varphi})_{\tau^a_{\varepsilon_x},\varepsilon^\Phi_{\varepsilon_x}} \in \mathfrak{I}^c_\epsilon$

Remark. From this definition one notices that the Formulae Relation Encoding Function preserves (i) the arity, i.e. all grammatical rules of a Formulae Relation 1.3 and (ii) all its Inference Rules* with respect to the theory $\mathcal{T}_x$. In the Example* this function is fairly simple (so simple, that I avoid writing it down in the proof), since it maps elements in Σ^Φ_{gr} to Σ^Φ_ϵ and $\Sigma^\Phi_{gr} = \Sigma^\Phi_\epsilon$, which also have the same Inference Rules* since $\mathfrak{I}_c = \mathfrak{I}_{gr} = \mathfrak{I}_\epsilon$. Therefore $\varepsilon^\Phi_{\varepsilon_{gr}} = id_{\Sigma^\Phi_{gr}}$.

Note that I require $(\dot{\mathcal{T}}_x \dot{\vdash} \dot{\varphi})_{\tau^a_{\varepsilon_x},\varepsilon^\Phi_{\varepsilon_x}}$ to be in the Closed Inference Rules* of language of set theory $\mathfrak{I}^c_\epsilon$, since that gives slightly more generality. There are though many ways of giving a more general version of this definition, that is though not required for the current aim.

As I remarked under the Definition of Formulae Relations 1.3, one might add another grammatical feature to Formulae Relations 1.3, i.e. the one that enables the distinction of quantifiers from other Formulae Relations 1.3, that is the function α^Φ. If one is therefore in disagreement with the arguments exposed in the Appendix I. Bounded and Free Variables*, where I argue why I decided to avoid this distinction, one shall introduce a second grammatical constraint to this function, that is (iii) $\forall_{\dot{\circ}\in\Sigma^\Phi_1}\alpha^\Phi(\dot{\circ}) = \alpha^\Phi(\varepsilon^\Phi_{\varepsilon_x}(\dot{\circ}))$.

Now that I provided all necessary definitions, I can present the notion of n-Canonical Partial Strong Translation from Set Theoretic Encoding 3.10. That is the recursion on complexity of formulae from the 0th atomic case of the Canonical Term-Atomic Partial Strong Translation from a Set Theoretic Encoding 3.7 to its extensions for both successor and limit ordinals. The reader is invited to recall the Definitions of n-Language* and the of Language* and the Remarks on those.

Definition 3.10 (n-Canonical Partial Strong Translation from Set Theoretic Encoding)**.** For $\varepsilon_x : \Lambda_1 \cup \Sigma^\Lambda_1 \to \Lambda_\epsilon \cup \mathcal{U}_{\epsilon,ZFC}$ a Set Theoretic Encoding Function 3.6 and $\varepsilon^\Phi_{\varepsilon_x} : \Sigma^\Phi_1 \to \Sigma^\Phi_\epsilon$ a Formulae Relations Encoding Function 3.9, first define $\tau^0_{\varepsilon_x} := \tau^a_{\varepsilon_x}$; then, for n a successor ordinal in $[1, \breve{\lambda}^2_{\mathcal{L}}]$, define $\tau^n_{\varepsilon_x} : \mathcal{L}^n \to \mathcal{L}_\epsilon, \circ(\varphi_1, ..., \varphi_{\alpha(\dot{\circ})}) \mapsto \varepsilon^\Phi_{\varepsilon_x}(\dot{\circ})(\tau^{n-1}_{\varepsilon_x}(\varphi_1), ..., \tau^{n-1}_{\varepsilon_x}(\varphi_{\alpha(\dot{\circ})}))$; for n^* a limit ordinal in $[1, \breve{\lambda}^2_{\mathcal{L}}]$, there exists a $n^*_- < n^*$ s.t. the following is well-defined: $\tau^{n^*}_{\varepsilon_x} : \mathcal{L}^{n^*} \to \mathcal{L}_\epsilon, \circ(\varphi_1, ..., \varphi_{\alpha(\dot{\circ})}) \mapsto \varepsilon^\Phi_{\varepsilon_x}(\dot{\circ})(\tau^{n^*_-}_{\varepsilon_x}(\varphi_1), ..., \tau^{n^*_-}_{\varepsilon_x}(\varphi_{\alpha(\dot{\circ})}))$

In this definition I make a claim that requires a proof, I give it here:

Claim. For n^* a limit ordinal in $[1, \breve{\lambda}^2_{\mathcal{L}}]$, there exists a $n^*_- < n^*$ s.t. the following is well-defined[11]:

[11]Recall the second footnote after 3.8 where I compare "$f(x) = y := \{x, \{y\}\} \in f$" with "$f(x) := y$ if $\{x, \{y\}\} \in f$". Since in the upcoming formulation I give no "=" sign but just construct another function, the formulation "$\tau^{n^*_-}_{\varepsilon_x}(\varphi_1)$" shall be understood as "$\tau^{n^*_-}_{\varepsilon_x}(\varphi_1) := y$ if $\{\varphi_1, \{y\}\} \in \tau^{n^*_-}_{\varepsilon_x}(\varphi_1)$". I am hence here using a different notation of function then the one I sued before, that is though trivially the case (and hence usually omitted in mathematical praxis) since I am not proving an equality but just giving the unique construction of a function. Hence, what I need to prove in order to conclude well-definiteness is that $\varphi_1, ..., \varphi_{\alpha(\dot{\circ})}$ are in the domain of the function.

$\tau^{n^*}_{\varepsilon_x} : \mathcal{L}^{n^*} \to \mathcal{L}_\epsilon, \circ(\varphi_1, ..., \varphi_{\alpha(\dot\circ)}) \mapsto \varepsilon^\Phi_{\varepsilon_x}(\dot\circ)(\tau^{n^*_-}_{\varepsilon_x}(\varphi_1), ..., \tau^{n^*_-}_{\varepsilon_x}(\varphi_{\alpha(\dot\circ)}))$

Proof. Note that in order to prove $\tau^{n^*}_{\varepsilon_x}$ well defined, I need only to prove $\varepsilon^\Phi_{\varepsilon_x}(\dot\circ)(\tau^{n^*_-}_{\varepsilon_x}(\varphi_1), ..., \tau^{n^*_-}_{\varepsilon_x}(\varphi_{\alpha(\dot\circ)}))$ well defined, that is to say that each $\tau^{n^*_-}_{\varepsilon_x}(\varphi_i)$ for $i \in [1, \alpha(\dot\circ)]$ and $\varphi_i \in \mathcal{L}^{n^*}$ is well defined. This is equivalent to the claim that there exists an ordinal $n^*_- < n^*$ s.t. $\forall_{i\in[1,\alpha(\dot\circ)]}\varphi_i \in \mathcal{L}^{n^*_-}$, since $\mathcal{L}^{n^*_-}$ is the domain of $\tau^{n^*_-}_{\varepsilon_x}$. Since, by Definition of n-Language*, for a limit ordinal n^*, $\mathcal{L}^{n^*} = \bigcup_{n^\times_- < n^*} \mathcal{L}^{n^\times_-}$, then note $\varphi \in \mathcal{L}^{n^*} \to \exists_{n^\times_- < n^*}\varphi \in \mathcal{L}^{n^\times_-}$. Since the Definition of n-Language is hierarchical (i.e. $m < m' \to \mathcal{L}^m \subseteq \mathcal{L}^{m'}$), call n^*_- the maximal such $n^\times_-$ and conclude $\forall_{i\in[1,\alpha(\dot\circ)]}\varphi_i \in \mathcal{L}^{n^*_-}$. Hence each $\tau^{n^*_-}_{\varepsilon_x}(\varphi_i)$ for $i \in [1, \alpha(\dot\circ)]$ and $\varphi_i \in \mathcal{L}^{n^*}$ is well defined. □

To summarise the bigger picture, one shall here recall that the aim is to define the Canonical Partial Strong Translation from Set Theoretic Encoding 3.11 that is a systematic procedure to define a Partial Translation* to the language of set theory. Now that I defined the nth case full generality using recursion on complexity of formulae, I am only left with defining the *final* case, i.e. the one whose maximal complexity corresponds to the maximal complexity of a formula in the language. For this to work it is necessary that the complexity of the source language (in the example $\mathcal{L}_{gr}$) has no greater complexity than the one of the target language, i.e. language of set theory.

Definition 3.11 (Canonical Partial Strong Translation from Set Theoretic Encoding)**.** If $\breve\lambda^2_1 \leq \breve\lambda^2_\epsilon$, define $\tau_{\varepsilon_x} := \tau^{\breve\lambda^2_1}_{\varepsilon_x}$

Once all these definitions have been laid down, it is still important to check whether they are coherent with the rest of the definitions given up to now, that is to say, one shall prove that the function I call Canonical Partial Strong Translation from Set Theoretic Encoding 3.11 is properly a Partial Strong Translation 2.1.

Theorem 3.12 (Syntactic Enrichment)**.** *A Canonical Partial Strong Translation from Set Theoretic Encoding* is a Partial Strong Translation 2.1.*

Proof. Let τ_{ε_x} be a Canonical Partial Strong Translation from Set Theoretic Encoding* hence assume $\breve\lambda^2_1 \leq \breve\lambda^2_\epsilon$, and claim $\forall_{\varphi\in\mathcal{L}_1}\mathcal{T}_x \vdash_1 \varphi \leftrightarrow \tau_{\varepsilon_x}(\mathcal{T}_x) \vdash_\epsilon \tau_{\varepsilon_x}(\varphi)$. I prove here both directions, let $\varphi \in \mathcal{L}_1$:

"→" *Claim.* $\forall_{\varphi\in\mathcal{L}_1}\mathcal{T}_x \vdash_1 \varphi \to \tau_{\varepsilon_x}(\mathcal{T}_x) \vdash_\epsilon \tau_{\varepsilon_x}(\varphi)$.
Assume $\mathcal{T}_x \vdash_1 \varphi$, it is equivalent to $(\dot\emptyset \dot\vdash \dot\varphi) \in \mathfrak{T}^x_1$, which is equivalent to $(\dot{\mathcal{T}}_x \dot\vdash \dot\varphi) \in \mathfrak{I}^c_1$; this implies [3.9] $(\dot{\mathcal{T}}_x \dot\vdash \dot\varphi)_{\tau^a_{\varepsilon_x}, \varepsilon^\Phi_{\varepsilon_x}} \in \mathfrak{I}^c_\epsilon$. From the other end, note that $\tau_{\varepsilon_x}(\mathcal{T}_x) \vdash_\epsilon \tau_{\varepsilon_x}(\varphi)$ is equivalent to $(\tau_{\varepsilon_x}\dot(\mathcal{T}_x) \dot\vdash \tau_{\varepsilon_x}\dot(\varphi)) \in \mathfrak{I}^c_\epsilon$. The claim is equivalent[12] to $(\dot{\mathcal{T}}_x \dot\vdash \dot\varphi)_{\tau^a_{\varepsilon_x}, \varepsilon^\Phi_{\varepsilon_x}} \in \mathfrak{I}^c_\epsilon \to (\tau_{\varepsilon_x}\dot(\mathcal{T}_x) \dot\vdash \tau_{\varepsilon_x}\dot(\varphi)) \in \mathfrak{I}^c_\epsilon$; I proceed by recursion on complexity of formulae on both φ and elements of $\mathcal{T}_x$, and prove the claim first for the atomic, successor and the limit case.

At. Case *Claim.* For $p \in \mathcal{A}^\Lambda_1$ and $\mathcal{T}^a_x \subseteq \mathcal{A}^\Lambda_1$, $(\dot{\mathcal{T}}^a_x \dot\vdash \dot p)_{\tau^a_{\varepsilon_x}, \varepsilon^\Phi_{\varepsilon_x}} \in \mathfrak{I}^c_\epsilon \to (\tau_{\varepsilon_x}\dot(\mathcal{T}^a_x) \dot\vdash \tau_{\varepsilon_x}\dot(p)) \in \mathfrak{I}^c_\epsilon$.
Proof. Note [3.8 c] that $(\dot p)_{\tau^a_{\varepsilon_x}, \varepsilon^\Phi_{\varepsilon_x}} = \tau^a_{\varepsilon_x}(p)$ and [3.8 c] $(\dot{\mathcal{T}}^a_x)_{\tau^a_{\varepsilon_x}, \varepsilon^\Phi_{\varepsilon_x}} = \{\tau^a_{\varepsilon_x}(p_x) : p_x \in \mathcal{T}^a_x\} = \tau^a_{\varepsilon_x}(\mathcal{T}^a_x)$, conclude $(\dot{\mathcal{T}}^a_x \dot\vdash \dot p)_{\tau^a_{\varepsilon_x}, \varepsilon^\Phi_{\varepsilon_x}} = (\tau_{\varepsilon_x}\dot(\mathcal{T}^a_x) \dot\vdash \tau_{\varepsilon_x}\dot(p))$ and derive the claim.

Succ. Case *Claim.* For n a successor ordinal, $\varphi \in \mathcal{L}^n_1$ and $\mathcal{T}^n_x \subseteq \mathcal{L}^n$, $(\dot{\mathcal{T}}^n_x \dot\vdash \dot\varphi)_{\tau^a_{\varepsilon_x}, \varepsilon^\Phi_{\varepsilon_x}} \in \mathfrak{I}^c_\epsilon \to (\tau_{\varepsilon_x}\dot(\mathcal{T}^n_x) \dot\vdash \tau_{\varepsilon_x}\dot(\varphi)) \in \mathfrak{I}^c_\epsilon$. This is proven by induction with the atomic case as the induction start and the induction step that I prove here: $(\forall_{\psi\in\mathcal{L}^{n-1}}\tau^{n-1}_{\varepsilon_x}(\psi) = (\psi)_{\tau^a_{\varepsilon_x}, \varepsilon^\Phi_{\varepsilon_x}}) \to (\forall_{\varphi\in\mathcal{L}^n}\tau^n_{\varepsilon_x}(\varphi) = (\varphi)_{\tau^a_{\varepsilon_x}, \varepsilon^\Phi_{\varepsilon_x}})$.
Proof. Clearly $\exists_{\dot\circ\in\Sigma^\Phi_1}\exists_{\psi_1,...,\psi_{\alpha(\dot\circ)}\in\mathcal{L}^{n-1}_1}\dot\varphi = \circ(\psi_1, ..., \dot\psi_{\alpha(\dot\circ)})$, note [3.8 c] that $(\dot\varphi)_{\tau^a_{\varepsilon_x}, \varepsilon^\Phi_{\varepsilon_x}} = \varepsilon^\Phi_{\varepsilon_x}(\circ)((\dot\psi_1)_{\tau^a_{\varepsilon_x}, \varepsilon^\Phi_{\varepsilon_x}}, ..., (\dot\psi_{\alpha(\dot\circ)})_{\tau^a_{\varepsilon_x}, \varepsilon^\Phi_{\varepsilon_x}})$. From the other end, note [3.11] $\tau_{\varepsilon_x}(\varphi) = \tau^n_{\varepsilon_x}(\varphi)$ and [3.10] $\tau^n_{\varepsilon_x}(\varphi) = \varepsilon^\Phi_{\varepsilon_x}(\circ)(\tau^{n-1}_{\varepsilon_x}(\psi_1), ..., \tau^{n-1}_{\varepsilon_x}(\psi_{\alpha(\dot\circ)}))$, by the induction hypothesis (i.e. $\forall_{\psi\in\mathcal{L}^{n-1}}\tau^{n-1}_{\varepsilon_x}(\psi) = (\psi)_{\tau^a_{\varepsilon_x}, \varepsilon^\Phi_{\varepsilon_x}}$), conclude $\forall_{\varphi\in\mathcal{L}^n}\tau^n_{\varepsilon_x}(\varphi) = (\varphi)_{\tau^a_{\varepsilon_x}, \varepsilon^\Phi_{\varepsilon_x}}$ and hence the claim.

[12] If one wishes to see how this proof may be conducted for a similar Canonical Translation but defined from a Relational Encoding 3.1, one shall visit the Formal Appendix of Hypotheses: Canonical Partial Strong Translation from a Relational Encoding*

Lim. Case *Claim.* For n a limit ordinal, $\varphi \in \mathcal{L}_1^n$ and $\mathcal{T}_x^n \subseteq \mathcal{L}^n$, $(\dot{\mathcal{T}}_x^n \dot{\vdash} \dot{\varphi})_{\tau^a_{\varepsilon_x}, \varepsilon^\Phi_{\varepsilon_x}} \in \mathfrak{I}_\epsilon^c \to (\tau_{\varepsilon_x}(\dot{\mathcal{T}}_x^n) \dot{\vdash} \tau_{\varepsilon_x}(\dot{\varphi})) \in \mathfrak{I}_\epsilon^c$. This is proven by transfinite induction with the atomic case and the successor case together with what I prove here: $(\forall_{n_- < n} \forall_{\psi \in \mathcal{L}^{n_-}} \tau_{\varepsilon_x}^{n_-}(\psi) = (\psi)_{\tau^a_{\varepsilon_x}, \varepsilon^\Phi_{\varepsilon_x}}) \to (\forall_{\varphi \in \mathcal{L}^n} \tau^n_{\varepsilon_x}(\varphi) = (\varphi)_{\tau^a_{\varepsilon_x}, \varepsilon^\Phi_{\varepsilon_x}})$.

Proof. Clearly, for some $n_- < n$, $\exists_{\dot{\circ} \in \Sigma_1^\Phi} \exists_{\psi_1, ..., \psi_{\alpha(\dot{\circ})} \in \mathcal{L}_1^{n_-}} \dot{\varphi} = \circ(\psi_1, ..., \psi_{\alpha(\dot{\circ})})$, note [3.8 c] that $(\dot{\varphi})_{\tau^a_{\varepsilon_x}, \varepsilon^\Phi_{\varepsilon_x}} = \varepsilon^\Phi_{\varepsilon_x}(\circ)((\dot{\psi}_1)_{\tau^a_{\varepsilon_x}, \varepsilon^\Phi_{\varepsilon_x}}, ..., (\dot{\psi}_{\alpha(\dot{\circ})})_{\tau^a_{\varepsilon_x}, \varepsilon^\Phi_{\varepsilon_x}})$. From the other end, note [3.11] $\tau_{\varepsilon_x}(\varphi) = \tau^n_{\varepsilon_x}(\varphi)$ and [3.10] $\tau^n_{\varepsilon_x}(\varphi) = \varepsilon^\Phi_{\varepsilon_x}(\circ)(\tau^{n_-}_{\varepsilon_x}(\psi_1), ..., \tau^{n_-}_{\varepsilon_x}(\psi_{\alpha(\dot{\circ})}))$, by the induction hypothesis (i.e. $\forall_{n_- < n} \forall_{\psi \in \mathcal{L}^{n-1}} \tau^{n-1}_{\varepsilon_x}(\psi) = (\psi)_{\tau^a_{\varepsilon_x}, \varepsilon^\Phi_{\varepsilon_x}}$), conclude $\forall_{\varphi \in \mathcal{L}^n} \tau^n_{\varepsilon_x}(\varphi) = (\varphi)_{\tau^a_{\varepsilon_x}, \varepsilon^\Phi_{\varepsilon_x}}$ and hence the claim.

"←" *Claim.* $\forall_{\varphi \in \mathcal{L}_1} \mathcal{T}_x \vdash_1 \varphi \leftarrow \tau_{\varepsilon_x}(\mathcal{T}_x) \vdash_\epsilon \tau_{\varepsilon_x}(\varphi)$.
Note that the previously conducted induction, proves that $\forall_{\varphi \in \mathcal{L}} \tau_{\varepsilon_x}(\varphi) = (\varphi)_{\tau^a_{\varepsilon_x}, \varepsilon^\Phi_{\varepsilon_x}}$ which analogously proves this direction as well.

□

This is the second important and original result that I prove in this work, after presenting the constructions that follow the Complete Descriptions 1.7. This theorem, together with its counterpart contained in the original work, Syntactic Reduction*, provides both practical and theoretical insights into the nature of morphisms between syntactic structures: on one hand, it allows for a formal understanding of the foundations of mathematics and other formal languages within Set Theory, and on the other hand, it provides an explicit procedure for constructing those morphisms. More results of this sort can be found both in the original work, the Bachelor Thesis, and in the Master Thesis I am currently working on.

References

[Cherniak 1986] Cherniak, Christopher. (1986). *Minimal Rationality*. MIT Press.

[Dzhaparidze 1993] Dzhaparidze, Giorgie. (1993). A generalized notion of weak interpretability and the corresponding modal logic. *Annals of Pure and Applied Logic*, 61, 113-160. North-Holland.

[Leitgeb 2024] Leitgeb, Hannes, Nodelman, Uri, & Zalta, N. Edward. (2024). *A Defense to Logicism*. Available at: mally.stanford.edu

[Quine 1986] Quine, W. V. (1986). *Philosophy of Logic: Second Edition*. Harvard University Press.

[Shapiro & Kissel 2024] Shapiro, Stewart, & Kissel, Teresa Kouri. (2024). *Classical Logic*. In E. N. Zalta & U. Nodelman (Eds.), *The Stanford Encyclopedia of Philosophy* (Spring 2024 Edition). Retrieved from SEP Classical Logic.

[Speaks 2021] Speaks, Jeff. (2021). *Theories of Meaning*. In E. N. Zalta (Ed.), *The Stanford Encyclopedia of Philosophy* (Spring 2021 Edition). Retrieved from SEP Meaning.

[Tarski 1933] Tarski, Alfred. (1983). *On the Concept of Logical Consequence*. Translation of Tarski 1936b by J.H. Woodger in Tarski 1983a, pp. 409–20.

[Tarski 1936] Tarski, Alfred. (1936). *Über den Begriff der logischen Folgerung*. In *Actes du Congrès International de Philosophie Scientifique*, fasc. 7 (*Actualités Scientifiques et Industrielles*, vol. 394), Paris: Hermann et Cie, pp. 1–11.

[Tarski 1946] Tarski, Alfred. (1946). *Introduction to Logic and the Methodology of the Deductive Sciences*. Oxford University Press.

[Tarski, Mostovski, & Robinson 1953] Tarski, Alfred, Andrzej Mostovski, & Raphael M. Robinson. (1953). *Undecidable Theories*. North-Holland.

[Tennant 2023] Tennant, Neil. (2023). *Logicism and Neologicism*. In E. N. Zalta & U. Nodelman (Eds.), *The Stanford Encyclopedia of Philosophy* (Winter 2023 Edition). Retrieved from SEP Logicism.

[Warren 2020] Warren, Jared. (2020). *Shadows of Syntax*. New York: Oxford University Press.

[Visser 1988] Visser, Albert. (1988). *Preliminary Notes on Interpretability Logic*. Department of Philosophy, University of Utrecht, January 1988.