Bealer’s Intensional Logic

Clarence Protin

August 2021

Abstract

Many intuitively valid arguments involving intensionality cannot be captured by first-order logic, even when extended by modal and epistemic operators. Indeed, previous attempts at providing an adequate treatment of the phenomenon of intensionality in logic and language, such as those of Frege, Church, Russell, Carnap, Quine, Montague and others are fraught with numerous philosophical and technical difficulties and shortcomings. We present Bealer’s solution to this problem which hinges on an ontological commitment to theory of Properties, Propositions and Relations (PRP). At the most basic level we can distinguish two conceptions in the theory of PRPs. An objective one tied to modality and necessary equivalence, and a mental (intentional) one tied to concepts and the requirement of non-circularity in definitions. Building on the work of Russell, Church and Quine, Bealer proposes two distinct intensional logics T1 and T2 (presented in Hilbert form) corresponding to these two conceptions, both based on the language of first-order logic extended with an intensional abstraction operator. In T1 necessitation can be directly defined and the axioms entail that we obtain standard S5 modal logic. These logics have a series of striking features and desirable aspects which set them apart from higher-order approaches. Bealer constructs a non-Tarskian algebraic semantic framework, distinct from possible worlds semantics and its problematic ontological commitments, yielding two classes of models for which T1 and T2 are both sound and complete. Other features include being able to deal with quantifying-in, and the various substitution puzzles, being free from artificial type restrictions, having a Russelian semantics, satisfying Davidson’s learnability requirement, etc. Bealer proposes his logic as the basis of a larger philosophical project in the tradition of logicism (or logical realism) concerning which we refer to his book Quality and Concept (1982). This includes a neo-Fregean logicist foundation of arithmetic and set-theory in which various (according to him) purely logical predications axioms (and intensional analogues of ZF, NGB, or Kelley-Morse axioms) are adjoined to T2, thereby explaining incompleteness as a property of pure logic rather than of mathematics. Surprisingly, and rather ironically, Bealer’s logic also fulfills Carnap’s thesis of extensionality due precisely to its ontological commitment to the reality of PRPs. The proof of these results consists either in lemmas which are merely stated or which are given but brief sketches of a proof. We aim to give detailed proofs of all the mathematical logical results that appear in Bealer’s Quality and Concept and in [2] and to clarify and simplify some of the concepts and techniques so as to bring Bealer’s work to a larger audience of philosophers, logicians, linguists and mathematicians and to be better equipped to address some of the unsolved problems and challenges. We also include a brief introduction to other approaches to intensionality in natural language and discuss how Bealer’s approach compares favourably to some them and is likely to benefit from the insights offered by others.
1 Approaches to Intensionality

Ever since the pioneering work of Montague\[?\] there has been interest in applying mathematical logic to the formalisation of natural language. This involves setting up a system of axioms and rules in a certain formal language together with a semantics which we would expect to be sound. The most basic requirement is that informally valid inferences in natural language should be mirrored and captured faithfully in our formal system. We are at once confronted with a major problem: that of the *denotation* and *meaning* of a sentence or parts of a sentence in natural language. If we are to formally ascribe denotation and meaning (as defined by Frege\[14\] and formalised by Church\[10, 11, 12\]) to sentences in our formal language, then these must be faithful to the actual denotations and meanings in natural language: for instance, if two expressions have different meanings then their corresponding formal expressions must likewise have different meanings. Is standard extensionalist Tarskian semantics, or a modification thereof, which discards meaning in favour of denotation, adequate for this task? Or must we look for an entirely new kind of formal semantics? Also, to what extent does this conflict of different semantics have implications for or is influenced by ontological and epistemic positions? We will focus on the adequacy of logical systems to represent and mirror faithfully informally valid reasoning in natural language together with factors of economy, simplicity and elegance, both of the logical system and its semantics. We refer the reader to the table of desiderata for such logical systems in \[1\][p.35-15] for a list of 25 features which include classical paradoxes which need to be accounted for, aspects of informal reasoning which must be captured, various ontological and epistemic contraints as well as philosophically desirable but not necessarily essential features, such as furnishing a foundation for mathematics.

Natural language abounds with syntactic constructions which *prima facie* have no analogue in the restricted domain of mathematical discourse. Our major example will be propositional attitudes (for example, an agent knowing or believing something), modalities and that-clauses. These constructions are all examples of *intensionality*: the seem to involve the meaning of the embedded sentence rather than its denotation. This is the cause of many serious problems. If two terms are merely co-extensive then we cannot substitute one for another according to the standard Leibniz rule. Suppose John believes that Scott is not the author of *Waverley*. Now "the author of *Waverley*" and Scott designate the same individual. But if we substitute one for the other in the sentence above we obtain \(\therefore\) John believes that Scott is not Scott. Following \[27\] we call this type of problem (A), a particular case of a wide class of *substitutivity problems* such as Mate’s puzzle and the paradox of analysis(see \[1\][ch.1 and 3] for a good introduction to these problems). Another problem involves terms which have no denotation when we attempt to apply the standard \(\exists\)-introduction rule. From John desires a unicorn we get \(\therefore\) \(\exists\) x. John desires x. How are we to interpret this existential quantifier? What precisely does it quantify? We refer to this type of problem as (B).

Historically \[13, 20\] modalities were some of the first of these intentional constructs to be studied formally, but only in the context of a fragment consisting of adjoining modal operators to already known systems of formal logic such as propositional or classical first-order logic. Epistemic and temporal constructs are dealt with in an *ad hoc* fashion by introducing the corresponding operators. With this approach it is clear that if we wish to have a general logic capable of dealing with all these constructs then such a logic would have to be rather eclectic and uneconomical in form, abounding in undefinable primitive symbols. On the other hand a new kind of semantics was proposed, the *possible worlds* semantics associated with Carnap, Kripke and Hintikka. This is the semantics that served as the basis for Montague’s approach, which was later axiomatised by Gallin\[15\]. The key features are that meaning is a *function* from states to extensions involving individuals and that we have an elaborate type-hierarchy and thus a limitation on self-applicability of predicates. For instance, the expression *number of the planets*
denoted different numbers at different historical epochs. The many-worlds approach has served as the basis for the majority of general formal approaches to intensionality in language until today. For example, the work of Cresswell\[13\], Bressan\[8\] and Tichý\[29, 30\].

Another approach is the sententialist (or inscriptionalist) one which is associated to Quine, Davidson and Scheffler. The sentences which occur within intentional constructs are taken as concrete inscriptions. In a recent formulation of Scheffler’s approach by David Parsons\[27\] we have a considerably complex system which is first-order and employs standard Tarskian semantics. In this system belief is taken as a binary relation between an agent and an inscription. Sentences are the truth-bearers.

The third class of approaches are those which are neither based on many-worlds semantics nor sententialism. Within this class are the very interesting approaches of Moschovakis\[23, 24, 25\] based on the "sense as algorithm" idea, the Meinongian approaches of Zalta\[31\] and Priest\[31\][p.129-145] and finally the approach of George Bealer, based on the theory of Properties, Relations and Propositions (PRP). Bealer’s logic is a classical first-order logic with equality augmented with a term-forming abstraction bracket operator \([A]_{x_1,...,x_n}\) in the style of Quine’s corner-bracket. This operator is syntactically very similar to the extensional operator \([x : A]\) in the Kelley-Morse formulation of set theory. Bealer starts out with two axiomatic systems T1 and T2. The axiomatic systems are quite simple and natural. In T1 the box operator can be defined and with this definition we obtain an embedding of S5 logic. The semantics are directly inspired by the ontology of PRPs, propositions and \(n\)-ary relations appear as irreducible entities alongside individuals. Formulas within the bracket operator are interpreted by means of such entities. In the class of models adequate for T1 each entity is assigned a set of possible extensions ranging over the entire model. In T2 only one extension is considered.

We now consider in brief some arguments justifying the preferability of Bealer’s approach relative to the many-worlds and sententialist/inscriptionalist approaches. We will then mention some of the challenges and open problems which remain and how the approaches of Zalta and Moschovakis might shed some light on these questions.

Let us consider what seems to be the most interesting sententialist approach, Scheffler’s approach which has been lately formalised by D. Parsons in \[27\][185-208] as the first-order system \(ISM_{P}^{L}\). In order to deal with that-clauses \[27\][p.204] an infinite number of unanalyisable one-place predicates "That-P" need to be introduced, corresponding to the infinite number of sentence-inscriptions of the object language \(L\). This violates Davidson’s learnability requirement (cf. \[1\][p.28]). Although system \(ISM_{P}^{L}\) can deal with problems (A) and (B), it is considerably complex, having numerous axioms and additional recursive axiom schemes (rules) introduced \textit{ad hoc} to deal with the combinatorics of sentence-inscriptions and that-clauses. This might be remedied by considering the axioms for Peano arithmetic and using Gödel coding. Also, the question of completeness is not addressed. Leaving aside the violation of Davidson’s learnability requirement, all things being equal, a simpler and more elegant system which solves the same problems is clearly preferable to the Scheffler-Parsons approach. A fatal problem with the sententialist approach lies in the nature of sentence-inscriptions. The Scheffler-Parsons approach is in itself a purely formal one. Despite the name, a "sentence-inscription" may correspond to entities of an indeterminate nature. The burden of the sententialist is to give a precise account of such sentence-inscriptions and to justify the counter-intuitive stance that propositional attitudes relate to sentence-inscriptions conceived as physical objects or sense perceptions (or classes of such). As Bealer observes in \[6\][p.174], how can the prelinguistic cognitive states of a child be described as a relating to sentence-inscriptions? The PRP theory on the other hand complies with our intuition: in knowing something we relate to propositions or states of affairs, not sentence-inscriptions. There are many powerful arguments against a sententialist and nominalist account of propositional attitudes. See \[28\] for a refutation of nominalist accounts of mathematical knowledge and \[6\][174-185] for a very detailed refutation of
Scheffler’s brand of sententialism. Also, Bealer has shown in [7][p.216-223] how Quine’s and Davidson’s scepticism regarding the need for entities such as PRPs can be fully met.

We now consider the many-worlds approach in which intensional constructs are interpreted as functions from worlds to extensions. It is a desideratum that an intensional logic should give a non-circular account for what is it for a proposition to be necessary, rather than leave the box as an undefined symbol. The Kripkean explanation in terms of truth in all possible worlds only sweeps the problem under the rug as it leaves open the question of nature of the accessibility relation $R$ between worlds. Bealer’s logic T1 makes the box a defined notion in terms of more primitive concepts and Bealer makes a convincing case for a non-circular definition of necessity in [1][p.204] based on the theory of PRPs. A serious problem in the many-worlds approach is fine-grain distinction between rigid concepts, predicates with have the same extension in all possible worlds yet clearly differ in sense. This is not a problem in Bealer’s PRP approach as sense is taken an ontological primitive. Another similar problem concerns the intensional interpretation of fictional names, predicates which arguably have empty extension in all possible worlds. What exactly are these objects which allow us in particular to make an application of rule (B)? To believe in unicorns is not the same as believing in fairies but both concepts correspond to the same function yielding the empty set in all worlds, a function which is difficult to attach any interpretation to. It is clear that the PRP approach is much more promising for it takes properties as primitives and unicorns and fairies correspond to different concepts. There are many more problems and objections to the many-worlds approach, such as regards to its higher-order presentation which imposes an artificial type-restriction on certain ”transcendental” predicates. Higher-order systems also have the disadvantage of being incomplete.

The main problem which confronts Bealer’s system (in particular in the form T2) is dealing with the wide class of substitutivity problems. While several of them are completely solved, others demand a further fine-grained distinction between PRPs. Bealer made further progress on these problems since [1] and in [7] he deploys the distinction between platonic and non-platonic modes of presentation in order to deal with the nature of proper names and the associated substitutivity puzzles. Although in [1] a pragmatic approach was proposed to solve Mate’s puzzle, in [3] he develops the idea of ”logical form” to allows a much finer distinction between intensional entities than is possible in T2. Although these theories are not entirely worked out and complete they are clearly among the most promising approaches. We believe that a comparison or combination with the ”sense as algorithm” approach of Moschavakis or the emphasis on sentences as constructions in Tichý may lead to the adequate concept of fine-grained distinction between intensional entities needed to solve the substitutivity puzzles.

2 The Nature of Bealer’s Logic

This paper aims to reformulate, clarify and simplify Bealer’s intensional logics, in particular by furnishing detailed proofs, so as to bring his work to a larger audience of logicians, linguists and philosophers. In this section we give our own motivation for Bealer’s logic and in section 3 we give a brief outline of the Bealer’s own logico-linguistic motivation for his logic, the details of which can be found in [1][Intr. and Ch.1]. Most of the lemmas in [1] are either merely stated or given a sketchy outline of a proof. The rest of the sections are dedicated to proving in detail the soundness and completeness of Bealer’s systems T1 and T2 as well as reformulating and simplifying certain concepts and details. We then conclude with a discussion of the work the remains to be done.

In certain versions of set theory (such as Kelley-Morse Set theory as presented in the classic book General Topology) one starts from standard first-order logic with equality endowed with
an extension operator, a term-forming operator which in general takes a formula \( \phi \) and forms the class \( \{ x : \phi(x) \} \) where \( \phi \) can have other free variables besides \( x \). Here the variable \( x \) functions like a quantified variable and the concepts of bound and free variable are extended in the expected way, including the usual proviso for performing substitution of terms for variables. The other fundamental feature is the presence of the distinguished binary membership predicate \( \epsilon \). The two fundamental axioms are the classification axiom scheme and the axiom of extensionality which express how \( \epsilon \) determines equality.

The basic idea of George Bealer’s first-order intensional logic is to consider an intensional version of a similar term-forming operator \( [x : \phi] \) (where we use square braces instead of curly ones). We call these “classes” intensional abstracts. Semantically, we can think of this logic as being a set theory with a plurality of membership predicates \( \epsilon_w \) (with some analogy to “possible worlds” semantics) and in which equality is intensional equality, a stronger property than mere membership equivalence (in this case, membership equivalence for all membership predicates). Such a membership equality for all membership predicates is called necessary extensional equality and is weaker than strict intensional equality. We also consider a special distinguished membership predicate representing the “actual world”.

Consider a version of Bealer’s logic without the intensional analogue of the membership predicate and consider two primitive notions of equality, ordinary equality = corresponding to strict intensional equality and \( \approx \) expressing necessary extensional equality. In this version we do not have the analogue of the Kuratowski construction of ordered-pairs and so we must take as primitives abstracts over (possible empty) sequences of distinct variables \([x_1...x_n : \phi]\). We can remedy this by introducing an ordered-pair operation as a primitive with the expected axiom \((x, y) = (x', y') \iff x = x' \& y = y'\). From this we can form arbitrary n-tuples \((x_1, ..., x_n)\) in the expected way. Thus we can define \([x_1, ..., x_n : \phi(x_1, ..., x_n)]\) as \[\{ x : \exists x_1, ..., x_n. x = (x_1, ..., x_n) \& \phi(x_1, ..., x_n) \}\]. We also consider the 0-tuple which we denote by \(\emptyset\), an additional primitive constant. Then we have that \([\emptyset : \phi]\) is \([x : x = \emptyset \& \phi]\). In Bealer’s logic this is interpreted as the proposition that \(\phi\). In extensional set theory, when \(\phi\) is closed, these classes can either be the ordinary empty set \(\emptyset\) or equal to the set \(\{\emptyset\}\) depending on whether \(\phi\) holds or not. Thus, in this case these kinds of abstract correspond to truth values. In general they correspond to taking the proposition \(\phi\) as a term. Consider the classical set-theoretic operations (functions) that can be performed on classes such as intersection, complement, projection (of a binary relation). These are the term analogues of the logical operations conjunction, negation and existential quantification. We can furthermore consider an expansion which involves transforming an \(n\)-ary relation \(R\) into a \(n + 1\)-ary one \(R'\) in a trivial way \((x_1, ..., x_n, x_{n+1}) \in R' \iff (x_1, ..., x_n) \in R\). There is also the inversion or, in general, a permutation of an \(n\)-ary relation as well as the identification of pairs of variables (generalised diagonals). In our situation these must be considered in principle as merely meta-syntactic operations. The central idea of first-order logic intensional logic, which allows us to define an adequate semantics, is the decomposing the structure of intensional abstracts in terms of such meta-syntactic operations, the most notable aspect being the use of a generalised form of substitution in which intensionally bound variables of the term to be inserted are added to the bound variables of the term they are substituted into.

3 Brief Motivation

3.1 Quantifying-in

Consider the following intuitively valid inference:
Whatever $x$ believes is necessary.
Whatever is necessary is true.
Whatever $x$ believes is true.

In standard quantifier logic this would be formalized as follows:

$$
\forall y. B(x, y) \rightarrow N(y) \\
\forall y. N(y) \rightarrow T(y) \\
\forall y. B(x, y) \rightarrow T(y)
$$

Now consider:

Whatever $x$ believes is true.
$x$ believes that $A$.
It is true that $A$.

If we treat that-clauses as singular terms, and represent for a formula $A$ the term 'the proposition that $A$' by $[A]$ then we can formalize this inference as follows:

$$
\forall y. B(x, y) \rightarrow T(y) \\
B(x, [A]) \\
T([A])
$$

But now consider:

Whatever $x$ believes is true.
$x$ believes that $v$ believes something.
It is true that $v$ believes something.

By analogy we would formalize this as follows:

$$
\forall y. B(x, y) \rightarrow T(y) \\
B(x, [\exists v. B(x, v)]) \\
T([\exists v. B(x, v)])
$$

But now consider the inference:

$x$ believes that he believes something.
There is someone $v$ such that $x$ believes that $v$ believes something.

A candidate for formalization would be:

$$
B(x, [\exists y. B(x, y)]) \\
\exists v. B(x, [\exists y. B(v, y)])
$$

In conclusion: that-clauses should be treated as singular terms which can contain externally quantifiable variables.
3.2 Semantic Problems

We need to find a way of dealing with substitutivity failures for materially equivalent and co-referential formulas. For instance:

\[
\begin{align*}
x & \text{ believes that everything runs.} \\
\text{Everything runs if and only if everything walks.} \\
x & \text{ believes that everything walks.}
\end{align*}
\]

\[
B(x, [\forall y. R(y)])
\]

\[
\forall y. R(y) \leftrightarrow \forall y. W(y)
\]

\[
B(x, [\forall y. W(y)])
\]

\[
x \text{ wonders whether } y \text{ is the author of } Waverley
\]

\[
y \text{ is the author of } Waverley
\]

\[
x \text{ wonders whether } y = y
\]

\[
W(x, [y = \iota z. A(z)])
\]

\[
y = \iota z. A(z)
\]

\[
W(x, [y = y])
\]

Intensional entities can thus be different even if their extension be the same. What kind of entities do the \([A]\) correspond to? Some approaches:

- Linguistic entities (Carnap, Quine): structural descriptions of a formula or inscription of a formula. But there is a difficulty with the Langford-Church translation test.

- Nothing at all: the above metalinguistics names and descriptions can be analysed away completely (Scheffler). According to Bealer these approaches seem to violate Davidson’s learnability requirement.

- Extra-linguistic entities such as propositions (Bealer’s neologicism).

According to Bealer there are two kinds of intensional entities. According to the first (traced back to Leibniz and one of Church’s formulations of Frege’s theory of sense) two intensional entities are equal if and if they are necessarily equivalent. This is related to possible-worlds semantics and deals with necessity and possibility. A definition need only be a necessary equivalence.

According to the second conception each intensional entity when defined completely must possess a unique non-circular definition. According to Bealer this is found in Leibniz’s distinction between simple and complex properties and Russell’s doctrine of logical atomism. This conception is used to treat intentionality: belief, desire, perception, etc.

Example of a deduction valid in the first conception but not in the second:

\[
x \text{ wonders whether there are trilaterals which are not triangles.}
\]

\[
\text{Necessarily all and only triangles are trilaterals.}
\]

\[
x \text{ wonders whether there are triangles which are not triangles.}
\]

Other nominative formations in language such as the gerundive and infinitive are formalized by intensional abstraction over variables \([A]_{x_1, \ldots, x_n}\). Intensional abstracts in general thus correspond to propositions, relations and properties.
3.3 Bealer’s Intensional Logic

What is Bealer’s intensional logic? The language of Bealer’s intensional logic is that of first-order logic with equality with a finite number of predicates symbols extended with term-forming intensional abstraction operators $[x_1...x_n]$. There are two kinds of intensional entities and thus two distinct axiomatic systems of intensional logic, $T_1$ and $T_2$. Bealer formulates two new classes of model for which $T_1$ and $T_2$ are both sound and complete. $T_1$ contains standard S5 modal logic. Using a single primitive binary predication predicate $\Delta$ and additional so-called “predication” axiom-schemes we obtain a neo-Fregean formulation of arithmetic. In [1][ch.7] Bealer argues that if we accept the ontology of intensional entities then we can translate intensional logic into standard extensional logic using standard Tarskian semantics (the thesis of extensionality).

The multi-modal approach originating with C. I. Lewis, Carnap, Hintikka and Kripke has some disadvantages. Operators apply only to formulas and not to singular terms so many intuitively valid arguments cannot be captured. It is eclectic and incomplete rather than based on a single intensional operator. According to Bealer, the higher-order approach has the following disadvantages:

- Does not have a sound and complete semantics.
- Predicates can be linguistic subjects which leads us to consider two sorts of meaning relation to avoid Frege’s ‘a=a’/‘a=b’ puzzle.
- Needs type theory to avoid paradoxes but this imposes implausible existence restrictions on intensional entities.
- Division of variables into sorts makes it impossible to deal with ”transcendental” predicates without violating Davidson’s learnability requirement.
- Requires possible-world semantics and commitment to non-actual possibilia.

We recommend the paper [3] for an interesting further discussion and defense of Bealer’s Intensional Logic.

4 Bealer Decomposition

**Definition 4.1** The language $L^\omega$ consists of a countable ordered collection of variables $x, y, z, ...$ and $n$-ary predicate symbols ($n \geq 1$) $F, G, H, ...$, with a distinguished binary predicate $=$, logical connectives $\&$, $\neg$, $\exists$ and the intensional abstraction operator $[x_1...x_n]$ where $x_1...x_n$ is a possibly empty sequence of distinct variables.

We use the notation $\bar{x}$ for a (possibly empty) sequence of distinct variables. We use lowercase letters (possibly with subscripts) towards the end of the Latin alphabet to denote variables.

**Definition 4.2** Formulas and terms of $L^\omega$ are defined by simultaneous induction:

- Variables are terms.
- If $t_1, ..., t_n$ are terms and $F$ is $n$-ary predicate, then $F(t_1,...,t_n)$ is a formula.
- If $A$ and $B$ are formulas, $v$ a variable, then $(A&B), \neg A$ and $\exists v. A$ are formulas.
- If $A$ is a formula and $v_1,...,v_m, 0 \leq m$, is a sequence of distinct variables, then $[A]_{v_1...v_m}$ is a term.
A term of the form \([A]_{v_1, ..., v_n}\) is called an *intensional abstract*. Intensional abstracts of the form \([F(v_1, ..., v_n)]_{v_1, ..., v_n}\) are called *elementary*. Intensional abstracts \([A]_\alpha\) where \(A\) is not an atomic formula are called *complex*.

The intensional abstraction operator generalises the set-theoretic extension operator \(\{x : A\}\). The standard notions of bound and free variable, bound-variable renaming \(\alpha\)-equivalence) as well as a term being free for \(x\) in \(A\) carry over to \(L^\omega\) in the expected way (note that intensional abstraction binds variables like quantifiers). Terms that differ only by bound variable renaming will be called *variants*.

We will define a set of (partially defined) syntactic operations on intensional abstracts and an algorithm to decompose an intensional abstract in a unique way in terms of such syntactic operations, elementary intensional abstracts and variables.

A term \([A]_{\bar{x}}\) is *normalized* if all the variables in \(\bar{x}\) are free in \(A\) and they display the order in which these variables first occur free.

If \(A\) is atomic then we call \([A]_{\bar{x}}\) a *prime term* if the variables in \(\bar{x}\) are free in \(A\), independently of the order.

Given an atomic formula \(F(t_1, ..., t_n)\) if a variable occurs free in more than one of the terms \(t_i\) then it is called a *reflected variable*. A prime term \([A]_{\bar{x}}\) is called a *prime reflection term* if \(\bar{x}\) contains a reflected variable in \(A\).

Consider a prime term \([F(t_1, ..., [B]_{\bar{y}}, ..., t_j)]_{\bar{x}}\) which is not a prime reflection term. If there is a variable in \(\bar{x}\) which is free in \([B]_{\bar{y}}\) and all the previous arguments are variables in \(\bar{x}\), then the prime term is a *prime relativized predication term* and such a variable is called a *relativized variable*.

Consider a prime term \([F(t_1, ..., t_k, ..., t_j)]_{\bar{x}}\) which is not a prime reflection term. If there is term \(t_k\) which is a variable not in \(\bar{x}\) or which contains no free variables which are in \(\bar{x}\) and all the previous arguments are variables in \(\bar{x}\), then it is a *prime absolute predication term*.

These definitions allow us to divide all intensional abstracts into seven disjoint categories. The following is clear:

**Lemma 4.3** Given an intensional abstract \([A]_{\bar{x}}\) it falls into one and one only of the following seven categories which depend only on \(A\) and \(\bar{x}\) independently of order (only on the underlying set). They are also invariant under the renaming of bound variables.

1. Complex and \(\bar{x}\) contains variables not free in \(A\).
2. Complex and all the variables in \(\bar{x}\) are free in \(A\).
3. Not complex and \(\bar{x}\) contains variables not free in \(A\).
4. Prime reflection term.
5. Prime relativized predication term.
6. Prime absolute predication term.
7. \([A]_{\bar{x}'}\) is elementary for some permutation \(\bar{x}'\) of \(\bar{x}\).

Before defining Bealer decomposition we need the following considerations on permutations. It is is clear that any permutation \(\sigma^n \in S^n(n \geq 2)\) can be decomposed into permutations of the form \(\sigma^n_i = (1 \ 2 \ ... \ n)\) (for \(n \geq 3\)) and \(\sigma^n_i = (n - 1 \ n)\). However, we need a uniquely defined decomposition defined as follows. Assume that our permutations are acting on sequences \(x_1, ..., x_n\) of variables and let \(x'_1, ..., x'_n\) be a permutation of \(x_1, ..., x_n\). We use the notation \(\sigma^n x_1, ..., x_n\) to denote the results of apply \(\sigma^n\) to the sequence. We associate to \(x_1, ..., x_n\) and \(x'_1, ..., x'_n\) (or
equivalently the permutation which transforms the first sequence into the second) a uniquely defined decomposition as follows:

Suppose \( \pi' = \pi_n...\pi_2\pi_1 \) has already been defined, where \( \pi_n \) is either \( \sigma^n_c \) or \( \sigma^n_c \). Then let \( s \) be the maximal subsegment of \( \pi'x_1...x_n \) equal to an initial segment \( b = x_1'...x_k' \) of \( x_1...x_n' \). If \( b \) is the whole \( x_1'...x_n' \) we are done. Otherwise \( x_{k+1}' \) will correspond to a certain \( x \) in \( \pi'x_1...x_n \). If \( x \) is the last element of this sequence then we choose \( \sigma^n_c \), otherwise we chose \( \sigma^n_c \). This process terminates and the decomposition is clearly unique. Given a permutation \( \sigma \) (or two sequences \( \bar{x}, \bar{x}' \) of the same length), we denote its decomposition by \( \beta(\sigma)(\sigma(\bar{x}, \bar{x}')) \).

**Example 4.4** Consider \( x_1x_2x_3x_4 \) and \( x_2x_4x_3x_1 \). The the process yields \( \sigma^4 \) and \( x_1x_2x_4x_3, \sigma^4 \sigma^4 \) and \( x_3x_1x_2x_4, \sigma^4 \sigma^4 \) \( \sigma^4 \) and \( x_4x_3x_1x_2 \) and finally \( \sigma^4 \sigma^4 \sigma^4 \sigma^4 \) and \( x_2x_4x_3x_1 \).

**Definition 4.5** The Bealer operations consists in eight syntactic operations defined on intensional abstracts. These consists in six unary operations \( C, I, N, U, E, R \) and partially-defined binary operations \( P_n \) for \( n \geq 0 \). These operations can be seen as defined on equivalence classes modulo renaming of bound variables. Let \([A]_{\bar{x}}\) be an intensional abstract with \( \bar{x} \) of length \( n \). Then we have:

\[
C[A]_{\bar{x}} = [A]_{\sigma^n_c \bar{x}} \quad n \geq 3
\]

\[
I[A]_{\bar{x}} = [A]_{\sigma^n_c \bar{x}} \quad n \geq 2
\]

\[
N[A]_{\bar{x}} = [\neg A]_{\bar{x}}
\]

\[
U[A]_{\bar{x}} = [\exists x_nA]_{x_1...x_{n-1}} \quad n \geq 1
\]

\[
E[A]_{\bar{x}} = [A]_{xy} \quad y \text{ the first variable not in } A \text{ or } \bar{x}
\]

\[
R[A]_{\bar{x}} = [A[x_{n-1}/x_n]]_{x_1...x_{n-1}}
\]

\[
K[A]_{\bar{x}}[B]_{\bar{y}} = [A \& B']_{\bar{x}} \quad [B']_{\bar{x}} \text{ variant of } [B]_{\bar{y}}
\]

\[
P_0[A]_{\bar{x}}[B]_{\bar{y}} = [A[[B]_{\bar{y}}/x_n]]_{x_1...x_{n-1}} \quad \text{where } n \geq 1 \text{ and } [B]_{\bar{y}} \text{ is free for } x_n \text{ in } [A]_{x_1...x_{n-1}}
\]

\[
P_n[A]_{\bar{x}}[B]_{\bar{y}} = [A[[B]_{\bar{y}}/x_n]]_{x_1...x_{n-1}} \bar{w} \quad \text{where } \bar{y} = \bar{x} \bar{w} ; \bar{w} \text{ length } n \geq 1, [B]_{\bar{y}} \text{ is free for } x_n \text{ in } [A]_{x_1...x_{n-1}}
\]

Consider a sequence \( \bar{x} \) and a permutation \( \bar{x}' \). We associate to \( \beta(\bar{x}, \bar{x}') \) a sequence \( \Sigma(\bar{x}, \bar{x}') \) of operations \( I \) and \( C \) obtained by replacing \( \sigma^n_c \) with \( I \) and \( \sigma^n_c \) with \( C \) in \( \beta(\bar{x}, \bar{x}') \). Thus in the previous example we have \( \Sigma(x_1x_2x_3x_4, x_2x_4x_3x_1) = CCCI \). This sequence is to interpreted compositionally. In general we shall use the notation \( \Sigma \) for any (possibly empty) sequence of operations \( C \) and \( I \). We use the notation \( E^n \) to denote \( E \) composed \( n \) times. Also we shall use the notation \( P^n_{tt} \) to mean \( P_0...P_0(P_0tt_1)t_2...t_n \).

We now define Bealer decomposition.

**Definition 4.6** Given an intensional abstract \([A]_{\bar{x}}\) or variable \( x \) the Bealer decomposition \( B[A]_{\bar{x}} \) is defined inductively as follows:
• Suppose $[A]_x$ belongs to categories 1 or 3. And let $\bar{x}'$ be a permutation of $\bar{x}$ of the form
$\bar{y}\bar{z}$ so that $[A]_{\bar{y}}$ is normalized and $\bar{z}$ (of length $m$) consists of the variables in $\bar{x}$ that do
not occur free in $A$ in order of their occurrence in $\bar{x}$. Then $B[A]_{\bar{x}} := \Sigma(\bar{x}, \bar{x}')E^mB[A]_{\bar{y}}$.

• Suppose $[A]_{\bar{x}}$ belongs to category 2 and is not normalized. And let $\bar{x}'$ be a permutation of $\bar{x}$ so that $[A]_{\bar{x}'}$ is normalized. Then $B[A]_{\bar{x}} := \Sigma(\bar{x}, \bar{x}')B[A]_{\bar{x}'}$.

• Suppose the term belongs to category 2 and is normalized. Then we have
$B[A\&B]_{\bar{x}} := KB[A]_{\bar{x}}B[B]_{\bar{x}}$
$B[\neg A]_{\bar{x}} := NB[A]_{\bar{x}}$
$B[\exists vA]_{\bar{x}} := UB[A]_{\bar{x}v}$ where $v$ is the first variable not in $A$ or $\bar{x}$.

• Let the term belong to category 4. Of the reflected variables in $\bar{x}$ consider the one $v$ with
the right-most occurrence and let be $t$ the right-most argument of $A$ in which $v$ has a free
occurrence. Let $w$ be the alphabetically earliest variable not occurring in $A$ or $\bar{x}$. Let $\bar{x}'$
be the permutation of $\bar{x}$ such that $[F(t_1, ..., t_n)]_{\bar{x}'}$ is normalized and consider $\bar{y}v$ where $\bar{y}$
is $\bar{x}'$ with $v$ removed. Then
$B[F(t_1, ..., t_n)]_{\bar{x}} := \Sigma(\bar{x}, \bar{y}v)RB[F(...t[w/v]...)]_{\bar{y}v}$

• Let the term belong to category 6 and let $t$ be the argument which is not a variable in $\bar{x}$ or
which contains no free variables which are in $\bar{x}$ and such that all the previous arguments
are variables in $\bar{x}$. Let $v$ be the alphabetically earliest variable not occurring in $A$ or $\bar{x}$.
Let $\bar{x}'$ be the permutation of $\bar{x}$ such that $[F(t_1, ..., t, ..., t_n)]_{\bar{x}'}$ is normalized. Then
$B[F(..., t, ..., )]_{\bar{x}} := \Sigma(\bar{x}, \bar{x}')P_{\bar{y}}B[F(..., v, ...)]_{\bar{y}v}$

• Let the term belong to category 5. Then it is a prime term $[F(t_1, ..., [B]_{\bar{y}}, ..., t_j)]_{\bar{x}}$, where
there is a variable in $\bar{x}$ which is free in $[B]_{\bar{y}}$ and such that all the previous arguments
are variables in $\bar{x}$. Let $\bar{z}$ be the sequence, as they first occur, of all $m$ free variables in $[B]_{\bar{y}}$
which are in $\bar{x}$. Let $v$ be the alphabetically earliest variable not occurring in $A$ or $\bar{x}$. Then
if $\bar{x}'$ is the normalizing permutation of $\bar{x}$ it is of the form $\bar{x}''\bar{z}\bar{z}''$. Let $\bar{w} = \bar{x}''\bar{z}''\bar{z}$.
Then
$B[F(..., [B]_{\bar{y}}, ...)]_{\bar{x}} := \Sigma(\bar{x}, \bar{w})P_{\bar{y}}B[F(..., v, ...)]_{\bar{y}v\bar{z}\bar{z}}$

• Finally let the term be in category 7 and let $\bar{x}'$ be such that $[F(x_1, ..., x_n)]_{\bar{x}'}$ is elementary.
$B[F(x_1, ..., x_n)]_{\bar{x}} := \Sigma(\bar{x}, \bar{x}')[F(x_1, ..., x_n)]_{\bar{x}'}$

• In the case of a variable we have $Bx := x$.

It is easy to check that this decomposition is well-defined and unique modulo renaming of
bound variables. Bealer decomposition transforms an intensional abstract into an expression
(in Polish notation) involving the eight Bealer operations, elementary abstracts and variables.

**Example 4.7** Consider the term $[\neg F(x, [G(x, y)])]_y$. Then
$B[\neg F(x, [G(x, y)])]_y = NB[F(x, [G(x, y)])]_y = NP_0B[F(v, [G(x, y)])]_{yv}x$
$= NP_0IB[F(v, [G(x, y)])]_{yv}x = NP_0IP_1B[F(v, w)]_{vw}B[G(x, y)]_y x$
$= NP_0IP_1B[F(v, w)]_{vw}P_0B[G(u, y)]_{yw}xx = NP_0IP_1B[F(v, w)]_{vw}P_0IB[G(u, y)]_{yw}xx$
The following observation will be used further ahead:

**Lemma 4.8** Let \([F(t_1, ..., t_n)]_\bar{x}\) be a prime term. Then its Bealer decomposition has the form

\[ \Gamma_1 \ldots \Gamma_m[F(v_1, ..., v_n)]_{v_1 \ldots v_n} B t'_1 \ldots B t'_m \]

where the \(\Gamma_i\) consist in sequences of the form

\[ \Sigma_1 R \Sigma_2 R \Sigma_3 \ldots R \Sigma_{l-1} R \Sigma_l P_{n_i} \]

and \(t'_{ik} = s_{\bar{y}}\) where \(s = [A]_{\bar{x}}\) then \(s_{\bar{y}}\) is \([A]_{\bar{y}}\) where \(\bar{y}\) is the sequence of free variables (as they first occur) of \(s\) which are in \(\bar{x}\). If \(s\) is a variable then \(\bar{y}\) is empty and \(t'_{ik} = s\).

5 The Logic of Necessity

We now consider Bealer’s axiomatic system T1 over the language \(L^ω\). The conception of intensionality behind T1 is that there is only one necessary truth and that the intensional abstracts of necessarily equivalent formulas are equal and interchangeable. We define necessitation in terms of equality of intensional abstracts:

\[ \Box A \equiv df [A] = [[A] = [A]] \]

Here \([A] = [A]\) represents the one necessary truth.

A Hilbert system for T1 consists of the following axioms and rules:

1. All propositional tautologies
2. (Inst) \(\forall v. A(v) \rightarrow A(t)\), \(t\) free for \(v\) in \(A\).
3. (QImp) \(\forall x. (A \rightarrow B) \rightarrow (A \rightarrow \forall x.B)\), \(x\) not free in \(A\)
4. (Id) \(x = x\)
5. (L) \(x = y \rightarrow (A(x, x) \leftrightarrow A(x, y))\), \(y\) free for \(x\) in the positions it replaces \(x\).
6. \([A]_{\bar{x}} \neq [B]_{\bar{y}},\) for \(\bar{x}\) and \(\bar{y}\) of different lengths
7. \([A]_{\bar{x}} = [A']_{\bar{x}'}\) (equality modulo renaming bound variables)
8. (B) \([A]_{\bar{x}} = [B]_{\bar{x}} \leftrightarrow \Box \forall \bar{x}. (A \leftrightarrow B)\)
9. (T) \(\Box A \rightarrow A\)
10. (K) \(\Box (A \rightarrow B) \rightarrow (\Box A \rightarrow \Box B)\)
11. (S5) \(\Diamond A \rightarrow \Box \Diamond A\)
12. (MP) From \(A\) and \(A \rightarrow B\) infer \(B\)
13. (N) From \(A\) infer \(\Box A\)
14. (Gen) From \(A\) infer \(\forall x. A\)

Here, as usual, \(\Diamond A = df \neg \Box \neg A\).

In order to prove soundness and completeness it is convenient to replace axioms (L),(B) and (S5) with:
1. (L') \( x = y \rightarrow (A(x, x) \rightarrow A(x, y)) \), \( y \) free for \( x \) in the positions it replaces \( x \), \( A \) atomic.

2. (B1) \( \Box(A \leftrightarrow B) \leftrightarrow [A] = [B] \)

3. (B2) \( \forall v. [A(v)]_{x} = [B(v)]_{x} \leftrightarrow [A(x)]_{xv} = [B(v)]_{xv} \)

4. (S5') \( x \neq y \rightarrow \Box x \neq y \)

The rest of this section is devoted to proving that we obtain equivalent axiomatic systems.

**Lemma 5.1** In T1 we have:

i) \( \vdash x = y \rightarrow \Box x = y \)

ii) \( \vdash \neg \Box x = y \leftrightarrow x \neq y \)

**Proof.** For i), by the identity axiom we have \( x = x \). Hence using the necessitation rule we get

\[ \Box x = x \]

which is

\[ [x = x] = [[x = x] = [x = x]] \]

We denote this expression by \( B(x, x) \). Using \( B(x, x) \) in (L)

\[ x = y \rightarrow A(x, x) \leftrightarrow A(x, y) \]

which is equivalent to \( A(x, x) \rightarrow (x = y \rightarrow A(x, y)) \) we obtain as desired.

For ii), take the converse of the (T) instance \( \Box x = y \rightarrow x = y \) to obtain \( x \neq y \rightarrow \neg \Box x = y \).

For the other direction simply take the converse of i)

**Lemma 5.2** We obtain an equivalent system if in T1 we replace (S5) with

\( (*) \) \( x \neq y \rightarrow \Box x \neq y \).

**Proof.** Assuming \( (*) \), since \( \Diamond A \) is by definition \( \neg([\neg A] = [[\neg A] = [\neg A]]) \) we get (S5) immediately.

In the other direction take the (S5) instance

\[ \neg \Box \neg \neg x = y \rightarrow \Box \neg \neg \neg x = y \]

that is,

\[ (** ) \] \( \neg \Box x = y \rightarrow \Box \neg \Box x = y \)

Now by ii) of the previous lemma we have

\[ \neg \Box x = y \rightarrow \neg x = y \]

Applying necessitation and using (K) we get

\[ \Box \neg \neg \Box x = y \rightarrow \Box \neg x = y \]

and combining with \( (**) \) we get as desired.

We will use the standard properties of S5 modal logic:
Lemma 5.3 In T1 we have
i) $\vdash A \rightarrow \Diamond A$
ii) $\vdash A \rightarrow \Box \Diamond A$
iii) $\Diamond \Box A \rightarrow A$
iv) If $\vdash A \rightarrow B$ then we can derive $\vdash \Diamond A \rightarrow \Diamond B$
v) We have $\vdash \Diamond A \rightarrow B$ iff $\vdash A \rightarrow \Box B$

Proof. For i) take the converse of the (T) instance $\Box \neg A \rightarrow \neg A$. For ii) combine i) and (S5). For iii) take the converse of the (S5) instance $\neg \Box \Diamond \neg A \rightarrow \neg \Diamond \neg A$ and use (T). For iv) take the converse of the premise, apply (T) and take the converse again. Finally v) follows easily from (T), iv), iii) and ii).

Lemma 5.4 $\vdash \Box \forall v.A(v) \rightarrow \forall v.\Box A(v)$

Proof. Take the (Ins) instance

$$\forall v.A(v) \rightarrow A(v)$$

Applying necessitation, (K) and modus ponens yields:

$$\Box \forall v.A(v) \rightarrow \Box A(v)$$

Applying generalization:

$$\forall v.(\Box \forall v.A(v) \rightarrow \Box A(v))$$

Since $v$ is not free on the left side of the implication we can apply (QImp) and modus ponens to obtain

$$\Box \forall v.A(v) \rightarrow \forall v.\Box A(v)$$

Lemma 5.5 We have the Barcan Formula $\vdash \forall v.\Box A(v) \rightarrow \Box \forall v.A(v)$.

Proof. We have by (Ins)

$$\forall v.\Box A \rightarrow \Box A$$

Applying $\Diamond$ yields, by lemma 5.3 iii) and iv):

$$\Diamond \forall v.\Box A \rightarrow \Diamond \Box A \rightarrow A$$

Hence by generalization, (QImp) and modus ponens:

$$\Diamond \forall v.\Box A \rightarrow \forall v.A$$

which again by lemma 5.3 yields

$$\forall v.\Box A \rightarrow \Box \forall v.A$$

We can now prove:
Lemma 5.6 We obtain an equivalent system to $T_1$ if instead of (B) take

$$(B'1) \quad \Box (A \leftrightarrow B) \leftrightarrow ([A] = [B])$$

$$(B'2) \quad \forall v.([A(v)] = [B(v)]) \leftrightarrow [A(v)] = [B(v)]$$

Proof. Using Barcan’s formula and its converse it is easy to see that (B’1) and (B’2) follow from (B).

Assume (B’1) and (B’2). We must show that

$$[A] = [B] \leftrightarrow \Box \forall \bar{x}.(A \leftrightarrow B)$$

Let $\bar{x} = \bar{x}^v$.

Since $[A] = [B]$ follows using (B’2) repeatedly we get

$$[A] = [B] \leftrightarrow \forall \bar{x}.([A] = [B])$$

Using (B’1) we get

$$[A] = [B] \leftrightarrow \forall \bar{x}.\Box (A \leftrightarrow B)$$

Using repeatedly Barcan’s formula and its converse we get as desired.

Finally:

Lemma 5.7 We obtain an equivalent system to $T_1$ when the Leibniz axiom (L) is restricted to atomic predicates and restricted to the form (L’) $x = y \rightarrow (A(x, x) \rightarrow A(x, y))$.

Proof. L obviously implies (L’) as a particular case. In the other direction we first show that $x = y \rightarrow (A(x, x) \rightarrow A(x, y))$. We have that $x = y \rightarrow (A(x, x) \rightarrow A(x, y))$ and $A(x, y)$ arises from $A(x, x)$ by replacing some occurrences of $x$ by $y$. So $A(x, x)$ arises from replacing some occurrences of $y$ in $A(x, y)$ by $x$. Hence we have $y = x \rightarrow (A(x, y) \rightarrow A(x, x))$. Also from (L’) we can derive $x = y \rightarrow y = x$ and using this we get as desired. We can now proceed inductively on the structure of the formula $A$. The case of negation is immediate. For conjunction we use the tautology $(A \rightarrow B) \rightarrow (C \rightarrow D) \rightarrow (A \& C \rightarrow B \& D)$ and for quantification we use generalisation, (QImp) and modus ponens.

This lemma justifies that "there is only one necessary truth".

Lemma 5.8 We have $\forall x, y. [x = x] = [y = y]$.

Proof. We have $x = x \leftrightarrow y = y$. We apply necessitation and (B’1) and then generalisation to get as desired.
6 The Logic of Definition

The logic T1 is insufficient in many instances.

Lemma 6.1 In T1 if $\vdash W(x, [\exists y(T(y) \land \neg S(y))])$ and $\vdash \Box \forall y(S(y) \leftrightarrow T(y))$ then

$\vdash W(x, [\exists y(T(y) \land \neg T(y))]).$

Proof. We only need to show that $[\exists y(T(y) \land \neg S(y))] = [\exists y(T(y) \land \neg T(y))]$, that is, by (B), $\Box([\exists y(T(y) \land \neg S(y)) \leftrightarrow \exists y(T(y) \land \neg T(y))]$. From the second hypothesis we get $\vdash S(y) \leftrightarrow T(y)$ and using the tautology:

$$(T(y) \leftrightarrow S(y)) \rightarrow ((T(y) \land \neg S(y)) \leftrightarrow (T(y) \land \neg T(y)))$$

we get easily

$$\forall y(T(y) \leftrightarrow S(y)) \rightarrow \exists y((T(y) \land \neg S(y)) \leftrightarrow (T(y) \land \neg T(y)))$$

and the result follows from necessitation, (K), modus ponens and (B).

To avoid this in our intensional logic we accept only one implication in axiom (B). Bealer’s logic T2 is given the following Hilbert axiomatic system:

1. All propositional tautologies
2. (Ins) $\forall x. A(x) \rightarrow A(t)$, $t$ free for $v$ in $A$.
3. (QImp) $\forall x. (A \rightarrow B) \rightarrow (A \rightarrow \forall x.B)$, $x$ not free in $A$
4. (Id) $x = x$
5. (L) $x = y \rightarrow (A(x, x) \leftrightarrow A(x, y))$, $y$ free for $x$ in the positions it replaces $x$.
6. $[A]_x \neq [B]_y$, $\bar{x}$ and $\bar{y}$ of different lengths
7. $[A]_x = [A']_{x'}$ (equality modulo renaming bound variables)
8. $[A]_x = [B]_x \rightarrow (A \leftrightarrow B)$
9. $[A]_x \neq [B]_y$, non-elementary terms belonging to different categories.
10. Let $t' = Bt$ and $r' = Br$ where $B$ is the first operation in the Bealer decomposition of $t'$ and $r'$. Then $t = r \leftrightarrow t' = r'$
11. Let $t = Bt'r'$ and $r = Bt''r''$ where $B$ is the first operation in the Bealer decomposition of $t$ and $r$. The $t = r \leftrightarrow t' = t'' \land r' = r''$.
12. If $[F(x_1, ..., x_n)]_{x_1...x_n} = s$ and $G$ occurs in $s$ then $[G(y_1, ..., y_m)]_{y_1...y_m} \neq t$ whenever $F$ occurs in $t$ (Non-circularity)

1. (MP) From $A$ and $A \rightarrow B$ infer $B$
2. (TGen) Suppose that $F$ does not occur in $A(v)$. If $\vdash A([F(x_1, ..., x_n)]_{x_1...x_n})$ then we can infer $\vdash A(t)$ whenever $t$ is complex, of arity $n$ and free for $v$ in $A$.
3. (Gen) From $A$ infer $\forall x. A$


7 Model Structures

Let \( F = \emptyset \) and \( T = \{ \emptyset \} \). A model \( \mathcal{M} \) consists of a set \( \mathcal{D} \) with a decomposition into the union of disjoint sets \( \mathcal{D}_i \) for \( i \geq -1 \), a distinguished element \( id \in \mathcal{D}_2 \), a set \( \mathcal{H} \) of functions \( H \) whose domain is \( \mathcal{D} \) and such that \( H \) is the identify on \( \mathcal{D}_{-1} \) and on \( \mathcal{D}_0 \) has range \( \{ T, F \} \) and on \( \mathcal{D}_i \) has range \( \mathcal{P}(\mathcal{D}^i) \) where \( \mathcal{D}^i = \mathcal{D} \times \ldots \times \mathcal{D} \) (\( i \) terms) for \( i \geq 1 \). All these functions must satisfy \( H(id) = \{ (x, y) : x = y \} \). There is a distinguished function \( G \) representing the actual extension. Furthermore we have unary (partial) operations \( n, e, u, c, i, r \) and binary (partial) operations \( k, p_n \) with \( n \geq 0 \). These operations restrict to the following domains and ranges:

\[
\begin{align*}
n &: \mathcal{D}_i \rightarrow \mathcal{D}_i, i \geq 0 \\
e &: \mathcal{D}_i \rightarrow \mathcal{D}_{i+1}, i \geq 0 \\
u &: \mathcal{D}_i \rightarrow \mathcal{D}_{i-1}, i \geq 1 \text{ and } u : \mathcal{D}_0 \rightarrow \mathcal{D}_0. \\
c &: \mathcal{D}_i \rightarrow \mathcal{D}_i, i \geq 3 \\
i &: \mathcal{D}_i \rightarrow \mathcal{D}_i, i \geq 2 \\
r &: \mathcal{D}_i \rightarrow \mathcal{D}_{i-1}, i \geq 2 \\
a &: \mathcal{D}_i \times \mathcal{D}_j \rightarrow \mathcal{D}_i, i \geq 0 \\
p_n &: \mathcal{D}_i \times \mathcal{D}_j \rightarrow \mathcal{D}_{i+j-1}, i \geq 1, j \geq n
\end{align*}
\]

We define inductively \( p_0^n(d, x_1, \ldots, x_n) = p_0(p_0^{n-1}(d, x_2, \ldots, x_{n-1}), x_1) \) and \( p_0^0(d, x_1) = p_0(d, x_1) \).

Furthermore we have the important constraints on how the functions in \( \mathcal{H} \) relate to these operations. Here \( d \) denotes a suitable member of \( \mathcal{D}^i \). For all \( \mathcal{H} \in H \) we must have:

\[
\begin{align*}
d' \in \mathcal{H}n(d) \text{ iff } d' \notin \mathcal{H}d \\
(x_1, \ldots, x_i, x_{i+1}) \in \mathcal{H}e(d) \text{ iff } (x_1, \ldots, x_i) \in \mathcal{H}d \\
(x_1, \ldots, x_{i-1}, x_i) \in \mathcal{H}u(d) \text{ iff there is a } x_i \text{ such that } (x_1, \ldots, x_i) \in \mathcal{H}d \\
(x_1, \ldots, x_{i-1}, x_i) \in \mathcal{H}c(d) \text{ iff } (x_i, x_1, \ldots, x_{i-1}) \in \mathcal{H}d \\
(x_1, \ldots, x_{i-1}, x_i) \in \mathcal{H}i(d) \text{ iff } (x_1, \ldots, x_i, x_{i-1}) \in \mathcal{H}d \\
(x_1, \ldots, x_{i-1}, x_i) \in \mathcal{H}r(d) \text{ iff } (x_1, \ldots, x_i, x_{i-1}) \in \mathcal{H}d \\
d'' \in \mathcal{H}k(d, d') \text{ iff } d'' \in \mathcal{H}d \text{ and } d'' \in \mathcal{H}d' \\
(x_1, \ldots, x_n) \in \mathcal{H}p_0(d, d') \text{ iff } (x_1, \ldots, x_n, d') \in \mathcal{H}d \\
(x_1, \ldots, x_{i-1}, y_1, \ldots, y_n) \in \mathcal{H}p_n(d, d') \text{ iff } (x_1, \ldots, x_{i-1}, p_n^0(d', y_1, \ldots, y_n)) \in \mathcal{H}d
\end{align*}
\]

Here we consider also the 0-tuple as being \( \emptyset \). Notice that elements of \( \mathcal{D}^0 \) are propositions seen as objective entities according to their meaning. The extension functions \( \mathcal{H} \) determine their Boolean values which are \( T = \{ \emptyset \} \) and \( F = \emptyset \).

**Definition 7.1** Given a model \( \mathcal{M} \), an interpretation \( \mathcal{I} \) assigns \( i \)-ary predicate elements of \( \mathcal{D}_i \) and \( = \) to \( id \). An assignment \( \mathcal{A} \) assigns to variables elements in \( \mathcal{D} \).

Bealer decomposition is crucial to define the denotation of a term relative to a model, interpretation and assignment.

**Definition 7.2** Given \( \mathcal{M}, \mathcal{I} \) and \( \mathcal{A} \) we define the denotation \( D_{\mathcal{I}, \mathcal{A}, \mathcal{M}}t \) of a term \( t \) of \( L_\omega \) as follows. If \( t \) is a variable then it is \( \mathcal{A}(t) \). If \( t \) is an elementary term \( \mathcal{F}(v_1, \ldots, v_n)_{v_1 \ldots v_n} \) then it is \( \mathcal{I}(F) \). Otherwise, consider the Bealer decomposition \( Bt \) of \( t \). If \( Bt = Bt' \) (respectively \( Bt't'' \)) where \( B \) is a Bealer operation then we define inductively

\[
D_{\mathcal{I}, \mathcal{A}, \mathcal{M}}t = bD_{\mathcal{I}, \mathcal{A}, \mathcal{M}}t' \quad \text{(respectively } D_{\mathcal{I}, \mathcal{A}, \mathcal{M}}t = bD_{\mathcal{I}, \mathcal{A}, \mathcal{M}}t'D_{\mathcal{I}, \mathcal{A}, \mathcal{M}}t'')
\]

where \( b \) is \( i, c, e, u, n, r, k, p_n \) if \( B \) is \( I, C, E, U, N, R, K, P_n \) respectively.
Definition 7.3 We say that a formula $A$ is true for $\mathcal{M}, \mathcal{I}$ and $\mathcal{A}$ (we also write $T_{\mathcal{I}, \mathcal{A}, \mathcal{M}}(A)$) if $GD_{\mathcal{I}, \mathcal{A}, \mathcal{M}}[A] = T$.

It is simple to see that the denotation is invariant modulo renaming bound variables. From now one, we will in most cases drop the subscripts in $D_{\mathcal{I}, \mathcal{A}, \mathcal{M}}$ and assume we are working with a given model, interpretation and assignment.

We are now interested in special classes of models for which $T_1$ and $T_2$ are sound and complete.

Definition 7.4 A model $\mathcal{M}$ is called type 1 if for all $i \geq -1$, $x, y \in D_i$ (type 1 model condition) $\forall \mathcal{H} \in H \quad \mathcal{H}(x) = \mathcal{H}(x) \to x = y$

A model $\mathcal{M}$ is called type 2 if the operations are:

1. one-to-one
2. disjoint on their ranges
3. and non-cycling

By the first two conditions each element of the model has a unique (possibly infinite) decomposition in terms of the operations. This decomposition is a tree having leaves indecomposable elements. The non-cycling condition means that the same element cannot appear in more than one place in the tree.

A formula $A$ is $T_1$-valid (we write $\vdash_{T_1} A$) iff $A$ is true for all type 1 model $\mathcal{M}$, interpretation $\mathcal{I}$ and assignment $\mathcal{A}$. In the same way a formula $A$ is $T_2$-valid ($\vdash_{T_2} A$) iff $A$ is true for all type 2 model $\mathcal{M}$, interpretation $\mathcal{I}$ and assignment $\mathcal{A}$.

Note that for type 2 models only $\mathcal{G}$ is relevant. The type 1 condition restricts the possible operations on the model.

Example 7.5 A standard model for Kelley-Morse set theory furnishes an example of a type 1 model with a single extensional function. Type 1 models can be seen as generalised set theory models with a plurality of membership predicates $\in_H$ such that the axiom of extensionality is only valid globally. Consider a language with a single unary predicate $M$. We can construct a type 2 model with no constants, $D_0, D_1$ and $D_i = \emptyset$ for $i \geq 2$.

Lemma 7.6 Let $v$ be an externally quantifiable variable in $[B(v)]_x$ and let $t$ be free for $v$ in $[B(v)]_x$. Consider any model structure $\mathcal{M}$ and any interpretation $\mathcal{I}$ and assignment $\mathcal{A}$. Let $\mathcal{A}'$ be an assignment which is just like $\mathcal{A}$ except that $\mathcal{A}'(v) = Dt$. Then

$$D_{\mathcal{A}'}[B(v)]_x = D[B(t)]_x$$

Proof. By induction on the Bealer decomposition of $[B(v)]_x$. The base case is either an elementary abstract (in which case there are no externally quantifiable variables) or else a variable. If it is not $v$ we are done and if it is $v$ the result is trivial.

Suppose that $[B(v)]_x = Bs(v)$ were $B$ is some unary operation. If $B$ is $I$, $C$, $N$, $U$ or $E$ then $t$ is still free for $v$ in $d(v)$ and hence by the induction hypothesis $D_{\mathcal{A}'} s(v) = Ds(t)$ so that $D_{\mathcal{A}'}[B(v)]_x = bD_{\mathcal{A}'} s(v) = bDs(t) = DBs(t)$ where $b$ is the corresponding model function. If $B$ is $R$ then there is a problem that $s$ has a new bound variable which may occur in $t$ in such a way that $t$ is no longer free for $v$ in $s$. But we may rename this bound variable to obtain
an $\alpha$-equivalent term $s'$ where this problem does not occur. Note that the substitution does not affect which of the seven syntactic categories an intensional abstract belongs to. Suppose $[B(v)]_{\bar{x}} = Bs(v)s'(v)$ for a binary operation $B$. If $B$ is $A$ then $t$ remains free for $v$ in $s(v)$ and $s'(v)$ and the previous argument applies. If $B$ is $P_k$ then we have the problem of the new bound variable in the first argument possibly occurring in $t$. We thus need to use $\alpha$-equivalence so $t$ remains free for $v$ in $s(v)$. In the case of 0-predication we have

$$D_{A'}[B(v)]_{\bar{x}} = p_0(D_{A'}[B'(v)]_{\bar{x}w}, D_{A'}s'(v))$$

If $s'$ is not a variable or a variable distinct from $v$, then we may apply the induction hypothesis to obtain:

$$p_0(D_{A'}[B'(v)]_{\bar{x}w}, D_{A'}s'(v)) = p_0(D[B'(t)]_{\bar{x}w}, Ds'(t)) = D[B(t)]_{\bar{x}}$$

If $s'$ is $v$ then we obtain

$$p_0(D_{A'}[B'(v)]_{\bar{x}w}, D_{A'}v) = p_0(D_{A'}[B'(v)]_{\bar{x}w}, A'v) = p_0(D[B'(t)]_{\bar{x}w}, Dt) = D[B(t)]_{\bar{x}}$$

In the $k$-predication case for $k > 0$ we have, for $s'(v) = [C(v)]_{\bar{y}\bar{z}}$ where $\bar{z}$ are relativized variables of $[C(v)]_{\bar{y}}$ in $[B(v)]_{\bar{x}}$

$$D_{A'}[B(v)]_{\bar{x}} = p_k(D_{A'}[B'(v)]_{\bar{x}w}, D_{A'}[C(v)]_{\bar{y}\bar{z}})$$

where $\bar{x}'$ is obtained from $\bar{x}$ by omitting $\bar{z}$. The result now follows easily from induction.

**Lemma 7.7** For all $M$ and $I$, $A$ we have $D[A]_{x_1...x_k} \in D_k$

**Proof.** This follows by easy induction from the definition of $D$ and the model operations.

Given a sequence of Bealer operations of the form $\Sigma$ we denote its associated $n$-permutation by $\sigma$. This permutation can also be applied to $n$-tuples $(x_1, ..., x_n)$ and we denote such an application by $\sigma(x_1, ..., x_n)$ just as we denote its application to variable sequences by $\sigma x_1...x_n$.

We will make frequent use of the following:

**Lemma 7.8** For any permutation $\sigma$ of $(x_1, ..., x_n)$ we have

$$\sigma(v_1, ..., v_n) \in HD[F]_{x_1...x_n} \leftrightarrow (v_1, ..., v_n) \in HD[F]_{x_1...x_n}$$

**Proof.** Let the Bealer decomposition of $[F]_{x_1...x_n}$ be $\Sigma_1 B[F]_{x_1'...x_n'}$, where $B[F]_{x_1'...x_n'}$ does not begin with $C$ or $I$. Then Bealer decomposition of $[F]_{x_1...x_n}$ will be of the form $\Sigma_2 B[F]_{x_1'...x_n'}$. By the definition of denotation and model we have for any $(x_1, ..., x_n)$:

$$(x_1, ..., x_n) \in HD[F]_{x_1...x_n} \text{ iff } \sigma_1(x_1, ..., x_n) \in HD[F]_{x_1'...x_n'}$$

$$(x_1, ..., x_n) \in HD[F]_{\sigma x_1...x_n} \text{ iff } \sigma_2(x_1, ..., x_n) \in HD[F]_{x_1'...x_n'}$$

where $x_1'...x_n' = \sigma_1(x_1, ..., x_n)$ and $x_1'...x_n' = \sigma_2(x_1, ..., x_n)$ and hence $\sigma = \sigma_2^{-1}\sigma_1$ and $\sigma_1 = \sigma_2\sigma$.

Hence given $(x_1, ..., x_n)$ we have $(x_1, ..., x_n) \in HD[F]_{x_1...x_n}$ iff $\sigma_1(x_1, ..., x_n) = \sigma_2 \sigma(x_1, ..., x_n) \in HD[F]_{x_1'...x_n'}$ iff $\sigma(x_1, ..., x_n) \in HD[F]_{\sigma x_1...x_n}$. 

19
Lemma 7.9 For all $\mathcal{I}, \mathcal{A}, \mathcal{M}$, $F(t_1, ..., t_n)$ is true iff $(Dt_1, ..., Dt_n) \in \mathcal{I}(F)$.

Proof. By definition $F(t_1, ..., t_n)$ is true iff $GD([F(t_1, ..., t_n)]) = T$. But

$$[F(t_1, ..., t_n)] = P_0[F(v_1, ..., t_n)]v_1 t_1 = P_0 P_p[F(v_1, v_2, ..., t_n)]v_1 v_2 t_2 t_1 = P_0^n [F(v_1, ..., v_n)]v_1 ... v_n t_n ... t_2 t_1$$

hence by definition $GD([F(t_1, ..., t_n)]) = T$ iff

$$Dt_1 \in Gp_0(p_0(...p_0(D[F(v_1, ..., v_n)]v_1 ..., v_n), Dt_n), ..., Dt_3), Dt_2)$$

iff

$$(Dt_1, Dt_2) \in Gp_0(p_0(...p_0(D[F(v_1, ..., v_n)]v_1 ..., v_n), Dt_n), ..., Dt_3)$$

and so on until obtaining

$$(Dt_1, ..., Dt_n) \in D[F(v_1, ..., v_n)]v_1 ... v_n = \mathcal{I}(F)$$

Lemma 7.10 $T_{\mathcal{I}\mathcal{A}\mathcal{M}}(A \& B)$ iff $T_{\mathcal{I}\mathcal{A}\mathcal{M}}(A)$ and $T_{\mathcal{I}\mathcal{A}\mathcal{M}}(B)$. Also $T_{\mathcal{I}\mathcal{A}\mathcal{M}}(\neg A)$ iff it is not the case that $T_{\mathcal{I}\mathcal{A}\mathcal{M}}(A)$.

This is an immediate consequence of the definition of $T_{\mathcal{I}\mathcal{A}\mathcal{M}}$.

8 Soundness of T1

In this section we work in a type 1 model $\mathcal{M}$. We will use Polish notation for the model operations and omit parenthesis when possible. We note first that Lemma 7.9 (for the case of equality) and Lemma 7.10 yields immediately:

Lemma 8.1 For all models we have propositional tautologies, (MP), (Id) and (L’) are sound.

It is immediate by lemma 7.7 that axiom 6 is sound and the soundness of axiom 7 is obvious.

The following is of central importance

Lemma 8.2 (Bealer’s lemma) Let $v$ be free in $[A(v)]_{\vec{x}}$. Then for all $\mathcal{M}$ of type 1 and $\mathcal{I}$, $\mathcal{A}$ be an interpretation and assignment. Then

$$D[A(v)]_{\vec{x}} = p_0 D[A(v)]_{\vec{x}v} A(v)$$

Note that when we write $(x_1, ..., x_n)$ we include the case $n = 0$ in which case the sequence is taken to be the empty set $\emptyset$.

Proof. We proceed by induction on the Bealer decomposition of $[A(v)]_{\vec{x}}$.

Let $B[A(v)]_{\vec{x}} = KB[B(v)]_{\vec{b}} B[C(v)]_{\vec{b}}$. Then

$$D[A(v)]_{\vec{x}} = kD[B(v)]_{\vec{x}} D[C(v)]_{\vec{x}} = kp_0 D[B(v)]_{\vec{x}v} A(v) p_0 D[C(v)]_{\vec{x}v} A(v)$$

by the induction hypothesis. We use the type 1 condition to show that

$$kp_0 D[B(v)]_{\vec{x}v} A(v) p_0 D[C(v)]_{\vec{x}v} A(v) = p_0 D[A(v)]_{\vec{x}v} A(v)$$

For any $\mathcal{H}$ let $(x_1, ..., x_n) \in \mathcal{H} kp_0 D[B(v)]_{\vec{x}v} A(v) p_0 D[C(v)]_{\vec{x}v} A(v)$. Then this is equivalent to

$$(x_1, ..., x_n) \in \mathcal{H} p_0 D[B(v)]_{\vec{x}v} A(v)$$

and $(x_1, ..., x_n) \in \mathcal{H} p_0 D[C(v)]_{\vec{x}v} A(v)$$
which is equivalent to

\[(x_1, ..., x_n, A(v)) \in \mathcal{H}D[B(v)]_{x_v} \text{ and } (x_1, ..., x_n, A(v)) \in \mathcal{H}D[C(v)]_{x_v}\]

The Bealer decomposition of \([A(v)]_{x_v}\) is \(\Sigma K[B(v)]_{x'_v}[C(v)]_{x'_v}\) were \(\Sigma\) is such that \(\bar{x}' = \sigma \bar{x}v\) is the normalised permutation of \(\bar{x}v\) for \([A(v)]_{x_v}\). But then the above condition is equivalent to

\[\sigma(x_1, ..., x_n, A(v)) \in \mathcal{H}D[B(v)]_{x'_v} \text{ and } \sigma(x_1, ..., x_n, A(v)) \in \mathcal{H}D[C(v)]_{x'_v}\]

which is equivalent to \(\sigma(x_1, ..., x_n, A(v)) \in \mathcal{H}kDB[(v)]_{x'_v}DC[(v)]_{x'_v}\). Let \(s\) be the composition of operations corresponding to \(\Sigma\). Then the previous condition is equivalent to \((x_1, ..., x_n, A(v)) \in \mathcal{H}skDB[(v)]_{x'_v}DC[(v)]_{x'_v} = \mathcal{H}D[A(v)]_{x_v}\). But this is equivalent to \((x_1, ..., x_n) \in \mathcal{H}p_0D[A(v)]_{x_v}A(v)\). Hence the conclusion follows from the type 1 condition.

Let \(\mathcal{B}[A(v)]_{x} = NB[A'(v)]_{x}\) where \(A(v) = \neg A'(v)\). We have \(D[A(v)]_{x} = nD[A'(v)]_{x} = np_0D[A'(v)]_{x_v}A(v)\) by the induction hypothesis. We use the type 1 condition to show that

\[np_0D[A'(v)]_{x_v}A(v) = p_0D[A(v)]_{x_v}A(v)\]

Then for any \(\mathcal{H}\),

\[(x_1, ..., x_n) \in \mathcal{H}np_0D[A'(v)]_{x_v}A(v) \iff (x_1, ..., x_n) \notin \mathcal{H}p_0D[A'(v)]_{x_v}A(v)\]

iff \((x_1, ..., x_n, A(v)) \notin \mathcal{H}A'(v)]_{x_v}\)

Let \(\bar{x}'\) be the normalised sequence for \([A'(v)]_{\bar{x}_v}\) and \(\sigma\) the associated permutation. Then the above condition holds iff

\[\sigma(x_1, ..., x_n, A(v)) \notin \mathcal{H}A'(v)]_{x_v} \iff \sigma(x_1, ..., x_n, A(v)) \in \mathcal{H}nA'(v)]_{x_v}\]

The Bealer decomposition of \([A(v)]_{x_v}\) is \(\Sigma N[A'(v)]_{x_v}\) where \(\sigma\) corresponds to \(\Sigma\) so \(D[A(v)]_{x_v} = snD[A'(v)]_{x_v}\) where \(s\) is the composition of model operation corresponding to \(\Sigma\). Hence we have \(\sigma(x_1, ..., x_n, A(v)) \in \mathcal{H}nA'(v)]_{x_v} \iff (x_1, ..., x_n, A(v)) \in \mathcal{H}D[A(v)]_{x_v} \iff (x_1, ..., x_n) \in \mathcal{H}p_0D[A(v)]_{x_v}A(v)\) and our conclusion follows from the type 1 model condition.

Let \(\mathcal{B}[A(v)]_{x} = U[A'(v)]_{x_w}\) where \(A(v) = \exists w. A'(v)\). Then \(D[A(v)]_{x} = uD[A'(v)]_{x_w} = up_0D[A'(v)]_{x_vw}A(v)\) by induction. We show that \(up_0D[A'(v)]_{x_vw}A(v) = p_0[A(v)]_{x_v}A(v)\). Take a \(\mathcal{H}\).

\[(x_1, ..., x_n) \in \mathcal{H}up_0D[A'(v)]_{x_vw}A(v) \iff \text{there is } y \text{ such that } (x_1, ..., x_n, y) \in \mathcal{H}p_0D[A'(v)]_{x_vw}A(v)\]

iff there is a \(y\) such that \((x_1, ..., x_n, y, A(v)) \in \mathcal{H}D[A'(v)]_{x_vw}\)

As previously let \(\bar{x}' = \sigma \bar{x}v\) be the normalized sequence for \([A(v)]_{x_v}\). The Bealer decomposition of \([A(v)]_{x_v}\) is \(\Sigma UB[A'(v)]_{x_vw}\) so \(p_0[A(v)]_{x_v}A(v) = p_0suD[A'(v)]_{x_vw}A(v)\). The above condition is equivalent to

there is a \(y\) such that \((x'_1, ..., x'_{n+1}, y) \in \mathcal{H}D[A'(v)]_{x_vw}\)

where \(A(v)\) occupies the position \(x'_i\) of \(v\) in \(\bar{x}'\). But this is equivalent to

\[(x'_1, ..., x'_{n+1}) \in \mathcal{H}uD[A'(v)]_{x'_v}A(v)\]

which is equivalent to
\[(x_1, \ldots, x_n, \mathcal{A}(v)) \in \mathcal{H}suD[A'(v)]_{x^rv} \text{ iff } (x_1, \ldots, x_n) \in \mathcal{H}p_0suD[A'(v)]_{x^rv}\mathcal{A}(v) = \mathcal{H}p_0D[A(v)]_{x^rv}\mathcal{A}(v)\]

and the result follows from the type 1 model condition.

Let \(\mathcal{B}[A(v)]_{x^r} = \mathcal{E}\mathcal{B}[A(v)]_{x^r}\) with \(x^r = x^ry\) and \(y\) does not occur free in \(A(v)\). We have \(D[A(v)]_{x^r} = eD[A(v)]_{x^r} = ep_0[A(v)]_{x^rv}\mathcal{A}(v)\) by induction. As previously take a \(\mathcal{H}\). Then

\[(x_1, \ldots, x_n) \in \mathcal{H}p_0[A(v)]_{x^rv}\mathcal{A}(v) \text{ iff } (x_1, \ldots, x_{n-1}) \in \mathcal{H}p_0[A(v)]_{x^rv}\mathcal{A}(v)\]

iff \((x_1, \ldots, x_{n-1}, \mathcal{A}(v)) \in \mathcal{H}[A(v)]_{x^rv}\)

The Bejler decomposition of \([A(v)]_{x^rv}\) is \(\Sigma\mathcal{B}[A(v)]_{x^r} = \Sigma\mathcal{EB}[A(v)]_{x^r}\) where \(\bar{z} = \sigma\bar{x}v\) is the normalizing permutation which puts \(v\) in its proper place and \(\bar{z}'\) drops the last element of \(\bar{z}\) (not \(v\) and the same as \(x_n\)). Put \(\bar{z}' = \sigma'\bar{x}'v\) and let \(\sigma'\) correspond to composition of model operations \(s'\). Then the above condition is equivalent to

\[(x_1, \ldots, x_{n-1}, \mathcal{A}(v)) \in \mathcal{H}s'D[A(v)]_{\sigma'x^rv} = \mathcal{H}s'D[A(v)]_{x^rv}\]

which is equivalent to

\[(x_1, \ldots, x_{n-1}, x_n, \mathcal{A}(v)) \in \mathcal{H}ies'D[A(v)]_{x^rv} = \mathcal{H}seD[A(v)]_{x^rv}\]

which is turn is equivalent to:

\[(x_1, \ldots, x_n) \in \mathcal{H}p_0seD[A(v)]_{x^rv}\mathcal{A}(v) = \mathcal{H}p_0D[A(v)]_{x^rv}\mathcal{A}(v)\]

and the result follows as in the other cases.

Let \(\mathcal{B}[A(v)]_{x^r} = \mathcal{BB}[A(v)]_{x^r}\) with \(B\) equal to \(C\) or \(I\) where \(\bar{x}' = \sigma_b\bar{x}\) for \(b\) equal to \(c\) or \(i\). Then \(D[A(v)]_{x^r} = bD[A(v)]_{x^rv} = bp_0[A(v)]_{x^rv}\mathcal{A}(v)\) using the induction hypothesis. We use the type 1 condition to show that

\[bp_0[A(v)]_{x^rv}\mathcal{A}(v) = Dp_0[A(v)]_{x^rv}\mathcal{A}(v)\]

For any \(\mathcal{H}\) we have

\[(x_1, \ldots, x_n) \in \mathcal{H}b\mathcal{p}_0[A(v)]_{x^rv}\mathcal{A}(v) \text{ iff } (x_1', \ldots, x_n') \in \mathcal{H}\mathcal{p}_0[A(v)]_{x^rv}\mathcal{A}(v)\]

where \((x_1, \ldots, x_n) = \sigma_b(x_1, \ldots, x_n)\). But

\[(x_1', \ldots, x_n') \in \mathcal{H}\mathcal{p}_0[A(v)]_{x^rv}\mathcal{A}(v) \text{ iff } (x_1', \ldots, x_n', \mathcal{A}(v)) \in \mathcal{H}[A(v)]_{x^rv}\]

iff \((x_1, \ldots, x_n, \mathcal{A}(v)) \in \mathcal{H}[A(v)]_{x^rv} \text{ iff } (x_1, \ldots, x_n) \in \mathcal{H}\mathcal{p}_0[A(v)]_{x^rv}\mathcal{A}(v)\)

Hence by the type 1 model condition we get that \(bp_0[A(v)]_{x^rv}\mathcal{A}(v) = Dp_0[A(v)]_{x^rv}\mathcal{A}(v)\).

Now let \(\mathcal{B}[A(v)]_{x^r} = \mathcal{R}\mathcal{B}[A'(v)]_{x^rv}\). Then \([A(v)]_{x^r}\) has a reflected variable and hence so does \([A(v)]_{x^rv}\) (it is also belongs to category 4 and is a prime reflection term). We distinguish between two cases. Either \(v\) is the right-most reflected variable or it is not. Assume that is is not. Let \(t\) be the right-most argument in which \(t\) occurs. Then the right-most reflected variable is the last element \(x_n\) of \(x\). We have \(D[A(v)]_{x^r} = \mathcal{R}\mathcal{p}_0D[A'(v)]_{x^rv}\mathcal{A}(v)\) by induction, were \(A'(v)\) is obtained from \(A(v)\) by replacing the argument \(t\) with \(t[w/x_n]\). Fix \(\mathcal{H}\). Then

\[(x_1, \ldots, x_n) \in \mathcal{H}\mathcal{R}\mathcal{p}_0D[A'(v)]_{x^rv}\mathcal{A}(v) \text{ iff } (x_1, \ldots, x_n, \mathcal{A}(v)) \in \mathcal{H}D'[A(v)]_{x^rv}\)

22
Now the Bealer decomposition of $[A(v)]_{x_v}$ in this case will be $\Sigma R B[A'(v)]_{\bar{x'} w}$. Here $\bar{x'} = \sigma \bar{x''}$, $\bar{x''}$ moves $v$ to its proper place so that $\bar{x''}$ is the normalized permutation of $\bar{x'}$ except that $x_n$ is placed at the end, being the reflected variable with right-most occurrence. We have $D[A(v)]_{x_v} = sr D[A'(v)]_{\bar{x'} w}$. Then the condition above is equivalent to:

$$(x'_1, ..., x'_{n+1}, x'_{n+1}) \in H D[A'(v)]_{\bar{x'} w}$$

with $x'_{n+1} = x_n$. But this is equivalent to:

$$(x'_1, ..., x'_{n+1}) \in H r D[A'(v)]_{\bar{x'} w}$$

which is turn is equivalent to

$$(x_1, ..., x_n, A(v)) \in H sr D[A'(v)]_{\bar{x'} w}$$

which is finally equivalent to

$$(x_1, ..., x_n) \in H p_0 s r D[A'(v)]_{x_w} A(v) = H p_0 D[A(v)]_{x_w} A(v)$$

and the result follows as in the other cases. Now for the second case: $v$ is a reflected variable with the right-most occurrence. Let the right-most argument in which it occurs be $A$ from $x$ variable or else it is so that $\bar{x}$ notation) but has new variables $t$ right-most argument in which it occurs replaced with $A$ from $x$ variable or else it is so that $\bar{x}$ notation but has $x$ new variables but with $k$ omitted. Hence

$$(x'_1, ..., x'_{n+k+1}) \in H D[A^{(k)}(v)]_{\bar{x'} w_k}$$

where $\bar{x'}$ (normalized except for $w_k$ being placed at the end) contains the elements of $\bar{x'}, v$ and new variables $w_1, ..., w_{k-1}$. Here $[A^{(k)}(v)]_{\bar{x'} w_k}$ is now as the previous case (modulo a permutation).

It is easy to see that

$$(x_1, ..., x_n) \in H p_0 s_1 r s_2 r ... s_k R D[A^{(k)}(v)]_{\bar{x'} w_k} A(v)$$

iff

$$(x'_1, ..., x'_{n+k+1}) \in H D[A^{(k)}(v)]_{\bar{x'} w_k}$$

Here $(x'_1, ..., x'_{n+k+1})$ follows the sequence of $\bar{x'} w_k$, having $x_i$ for $x_i$ (here we are abusing notation) but has $x'_k = A(v)$ whenever $x'_k$ is a $w_i$ or $v$.

Now $D[A^{(k)}(v)]_{\bar{x'} w_k} = s D[A^{(k)}(v)]_{\bar{x'} w_k} = s r D[A^{(k)}(v)]_{\bar{x'} w_k} w_k$ where $\bar{u}$ is the normalized permutation of $\bar{x'} w_k$ but with $x_n$ omitted. Hence

$$(^*) (x'_1, ..., x'_{n+k+1}) \in H D[A^{(k)}(v)]_{\bar{x'} w_k} \text{ iff } (x''_1, ..., x''_{n+k}, x_n, x_n) \in H D[A^{(k)}(v)]_{\bar{x'} w_k} w_k$$

where the $x''_i$ follows the normalized sequence but with $x_n$ omitted and $A^{(k)}(v)$ has the right-most argument $t$ in which $x_n$ occurs replaced with $t[w/x_n]$. Now we have by induction,

$$(x_1, ..., x_n) \in H D[A(v)]_{x} \text{ iff } (x_1, ..., x_n) \in H r p_0 [A_1(v)]_{x_w} A(v)$$

iff $$(x_1, ..., x_n, A(v)) \in H D[A_1(v)]_{x_w}$$

23
where $A_1$ has the right-most argument $t$ in which $x_n$ appears replaced by $t[w/x_n]$. Now $[A_1(v)]_{xwv}$ has $v$ reflected in the same arguments and positions as $[A(v)]_x$. Hence

$$[A_1(v)]_{xwv} = \Sigma'_1 R \Sigma'_2 R \ldots \Sigma'_k R B [A_1^{(k)}(v)]_{\bar{q}w_k}$$

Hence

$$(x_1, ..., x_n, x_n, A(v)) \in \mathcal{H} D [A_1(v)]_{xwv} \text{ iff } (q_1, ..., q_{n+k+2}) \in \mathcal{H} D [A_1^{(k)}(v)]_{\bar{q}w_k}$$

Here $\bar{q}$ is the normalized permutation of $\bar{x}wv$ except that $w_k$ is moved to the empty sequence. It has new variables $w_1, ..., w_{k+1}$, $(q_1, ..., q_{n+k+2})$ follows this sequence and for $v$ or $w_i$ it has $A$ and for $w$ it has $x_n$. It is clear that $A_1^{(k)}(v)$ is the same as $A^{(k)}_0(v)$. Examining (*) and noticing that applying a permutation we have

$$(x_1', ..., x_n', x'_n, x_n) \in \mathcal{H} D [A_0^{(k)}(v)]_{x'_{x,w}} \text{ iff } (q_1, ..., q_{n+k+2}) \in \mathcal{H} D [A_0^{(k)}(v)]_{x'_{\bar{q}w_k}}$$

and the result follows.

Finally, consider the case in which $B[A(v)]_x = P_n B[A'(v)]_{xwv}$. Here $t_\bar{y}$ means that if $t = [B]_\bar{x}$ then $t_\bar{y} = [B]_\bar{y}$. Here $t_\bar{y}$ is variable or $\bar{y}$ is the empty sequence. Let the argument in $A(v)$ which $w$ replaced be $t$. Here $\bar{x} = \bar{x}' \bar{x}'_w \bar{y}$ is normalized except that the sequence $\bar{y}$ of variables in $\bar{x}$ which are free in $t$ placed at the end. All arguments before $t$ are variables in $\bar{x}$. So $v$ may occur in $t$ and in arguments after $t$ and $v$ may be reflected. Hence we will have (if $v$ is reflected)

$$[A(v)]_{\bar{x}'x'_wv} = \Sigma_1 R \Sigma R \ldots R \Sigma_k \Sigma [A^{(k)}(v)]_{x'_{\bar{y}'w}} = \Sigma' R \Sigma [A^{(k)}(v)]_{x'_{\bar{y}'w}}$$

Here $\bar{x}'\bar{y}'$ is the normalized permutation of $\bar{x}v$ and the $k$ new variables $w_k$ but with the sequence $\bar{y}'$ of variables of $\bar{q}$ which occur free in $t$ placed at the end. $\bar{y}'$ is either equal to $\bar{y}$ or contains in addition $v$. Notice how $\bar{x}'$ is not affected by the permutations. For clarity, we wrote the last permutation as a composition. If $v$ is not reflected then we have only $\Sigma$ which puts $v$ in its proper place (which is either in $\bar{y}$ or in $\bar{p}$).

We have that

$$(x_1', ..., x_a', x_1', ..., x_b', y_1, ..., y_c, v) \in \mathcal{H} S R S D [A^{(k)}(v)]_{x'_{\bar{y}'w}}$$

iff

$$(x_1', ..., x_a', x_1', ..., x_b', y_1', ..., y_c', v) \in \mathcal{H} D [A^{(k)}(v)]_{x'_{\bar{y}'w}}$$

where $j$ is either 0 and $v$ is in $\bar{p}$ or else is 1 and $v$ is in $\bar{y}'$. Here the $(p_1, ..., p_{b+k})$ is permutation of the $x_1''$ and $w_i$ (and possibly $v$) corresponding to the normalized sequence of these variables and which has $v$ for $w_i$.

W.l.o.g let $v$ be in $\bar{p}$ and $t$. Fix a $\mathcal{H}$. Then

$$(x_1', ..., x_a', x_1', ..., x_b', y_1, ..., y_c) \in \mathcal{H} D [A(v)]_{x'x'w} = \mathcal{H} P_n D [A_1'(v)]_{x'x'w} D t_\bar{y} \text{ iff }$$

$$(x_1', ..., x_a', x_1', ..., x_b', p'(D t_\bar{y}, y_1, ..., y_1)) \in \mathcal{H} D [A_1'(v)]_{x'x'w} = \mathcal{H} P_n D [A_1'(v)]_{x'x'w} A(v)$$

by induction. This is equivalent to

$$(x_1', ..., x_a', x_1', ..., x_b', p'(D t_\bar{y}, y_1, ..., y_1), A(v)) \in \mathcal{H} D [A_1'(v)]_{x'x'w}$$

Let the Bealer decomposition of $[A_1'(v)]_{x'x'w}$ be

$$\Sigma'_1 R \Sigma'_2 R ... R \Sigma'_{k-1} \Sigma [A^{(k-1)}(v)]_{x'\bar{y}} = \Sigma' R \Sigma [A^{(k)}(v)]_{x'\bar{q}}$$

24
where $\bar{q}$ consists of variables of $\bar{x}''$, $w$, $v$ and $k$ new variables $w_k$.

The previous condition is equivalent to

$$(x_1', ..., x_a', x_1'', ..., x_b'', p'(Dt_{\bar{g}}, y_1, ..., y_1), A(v)) \in \mathcal{H} \mathcal{s' R's'} D[A^{(k-1)}(v)]_{x'q}$$

which is equivalent to

(**) $$ (x_1', ..., x_a', r_1, ..., r_s) \in \mathcal{H} D[A^{(k-1)}(v)]_{x'q}$$

where $(r_1, ..., r_s)$ corresponds to the sequence $\bar{q}$ with $v$ and $w_k$ corresponding to $A(v)$ and $w$ corresponding to $p'(Dt_{\bar{g}}, y_1, ..., y_1)$.

Now consider

$$(x_1', ..., x_a', x_1'', ..., x_b'', y_1, ..., y_c) \in \mathcal{H} p_0 D[A(v)]_{x'qv} A(v)$$

this is equivalent to

$$(x_1', ..., x_a', x_1'', ..., x_b'', y_1, ..., y_c, A(v)) \in \mathcal{H} D[A(v)]_{x'qv} = \mathcal{H} \mathcal{s'r's'} D[A^{(k)}(v)]_{x'q\bar{y}}$$

where $\bar{y}' = \bar{y}' \bar{y} \bar{m}$. This is turn is equivalent to

$$(x_1', ..., x_a', x_1'', ..., x_b'', y_1, ..., y_c, A(v)) \in \mathcal{H} \mathcal{s'r's'} P_{n+1} D[A^{(k)}(v)]_{x'qw} t \bar{y'}$$

that is,

$$(x_1', ..., x_a', e_1, ..., e_{b+k}, y_1', ..., y_{c+1}) \in \mathcal{H} P_{n+1} D[A^{(k)}(v)]_{x'qw} t \bar{y'}$$

where $(e_1, ..., e_{b+k+1})$ corresponds to the sequence $\bar{p}$ and has $A(v)$ corresponding to $w_k$ and $(y_1', ..., y_{c+1})$ corresponds to $\bar{y}' = \bar{y}' \bar{y} \bar{m}$ with $A(v)$ corresponding to $v$. This in turn is the same as

$$(x_1', ..., x_a', e_1, ..., e_{b+k}, p_0^{k+1}(Dt_{\bar{g}'}, y_1', ..., y_{c+1})) \in \mathcal{H} D[A^{(k)}(v)]_{x'qw}$$

It is easy to check that by induction we have

$$p_0^{k+1}(Dt_{\bar{g}'}, y_1', ..., y_{c+1}) = p_0 p_0(Dt_{\bar{g}}, y_1, ..., y_c)) A(v) = p_0(Dt_{\bar{g}}, y_1, ..., y_c)$$

Inspecting (**), noticing that $[A^{(k)}(v)]_{x'qw} = [A^{(k)}(v)]_{x'qw}$ (modulo renaming bound variables) and applying a permutation we get as desired.

**Lemma 8.3** $T_{LAM}(\exists v.A)$ iff there is an assignment $A'$ like $A$ except perhaps for what it assigns to $v$ and such that $T_{LAM}(A'(v))$.

**Proof.** Assume w.l.o.g that $v$ occurs free in $A$. We have $T_{LAM}(\exists v.A)$ iff $G_D([\exists v.A]) = T$ iff $G_{D[A]} = T$. The last condition is equivalent to the existence of a $x \in D$ such that

$$x \in G_D[A(v)]$$

Let $A'$ be like $A$ but with $A'(v) = x$. We must show that $G_D[A'(v)] = T$. Notice that $D_{A'}[A(v)] = D_{A'}[A(v)]$. By the proof of Bealer's lemma (note that we are not using the type 1 condition) we have that:

$$G_D[A'(v)] = T$$

Hence we must show that $G_{D[A(v)]} A'(v) = T$. But this is equivalent to $A'(v) \in G_D[A(v)]$ and the result follows since $A'(v) = x$. The other direction is similar.
Hence if $T_{\lambda M}(\forall v.A(v))$ then for any assignment $A'$ which only differs at most on $v$ we have that $T_{\lambda A'M}(A(v))$. Hence lemma 8.3 yields:

**Lemma 8.4** For any models we have (Inst) $\models \forall v.A(v) \rightarrow A(t)$ where $t$ is free for $v$ in $A$.

Likewise we have

**Lemma 8.5** For any models we have (QImp) $\models \forall v.(A \rightarrow B(v)) \rightarrow (A \rightarrow \forall v.B(v))$ where $v$ does not occur free in $A$.

**Proof.** Assume $T_{\lambda M}(\forall v.A(v))$. Then for all assignment $A'$ differing from $A$ at most on $v$ we have that

\[(*)\quad T_{\lambda A'M}(A \rightarrow B(v))\]

We must show that $T_{\lambda M}(A \rightarrow \forall v.B(v))$. Assume then that $T_{\lambda M}(A)$. We must show that $T_{\lambda M}(A \rightarrow \forall v.B(v))$. Take any assignment $A'$ differing from $A$ at most on $v$. Since $v$ does not occur in $A$ we have that $T_{\lambda A'M}(A)$. Hence by $(*)$ we get that $T_{\lambda A'M}(\forall v.A(v))$. Since this holds for any $A'$ we get the conclusion by lemma 8.3.

Similarly we get the soundness of the (Gen) rule:

**Lemma 8.6** For any models if $\models A(v)$ then $\models \forall v.A(v)$.

**Lemma 8.7** For all $M$ of type 1 and $I$, $A$ and terms $t, t'$ we have

\[D([t = t]) = D([t' = t'])\]

**Proof.** Suppose that

\[D([t = t]) \neq D([t' = t'])\]

Then the type 1 condition implies that there is a $\mathcal{H}$ such that

\[\mathcal{H}D([t = t]) \neq \mathcal{H}D([t' = t'])\]

But

\[[t = t] = P_0(P_0([v = w]_{v,w}, t), t)\]

Hence applying the definition of $\mathcal{H}$ yields

\[\mathcal{H}D([t = t]) = T \leftrightarrow D(t) = D(t)\]

hence we have $\mathcal{H}D([t = t]) = T$ and also analogously $\mathcal{H}D([t' = t']) = T$ and so we obtain a contradiction.

**Lemma 8.8 (TA8*(a))** For type 1 models we have (B1) $\models \Box (A \leftrightarrow B) \leftrightarrow [A] = [B]$.

**Proof.** We must show that $GD_{\lambda M}(\Box (A \leftrightarrow B) \leftrightarrow [A] = [B])] = T$, which is equivalent to showing that

\[GD_{\lambda M}(\Box (A \leftrightarrow B))] = T \text{ iff } GD_{\lambda M}([A] = [B])] = T\]

that is,

\[GD_{\lambda M}( [A \leftrightarrow B)] = [[A = B] = [A = B])] = T \text{ iff } GD_{\lambda M}( [[A] = [B])] = T\]

which is equivalent to

\[D_{\lambda M}( [A \leftrightarrow B)] = D_{\lambda M}( [[A = B] = [A = B])] \text{ iff } D_{\lambda M}( [[A]) = D_{\lambda M}( [B])\]
Lemma 8.9 ([I] A8*(b)) For type 1 models we have \((B2) \vdash \forall v. [A(v)]_\alpha = [B(v)]_\alpha \leftrightarrow [A(v)]_{\alpha v} = [B(v)]_{\alpha v}\).

Proof. We have

\[D_{LAM}[A(v)]_\alpha = p_0(D_{LAM}[A(v)]_{\alpha v}, A(v))\]

and

\[D_{LAM}[B(v)]_\alpha = p_0(D_{LAM}[B(v)]_{\alpha v}, A(v))\]

Consider all assignments \(A'\) like \(A\) except for the assignment on \(v\). Then \(D_{LAM}[A(v)]_{\alpha v} = D_{LAM}[A'(v)]_{\alpha v}\) and \(D_{LAM}[B(v)]_{\alpha v} = D_{LAM}[B'(v)]_{\alpha v}\). So if \(D_{LAM}[A(v)]_{\alpha v} = D_{LAM}[B(v)]_{\alpha v}\) the for any \(A'\) we have \(D_{LAM}[A'(v)]_{\alpha v} = D_{LAM}[B'(v)]_{\alpha v}\) and one direction follows.

We now need to show that if for all such \(A'\) we have

\[D_{LAM}[A'(v)]_{\alpha v} = D_{LAM}[B'(v)]_{\alpha v}\]

then \(D_{LAM}[A(v)]_{\alpha v} = D_{LAM}[B(v)]_{\alpha v}\). But if \(D_{LAM}[A(v)]_{\alpha v} \neq D_{LAM}[B(v)]_{\alpha v}\) then by the type 1 condition there would be a \(\mathcal{H}\) and a \((x_1, ..., x_m)\) such that \((x_1, ..., x_m) \in \mathcal{H}D_{LAM}[A(v)]_{\alpha v}\) but \((x_1, ..., x_m) \notin \mathcal{H}D_{LAM}[B(v)]_{\alpha v}\). But taking \(A'(v) = x_n\) this means that

\[(x_1, ..., x_{m-1}) \in \mathcal{H}p_0(D_{LAM}[A(v)]_{\alpha v}, A'(v)) = \mathcal{H}D_{LAM}[A(v)]_{\alpha v}\]

but

\[(x_1, ..., x_{m-1}) \notin \mathcal{H}p_0(D_{LAM}[B(v)]_{\alpha v}, A'(v)) = \mathcal{H}D_{LAM}[B(v)]_{\alpha v}\]

and we obtain a contradiction.

Lemma 8.10 ([I] A11*) In the same conditions we have \((S5') \vdash v_i \neq v_j \rightarrow \Box v_i \neq v_j\).

Proof. We must show that if \(T_{LAM}(v_i \neq v_j)\) then \(T_{LAM}(\Box v_i \neq v_j)\).

But if \(T_{LAM}(v_i \neq v_j)\) then \(D_{LAM}(v_i) \neq D_{LAM}(v_k)\). But this is equivalent to \(A(v_i) \neq A(v_k)\). This means that for all \(\mathcal{H} \in H\) we have \(\mathcal{H}([v_i \neq v_j]) = T\) and the conclusion follows.

Lemma 8.11 ([I] A9) In the same conditions we have \((T) \vdash \Box A \rightarrow A\).

Proof. If \(T_{LAM}(\Box A)\) then \(\forall \mathcal{H} \in H\) we have \(\mathcal{H}D_{LAM}([A]) = T\) and so in particular \(T_{LAM}(A)\)

Lemma 8.12 ([I] A10) In the same conditions we have \((K) \vdash \Box(A \rightarrow B) \rightarrow (\Box A \rightarrow \Box B)\).

Proof. Assume \(T_{LAM}(\Box(A \rightarrow B))\). Then \(\forall \mathcal{H} \in H\) we have \(\mathcal{H}D_{LAM}([A \rightarrow B]) = T\) which means that if \(\mathcal{H}D_{LAM}([A]) = T\) then \(\mathcal{H}D_{LAM}([B])\). To show that \(T_{LAM}(\Box(A \rightarrow B))\) consider any \(\mathcal{K} \in H\). Assume that \(\forall \mathcal{H} \in H\) we have \(\mathcal{H}D_{LAM}([A]) = T\). Then \(\mathcal{K}D_{LAM}([A]) = T\) and hence \(\mathcal{K}D_{LAM}([B]) = T\). Hence we have \(\forall \mathcal{H} \in H\) we have \(\mathcal{H}D_{LAM}([B]) = T\) and thus \(T_{LAM}(\Box A \rightarrow \Box B)\).

9 Completeness of T1

In this section we show completeness:

**Lemma 9.1** If a sentence \(A\) is T1-valid then \(\vdash A\).

It is enough to show that:

**Lemma 9.2** A set \(\Gamma\) of sentences of T1 is consistent iff \(\Gamma\) has a model.
By soundness an inconsistent set of formulas cannot have a model so we only need prove one implication. This lemma is enough to prove completeness for is a formula $A$ is $T$-valid then $\neg A$ cannot have a model, otherwise there would be a model for which both $A$ and $\neg A$ held. Since $\neg A$ does not have a model by the lemma above $\neg A$ is inconsistent and hence we have $\neg A \vdash \neg x = x$, that is $\vdash \neg A \rightarrow \neg x = x$ or $\vdash x = x \rightarrow A$ from which it follows that $\vdash A$.

Given a consistent set $\Gamma$ of sentences we form an extension $L^\omega$ of $L^\omega$ consisting of countably infinite constants and $n$-ary predicate symbols $F_i$ (including 0-ary propositional constants) and consider any enumeration $S_1, S_2, \ldots$ of all sentences. Consider also any enumeration of all closed terms $[B]_x$. We apply the following process to the sequence $S_1, S_2, \ldots$. Consider the first closed term $[B]_x$ and chose a $F$ and let $[F(x_1, \ldots x_n)]_x = [B]_x$ be $S_k$. If $F$ does not occur in a sentence before $S_k$ or in $B$ or in $\Gamma$ then we proceed to the next closed term. Otherwise chose any $F'$ which satisfies this condition and let $S_m$ be $[F'(x_1, \ldots x_n)]_x = [B]_x$. Then switch the positions of $S_m$ and $S_k$ in the enumeration $S_1, S_2, \ldots$ and proceed to the next closed term. Any finite initial segments of the resulting transformations of the resulting sequence $S'_1, S'_2, \ldots$ will stabilise after a finite number of iterations (we do not move an element of a sequence twice).

From now on we assume that we have a consistent set $\Gamma$ and that we have an enumeration $S_1, S_2, \ldots$ of sentences of $L^\omega$ satisfying the property that for any closed term $t$ there is a predicate symbol $F$ and a $S_k$ of the form $[F(x_1, \ldots x_n)]_x = t$ such that $F$ does not occur in $\Gamma$, $t$ or in any $S_j$ before $S_k$.

We construct an array $\Delta_{ij}(i,j \geq 1)$ of sets of sentences is the following form:

- $\Delta_{i1} = \Gamma$.
- $\Delta_{ij} = \Delta_{ij-1} \cup C_{ij} \cup \{S_j\}$ if $j > 1$ and $\{S_j\} \cup \Delta_{ij-1} \cup E_{ij}$ is consistent.
- $\Delta_{ij} = \Delta_{ij-1} \cup C_{ij}$ if $j > 1$ and $\{S_j\} \cup \Delta_{ij-1} \cup E_{ij}$ is inconsistent.
- $\Delta_{i1} = \{\exists x. \neg A \leftrightarrow B\}$ if $S_{i-1}$ is $[A]_x \neq [B]_x$ and $S_{i-1} \in \Delta_{i-1i-1}$
- $\Delta_{i1} = \Delta_{i-1i-1}$ otherwise.

Here

$E_{ij} = \{[A]_x = [B]_y : \exists k, m, k \leq i, m \leq j, (k, m) \neq (i, j) \& \Delta_{km} \vdash [A]_x = [B]_y\}$

for $j > 1$ and

$E_{ij} = \{[A]_x = [B]_y : \exists k, m, k \leq i, m \leq i - 1 \& \Delta_{km} \vdash [A]_x = [B]_y\}$

for $j = 1$. Consider the set (in the order of $S_1, S_2, \ldots$) of all existential sentences $\exists x.A(x), \exists x.A'(x), \ldots$ in $\Delta_{ij-1}$ which occur before $S_j$. We proceed as follows. We start by picking a constant $a_1$ such that $A(a_1)$ is the first instantiation occurring after $S_j$ such that $a_i$ does not occur before $S_j$ in $\Delta_{ij-1}$ (or $\Delta_{i-1i-1}$ for $j = 1$). We then proceed to chose an constant $a_2$ such that $A'(a_2)$ is the first instantiation after $A(a_1)$ such that $a_2$ does not occur before or in $A(a_1)$ and so forth. $C_{ij}$ consists of the set of such instantiations $\{A(a_1), A'(a_2), \ldots\}$.

Lemma 9.3 $\Delta_{ij} \cup E_{ij}$ is consistent.

Proof. By induction on the construction of $\Delta_{ij}$. The case $\Delta_{i1}$ is trivial since $\Gamma$ is consistent. Let $j \geq 1$ and assume by induction that $\Delta_{ij} \cup E_{ij}$ is consistent.

Suppose that $\Delta_{ij+1} \cup E_{ij+1}$ were inconsistent. Then $\Delta_{ij+1} \cup E_{ij+1} \vdash \neg v = v$ which means that for some finite set of formulas $F_1, \ldots, F_k$ of $\Delta_{ij+1} \cup E_{ij+1}$ we have

(*) $\vdash (F_1 \& F_2 \& \ldots \& F_k) \rightarrow \neg v = v$

28
Suppose that $\Delta_{ij+1} = \Delta_{ij} \cup C_{ij+1}$. If all the $F_i$'s belong $\Delta_{ij}$ then this contradicts the consistency of $\Delta_{ij} \cup E_{ij}$. Hence there is at least one $F_i$ belonging to $C_{ij+1} \cup E_{ij+1}$.

If $F_i$ is in $C_{ij+1}$ then it is a $C(a)$ where $\exists v.C(v)$ is in $\Delta_{ij}$. We may reorganise (*) into the form
\[
\vdash C_1(a_1) \rightarrow (C_2(a_2) \rightarrow ... \rightarrow (F_k \rightarrow \neg v = v)...
\]
so that each instantiation constant $a_i$ in the expression does not occur further ahead. Since $a_1$ does not occur further ahead we may substitute it for a fresh variable and then apply the $\exists$-introduction rule
\[
\vdash \exists x.C_1(x) \rightarrow (C_2(a_2) \rightarrow ... \rightarrow (F_k \rightarrow \neg v = v)...
\]
and so on until we have replaced all formulas in $C_{ij+1}$ by formulas in $\Delta_{ij}$.

If an $F_i$ belongs to $E_{ij+1}$ then it is an equality $t = s$ such that we have $\Delta_{km} \vdash t = s$ for $k \leq i, m \leq j + 1, (k, m) \neq (i, j + 1)$, that is $\vdash (G_1 \& ... \& G_n) \rightarrow t = s$ for $G_i \in \Delta_{km}$. We replace $t = s$ by $G_1 \& ... \& G_n$. Thus we may eliminate all such $F_i$ and we obtain a contradiction.

Now suppose that $\Delta_{ij+1} = \Delta_{ij} \cup C_{ij+1} \cup \{S_{j+1}\}$. Assume that $\Delta_{ij+1} \cup E_{ij+1}$ is inconsistent. Then $\Delta_{ij} \cup C_{ij+1} \cup \{S_{j+1}\} \cup E_{ij+1}$ is inconsistent and using the previous argument we find $F_1, ..., F_k$ in $\Delta_{ij} \cup \{S_{j+1}\}$ such that
\[
\vdash (F_1 \& F_2 \& ... \& F_k) \rightarrow \neg v = v
\]
which contradicts the way $\Delta_{ij+1}$ is formed in this case.

Finally we must consider the case of $\Delta_{ij}$. We assume by induction that $\Delta_{i-1i-1} \cup E_{i-1i-1}$ is consistent. If $\Delta_{i1} = \Delta_{i-1i-1}$ then assume that $\Delta_{i-1i-1} \cup E_{i1}$ were inconsistent and we have
\[
\vdash (F_1 \& F_2 \& ... \& F_k) \rightarrow \neg v = v
\]
where the $F_i$ are in $\Delta_{i-1i-1} \cup E_{i1}$. Now by definition $E_{i1} = E_{i-1i-1} \cup \{[A]_x = [B]_y : \Delta_{i-1i-1} \vdash [A]_z = [B]_z\}$ so an argument similar to the one above shows that we obtain a contradiction to the consistency of $\Delta_{i-1i-1} \cup E_{i-1i-1}$.

If $\Delta_{i1} = \{\exists \bar{x}. \neg A \leftrightarrow B\}$ then $[A]_z \neq [B]_z \in \Delta_{i-1i-1}$. Assume $\{\exists \bar{x}. \neg A \leftrightarrow B\} \cup E_{i1}$ were inconsistent. Then there are equality sentences $E_1, ..., E_k$ in $E_{i1}$ such that $\vdash E_1 \rightarrow ... \rightarrow E_k \rightarrow \forall \bar{x}. A \leftrightarrow B$. We then apply necessitation, (T) and (B) to obtain $E_{i1} \vdash [A]_z = [B]_z$, a contradiction.

We now define $\Delta^k = \bigcup_j \Delta_{kj}$ and consider the set $\mathcal{A}$ consisting of all such $\Delta^k$ for $k \geq 1$.

**Definition 9.4** Let $L^*_\omega$ be an extension of $L_\omega$. A set $\mathcal{A}$ of sets of sentences in $L^*_\omega$ is $T1$-perfect if

- Every set in $\mathcal{A}$ is maximal, consistent and $\omega$-complete.
- If an identity sentence $t = t'$ is in one of the sets of $\mathcal{A}$ it is in all of them.
- If a sentence $[A]_x = [B]_x$ belongs to one of the sets of $\mathcal{A}$ there is a set in $\mathcal{A}$ which contains the sentence $\exists \bar{x}. \sim A \leftrightarrow B$.
- For every closed term $[B]_{x_1...x_n}$ there is an atomic predicate $F$ such that $[B]_{x_1...x_n} = [F(x_1, ..., x_n)]_{x_1...x_n}$ belongs to a set in $\mathcal{A}$.

**Lemma 9.5** $\mathcal{A}$ is $T1$-perfect.
Proof. Let $\Delta^k \in \mathcal{A}$. We show first that $\Delta^k$ is maximal, consistent and $\omega$-complete. It is obviously consistent because each $\Delta_{kj}$ is consistent by the above lemma. To show that it is maximal we need to show that for any sentence $S$, either $S$ or $\neg S = S$ is in $\Delta^k$. Suppose then that neither formula was in $\Delta^k$. Then we must have that $S_i \cup \Delta_{kj-1} \cup E_{kj}$ and $S_j \cup \Delta_{kj-1} \cup E_{kj}$ are inconsistent. Suppose w.l.o.g. that $i < j$. Then since $\Delta_{ki} \subseteq \Delta_{kj}$ we get a contradiction. For if both a sentence and it’s negation are inconsistent with a set of sentences, that set is inconsistent.

To show that $\Delta^k$ is $\omega$-complete we need to show that if $\Delta^k \vdash \exists v. F(v)$ then $\Delta^k \vdash F(t)$ for some term $t$. If $\Delta^k \vdash \exists v. F(v)$ then by maximality $\exists v. F(v) \in \Delta^k$ and let it have been introduced in stage $\Delta_{ki}$. Then in a subsequent stage $j$, by construction it is clear that $C_{ij}$ and hence $\Delta^k$ will contain a formula of the form $F(c)$ for some constant $c$.

We now show that if an equality $t = s$ is in one $\Delta^k \in \mathcal{A}$ then it is in them all. Let then $t = s$ be in a certain $\Delta^k$ and consider any other $\Delta^j$. If $\Delta^j$ is in a previous row then obviously we cannot have $\neg t = s \in \Delta^j$ because $t = s$ must belong to a certain $\Delta_{km}$ and $\Delta_{km} \cup E_{km}$ is consistent by the previous lemma. A similar argument show that $\neg t = s$ cannot belong to a $\Delta^j$ which is on a subsequent row.

We now show that if $[F]_\bar{x} \neq [G]_\bar{x}$ is in a $\Delta^k$ then there is a $\Delta^j$ containing $\exists \bar{x} \sim A \leftrightarrow B$. Suppose $[F]_\bar{x} \neq [G]_\bar{x}$ is in $\Delta_{ki}$ (minimal $i$ for that condition) which is under or on the diagonal ($i \leq k$). Then the result follows directly from the construction of the transition between rows. Suppose then that it is above the diagonal. Notice that in all subsequent $\Delta_{kj}$’s we cannot have $\Delta_{kj} \vdash [F]_\bar{x} = [G]_\bar{x}$ and we just “slide” down the column until reaching below the diagonal and get as desired.

Finally we show that for any closed term $[G]_\bar{x}$ there is a primitive predicate $S$ such that $[G]_\bar{x} = [F(x_1, \ldots, x_k)]_{x_1 \ldots x_k}$. We can choose such a $S$ (let $[G]_\bar{x} = [F(x_1, \ldots, x_k)]_{x_1 \ldots x_k}$ be $S_i$) so that $F$ does not occur in any of the previous $S_j$ in $G$ or in $\Gamma$. Hence in the stage where $S_i$ may be introduced this formula is consistent with the previous $\Delta_{kj}$ in the same row (including for any previous column) and thus belongs to $\Delta^k$.

Definition 9.6 Consider a T1-perfect set $\mathcal{A}$ and a $\Delta \in \mathcal{A}$. We define an order $<$ on predicate symbols (such that the first element is $=$) and variables. We define the canonical model as follows. Consider the set $D_\Delta$ consisting of those symbols $S$ such that there are no $G < F$ for which $[F(x_1, \ldots, x_n)]_{x_1 \ldots x_n} = [G(x_1, \ldots, x_n)]_{x_1 \ldots x_n} \in \Delta$ and those constants $a$ such that there is no $[F(x_1, \ldots, x_n)]_{x_1 \ldots x_n} = a \in \Delta$ and there are no constants $b$ such that $b < a$ and $a = b \in \Delta$. We let $D_\Delta$ divide the symbols according to arity for $i \geq 0$, the case $i = -1$ corresponding to constants. The distinguished equality element is simply $=$. The extension functions $H \in \mathcal{H}$ are determined by the $\Delta$’s in $\mathcal{A}$ and the actual extension function by $\Delta$. We have that $\mathcal{H}_\Delta(F)$ consists of all $(F_1, \ldots, F_n)$ such that $[F_i(v_1, \ldots, v_n)]_{v_1 \ldots v_n} = t_i \in \Delta$ or $F_i$ is a constant $c_i$ and $F_i(t_1, \ldots, t_n) \in \Delta$. We now define the eight model operations.

$$k(F, G) = H \text{ iff } [H(v_1, \ldots, v_k, u_1, \ldots, u_m)]_{v_1 \ldots v_k u_1 \ldots u_m} = [F'(v_1, \ldots, v_k) \& G'(u_1, \ldots, u_k)]_{v_1 \ldots v_k u_1 \ldots u_m} \in \Delta$$

where $[F'(v_1, \ldots, v_k, u_1, \ldots, u_k)]_{v_1 \ldots v_k u_1 \ldots u_k} = [F'(v_1, \ldots, v_k)]_{v_1 \ldots v_k u_1 \ldots u_k} \in \Delta$ and $[G'(v_1, \ldots, v_k, u_1, \ldots, u_k)]_{v_1 \ldots v_k u_1 \ldots u_k} = [G'(u_1, \ldots, u_k)]_{v_1 \ldots v_k u_1 \ldots u_k} \in \Delta$

$$n(F) = H \text{ iff } [H(v_1, \ldots, v_k)]_{v_1 \ldots v_k} = [\sim F(v_1, \ldots, v_k)]_{v_1 \ldots v_k} \in \Delta$$

$$u(F) = H \text{ iff } [H(v_1, \ldots, v_k)]_{v_1 \ldots v_k} = [\exists v. F(v_1, \ldots, v_k)]_{v_1 \ldots v_k} \in \Delta$$

$$e(F) = H \text{ iff } [H(v_1, \ldots, v_k)]_{v_1 \ldots v_k} = [F(v_1, \ldots, v_k)]_{v_1 \ldots v_k} \in \Delta$$

30
Proof. We need to prove that \( (F, G) \) satisfies (B) and (T) for \( H \) in degree \( \Delta \) such that \( \{G(x_1, \ldots, x_n) \}_{x_1 \cdots x_n} = [F(x_1, \ldots, x_n)]_{x_1 \cdots x_n} \in \Delta \) and consequent \( c \) the constant \( d \in \Delta \) such that \( c = d \in \Delta \).

**Lemma 9.7** The canonical model defined above with the canonical interpretation is a T1-model and a model for \( \Delta \).

Proof. The conditions of the extension functions are verified as follows.

Suppose that \( (F_1, \ldots, F_k, G_1, \ldots, G_m) \in \mathcal{H}_H \) for \( H = k(F, G) \). Then we have closed terms \( t_i, t'_i \) such that \( F_i[x_1w, \ldots, x_n]_{x_1 \cdots x_n} = t_i, G_i[x_1, \ldots, x_n]_{x_1 \cdots x_n} = t'_i \), and \( H(t_1, \ldots, t_k, t'_1, \ldots, t'_m) \in \Delta \).

But since \( [H(v_1, \ldots, v_k, u_1, \ldots, u_m)]_{v_1 \cdots v_k u_1 \cdots u_m} = [F(v_1, \ldots, v_k)G'(u_1, \ldots, u_k)]_{v_1 \cdots v_k u_1 \cdots u_m} \in \Delta \) where

\[
[F(v_1, \ldots, v_k, u_1, \ldots, u_k)]_{v_1 \cdots v_k u_1 \cdots u_m} = [F'(v_1, \ldots, v_k)]_{v_1 \cdots v_k u_1 \cdots u_m} \in \Delta \quad \text{and} \quad [G(v_1, \ldots, v_k, u_1, \ldots, u_k)]_{v_1 \cdots v_k u_1 \cdots u_m} = [G'(u_1, \ldots, u_k)]_{v_1 \cdots v_k u_1 \cdots u_m} \in \Delta
\]

one direction of axiom (B) and (T) yield

\[
\forall v_1, \ldots, v_k, u_1, \ldots, u_m. F'(v_1, \ldots, v_k)G'(u_1, \ldots, u_k) \leftrightarrow H(v_1, \ldots, v_k, u_1, \ldots, u_m) \quad \text{and} \quad \forall v_1, \ldots, v_k, u_1, \ldots, u_m. F'(v_1, \ldots, v_k)G'(u_1, \ldots, u_k) \leftrightarrow G'(u_1, \ldots, u_k) \in \Delta.
\]

whence we deduce that \( F'(t_1, \ldots, t_k) \) and \( G'(t'_1, \ldots, t'_k) \) are in \( \Delta \) which is equivalent to \( (F_1, \ldots, F_k, G_1, \ldots, G_m) \in \mathcal{H}_F \) and \( (G_1, \ldots, G_m) \in \mathcal{H}_G \). The other direction is similar. The conditions for \( u, e, i, c, r \) are proven in the same way. We now show the condition for \( p_i \). Suppose we had \( (F_1, \ldots, F_k, G_1, \ldots, G_m) \in \mathcal{H}_F \) where \( p_i(F, G) = H \). Then we have closed terms \( t_i, t'_i \) such that \( F_i[x_1w, \ldots, x_n]_{x_1 \cdots x_n} = t_i, G_i[x_1, \ldots, x_n]_{x_1 \cdots x_n} = t'_i \), and \( H(t_1, \ldots, t_k, t'_1, \ldots, t'_m) \in \Delta \).

But since \( [H(v_1, \ldots, v_k, u_1, \ldots, u_m)]_{v_1 \cdots v_k u_1 \cdots u_m} = [F(v_1, \ldots, v_k, G(x_1, \ldots, x_j, w_1, \ldots, w_i)]_{x_1 \cdots x_j} \in \Delta \) using as previously (B) and (T) we get that

\[
[F(t_1, \ldots, t_k), G(x_1, \ldots, x_j, t'_1, \ldots, t'_i)]_{x_1 \cdots x_j} \in \Delta
\]

We need to prove that \( (F_1, \ldots, F_k, p'_i(G, G_1, \ldots, G_j)) \in \mathcal{H}_F \). This is equivalent to proving that, for \( H = p'_i(G, G_1, \ldots, G_j) \) there is a \( t = [H(y_1, \ldots, y_j)]_{y_1 \cdots y_j} \) such that \( F(t_1, \ldots, t_k, k) \in \Delta \). But this follows from taking the \( H \in \Delta \) such that \( [H(y_1, \ldots, y_j)]_{y_1 \cdots y_j} = [G(x_1, \ldots, x_j, t'_1, \ldots, t'_i)]_{x_1 \cdots x_j} = G(x_1, \ldots, x_j, G(x_1, \ldots, x_j, x_{p_1}[x_1 \cdots x_{p_1}], \ldots, G(x_1, \ldots, x_{p_1}[x_1 \cdots x_{p_1}])_{x_1 \cdots x_j} \in \Delta.

We now need prove the T1 condition. Let \( F \neq G \) for \( F, G \in \Delta \). Then \([F(x_1, \ldots, x_n)]_{x_1 \cdots x_n} \neq [G(x_1, \ldots, x_n)]_{x_1 \cdots x_n} \in \Delta \). Since \( A \) is T1 perfect there exists a \( \Delta' \) such that \( \exists \overline{x}. \neg F(x_1, \ldots, x_n) \leftrightarrow G(x_1, \ldots, x_n) \in \Delta' \). Hence by \( \omega \)-completeness there are terms \( t_1, \ldots, t_n \) such that \( \neg F(t_1, \ldots, t_n) \leftrightarrow G(t_1, \ldots, t_n) \in \Delta' \) and the result follows.

We now prove by induction on complexity that we obtain a model of the sentences in \( \Delta \). For the atomic case note that if \( t = [F(x_1, \ldots, x_n)]_{x_1 \cdots x_n} \in \Delta \) and \( F \in \Delta \) then \( Dt = F \). Thus if \( A(t_1, \ldots, t_n) \in \Delta \) consider \( t_i = [F_i(x_1, \ldots, x_n)]_{x_1 \cdots x_n} \in \Delta \) with \( F_i \in \Delta \). Then \( (Dt_1, \ldots, Dt_n) = (F_1, \ldots, F_n) \in G\mathcal{T}(F) \) so \( \mathcal{G}(A(t_1, \ldots, t_n)) = T \). Consider \( A \& B \in \Delta \). Then by completeness \( A \) and \( B \) belong to \( \Delta \) and thus by induction \( \mathcal{G}[A] = T \) and \( \mathcal{G}[B] = T \) and thus \( \mathcal{G}[A \& B] = T \). The case of \( \neg A \) is similar. Finally if \( \exists \overline{x}. A(x) \in \Delta \) then \( \omega \)-completeness implies that there is a closed term \( t \) such that \( A(t) \in \Delta \) and so by induction \( \mathcal{G}[A(t)] = T \). Use lemma 7.6 and 8.3 with \( A(v) = F \) for \( t = [F(x_1, \ldots, x_n)]_{x_1 \cdots x_n} \in \Delta \).
10 Soundness and Completeness of T2

We have already proven the soundness of the standard first-order logical axioms and rules for any model. Also the soundness of axioms 6 and 7. Axioms 9 to 11 are clearly sound by the type 2 model condition.

Lemma 10.1 \textit{TGen is sound: suppose that } F \textit{ does not occur in } A(v). \textit{If } \not \models A([F(x_1, ..., x_n)]_{x_1...x_n}) \textit{ then we can infer } \not \models A(t) \textit{ whenever } t \textit{ complex, of arity } n \textit{ and is free for } v \textit{ in } A.

\textit{Proof.} By (L) we have

\[ \models [F(x_1, ..., x_n)]_{x_1...x_n} = t \rightarrow A([F(x_1, ..., x_n)]_{x_1...x_n}) \rightarrow A(t) \]

from which it follows that

\[ \models A([F(x_1, ..., x_n)]_{x_1...x_n}) \rightarrow [F(x_1, ..., x_n)]_{x_1...x_n} = t \rightarrow A(t) \]

Assume \[ \models A([F(x_1, ..., x_n)]_{x_1...x_n}). \] Then by soundness of (L), (MP) and tautologies we get

\[ \models [F(x_1, ..., x_n)]_{x_1...x_n} = t \rightarrow A(t) \]

By the non-cycling condition for T2 models \( F \) does not occur in \( t \) and hence the interpretation of \( F \) does not affect the denotation of \( Dt \). But choose an interpretation \( \mathcal{I} \) assigning to \([F(x_1, ..., x_n)]_{x_1...x_n} \) the value \( Dt \). In this case we have \[ \models [F(x_1, ..., x_n)]_{x_1...x_n} = t \] and hence \( T_\mathcal{I}(A(t)) \). But since \( D[A(t)] \) cannot depend on the interpretation we must have \[ \models A(t) \].

Lemma 10.2 We have that \([A]_x = [B]_x \rightarrow A \leftrightarrow B \) is sound.

\textit{Proof.} For any T2 model, interpretation and assignment we must show that \( T([A]_x = [B]_x \rightarrow A \leftrightarrow B) \). We must show that if \( T([A]_x = [B]_x) \rightarrow T(A \leftrightarrow B) \). Now if \( T([A]_x = [B]_x) \rightarrow T(A \leftrightarrow B) \). We now show that \( T(A \leftrightarrow B) \). We show only one implication, the other is similar. We need only show that \( T(\forall x.A \rightarrow B) \), that is, \( GD[\forall x.A \rightarrow B] = T \). Now

\[ D[\forall x.A \rightarrow B] = D[\neg \exists \bar{x} \cdot \neg (A \rightarrow B)] = D[\neg \exists \bar{x}.A \& \neg B] \]

\[ = ne...eD[A \& \neg B] = ue...ekD[A]_x nD[B]_x = ue...ekD[A]_x nD[A]_x \]

So \( \emptyset \in \mathcal{H}ue...ekD[A]_x nD[A]_x \) iff there is no \( (x_1, ..., x_n) \) such that \( (x_1, ..., x_n) \in \mathcal{H}D[A]_x \) and \( (x_1, ..., x_n) \notin \mathcal{H}D[A]_x \). Hence \( GD[\forall x.A \rightarrow B] = T \).

Note that this rule is sound for any model.

Lemma 10.3 Let \( \Gamma \) be a consistent set of sentences in T2. Then it is included in a maximal, consistent, \( \omega \)-complete set of sentences \( \Delta \) such that for each closed term \([A]_x \) the sentence \([A]_{x_1...x_n} = [F(x_1, ..., x_n)]_{x_1...x_n} \) belongs to \( \Delta \), for some symbol \( F \).

\textit{Proof.} We assume we have an enumeration \( S_1, S_2, ... \) of the sentences of \( L^\omega \) as in the proof of completeness of T1. Let \( \Delta_0 = \Gamma \). Then given \( \Delta_i \), consider the sentence \( S_{i+1} \). If it is consistent with \( \Delta_i \) then \( \Delta_{i+1} = \Delta_i \cup \{ S_{i+1} \} \cup C_i \) otherwise \( \Delta_{i+1} = \Delta_i \cup C_i \). Consider the set (in the order of \( S_1, S_2, ... \) of all existential sentences \( \exists x.A(x), \exists x.A'(x), ... \) in \( \Delta_i \) which occur before \( S_{i+1} \). We proceed as follows. We start by picking a constant \( a_1 \) such that \( A(a_1) \) is the first instantiation occurring after \( S_{i+1} \) such that \( a_1 \) does not occur before \( S_{i+1} \) or in \( \Delta_i \). We then proceed to choose a constant \( a_2 \) such that \( A'(a_2) \) is the first instantiation after \( A(a_1) \) such that \( a_2 \) does not occur before or in \( A(a_1) \) and so forth. \( C_i \) consists of the set of such instantiations \( \{ A(a_1), A'(a_2), ... \} \).
Each $\Delta_i$ is consistent, by a similar argument as in the T1 case.

We show first that $\Delta = \bigcup \Delta_i$ is maximal, consistent and $\omega$-complete. It is obviously consistent because each $\Delta_i$ is consistent. To show that it is maximal we need to show that for any sentence $S_i$ either $S_i$ or $\neg S_i = S_j$ is in $\Delta$. Suppose then that neither formula was in $\Delta$. Then we must have that $S_i \cup \Delta_{i-1}$ and $S_j \cup \Delta_{j-1}$ are inconsistent. Suppose w.l.o.g. that $i < j$. Then since $\Delta_i \subseteq \Delta_j$ we get a contradiction. For if both a sentence and its negation are inconsistent with a set of sentences, that set is inconsistent.

To show that $\Delta$ is $\omega$-complete we need to show that if $\Delta \vdash \exists v. F(v)$ then $\Delta \vdash F(t)$ for some term $t$. If $\Delta \vdash \exists v. F(v)$ then by maximality $\exists v. F(v) \in \Delta$ and let it have been introduced in stage $\Delta_i$. Then in a subsequent stage $j$, by construction it is clear that $C_j$ and hence $\Delta$ will contain a formula of the form $F(c)$ for some constant $c$.

Finally we show that for any closed term $[G]_x$ there is a primitive predicate $F$ such that $[G]_x = [F(x_1, \ldots, x_k)]_{x_1, \ldots, x_k}$. We can choose such a $F$ (let $[G]_x = [F(x_1, \ldots, x_k)]_{x_1, \ldots, x_k}$ be $S_i$) so that $F$ does not occur in any of the previous $S_i$, in $G$ or in $\Gamma$. Hence in the stage where $S_i$ may be introduced this formula is consistent with the previous $\Delta_j$ and thus belongs to $\Delta$.

We form the canonical model and canonical interpretation as in the T1 case, using $\Delta$. Verification that we obtain in fact a model is the same. Note that were in the T1 case we used (B) and (T) here we can make use of axiom 8. The only difference is that we have only one extension function $G \in H$ corresponding to $\Delta$ itself.

**Lemma 10.4** The canonical model defined above with the canonical interpretation is a T2-model and a model for $\Delta$.

**Proof.** We check the type 2 condition. The condition of being one-to-one clearly follows from axioms 10 and 11. For instance if we had $F = n(G) = n(G')$ for $F, G, G' \in D_\Delta$ then $[F(x_1, \ldots, x_n)]_{x_1, \ldots, x_n} = \neg G(x_1, \ldots, x_n)$ and $[F(x_1, \ldots, x_n)]_{x_1, \ldots, x_n} = \neg G'(x_1, \ldots, x_n)$ are in $\Delta$. But since $\Delta$ is maximal we have then $\neg G(x_1, \ldots, x_n) = \neg G'(x_1, \ldots, x_n) \in \Delta$ and hence by axiom 10 that $[G(x_1, \ldots, x_n)]_{x_1, \ldots, x_n} = G'(x_1, \ldots, x_n) \in \Delta$. Now the conditions on elements of $D_\Delta$ imply then that $G = G'$. The condition of having disjoint ranges similarly follows from axiom 9. The non-cycling condition follows from axiom 12.

We prove by induction on complexity that we obtain a model of the sentences in $\Delta$ in the same way as in lemma 9.7.

## 11 Fregean Arithmetic in T2

Let $\Delta$ be a marked binary predicate in T2, where we write $x\Delta y$ for $\Delta(x, y)$.

**Lemma 11.1** Bealer, Quality and Concept, p122.

**Proof.** We must show that N0N, where

$$0 = \neg \exists v(\neg \Delta y)_y$$

$$\text{NN}x \equiv_{df} \forall z(I(z) \rightarrow x\Delta z)$$

where

$$I(z) \equiv_{df} 0\Delta z \ & \ (\forall y(y\Delta z \rightarrow S(y)\Delta z))$$

and

$$S(x) \equiv_{df} [\exists u(u\Delta x \ & \ \exists v(\neg v\Delta u \ & \ y \equiv [v = v \lor w\Delta u]_w)]_y$$
\[ x \equiv y \equiv_{df} \forall w(w\Delta x \leftrightarrow w\Delta y) \]

We have the tautology \(0\Delta z \land (\forall y(y\Delta z \rightarrow S(y)\Delta z)) \rightarrow 0\Delta z\) to which applying \(\forall\)-generalization we get immediately the formula \(NN0\).

We now show that \(NNx \rightarrow NN(S(x))\). We must show that
\[
\forall z(I(z) \rightarrow x\Delta z) \rightarrow \forall z(I(z) \rightarrow S(x)\Delta z)
\]

By tautologies and a quantifier axiom we have
\[
I(z) \rightarrow \forall y(y\Delta z \rightarrow S(y)\Delta z) \rightarrow (x\Delta z \rightarrow S(x)\Delta z)
\]

If we consider the tautology \(A \rightarrow (B \rightarrow C) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C))\) then we get
\[
\vdash (I(z) \rightarrow x\Delta z) \rightarrow (I(z) \rightarrow S(x)\Delta z)
\]
and hence using generalization and \(\forall x(A \rightarrow B) \rightarrow (\forall xA \rightarrow \forall xB)\):
\[
\vdash \forall z(I(z) \rightarrow x\Delta z) \rightarrow \forall z(I(z) \rightarrow S(x)\Delta z))
\]
which is \(NN(x) \rightarrow N(S(x))\).

We now show that \(\neg \exists x : 0 = S(x)\). But this equivalent to
\[
\forall x \neg([\neg \exists v(v\Delta y)]_y = [\exists u(u\Delta x \land \exists v(v\Delta u \land y \equiv [w = v \lor w\Delta u])_w)]_y
\]
which is an immediate consequence of the T2 axiom for these intensional abstracts belong to different domains.

We now show that \(x = y \rightarrow S(x) = S(y)\).

Consider \([u\Delta v]_v\). Then \([u\Delta x]\) is the predication of \([u\Delta v]_v\) of \(x\) and \([u\Delta y]\) is the predication of \([u\Delta v]_v\) of \(y\). The axiom (A11) yields immediately that \(x = y \rightarrow [u\Delta x] = [u\Delta y]\). Denoting \([\exists v(-v\Delta u \land z \equiv [w = v \lor w\Delta u])_w]_z\) by \(s\) we get by the same axiom that \(x = y \rightarrow ([u\Delta x \land s])_z = ([u\Delta y \land s])_z\) where we first apply (A10) to obtain the expansions \([u\Delta x]_z\) and \([u\Delta y]_z\). Finally we apply the same axiom for existential generalization to obtain as desired.

Finally we must show that \(\forall z(I(z) \rightarrow \forall x(NNx \rightarrow x\Delta z))\). The result follows easily from quantifier rules and axioms from \(I(z) \rightarrow (NNx \rightarrow x\Delta z)\) which is just the tautology
\[
A \rightarrow ((A \rightarrow B) \rightarrow B)
\]
Let \(A(x) \equiv_{df} NNx \land \exists y. (NNy \land y = S(x))\). Consider the following predication axiom
\[
x\Delta [A(u)]_u \leftrightarrow A(x)
\]
Then according to Bealer we have:

**Theorem 11.2** Quality and Concept, p127. *If we add the above predication axiom the complete theory for the structure <NN,=,S,0> is derivable in T2.*

Frege’s Law V is generally blamed for the paradoxes. But the paradoxes can be avoided if the law is retained but higher-order variables and quantifiers are dropped, for then we obtain the sound and complete system of first-order logic with extensional abstraction (T1 with modal collapse), law V corresponding to axiom (B).
12 Conclusion and Future Work

We have given detailed proofs of the soundness and completeness of Bealer’s systems T1 and T2 as well as simplifying the original presentation of Bealer Decomposition in [1]. There are many other results which are stated (with or without a sketch of a proof) in Quality and Concept as well as open problems. There is a whole avenue to explore regarding the addition of ”predication” axioms to T2. In particular, we should give a detailed proof of Theorem 11.2 of the previous section. In [1][pp.67 - 68] Bealer proposes a way of combining T1 and T2 into a single logic. Here equality means T2 equality (which is finer) and T1 equality is represented by an additional equivalence relation \( \approx \). One problem with proving soundness and completeness seems to be that we need to use Bealer’s lemma. But in this case Bealer lemma will be an identity involving \( \approx \) rather than =. Thus it would seem that we need to add an extra condition on the extension functions: \( x \in \mathcal{H}y \) and \( z \approx x \) implies that \( z \in \mathcal{H}y \). To prove the completeness of system T2’ seems to the most logical follow-up to the present work. Another interesting aspect would be to abandon the rather cumbersome Hilbert-style axiomatic system and formulate a natural deduction version of T1 and T2 instead. The major challenge in Quality and Concept is to find an axiomatization and prove soundness and completeness for Bealer’s System T3. Syntactically T3 is like T1 and T2 except that there are now two distinct term-forming operators \([A]_\lambda\) and \([A]_\xi\). Hence we will have two version of the Bealer operators and the semantics must have two sets of the corresponding functions. We now give a brief descriptions of the structure of the models envisioned by Bealer.

According to Bealer the building blocks of reality consist in primitive qualities, connections and conditions. Bealer, in his realist, anti-representationalist approach in Quality and Concept, ch.8, argues that we should consider two sets of syntactic operators, one for forming relations, properties and conditions (pertaining to objective phenomenality) out of the primitives \( C, I, N, E, U, R, P_n, K\) and \( C', I', N', E', U', R', P'_n, K'\) for forming thoughts and ideas (pertaining to the subjective or conceptual), which includes Cambridge properties. Our thoughts are ultimately grounded in reality itself: they should possess a decomposition tree into objective phenomenal components.

We now consider the model-theoretic counterpart in which to the two classes of syntactic operations \( C, I, N, E, U, R, P_n, K\) and \( C', I', N', E', U', R', P'_n, K'\) correspond to operations \( c, i, n, e, u, r, p_n, k\) and \( c', i', n', e', u', r', p'_n, k'\) on a set \( D\) partitioned into disjoint sets \( D_i\) corresponding to the different arities of intensional terms. We will call the first set of operations primary entity operations \( o_2\) and the second set secondary entity operations \( o_1\). Secondary-entity operations must satisfy injectivity, disjoint range and non-cycling conditions (Postulate I). In \( D\) we consider two fundamental domains: secondary entities \( E_2\) which are defined as the range of \( o_2\) and primary entities \( E_1\) defined as the range of \( o_1\). Given a \( d \in D\) we can seek for various possible decompositions through the inverse of the operations \( o_1\) and \( o_2\). Note that such decompositions do not need to follow the order of the (in this case generalised) Bealer decomposition. We denote a decomposition by \( o_1(d_1, ..., d_k)\) for \( d_i \in D\) and we can distinguish those \( d_i\) which are the second or further argument of a \( P_n\) or \( P'_n\) and those that are not (Bealer calls these subject nodes). Our decompositions may involve the intertwining of both \( o_1\) and \( o_2\).

Then there are two kinds of ”primes”, those that cannot be decomposed in terms of \( o_1\), which we call simple primary entities \( \mathcal{P}_1\) and those that cannot be decomposed in terms of \( o_2\) which we call secondary primary entities \( \mathcal{P}_2\). For each of these sets we can consider their restrictions to the various \( D_i\). We have obviously

\[
D = E_1 \cup E_2 \cup \mathcal{P}_1 \cup \mathcal{P}_2
\]

Clearly \( \mathcal{P}_1 \cap E_1 = \emptyset \) and \( \mathcal{P}_2 \cap E_2 = \emptyset \). We put \( C = E_2 \cap E_1\) (Cambridge entities). What is the relationship of \( \mathcal{P}_2\) to \( E_1\)? We consider the \( E_1\) which are not Cambridge entities, that
is, $\mathcal{P} = \mathcal{E}_1 \cap \mathfrak{P}_2$, to be primitive entities which cannot be further decomposed via an abstract thought-operation.

The other postulate (Postulate II) is that all secondary entities must admit a decomposition $o_2(d_1, \ldots, d_2)$ in which the $d_i$ are either in $\mathcal{P}$ or subject nodes. It would be interesting to understand better the significance of this postulate of Bealer’s.

We note that $\mathcal{P}$ need not be closed under the operations $o_1$. If we consider its intersection with the various $D_i$ we get primitive beings for $i = -1$, propositions for $i = 0$, qualities for $i = 1$ and connections for $i > 1$. A part $\mathcal{M} \subset \mathcal{P} \cap D_{-1}$ consists of minds, those beings capable of entering into transcendental connections $[1][245-248]$.

After the publication of [1] in 1982 Bealer continued to develop his theory of PRPs for over two decades, in particular in [3] [7] where the distinction between platonic and non-platonic modes of presentation is made as well as a further definition of ”logical form”. It is clear that an analysis of different forms of syntactic decomposition of formulas and terms plays a central role. It seems natural that a combination with approaches which focus (in a non-nominalistic way) on the constructive or algorithmic meaning of the syntactic structure of expressions, such as those of Tichý [29, 30] or Moschovakis [24], are likely to prove fruitful is bringing the theory of PRPs to a conclusive and satisfactory form.

### 13 Bealer’s Logic and Operads

Bealer introduces various syntactic operations on intensional abstracts. We can present these in alternative form. Let $I(n)$ denote abstracts with $n$ variables. Then we have an action $\sigma$ of the symmetric group $\Sigma_n$ on $I(n)$ which simply switches the order of the variables. Thus if we had $[R(x, y)]_{xy}$ then $\sigma(12)$ yields the inverse relation $[R(x, y)]_{yx} = [R(y, x)]_{xy}$ where we consider equivalence modulo renaming bound variables. Then we have composition

$$\kappa : I(n) \times I(k_1) \times \ldots \times I(k_n) \to I(k_1 + k_2 + \ldots + k_n)$$

Let us illustrate this for $I(1)$ and $I(2)$.

We have

$$\kappa([P(x)]_x, [Q(y, z)]_{yz}) = [P([Q(y, z)])]_{yz}$$

However we would like to obtain terms such as $[P([Q(x)]_x, y)]_y$. This suggests we consider

$$I(0) \leftarrow I(1) \leftarrow I(2) \leftarrow \ldots \leftarrow I(n) \leftarrow I(n + 1) \leftarrow \ldots$$

That is, an element of $I(n + 1)$ can always be considered an element of $I(n)$ in which we ignore the inner most variable in the list.

If we wished to obtain an symmetric operad over $\text{Set}$ we need an identity!

This suggests we reformulate the language as follows. We are given a plain intensional term-forming operator $[\phi]$ where $\phi$ is a formula and $\lambda$-abstraction of any term $t$. Thus we interpret $[\phi]_{x_1, \ldots, x_n}$ as $\lambda x_n \lambda x_{n-1} \ldots \lambda x_1.[\phi]$. $I(n)$ consists of terms which are preceeded by at most $n$ lambdas. But lambdas can apply to any terms, including variables. Our operad unit is just $\lambda x.x \in I(1)$.

The problem here is that can variables be ’entangled’ (Bealer’s ‘reflection’). How would we obtain $\lambda x.[R(x, x)]$ as an image of compositions and permutations?

If we temporarily restrict ourselves to non-entangled abstracts over atomic formulas (those that can be obtained via composition and permutation) then we get indeed an operad $\mathcal{I}$ in $\text{Set}$. Bealer’s intensional models which consist in domains $D_i$ and maps between them can be easily adapted so that models become morphisms of $\text{Set}$- operads $\mathcal{I} \to \mathcal{D}$. 

36
The above considerations suggest we extend the concept of operad to deal with 'contractions' or entanglements. Let $O$ be an operad over $\text{Set}$. Then we need morphisms

$$\rho : O(n) \to O(n - 1)$$

for $n > 1$.

Let us say intuitively that $\rho$ identifies (contracts) the last two variables, that is, it takes an operation $f : X_1 \times \ldots \times X_n$ and gives us an operation $\rho(f) : X_1 \times \ldots \times X_{n-1}$ given by $\rho(f)(x_1, \ldots, x_{n-1}, x_n) = f(x_1, \ldots, x_{n-1}, x_n)$. Other identifications can be obtained via composition with $\sigma$.

Hence for the Bealer $c$-operad we would have $\lambda x. [R(x, x)] = \rho(\lambda x \lambda y. [R(x, y)])$.

Let us call this new operad a $c$-operad (short for contraction operads).

TO DO: set up axioms expression how $\rho$ relates to $\kappa$ and $\sigma$ and $I(1)$.

References

[1] Bealer, G. (1982). Quality and Concept, Oxford University Press.

[2] Bealer, G. (1983). Completeness in the theory of properties, relations, and propositions, Journal of Symbolic Logic 48 (2): 415-426.

[3] Bealer, G. (1994). Property theory: the type-free approach v. the Church approach, Journal of Philosophical Logic 23: 139-171.

[4] Bealer, G. (1993). A solution to Frege's puzzle. In J. E. Tomberlin (Ed.), Philosophical perspectives 7: Language and logic (pp. 17–60). Atascadero, CA: Ridgeview Publishing Company.

[5] Bealer, G. (1998). Propositions. Mind, 107(425), 1–32.

[6] Bealer, G. and Mönich, U (1989). Property Theories, In Dov Gabbay & Franz Guenthner (eds.), Handbook of Philosophical Logic, Volume IV. Kluwer Academic Publishers. pp. 133-251.

[7] Bealer, G. (1989). Fine-Grained Type-Free Intensionality, in Gennero Chierchia, Barbara H. Partee & Raymond Turner (eds.), Properties, Types, and Meaning, Volume 1. Kluwer Academic Publishers. pp. 177-230.

[8] Bressan, A. (1972). A General Interpreted Modal Calculus, New Haven: Yale University Press.

[9] Carnap, R. (1947). Meaning and Necessity, Chicago: University of Chicago Press. Enlarged edition 1956.

[10] Church, A. (1951). A formulation of the logic of sense and denotation, in P. Henle (Ed.), Structure, Method and Meaning, New York: The Liberal Arts Press. pp 3–24.

[11] Church, A. (1973). Outline of a revised formulation of the logic of sense and denotation (part I), Noûs, 7: 24–33.

[12] Church, A. (1974). Outline of a revised formulation of the logic of sense and denotation (part II), Noûs, 8: 135–156.

[13] Cresswell, M. J. (1973). Logic and languages. London: Meuthen.
