Alternative Technologies™

Logic for Serious Database Folks Series

by David McGoveran, Alternative Technologies

LEGAL NOTICE AND LICENSE

IMPORTANT CONTENT NOTICE

It is anticipated that many readers of this document will have some familiarity with the writings of other authors, especially but not limited to E. F. Codd ("EFC"), C. J. Date ("CJD"), and H. Darwen ("HD"), or familiarity with Date and Darwen's The Third Manifesto ("TTM") and related writings. Material in this series will not comply with all of TTM and definitions therein, nor with any other interpretation of the work of EFC. Readers should not assume otherwise of an independent work. I will, from time to time, refer to specific passages in the works of EFC, CJD and HD for purposes of commentary or to illustrate some issue. When I disagree with a some construct by EFC, CJD, or HD and am aware of it, I will say so and give a justification.

CONTACT BY REVIEWERS

Reviewers are encouraged to convey their comments and criticisms to me directly via email addressed to **mcgoveran@AlternativeTech.com**., and including "REVIEW" in the subject line. Please provide document title, revision date, and page when referencing any passage. I will do my best to respond in a timely manner and will try to answer specific questions if there is a reasonably short answer.

Chapter 4: Formal Interpretation
Logic for Database Folks Series

by David McGoveran, Alternative Technologies

"*The logic of the world is prior to all truth and falsehood.*" **-- Ludwig Wittgenstein**

I. INTRODUCTION

So far in this series we have discussed the structure of formal systems at a high level, the fundamentals of meta-languages with attention to set-theoretic terminology, and the first of two major *structural components of formal systems*, the Deduction Subsystem. The structural split of formal systems into a Deduction Subsystem and an Interpretation Subsystem is pedagogical, being constructed and defined to clearly separate elements having a purely syntactic function from elements having a semantic function. We now turn to the second major and semantic component, the Interpretation Subsystem.

The Interpretation Subsystem provides the apparatus that enables us to apply a formal system by interpreting it in terms of some subject. Through interpretation, a formal system becomes useful. Interpretation imbues the system with semantics – i.e., with meaning. Formal interpretation permits us to create or identify an initial well-defined model of the Deductive Subsystem and then, assuming the formal system is *constructive*, engage in iterative refinement. We will focus on interpretation of formal logical systems, following with comments on the interpretation of more general formal systems. Although we will not be ready to apply formal interpretation to some particular discipline like database theory (let alone representing a specific subject via a database design) until we have learned much more, interpretation is the essence of relational database design and formal interpretation can make it closer to science than to the art database design is normally considered to be.

II. AN OVERVIEW OF THE INTERPRETATION PROCESS

As noted earlier, the *study of formal systems* divides into two disciplines, the first being called *proof theory* and a second discipline – that pertaining to the present article – being called ***model theory***. An interpretation gives *meaning* (called an *interpretive semantics*) to the structural components of a particular Deduction Subsystem by relating them to some subject. Until assigned a distinguishing interpretation, any non-logical symbol of a given type in an object language may be substituted for any other. An interpretation restricts such substitutions, causing some to be judged true and others to be judged false in terms of the subject. A proposed interpretation is a ***permissible interpretation*** only if the object language and subject are compatible in every way. The *permissible interpretations* of expressions incorporating non-logical symbols are studied in *model theory*. Model theory examines semantic elements (such as those conveying meaning and truth) of a *subject language* (i.e., the language used for expressions about the subject), and how those semantic elements may be represented using syntactic elements (i.e., symbols, terms, formulas and proofs) of a corresponding object language (the language used for expressions within the formal system). Model theory's objects of study are, unsurprisingly, models of uninterpreted formal systems (a.k.a. theories) expressed in a formal language. As a result of such study, we can identify certain semantic properties of a formal system and relationships between those semantics properties and the purely syntactic properties of that formal system.

An application of model theory comprises a subject, a subject language, an evaluation language, a set of rules of correspondence, and an evaluation procedure, relating them to an *uninterpreted formal system* to create an *interpreted formal system*. The *subject language* is the "native" language used within the subject. Its terms denote subject entities and its operators denote subject relationships either about or among subject entities. The evaluation language is used to express, formally, an evaluation or judgment regarding an object language expression in light of its interpretation in terms of the subject language. The evaluation language will have a vocabulary that includes a set of truth values, a set of semantic operators, and a set of grouping indicators. The rules of correspondence will include a set of meaning assignments and a set of truth assignments.

A subject system[1] is either an informal system or a formal system having elements and relationships among (or operations on) those elements. A subject system to which a formal logical system (a.k.a. a *deductive theory*) may be applied (successfully[2]) is said to be a ***model*** of the theory. We call the portion of a formal system that includes a formal language and a set of terms and wffs in that formal language that can be assigned truth values, an ***interpretable theory***. We call a structure (e.g. an interpretation) that satisfies an interpretable theory is called a ***model***

[1] We will often use the term "subject" in place of "subject system" when it is clear we are referring to a system.
[2] We will explain what we mean by "successfully" below.

of the theory. The model must entail at least one (but possibly many) instantiation of the theory's variables by assignment of values corresponding to entities in the subject. A model is often discussed using an informal meta-language which incorporates and relates expressions in the formal system's *object language* to expressions in the *subject language*.

How the model is to be understood as an application of the theory is established through *rules of correspondence*. A ***rule of correspondence*** is a binary relation that relates or *interprets* a symbol of the theory (including those representing operators and relationships) to one or more elements of the model (including operations and relationships). At most one rule of correspondence interprets any particular symbol. A particular set of rules of correspondence that interpret each of the symbols of a theory constitute an ***interpretation*** of the theory. The study of interpretations is called ***formal semantics***. In mathematical logic, the appropriate discipline is called ***model theory***.

If all the symbols of the theory are interpreted, the rules of correspondence constitute a ***full interpretation*** of the formal system and we say the formal system is ***fully interpreted***. If at least some but not all of the symbols of the formal system are interpreted, the set of rules of correspondence constitute a ***partial interpretation*** of the formal system and we say the formal system is ***partially interpreted***. In practice, it must be understood that formal systems are often defined with an infinite number of non-logical symbols, so that it is expressively powerful enough for any finite subject. A specific subject will typically have only a finite number of entities, properties and relationships, so the formal system is redefined (often implicitly) to have a finite number of non-logical symbols. This "restriction" of the formal system creates what is called a *fragment* of it. The fragment may or may not have properties that differ significantly from those of the original formal system. Similar comments apply when the subject does not obey relationships corresponding to all of the axioms of the formal system.

Once a formal logical system has been interpreted, a fully interpreted formula can be evaluated according to an ***evaluation procedure***. The first step of the evaluation procedure is to translate the terms of the formula into the ***evaluation language***. The resulting interpreted and translated formulas of the theory constitute *evaluation language expressions*. The evaluation language has different vocabulary and formation rules than those of the object language.

For formal logical systems, expressions in the evaluation language evaluate to a *truth value;* that is, a truth value is computed for the expression. Truth valued component terms of the evaluation language expression are each assigned a *truth value* in accordance with the definition of the evaluation procedure. Each assignment of a truth value to a term is called a ***truth value assignment***. A particular truth value is assigned to a truth valued expression according to a *theory of truth*[3]. For example, the degree to which the relationship or property denoted by the expression is asserted to hold in the subject might, according to a common theory of truth,

[3] Examples of *theories of truth* are given below.

correspond to **/T/** if it holds and **/F/** otherwise. Likewise, the object language ***n***-ary operators (on terms) are translated into their truth valued counterparts, which we will call ***semantic operators*** to distinguish them from the logical operators of the object language.

Once an interpretation has been supplied, it is tempting to use the object language when speaking of the model. Although writers often fall prey to this temptation, it is incorrect to do so and is the cause of many subtle errors. Many formal concepts (e.g., truth, falsity, correctness, completeness, etc.) refer to relationships between certain aspects of the theory and/or the model, and should not be confused with similar concepts in either the model or in common informal language. Such confusion abounds. For example, some writers on relational theory assert that a DBMS, as an implementation of a formal logical system, cannot "know" about the real world and so cannot be expected to portray what is or is not true. This argument confuses various notions of true and false, validity and invalidity, and object language expressions vs. their evaluations. Ultimately, the argument is irrelevant because results from the DBMS are exclusively within the object language. It is only through formal interpretation that any asserted relationship to a concept (e.g., in the physical world) is possible. Then, relationships encoded in the formal logical system and enforced by its rules of inference either reflect corresponding relationships among elements of the subject or they do not. If they do, we say the interpretation is ***faithful*** and otherwise, *not faithful*. As we shall see, it is the responsibility of the database designer to make certain they do through adequate conceptual and logical modeling.

Sometimes logicians short circuit the process of identifying a model and establishing the rules of correspondence. Instead, they simply assigning truth-values to truth-valued symbols (or groups of symbols) and either call the result (or treat it in a certain abstract sense as) an interpretation. Of course, what they really mean is any interpretation that would result in this particular set of truth-value assignments. Thus, it is important to remember that a formal logical system may be used for computing truth or falsity only in light of an interpretation and an evaluation procedure.

When writing about uninterpreted formal logical systems, logicians may take yet a further short cut. They will often forego translation of interpreted object language into an evaluation language, treating object language non-logical truth value constants like **TRUE** and **FALSE** as if they were truth values like **/T/** and **/F/**, and treating operators as having an alternate definition as truth tables. In this way, the object language becomes both an algebra of deduction and a calculus of truth values. When it behaves as an algebra and when as a calculus must be inferred by example. For our purposes, such short cuts obscure important aspects of formal logical systems, their properties, and their application to database theory and practice.

Given these two disciplines of proof theory (relating to syntax) and model theory (relating to semantics), it is useful to divide formal logical systems into two portions – Deduction Subsystem and Interpretation Subsystem. The Deduction Subsystem corresponds to the subject matter of proof theory and syntax. The Interpretation Subsystem corresponds to the subject matter peculiar to model theory and semantics, which are necessarily related to the Deduction Subsystem in specific ways. We will address the components assumed by the interpretation procedure, and related issues and concepts discussed above a bit more formally and in detail below.

III. THE INTERPRETATION SUBSYSTEM

As with the Deduction Subsystem, we begin with some definitions, to be elaborated upon later. For pedagogical purposes, we will consider the Interpretation Subsystem of a formal logical system as comprising certain types of objects (see Figure 4.1): a Subject with a Subject Language, an Evaluation Language, Rules of Correspondence, and an Evaluation Procedure.[4]

How each of these is defined in detail depends on the logical system. The Subject, Evaluation Language, Rules of Correspondence, and Evaluation Procedure may be said to provide a theory with *semantics* (through interpretation).

Subjects and Models

Ordinarily, logicians think of a formal system as a theory that exists independent of any interpretation. They then look for exemplars (other systems) of this theory according to certain criteria (elucidated below). Those other systems are understood as possible ***subject systems***. When a particular exemplar is intended (whether specifically identified or not), we will refer to it as the ***subject***. If the subject's relationship to the theory is formally specified and meets all the required criteria of an exemplar, the subject is called a ***model of the theory.*** The subject system may itself be either an informal system or a formal system. If the subject is part of the alleged physical world, the model is said to be a ***concrete model***. If the subject is a formal system, the model is said to be an ***abstract model***. However informal or formal, abstract or concrete a subject may be, it is a system in its own right.

In the literature it is common practice to refer to the subject system as a *model*, and also to refer to the combination of the subject system and its interpretation (see below) as a model, relying on context to clarify the ambiguity. Although the language will be a bit stilted at times, we will try to use "model" herein when referring to the subject system that is an exemplar of a formal system, and to use "interpretation" when referring to the formal relationships between theory and model. By a subject or subject system, we will mean a system independent of any relationship to any external formal system.

[4] Other authors, if they identify semantics in terms of a distinct subsystem at all, may divide it into a slightly different set of components, or use different names for them. We prefer this one for pedagogical reasons. It will help us explain and categorize properties of formal logical systems, and some of the metamathematical results like Gödel's Incompleteness Theorems.

Subject:
A system intended to be an exemplar of a formal theory, but independent of any particular formal system.

Subject Language:
The language used to describe the subject system, comprising a Set of Denotations (i.e.., symbols or terms identifying components of the Model).

Evaluation Language:
A calculus for computing truth value relationships, having expressions formed from the vocabulary according to a formal syntax (formation rules).

Vocabulary – **"Semantic" or "Model Theoretic":**
A Set of Truth Values - symbols (e.g., /T/ or /F/) representing degrees of truth or falsity in the subject
A Set of Semantic Operators – maps from arguments to a truth value
A Set of Grouping Indicators – symbols such as parentheses and brackets

Rules of Correspondence (Interpretation):
A Set of Meaning Assignments relating elements of the subject to non-logical symbols and operators of the object language, respectively.
A Set of Truth Value Assignments assigning truth values to terms and truth value constants of the object language.

Evaluation Procedure:
A procedure for translating object language wffs into Evaluation Language expressions via Meaning Assignments, Truth Value Assignments and Semantic Operators, and then *calculating* a truth value for the resulting Evaluation Language expression.

Figure 4.1. Components of a Formal Logical System's Interpretation Subsystem.

A formal logical system is generally intended for some practical use (albeit perhaps an abstract one). For convenience, we will say that our understanding of the subject of that practical use (for example, accounting) is a subject theory. The subject theory is typically an ***informal*** *theory*, incorporating at least some informal set of rules, requirements, descriptions, and so forth about the subject. If the subject theory is sufficiently precise regarding the subject, it will rise to the level of a ***formal*** *theory*. This subject theory becomes a model of the formal logical system which, by contrast, is always a formal theory. The theory's scope is sometimes called the ***universe*** (or ***domain***) ***of discourse*** (a use of set theoretic terminology in the meta-langauge). Unless we have clearly identified the appropriate universe of discourse for an interpreted formal logical system, we cannot reason about the subject in the object language and then translate conclusions back into the subject in any reliable manner. This issue is discussed further in the articles on propositional logic and first order predicate logic, and in those articles addressing database design.

In order to apply the structures described in this series for understanding interpreted formal systems, it is necessary to establish which is the theory and which is the model. When reading the literature, what constitutes "the theory" and what constitutes "the model" is often somewhat relative. Furthermore, in certain contexts, the structure having the designated role of "the theory" may not be completely formalized. Some examples may help clarify this idea:

- Empirical Modeling: A mathematical system is sometimes said to model or be a model of some portion of the physical world, treating the physical world as if it were a formal system. The physical world is then assumed to have some, perhaps unknown, formal structure evidenced by empirical data. A so-called empirical mathematical model is then constructed along with rules of correspondence to elements of the physical world. The faithfulness of model[5] based on the rules of correspondence can then be compared to the presumed theory by observation. The mathematical model, including the rules of correspondence, are then adjusted so that the model is (hopefully) increasingly more faithful. If the mathematical model is faithful, we then often say that we have an explanatory or even a predictive theory of the physical system – in essence claiming that the model is equivalent to the formal system underlying that physical system.

- Physical Theories: A mathematical system is often said to be a theory of some portion of the physical world, implying that the physical world is a model for that theory. Although the terminology treats the theory as the more formal system and the model as the less formal, it is usually the model for which many (competing) theories are found, whereas in logic, it is usually the theory for which many models are found.

- Model Making: Similar to empirical models, model makers and other artists speak of creating a model or a caricature when sculpting or drawing. In this case, the subject of the modeling is usually something physical and is treated as the more formal system (you might say an inherently correct, immutable and unreachable theory). The model is then something the artist tries to make into a faithful representation (i.e., an interpretation) of the subject under some (perhaps intuitive or abstract) rules of correspondence.

[5] Notice that I'm using faithfulness and faithful a bit loosely. In practice, we have to identify a test that establishes sufficient correspondence to constitute faithfulness. In the practice of science, that test often incorporates consensus among scientists in the relevant community.

- Formal Models: Mathematicians and logicians often find that one formal (and usually abstract) system is can be a model for another. For example, propositional logic is a model for simple set theory and vice-versa.

- Data Modeling: A data model in the database disciplines is both an establishment of and an interpretation of the non-logical symbols of the formal logical system implemented by a DBMS[6] – i.e., the theory. In effect, it makes meaning assignments to variables that are instantiated by some set of values (the data in a specific database conforming to the model). In principle, the formal logical system may be defined with a very large but finite number of non-logical symbols. The data modeler has the task of selecting and interpreting some subset of these, thereby restricting the formal system to a smaller set of non-logical symbols. Additionally, the data modeler may establish certain axioms for the formal logical system, these being interpreted as relationships among elements of the model. More often than not, the model is intended to represent some aspect of the physical world (and so is sometimes said to be a model of the physical world as understood in the empirical modeling example above). Conflating these two uses of "model" can lead to lead to confusion (and erroneous reasoning) among members of a data modeling team, business users, database theoreticians, and so on.

The assignment of semantics to a formal logical system is not concerned with any real-world meaning of formulae, but with relationships among them under an interpretation. Logical methods are designed to preserve these relationships, no matter what meanings are assigned to the constituent terms. On the other hand, assignments that would cause the preservation of a relationship to fail are either indicative that the subject is not a proper model of the theory or, conversely, that the formal system is inadequate or inappropriate for the subject.

Subject Language

A *subject language* is the language used to make statements within the *subject*. If the *subject* is a formal system, then this language is that subject formal system's object language (to be distinguished from the object language of the theory we intend to interpret). If the *subject* is an informal system, then the *subject language* is often a natural language, or perhaps some hybrid combination of natural, artificial, and formal languages. For example, if we our subject is accounting practice, then a subset of a natural language like English might be the subject language possibly enhanced with specialized accounting terminology.

The vocabulary of the subject language comprises a set of symbols and/or terms denoting the

[6] Notice that this definition indicts any DBMS for which the formal logical system is ill-defined. For a DBMS that implements a procedural or imperative query language, the model is probably best understood as the lambda calculus (to be discussed in brief later).

entities and relationships of the subject. Those symbols and terms will be used to interpret the non-logical terms of the theory. If the subject is a formal system, then these symbols are often taken from the formal vocabulary of the subject language (e.g., the object language of that subject formal system). If the subject is an informal system, then the symbols may be created as denotations for entities and relationships in the subject language, thereby at least partially formalizing the subject system.

Evaluation Language

An evaluation language is a calculus (a formal language for computation rather than for deduction). It comprises a vocabulary and a set of formation rules. A well-formed expression in the evaluation language denotes a computational relationship among values, rather than relationships among object types as in the object language.

Evaluation Language Vocabulary

The evaluation language vocabulary consists of a set of grouping indicators, a set of truth values and a set of semantic operators. Most evaluation languages use the same grouping indicators (such as parentheses and brackets) as those in the corresponding object language and so will not be discussed here.

- **Truth Values**

The ***set of truth values*** is a set of symbols in the evaluation language. There will be two such values (e.g., **/T/** and **/F/**) for a classical *two-valued logic*, and more than two truth values for a *many-valued logic*). The number of truth values in the set must match the number of truth-value constants in the uninterpreted formal system. How one is to understand the meanings of those truth values is a semantic issue that hinges upon the *theory of truth* being used. Each *truth value* assigned to a term of the object language denotes a judgment of truth or falsity regarding the aspect of the subject that term represents (see rules of correspondence). During evaluation, each interpreted non-logical term in an object language expression, including each truth value constant, is replaced with a *truth value*.

Every formal logical system has an associated set of *truth values* which may be assigned to the non-logical terms of the system and which are used in defining the semantic operators (e.g., truth functions) corresponding to logical operators. It is only within model theory that the concept of truth values arises.

It is important to distinguish between the non-logical symbols for truth value constants (e.g., **TRUE** and **FALSE**) in the object language, those names used for values of Boolean domains (e.g., 1 and 0, “Yes” and “No”, “T” and “F”, etc.), the set of truth values (e.g., **/T/** and **/F/**) in the evaluation language that are used to represent truth or falsity under an interpretation, and those signs used in the meta-language for concepts like truth or falsity (e.g., "*true"* or "*false"*). Logic texts often do not differentiate truth value constants from truth values, and texts on programming languages often fail to distinguish between truth value constants and values of Boolean domains. When such glosses are introduced in the subject of database theory and practice, confusions about the distinction between logical truth, assertions, true propositions, and facts (observations) about the physical world arise. More consequential, when this is done in a formal logical system, the boundary between first order predication and 2nd, … nth order predication is blurred, if not lost altogether.

So-called *classical* or *traditional* logical systems are ***two-valued logics***: they have exactly two possible truth values, the canonical meaning of which is "*true*" and "*false*". For example, the propositional calculus has exactly two truth value constants, TRUE and FALSE, and two truth values **/T/** and **/F/**. These meanings of these pairs of values correspond, as may seem intuitively obvious, to our common notions of truth and falsity. To see that this is a bit arbitrary, consider that the two values could just as easily be taken to mean set membership or the lack thereof.

One type of non-classical system has a more than two truth values, with corresponding truth value constants in their object language. When a formal logical system has more than two truth value from the set, it is said to be a *many-valued logic.* To understand the role played by these other truth values better, logicians classify them as being "true-like", "false-like", or neither. Every truth value is then said to be ***designated, anti-designated,*** or ***undesignated***, according to whether it is treated as true-like, false-like, or neither. This subtlety is necessary for many-valued logics in which the notions of "degrees of truth" and "degrees of falsity" may be intended. We will return to many-valued logics and their properties in a future article. In the meantime, when a definition involving truth-values is to be understood as applying to many-valued logics, we will use the symbol **/𝒯/** for an arbitrary designated truth value and the symbol **/ℱ/** for an arbitrary anti-designated truth value with the understanding that these reduce to **/T/** and **/F/** respectively when the definition is applied to a formal logical system with exactly two truth-values.

Truth Value Notation

/𝒯/ – indicates an arbitrary ***designated*** truth value (**/T/** in the two-valued case)
/ℱ/ – indicates an arbitrary ***anti-designated*** truth value (**/F/** in the two-valued case)

Theories of Truth

In addition to subtleties regarding truth values such as those introduced by many-valued logics, systems can entail different *theories of truth* (mentioned above). A ***theory of truth*** establishes the grounds for deciding how some object in the subject (e.g., a concept, declarative statement, or a belief may be judged "true" or "false" (or possibly something in between). Theories of truth are often philosophical in nature. Two such theories, among the most common, are the *correspondence theory* of truth and the *coherence theory* of truth.

The ***correspondence theory*** of truth defines a statement as "true" if it corresponds to some actual state of affairs in the physical world. Clearly, such a definition of truth is very limited – it relies on multiple ontological assumptions regarding the nature of entities in the physical “reality”, our ability to access them (e.g., observe them), their relation to abstract constructs, and so on. The correspondence theory of truth arguably does not support models that are abstract such as formal logical systems, sets of truth values with other than two simple “true” or “false” values, and so on.

Alternatively, the ***coherence theory*** of truth requires only that the set of truth values and their assignment preserve internal coherence of the model. In some versions, it is only internal coherence of the theory as a whole. Unfortunately, this freedom comes with a price – it becomes possible to define sets of truth values that are very non-intuitive and difficult to understand, let alone use.

Generally, the judgments “true” and “false” reflect our commonly held understanding of these words. The reader is warned, however, that there exist formal logical systems in which the members of the set of “truth” values do not seem to relate to notions of truth and falsity at all. An understanding of the “truth” values depends at least on the logical system’s canonical interpretation, and possibly on the specific interpretation at hand. Such systems are abstract and, although interesting to study, difficult for most to relate to common experience. Nonetheless, they have, for example, sometimes been rather naïvely proposed as the formal logical systems to be implemented by a DBMS.

- ## Semantic Operators

For each logical ***n***-ary operator in the object language there must be a corresponding ***n***-place semantic operator. Each semantic operator transforms its ***n*** truth-valued arguments into a single truth value. Each semantic operator must correspond to an operation in the subject system pertaining to reasoning about *judgments, beliefs, assertions*, or *assumptions*. Inasmuch as the evaluation language permits computation of a truth value for any interpreted and translated formula of the theory, it constitutes a ***calculus***.[7]

A *semantic operator* in the evaluation language may or may not be *truth functional*. When semantic operators are not truth functional, the truth value of expressions may depend on syntactic and/or semantic elements other than its truth valued components or operands. For example, certain *probability logics* compute truth value in a manner which depends on how the

[7] The terms logic and calculus are often used interchangeably in the literature, but this is sloppy usage. Relationships are deduced. Evaluations are calculated. I will return to this point in more detail when discussing predicate logic vs. predicate calculus.

terms are related (joint versus independent).

All of the formal logical systems we will address in this series excepting certain many-valued logics are said to have truth functional semantics, meaning that the semantic operators are *truth functions* as defined below.

• Truth Functions

Truth functions are an important and very common type of semantic operator. A ***truth function*** is a truth-valued function of ***n*** truth valued operands which behave as formal parameters normally called "variables"[8]. The set of truth functions belongs to the *evaluation language*. Each *truth function* may be understood as a mapping from a set of ***n***-tuples of truth values to a set of unary truth values. Like other semantic operators, every truth function corresponds to an operation in the subject system pertaining to reasoning about *judgments, beliefs, assertions,* or *assumptions*. The *evaluation procedure* will replace each logical operator in an object language formula with a corresponding truth function. When the truth value of any wff or any term can be evaluated in a mechanical manner from, and is completely determined by, the truth values of its components terms, a logical system is said to be ***truth functional***.

Even a classical logic with two truth value constants has many truth functions. The number of truth functions is determined by the number of parameters. For example, suppose the formal logical system has exactly two logical constants. Then, we say that every ***n***-ary logical operator has a corresponding ***n***-place truth function with two possible truth values per operand (corresponding to the two truth value constants). There will then be $\mathbf{2\uparrow 2^{\mathit{n}}}$ truth functions in the evaluation language. <TBD: Add a table showing the number of truth functions vs. parameters.>

Although logical operators are often erroneously defined in terms of truth tables in logic texts, it is important to remember that the truth tables are not the deductive operators for which they are named. That said, some authors consider an ***n***-place truth table to be an *extensional definition* of an ***n***-ary logical operator, whereas defining ***n***-ary logical operators in terms of the rules of inference that apply to them can be considered an *intensional definition*. Strictly speaking, however, reasoning based on truth tables usually belongs to semantics (because we select an entry in a truth table based on an interpretation or assignment) and the evaluation language, while reasoning based on rules of inference belongs to syntax and a deductive object language.

A truth valued expression incorporating a truth functional operator is called a ***truth functional expression***. Given truth assignments for all the terms in a truth functional expression and a truth functional definition of any operators appearing in the expression, we can compute the truth value of the expression simply by looking up the truth value of the semantic operator for the assigned truth values of its operands.

---

[8] How this term is to be understood depends on the formal logical system in question.

Truth functions are often given (where possible) as ***truth tables***, which summarize a set of mappings or reduction rules (e.g., if "/T/ AND /F/" then /F/). A truth table is a tabular representation of a truth function (expressing a *semantic* formation rule and, possibly, certain *semantic* rules of inference) on abstract constants in the *semantic* vocabulary, namely *truth values*. They specify the operator's truth value, given the operands' truth values (see Figure 4.2), and permit the evaluation (i.e., computation) of the value of an expression in the evaluation language in accordance with the evaluation procedure.

An explanatory metaphor: In much the same way that the truth tables correspond to logical operators and serve in the evaluation of truth valued expressions, the addition and multiplication tables in the language of integer arithmetic (a calculus or evaluation language) correspond to the integer arithmetic operators "+" and "*" and serve in the evaluation of numeric valued expressions. Both logic and *algebraic* arithmetic have an algebra (or deductive language) for expressing and reasoning about relationships and both have a calculus (an evaluation language) for computing values. In both, the non-logical symbols (e.g., truth valued or integer valued) and the logical operators of an expression in the algebra (the deductive language) must be given interpretation in the appropriate evaluation language as constants and functions (as tables) in order to be evaluated or "solved".

Expression: **P OR (R AND Q)**

Truth Table:

P	Q	R	P OR (R AND Q)
T	T	T	T
T	T	F	T
T	T	T	T
T	F	F	T
T	F	T	T
T	F	F	T
F	T	T	T
F	T	F	F
F	T	T	T
F	F	F	F
F	F	T	F
F	F	F	F

Figure 4.2: The truth table for an 3-term expression.

With the foregoing in mind, suppose that we have an expression in the evaluation language (derived from a wff in the deductive language) with ***n*** distinct truth-valued symbols (corresponding to terms in the deductive language). In the absence of specific truth value assignments to these ***n*** terms, the entire expression behaves as a function with ***n*** arguments.[9] As such, it can be represented by truth table (see Figure 4.2 above).

We say that a set of operators is *independent* if it is not possible to express the truth table for any given operator in terms of the truth tables for the other operators in the set. For example, the set of truth tables (restricted to two-valued logic) for the operators “NOT” and “OR” (shown in Figure 4.3a) and the connective called *material implication* (shown in Figure 4.3b) are not independent, since material implication has the same truth table as “(NOT P) OR Q.” Readers can show this fact by substitution using the truth tables in Figures 4.3a and 4.3b for “NOT” and “OR.” In fact, the standard truth tables for “AND,” “OR,” and “NOT” also fail the test of independence. The truth tables shown in Figure 4.3a and 4.3b are constructed according to our intuitive understanding of these logical operations.

AND		
P \ Q	T	F
T	T	F
F	F	F

OR		
P \ Q	T	F
T	T	T
F	T	F

NOT	
P	NOT P
T	F
F	T

FIGURE 4.3a. Two-valued truth tables assigned to AND, OR, and NOT.

IMPLIES		
P \ Q	T	F
T	T	F
F	T	T

BI-IMPLIES		
P \ Q	T	F
T	T	T
F	F	T

FIGURE 4.3b. Two-valued truth-tables assigned to IMPLIES (material implication) and BI-IMPLIES (material equivalence).

• Evaluation Language Formation Rules

As with the object language of the Deduction Subsystem, evaluation language expressions in the Interpretation Subsystem are said to be *well-formed* if they can be created from vocabulary

[9] This realization is due to Frege.

terms or other expressions according to evaluation language *formation rules*. Typically, the evaluation language formation rules are an abstraction of the formation rules for the object language. As such, they are satisfied automatically when, in an interpreted wff of the object language, truth values and semantic operators are substituted in place of truth valued terms (including truth value constants) and logical operators, respectively. The substitutions for atomic terms are to be made according to truth value assignments.

Rules of Correspondence

A formal logical system has no meaning until it is interpreted in the context of a subject. A *specification* of exactly how a uninterpreted formal logical system (a theory) and a subject system interrelate is called an ***interpretation***. Such a *specification* is given by a set of *rules of correspondence*. Interpretation provides the semantics of the theory. The formal mechanism by which an interpretation is spelled out is a set of *rules of correspondence*. ***Rules of correspondence*** establish assignments of components of the subject to components of the theory and, in particular, to the theory's non-logical symbols. Note that a rule of correspondence is a bi-directional relationship. Given rules of correspondence for each of the non-logical symbols of a language or theory, a logical system fixes the interpretation for all compound expressions in that language via connectives and the rules of inference.

If, under an interpretation, every symbol and every derivable wff of the formal logical system's object language has a corresponding entity or relationship in the subject, and if every axiom and derived wff that evaluates to $/\mathcal{T}/$ corresponds to a component of the subject that is judged *true*, we call the subject a ***model*** of the formal logical system. Such an interpretation is said to be a ***faithful interpretation*** (of the object language).

Typically, the developers and users of a theory will have one or more interpretations in mind, referred to as the theory's ***intended interpretations***. In deference to the person or community establishing the formalism usually has in mind an intended interpretation, and as they are the originators of the formalism, the intended interpretation rises to the level of, and is often referred to as, a ***standard interpretation*** or ***canonical***[10] ***interpretation***. It is common practice to use an intended interpretation as a kind of "acid test" regarding the concept of *validity* as applied to the formalism's object language: The intended interpretation must provide a faithful interpretation of the object language, else the formalism is not "successful" and needs changing. It is the intended interpretation that one is expected to keep in mind when learning the formal logical system.

The formal logical system is interpreted at most once for any deduction and cannot be changed throughout that deduction (or any set of deductions to be related to that model). Formal interpretation takes place only via explicitly stated rules of correspondence.

[10] Canonical is used here in the senses of "generally accepted", "primary", and "is more or less the most simple without losing generality".

For example, in the art of programming, a business process or activity in the "real world" often plays the role of an informal model. The program that is designed to symbolize the entities, data, and operations in that business process or activity form a corresponding theory. Similarly, when designing a relational database pertaining to some business activity, it is the database design and the relational model that constitute the theory. A dictionary is required to record the rules of correspondence.

Rules of correspondence also establish a formal association between the operators in the formal system and relationships (or operations) among objects or concepts in the subject. Within the study of formal logical systems, this type of assignment is of *a semantic* character. If the model is informal, then from an epistemological view, a rule of correspondence is essentially an *explication* in which a somewhat vague *explicandum* (given in the subject language of the informal model) is related to a precise *explicans* in the object language of the formal theory. If the set of explications are <u>successful</u> at preserving the observed relationships, operations and perceived truths of the subject, we obtain an *explanatory theory*. Alternatively, we sometimes say the theory faithfully represents the subject.

If a formal logical system includes *free variables*[11] in its wffs, a meaning assignment replaces the variable with a constant, representing an individual in the subject, and the variable is then said to be ***instantiated***. Often logicians ignore the interpretation and simply use instantiation in the abstract with a arbitrary constant symbol. Once all the free variables in a term (belonging to a wff) have been instantiated, that term may be given a *truth assignment* as described below.

We define a ***valid interpretation*** of a theory as one that makes each axiom of that theory evaluate to /$\mathcal{T}$/ (or some other designated truth value if a many-valued logic) under the *evaluation procedure*. Every intended interpretation of a theory must be a valid interpretation. In other words, for an interpretation to be an *intended interpretation* of the theory, the axioms of the system must evaluate to /$\mathcal{T}$/.

Note that in both programming and relational database design, the process is somewhat the reverse of the usual logician's activity: One starts with the subject, and then presumes it is a model of some fragment of the more general formal logical system (e.g., the lambda calculus for computationally complete programming languages or the relational algebra for relational databases). In each case, the designer selects a portion of the abstract vocabulary of the object language as those symbols that must be interpreted.

Although the perception is one of creating that vocabulary by declarations and definitions, the abstract vocabulary is defined by the object language (including potential variable names and

[11] A ***free variable*** is one for which a specific value or constant can be substituted. By contrast, a ***bound variable*** ranges of a predefined set of values. In some systems, all variables are free (there being no notion of bound).

constant names). When the designer specifies a particular name to represent a business quantity, they are constructing the vocabulary subset that will be interpreted (often by some user interface). This is sometimes confusing when applying formal logical systems to programming or database theory and practice, because we often speak of the program or database as being a model of something in the physical world.[12]

Evaluation Procedure

An *evaluation procedure* is a mechanical process by which an *evaluation* (a.k.a., *truth valuation*) of an interpreted object language wff is established. It is used to compute the truth value of an instantiated object language wff mechanically from the truth value equivalents of the operators and operands in that wff. The first step in the determination of the truth or falsity of a wff (the computation step of evaluation) requires the wff first be translated into the *evaluation language*. The *non-logical symbols* of a formal system include certain symbols used to represent truth-valued formal objects. Additionally, certain *terms* of the formal system also represent truth-valued formal objects. The process using these facts to evaluate a wff may or may not be trivial depending on the complexity of the formal logical system and the wff to be evaluated.

The procedure may be specified in the form of a specific algorithm, as a set of evaluation rules, or by some other well-defined (i.e., unambiguous) method. It is the process of translating (via truth value assignments and truth function assignments) an *interpreted object language* wff into an *evaluation language* expression, and then calculating the truth value of that expression according to that calculus. Informally, the *evaluation procedure* [13]involves:

1. interpreting the object language wff by substituting, according to the rules of correspondence, denotations of components of the subject for the variables of the wff (known as *instantiation*),
2. substituting truth values for the truth value constants in the interpreted wff ,
3. substituting semantic operators (usually truth functions) for the logical operators in the interpreted wff, and,
4. replacing complex expressions with the corresponding truth values as defined by the semantic operators (e.g., truth tables), working from the inside out as defined by grouping indicators.

From time to time, logicians are presented with one or more compound sentences in the object language, and then try to determine which possible truth value assignments will *satisfy* those sentences, given truth functions in place of the operators. This process is called *reverse evaluation* and is similar to solving an algebraic expression with some variables assigned values and one or more other variables treated as unknowns.

[12] This is a confusion I warned against earlier.
[13] We will see specific examples in the articles on propositional logic and first order predicate logic.

Again, evaluation belongs to model theory inasmuch as it involves the *set of meaning assignments*, *set of truth values*, the *set of semantic operators (e.g., truth functions),* and the *set of truth value assignments, all of which are semantic in nature*.

Meaning Assignments

For each non-logical free variable in the object language formula, the appropriate variable must be instantiated in a manner consistent with the appropriate rule of correspondence. In this context, a rule of correspondence that establishes a formal association between a non-logical symbol in the formal system and an object or concept in the subject is called a ***meaning assignment***. Thus, the practical use of a formal logical system is implemented through meaning assignments for the non-logical symbols of the vocabulary. Once meaning assignments are used to instantiate a wff, it should be possible to assign to each truth-valued variable or term an obvious truth value constituting a *judgment* about the corresponding aspect of the subject. For example, if the object language were elementary algebra, then each variable in an algebraic formula with equality (i.e., an equation) would be replaced with an appropriate number consistent with the interpretation.

Abstraction in Meaning Assignments

Abstraction in an interpretation may be used to capture concepts ordinarily considered impossible to express in the object language. Although we do not yet have all the tools in place to understand the interpretation of symbols, I want to caution the reader that the expressive power of a formal logical system does not limit the possible meanings assigned to non-logical symbols.

A non-logical symbol in *propositional logic* can represent anything that can be judged true or false in the subject, however simple or complex. These may include, for example, formulas in a language that is expressively more powerful than propositional logic. For example, it is not possible to express arithmetic in propositional logic. However, it is possible to designate any number of non-logical symbols of propositional logic as representing selected truths (falsehoods) of arithmetic, one symbol per true (false) arithmetic equation. We will discuss this topic further in later articles.

Likewise, an n-ary predicate or a constant in *first order predicate logic* can represent, in the model, facts that have an extremely complex justification. While that justification might require a *higher ordered predicate logic* to express symbolically, that fact is irrelevant because such justifications are in the language of the model, not of in the object language of the formal logical system! This is a powerful *abstraction property* of formal logical systems and one to which we will return when discussing the desirable properties of so-called declarative data sublanguages relative to general purpose imperative programming languages. Again, we will discuss this topic further in later articles.

Truth Value Assignments

A ***truth value assignment*** is a type of rule of correspondence which assigns a unique truth-value (corresponding to a judgment from the set of judgments in the subject) to some component of the object language. That component can be any of certain non-logical symbols (e.g., truth valued constants), terms, and formulae. The formulas and terms of a proof theory are often understood as being subject to truth assignments in the model theory.

It is only via a prior interpretation of any symbol in a term that an assignment of a truth value to that term is anything other than arbitrary. An interpretation for a specific formula is therefore a set of meaning assignments (in effect, one for each symbol of the formula) so that the corresponding relationships expressed by the formula are judged "true" in the subject system and so assigned and evaluated as "/$\mathcal{T}$/" in the evaluation language. Again, *meaning assignments* to free variables are effectively instantiations – they select a particular member of the relevant domain and then associate it with a element of the subject system. Writers often rely on context to differentiate an interpretation of an entire theory from a interpretation of a specific formula.

Care should be taken to differentiate the symbols in the formal system for truth-value constants (e.g., "TRUE", "FALSE") from the semantic notion of truth they will represent under interpretation (*true*, *false*) and evaluation (/$\mathcal{T}$/, /$\mathcal{F}$/). Similarly, one must differentiate between truth-value assignments and the related meaning assignment. Having failed to differentiate these concepts, logicians, both amateur and professional alike, often treat the both assignment and manipulation of truth-values as part of the formal logical system. While this treatment may have a formal justification for some formal logical systems, in general it does not.

Just as misleading, a specific set of truth value assignments may be treated as if it were an interpretation, even when the subject has not been identified and rules of correspondence that assign meanings to the non-logical symbols of the theory have not been explicated. For more on this topic, see the section below on Formal Logical Systems as Calculi.

For a particular logical system, the literature often uses the foregoing gloss. It will refer to assigning a truth value to each and every truth-valued variable, constant, or term in a set of wffs as an interpretation of that set of wffs. Each assignment may be compound and perhaps even very complex. The following is an example of such an "interpretation" of wff in propositional logic:

Given Wffs:	"P AND Q," "P OR Q"
Truth value assignments:	"P" is assigned /T/; "Q" is assigned /F/.

FIGURE 4.4. Example of The Truth Value Assignment "Gloss"

Note that in some logical systems (including, in particular, the system supposedly underlying the query language SQL), a wff may consist of truth-valued expressions containing variables that are not themselves truth-valued, but whose assigned values imply the truth or falsity of the expression**.** The interpretation of such expressions then includes the values assigned to these non-truth-valued variables.

Logic texts often limit discussion to a type of formula called a *sentence*. In general, a ***sentence*** is a wff to which a truth value (e.g., /$\mathcal{T}$/ or /$\mathcal{F}$/) may be assigned. To evaluate the truth value of a sentence requires reference to a specific interpretation or model of the theory (often called a structure in first order theories). The remainder of this subsection anticipates terminology that will be introduced formally in later articles.

In propositional logic, every wff is a sentence. In predicate logic, only a ***closed wff*** – a wff with no free variables or in which every free variable has been interpreted (assigned a value) – is a ***sentence***. A ***non-closed wff*** contains free variables and is possibly an open wff. An ***open wff*** contains no quantifiers. An open wff has no truth value until its free variables are assigned values under an interpretation thereby yielding a sentence which is a ***substitution instance*** of the wff. It is important to note that "sentence" may have other uses in some formal logical systems such as first order predicate logic.[14]

Under a class of interpretations as determined by a set of truth value assignments, the evaluation of a wff might be either *tautology-like* or a *contradiction*. Within any formal logical system, a wff is ***tautology-like*** if it evaluates to a true-like truth value for every interpretation,[15] assuming it can be evaluated[16]. In systems in which it can be evaluated, the wff is a ***tautology***. In other words, the wff evaluates to a true-like truth value for every possible combination of truth assignments to its components. For example, according to our usual understanding of two-valued logic, "P OR (NOT P)" is always true, irrespective of the meaning assignment or truth assignment (i.e., whether assigned true or false) of "P". Given our definition of axiom, it follows that the axioms of a logical system are tautologies under an interpretation. It is important to understand that, in many formal logical systems, only certain wffs can be a tautology.

By contrast with tautology, a ***contradiction*** is a wff that evaluates to a false-like truth value under every interpretation. In other words, the wff evaluates to a false-like truth value for every possible combination of truth assignments to its components. As with the example of a tautology, and according to our usual understanding of two-valued propositional logic, "P AND (NOT P)" is always false, irrespective of the meaning assignment or truth assignment (i.e.,

[14] For example, a formula may be designated as an "open sentence" or a "closed sentence" in the system of first order predicate logic. We will revisit and explain such alternative uses as the need arises.
[15] However, tautology may not be the preferred designation for technical reasons. For example, in first order predicate logic, the equivalent concept is to refer to the formula as ***valid***.
[16] In first order predicate logic, only certain wffs are tautology-like. The more general concept of *validity* is used.

whether assigned true or false) of “P”.

Truth Function Assignments

Each logical operator in the Deductive Subsystem is assigned (but is not the same as) the set of rules of truth evaluation of a corresponding semantic operator. If the semantic operator is *truth functional*, its rules define a *truth function* and so the assignment is called a *truth function assignment*. Semantic operators have a computational definition and belong to model theory. Even when a logic text does not make evaluation explicit and treat a truth function as a secondary definition of a logical operator, the truth function provides a *canonical semantics* for that logical operator. The truth function definition of logical operators becomes an abstract interpretation in that they define how the logical operators are to be evaluated, irrespective of the particular interpretation, when their parameters are assigned a truth value.

It is useful to use different symbols for logical operators in the object language on the one hand and their semantic operator counterparts in the evaluation language on the other. For example, as a matter of typographical convenience, if we need to distinguish between the logical operator and its assigned truth function, we could indicate the truth function with the English name for the operator to which it is assigned but underscored. Thus, when referring to the truth function assigned to the propositional logic dyadic operator OR, we would use $\underline{\text{OR}}$.

Computation

The computation step of an evaluation procedure computes a truth value for the evaluation language expression that has resulted from the foregoing translation steps. Respecting grouping indicators (or equivalent rules regarding operator precedence), each truth function is evaluated given the truth value assignments to its parameters. This process proceeds from the inside out, evaluating the most deeply nested truth functions first, progressively creating new expressions until a single truth value results.

IV. INTERPRETATION OF GENERAL FORMAL SYSTEMS

When the formal system to be interpreted is not a logical system, evaluation results in something other than a truth value. For example, most systems of mathematics involve the assignment of instances of some class of mathematical object to the variables of the formulae. The class of objects relevant to a particular system may be integers, real numbers, complex numbers, vectors, matrices, tensors, and so on. The evaluation procedure applied to some (well-formed) formula will then result in a specific value of the class.

As a simple, familiar example, consider the formal system of elementary algebra. The formulae of elementary algebra (excluding equality) include operators (corresponding to the logical symbols of the system), variables (corresponding to non-logical symbols), and grouping indicators. The rules of algebraic manipulation are the inference rules of the system. When variables are instantiated with numbers, the system is (at least partially) interpreted, and results in an arithmetic equation. This arithmetic equation may then be evaluated according to the rules of arithmetic, for example by applying tables for addition, multiplication, etc. These arithmetic tables correspond to the truth tables used in evaluating logical formulae. The result of the evaluation procedure applied to a particular formula is then a number.

Of course, if we include equality among the valid symbols in algebraic formula, we introduce truth and falsity and a second part to the evaluation procedure in which two formulae are compared for equivalence of numerical value. We call the overall formula, incorporating at most one equality symbol between two formulae, an equation. We then say the equation is either true or false according to whether or not the two formulae evaluate to the same number or not.

V. FORMAL LOGICAL SYSTEMS AS CALCULI

As mentioned earlier, logicians sometimes short circuit the formal process of interpretation[17], studying the properties of a formal logical system under some set of truth value assignments. This is equivalent to studying the formal logical system for the entire *class of interpretations* that would result in those truth value assignments. This is a useful procedure even for the topic of this series, as long as it is understood that a class of interpretations is intended.

Any specific set of truth value assignments are therefore semantic, although the set of all possible truth value assignments is reduced to a purely syntactic procedure. Once a set of truth-value assignments has been made, any meaning assignments to symbols and terms from a given subject system must be consistent with those truth-value assignments (as ex post facto *judgments*). In consequence, a set of truth-value assignments prior to identification of a subject and meaning assignments selects a class of models from among all possible models.

Yet another short cut is sometimes taken, in which the logical constants corresponding to truth values are treated as if they were truth values, as if logical operators were definable as truth functions, and evaluation of a formula were part of the object language. With such an understanding, and assuming that certain properties of the formal logical system hold[18], it is then reasonable to treat the process of deduction as the computation of "true" formulas. For example, a system of propositional logic might then reasonably be called the propositional calculus. Such methods are quite common and sometimes understood as reducing semantics to syntax[19]. They are particularly useful when proving meta-theorems about formal logical systems (i.e., proving

[17] Such short cuts were usually unintentional before the formal interpretation and model theory were understood.
[18] We will identify these specific properties in the next article.
[19] This approach was formalized by Tarski. Church (1958, 1996) explains the use of function symbols for this purpose.

their properties) which do not depend on any specific interpretation, but may characterize the system for some specific class of interpretations or for all interpretations.

Unfortunately, the literature is not always clear as to why an author chooses to identify the system under study as a *calculus* instead of a *logic*, or vice-versa. In consequence, less formal authors have often used the terms interchangeably (and inappropriately).

VI. CONCLUSIONS

We remind the reader once again that the process of interpretation is one of associating meaning with abstract formulae in a structured manner. The result is a model of the deductive subsystem – an associated system that exemplifies the relationships found among the elements of the deductive subsystem that have been interpreted. As we will see later, a relational DBMS provides a generic deductive subsystem. The process of database design supplements or refines that deductive subsystem by providing a set of axioms. The application data model establishes rules of correspondence between properties, sets, and their relationships in deductive subsystem and those idealizations in the subject. The degree to which database design and modeling activities are informal and ad hoc determines the degree to which the resulting system is predictable and faithful to the subject. It follows that database technologies without a formal database design methodology are necessarily unable to guarantee any particular behavior or preserve any meaning.