Ben Goertzel
Matthew Iklé
Izabela Freire Goertzel
Ari Heljakka

Probabilistic Logic Networks

A Comprehensive Framework for Uncertain Inference

Springer

Probabilistic Logic Networks

A Comprehensive Framework for Uncertain Inference

Probabilistic Logic Networks

A Comprehensive Framework for Uncertain Inference

Ben Goertzel
Matthew Iklé
Izabela Freire Goertzel
Ari Heljakka

Springer

Authors
Ben Goertzel
Novamente LLC
1405 Bernerd Place
Rockville, MD 20851
ben@goertzel.org

Matthew Iklé
Department of Chemistry, Computer Science, and Mathematics
Adams State College
Alamosa, CO 81102
moikle@adams.edu

Izabela Freire Goertzel
Novamente LLC
1405 Bernerd Place
Rockville, MD 20851
izabela@goertzel.org

Ari Heljakka
Novamente LLC
1405 Bernerd Place
Rockville, MD 20851
heljakka@iki.fi

ISBN: 978-0-387-76871-7 e-ISBN: 978-0-387-76872-4
DOI: 10.1007/978-0-387-76872-4

Library of Congress Control Number: 2008924650

Printed on acid-free paper

springer.com

Dedicated to the memory of Jeff Pressing

Physicist, psychologist, musician, composer, athlete, polyglot and so much more --

Jeff was one of the most brilliant, fascinating, multidimensional and fully alive human beings any of us will ever know. He was killed in his sleep by a sudden, freak meningitis infection in 2002, while still young and in perfect health, and while in the early stages of co-developing the approach to probabilistic reasoning described in this book.

Jeff saw nearly none of the words of this book and perhaps 25% of the equations. We considered including him as a posthumous coauthor, but decided against this because many of the approaches and ideas we introduced after his death are somewhat radical and we can't be sure he would have approved them. Instead we included him as co-author on the two chapters to whose material he directly contributed. But nonetheless, there are many ways in which the overall PLN theory presented here – with its combination of innovation, formality and practicality -- embodies Jeff's "spirit" as an intellect and as a human being. Jeff, we miss you in so many ways!

Contents

Chapter 1: Introduction

Abstract In this chapter we provide an overview of probabilistic logic networks (PLN), including our motivations for developing PLN and the guiding principles underlying PLN. We discuss foundational choices we made, introduce PLN knowledge representation, and briefly introduce inference rules and truth-values. We also place PLN in context with other approaches to uncertain inference.

1.1 Motivations

This book presents Probabilistic Logic Networks (PLN), a systematic and pragmatic framework for computationally carrying out uncertain reasoning – reasoning about uncertain data, and/or reasoning involving uncertain conclusions. We begin with a few comments about why we believe this is such an interesting and important domain of investigation.

First of all, we hold to a philosophical perspective in which "reasoning" – properly understood – plays a central role in cognitive activity. We realize that other perspectives exist; in particular, logical reasoning is sometimes construed as a special kind of cognition that humans carry out only occasionally, as a deviation from their usual (intuitive, emotional, pragmatic, sensorimotor, etc.) modes of thought. However, we consider this alternative view to be valid only according to a very limited definition of "logic." Construed properly, we suggest, logical reasoning may be understood as the basic framework underlying all forms of cognition, including those conventionally thought of as illogical and irrational. The key to this kind of extended understanding of logic, we argue, is the formulation of an appropriately general theory of uncertain reasoning – where what is meant by the latter phrase is: reasoning based on uncertain knowledge, and/or reasoning leading to uncertain conclusions (whether from certain or uncertain knowledge). Moving from certain to uncertain reasoning opens up a Pandora's box of possibilities, including the ability to encompass within logic things such as induction, abduction, analogy and speculation, and reasoning about time and causality.

While not necessarily pertinent to the technical details of PLN, it is perhaps worth noting that the authors' main focus in exploring uncertain inference has been its pertinence to our broader work on artificial general intelligence (AGI). As elaborated in (Goertzel and Pennachin 2007; Goertzel and Wang 2007; Wang et al 2008), AGI refers to the construction of intelligent systems that can carry out a variety of complex goals in complex environments, based on a rich contextual understanding of themselves, their tasks and their environments. AGI was the original motivating goal of theAI research field, but at the moment it is one among multiple streams of AI research, living alongside other subfields focused on more narrow and specialized problem-solving. One viable approach to achieving powerful AGI, we believe, is to create integrative software systems with uncertain inference at their core. Specifically, PLN has been developed within the context of a larger artificial intelligence project, the Novamente Cognition Engine or NCE (Goertzel 2006), which seeks to achieve general forms of cognition by integrating PLN with several other processes. Recently, the NCE has spawned an open-source sister project called OpenCog, as well (Hart and Goertzel 2008). In the final two chapters we will briefly discuss the implementation of PLN within the NCE, and give a few relevant details of the NCE architecture. However, the vast majority of the discussion of PLN here is independent of the utilization of PLN as a component of the NCE. PLN stands as a conceptual and mathematical construct in its own right, with potential usefulness in a wide variety of AI and AGI applications.

We also feel that the mathematical and conceptual aspects of PLN have the potential to be useful outside the AI context, both as purely mathematical content and as guidance for understanding the nature of probabilistic inference in humans and other natural intelligences. These aspects are not emphasized here but we may address them more thoroughly in future works.

Of course, there is nothing new about the idea that uncertain inference is broadly important and relevant to AI and other domains. Over the past few decades a number of lines of research have been pursued, aimed at formalizing uncertain inference in a manner capable of application across the broad scope of varieties of cognition. PLN incorporates ideas derived from many of these other lines of inquiry, including standard ones like Bayesian probability theory (Jaynes, 2003), fuzzy logic (Zadeh 1989), and less standard ones like the theory of imprecise probabilities (Walley 1991), term logic (Sommers and Englebretsen 2000), Pei Wang's Non-Axiomatic Reasoning System (NARS) (Wang 1996), and algorithmic information theory (Chaitin 1987). For various reasons, which will come out as the book proceeds, we have found each of these prior attempts (and other ones, from which we have not seen fit to appropriate ideas, some of which we will mention below) unsatisfactory as a holistic approach to uncertain inference or as a guide to the creation of an uncertain inference component for use in integrative AGI systems.

Among the general high-level requirements underlying the development of PLN have been the following:

- To enable uncertainty-savvy versions of all known varieties of logical reasoning; including, for instance, higher-order reasoning involving quantifiers, higher-order functions, and so forth.
- To reduce to crisp "theorem prover" style behavior in the limiting case where uncertainty tends to zero.
- To encompass inductive and abductive as well as deductive reasoning.

- To agree with probability theory in those reasoning cases where probability theory, in its current state of development, provides solutions within reasonable calculational effort based on assumptions that are plausible in the context of real-world embodied software systems.
- To gracefully incorporate heuristics not explicitly based on probability theory, in cases where probability theory, at its current state of development, does not provide adequate pragmatic solutions.
- To provide "scalable" reasoning, in the sense of being able to carry out inferences involving at least billions of premises. Of course, when the number of premises is fewer, more intensive and accurate reasoning may be carried out.
- To easily accept input from, and send input to, natural language processing software systems.

The practical application of PLN is still at an early stage. Based on our evidence so far, however, we have found PLN to fulfill the above requirements adequately well, and our intuition remains that it will be found to do so in general. We stress, however, that PLN is an evolving framework, consisting of a conceptual core fleshed out by a heterogeneous combination of components. As PLN applications continue to be developed, it seems likely that various PLN components will be further refined and perhaps some of them replaced entirely. We have found the current component parts of PLN acceptable for our work so far, but we have also frequently been aware of more sophisticated alternative approaches to various subproblems (some drawn from the literature, and some our own inventions), and have avoided pursuing many of such due to a desire for initial simplicity.

The overall structure of PLN theory is relatively simple, and may be described as follows. First, PLN involves some important choices regarding knowledge representation, which lead to specific "schematic forms" for logical inference rules. The knowledge representation may be thought of as a definition of a set of "logical term types" and "logical relationship types," leading to a novel way of graphically modeling bodies of knowledge. It is this graphical interpretation of PLN knowledge representation that led to the "network" part of the name "Probabilistic Logic Networks." It is worth noting that the networks used to recognize knowledge in PLN are weighted directed hypergraphs (Bollobas 1998) much more general than, for example, the binary directed acyclic graphs used in Bayesian network theory (Pearl 1988).

Next, PLN involves specific mathematical formulas for calculating the probability value of the conclusion of an inference rule based on the probability values of the premises plus (in some cases) appropriate background assumptions. It also involves a particular approach to estimating the confidence values with which these probability values are held (weight of evidence, or second-order uncertainty). Finally, the implementation of PLN in software requires important choices regarding the structural representation of inference rules, and also regarding "inference control" – the strategies required to decide what inferences to do in what order, in each particular practical situation.

1.1.1 Why Probability Theory?

In the next few sections of this Introduction we review the conceptual foundations of PLN in a little more detail, beginning with the question: Why choose probability theory as a foundation for the "uncertain" part of uncertain inference?

We note that while probability theory is the foundation of PLN, not all aspects of PLN are based strictly on probability theory. The mathematics of probability theory (and its interconnection with other aspects of mathematics) has not yet been developed to the point where it is feasible to use explicitly probabilistic methods to handle every aspect of the uncertain inference process. Some researchers have reacted to this situation by disregarding probability theory altogether and introducing different conceptual foundations for uncertain inference, such as Dempster-Shafer theory (Dempster 1968; Shafer 1976), Pei Wang's Non-Axiomatic Reasoning System (Wang 1996), possibility theory (Zadeh 1978) and fuzzy set theory (Zadeh 1965). Others have reacted by working within a rigidly probabilistic framework, but limiting the scope of their work fairly severely based on the limitations of the available probabilistic mathematics, avoiding venturing into the more ambiguous domain of creating heuristics oriented toward making probabilistic inference more scalable and pragmatically applicable (this, for instance, is how we would characterize the mainstream work in probabilistic logic as summarized in Hailperin 1996; more comments on this below). Finally, a third reaction – and the one PLN embodies – is to create reasoning systems based on a probabilistic foundation and then layer non-probabilistic ideas on top of this foundation when this is the most convenient way to arrive at useful practical results.

Our faith in probability theory as the ultimately "right" way to quantify uncertainty lies not only in the demonstrated practical applications of probability theory to date, but also in Cox's (1961) axiomatic development of probability theory and ensuing refinements (Hardy 2002), and associated mathematical arguments due to de Finetti (1937) and others. These theorists have shown that if one makes some very basic assumptions about the nature of uncertainty quantification, the rules of elementary probability theory emerge as if by magic. In this section we briefly review these ideas, as they form a key part of the conceptual foundation of the PLN framework.

Cox's original demonstration involved describing a number of properties that should commonsensically hold for any quantification of the "plausibility" of a proposition, and then showing that these properties imply that plausibility must be a scaled version of conventional probability. The properties he specified are, in particular[1],

[1] The following list of properties is paraphrased from the Wikipedia entry for "Cox's Theorem."

1. The plausibility of a proposition determines the plausibility of the proposition's negation; either decreases as the other increases. Because "a double negative is an affirmative," this becomes a functional equation

$$f\left(f\left(x\right)\right)=x$$

saying that the function f that maps the probability of a proposition to the probability of the proposition's negation is an involution; i.e., it is its own inverse.

2. The plausibility of the conjunction [A & B] of two propositions A, B, depends only on the plausibility of B and that of A ***given*** that B is true. (From this Cox eventually infers that multiplication of probabilities is associative, and then that it may as well be ordinary multiplication of real numbers.) Because of the associative nature of the "and" operation in propositional logic, this becomes a functional equation saying that the function g such that

$$P\left(A \text{ and } B\right)=g\left(P\left(A\right),P\left(B|A\right)\right)$$

is an associative binary operation. All strictly increasing associative binary operations on the real numbers are isomorphic to multiplication of numbers in the interval [0, 1]. This function therefore may be taken to be multiplication.

3. Suppose [A & B] is equivalent to [C & D]. If we acquire new information A and then acquire further new information B, and update all probabilities each time, the updated probabilities will be the same as if we had first acquired new information C and then acquired further new information D. In view of the fact that multiplication of probabilities can be taken to be ordinary multiplication of real numbers, this becomes a functional equation

$$yf\left(\frac{f\left(z\right)}{y}\right)=zf\left(\frac{f\left(y\right)}{z}\right)$$

where f is as above.

Cox's theorem states, in essence, that any measure of plausibility that possesses the above three properties must be a rescaling of standard probability.

While it is impressive that so much (the machinery of probability theory) can be derived from so little (Cox's very commonsensical assumptions), mathematician Michael Hardy (2002) has expressed the opinion that in fact Cox's axioms are too strong, and has provided significantly weaker conditions that lead to the same end result as Cox's three properties. Hardy's conditions are more abstract and difficult to state without introducing a lot of mathematical mechanism, but essentially he studies mappings from propositions into ordered "plausibility" values, and he shows that if any such mapping obeys the properties of

1. If x implies y then $f(x) < f(y)$
2. If $f(x) < f(y)$ then $f(\neg x) > f(\neg y)$, where ¬ represents "not"
3. If $f(x|z) <= f(y|z)$ and $f(x|\neg z) <= f(y|\neg z)$ then $f(x) < f(y)$
4. For all x, y either $f(x) \leq f(y)$ or $f(y) \leq f(x)$

then it maps propositions into scaled probability values. Note that this property list mixes up absolute probabilities f() with conditional probabilities f(|), but this is not a problem because Hardy considers f(x) as equivalent to f(x|U) where U is the assumed universe of discourse.

Hardy expresses regret that his fourth property is required; however, Youssef's (1994) work related to Cox's axioms suggests that it is probably there in his mathematics for a very good conceptual reason. Youssef has shown that it is feasible to drop Cox's assumption that uncertainty must be quantified using real numbers, but retain Cox's other assumptions. He shows it is possible, consistent with Cox's other assumptions, to quantify uncertainty using "numbers" drawn from the complex, quaternion, or octonion number fields. Further, he argues that complex-valued "probabilities" are the right way to model quantum-level phenomena that have not been collapsed (decohered) into classical phenomena. We believe his line of argument is correct and quite possibly profound, yet it does not seem to cast doubt on the position of standard real-valued probability theory as the correct mathematics for reasoning about ordinary, decohered physical systems. If one wishes to reason about the uncertainty existing in pure, pre-decoherence quantum systems or other exotic states of being, then arguably these probability theories defined over different base fields than the real numbers may be applicable.

Next, while we are avid probabilists, we must distinguish ourselves from the most ardent advocates of the "Bayesian" approach to probabilistic inference. We understand the weakness of the traditional approach to statistics with its reliance on often unmotivated assumptions regarding the functional forms of probability distributions. On the other hand, we don't feel that the solution is to replace these assumptions with other, often unmotivated assumptions about prior probability distributions. Bayes' rule is an important part of probability theory, but the way that the Bayesian-statistical approach applies it is not always the most useful way. A major example of the shortcomings of the standard Bayesian approach lies in the domain of confidence assessment, an important aspect of PLN already mentioned above. As Wang (2001) has argued in detail, the standard Bayesian approach does not offer any generally viable way to assess or reason about the "second-order uncertainty" involved in a given uncertainty value. Walley (1991) sought to redress this problem via a subtler approach that avoids assuming a single prior distribution, and makes a weaker assumption involving drawing a prior from a parametrized family of possible prior distributions; others have followed up his work in interesting ways (Weichselberger 2003), but this line of research has not yet been developed to the point of yielding robustly applicable mathematics. Within PLN, we introduce a spectrum of approaches to confidence assessment ranging from indefinite probabilities (essentially a hybridization of Walley's imprecise probabilities with Bayesian credible intervals) to frankly non-probabilistic heuristics inspired partly by Wang's work. By utilizing this wide range of approaches, PLN can more gracefully assess confidence in diverse settings, providing pragmatic solutions where the Walley-type approach (in spite of its purer probabilism) currently fails.

Though Cox's theorem and related results argue convincingly that probability theory is the correct approach to reasoning under uncertainty, the particular ways of applying probability theory that have emerged in the contemporary scientific community (such as the "Bayesian approach") all rely on specific assumptions beyond those embodied in the axioms of probability theory. Some of these assumptions are explicit mathematical ones, and others are implicit assumptions about how to proceed in setting up a given problem in probabilistic terms; for instance, how to translate an intuitive understanding and/or a collection of quantitative data into mathematical probabilistic form.

1.2 PLN in the Context of Traditional Approaches to Probabilistic Logic

So, supposing one buys the notion that logic, adequately broadly construed, is essential (perhaps even central) to cognition; that appropriate integration of uncertainty into logic is an important aspect of construing logic in an adequately broad way; and also that probability theory is the correct foundation for treatment of uncertainty, what then? There is already a fairly well fleshed-out theory of probabilistic logic, so why does one need a substantial body of new theory such as Probabilistic Logic Networks?

The problem is that the traditional theories in the area of probabilistic logic don't directly provide a set of tools one can use to structure a broadly-applicable, powerful software system for probabilistic inferencing. They provide a number of interesting and important theorems and ideas, but are not sufficiently pragmatic in orientation, and also fail to cover some cognitively key aspects of uncertain inference such as intensional inference.

Halpern's (2003) book provides a clearly written, reasonably thorough overview of recent theories in probabilistic logic. The early chapters of Hailperin (1996) gives some complementary historical and theoretical background. Alongside other approaches such as possibility theory, Halpern gives an excellent summary of what in PLN terms would be called “first-order extensional probabilistic logic” – the interpretation and manipulation of simple logic formulas involving absolute and conditional probabilities among sets. Shortcomings of this work from a pragmatic AI perspective include:

- No guidance is provided as to which heuristic independence assumptions are most cognitively natural to introduce in order to deal with (the usual) situations where adequate data regarding dependencies is unavailable. Rather, exact probabilistic logic formulas are introduced, into which one can, if one wishes, articulate independence assumptions and then derive their consequences.
- Adequate methods for handling “second order uncertainty” are not presented, but this is critical for dealing with real-world inference situations where available data is incomplete and/or erroneous. Hailperin (1996) deals with this by looking at interval probabilities, but this turns out to rarely be useful in practice because the intervals corresponding to inference results are generally far too wide. Walley’s (1991) imprecise probabilities are more powerful but have a similar weakness, and we will discuss them in more detail in Chapter 4; they also have not been integrated into any sort of powerful, general, probabilistic logic framework, though integrating them into PLN if one wished to do so would not be problematic, as will become clear.
- Various sorts of truth-values are considered, including single values, intervals, and whole probability distributions, but the problem of finding the right way to summarize a probability distribution for logical inference without utilizing too much memory or sacrificing too much information has not been adequately resolved (and this is what we have tried to resolve with the “indefinite probabilities” utilized in PLN).
- The general probabilistic handling of intensional, temporal, and causal inference is not addressed. Of course, these topics are handled in various specialized theories; e.g., Pearl’s causal networks (2000), but there is no general theory of probabilistic intensional, temporal, or causal logic; yet the majority of commonsense logical inference involves these types of reasoning.
- The existing approaches to intermixing probabilistic truth-values with existential and universal quantification are conceptually flawed and often do not yield pragmatically useful results.

All in all, in terms of Halpern’s general formalism for what we call first-order extensional logic, what PLN constitutes is

- A specific compacted representation of sets of probability distributions (the indefinite truth-value)
- A specific way of deploying heuristic independence assumptions; e.g., within the PLN deduction and revision rules

- A way of counting the amount of evidence used in an inference (which is used in the revision rule, which itself uses amount of evidence together with heuristic independence assumptions)

But much of the value of PLN lies in the ease with which it extends beyond first-order extensional logic. Due to the nature of the conceptual and mathematical formalism involved, the same essential inference rules and formulas used for first-order extensional logic are extended far more broadly, to deal with intensional, temporal, and causal logic, and to deal with abstract higher-order inference involving complex predicates, higher-order functions, and universal, existential, and fuzzy quantifiers.

1.2.1 Why Term Logic?

One of the major ways in which PLN differs from traditional approaches to probabilistic logic (and one of the secrets of PLN's power) is its reliance on a formulation of logic called "term logic." The use of term logic is essential, for instance, to PLN's introduction of cognitively natural independence assumptions and to PLN's easy extension of first-order extensional inference rules to more general and abstract domains.

Predicate logic and term logic are two different but related forms of logic, each of which can be used both for crisp and uncertain logic. Predicate logic is the most familiar kind, where the basic entity under consideration is the "predicate," a function that maps argument variables into Boolean truth-values. The argument variables are quantified universally or existentially.

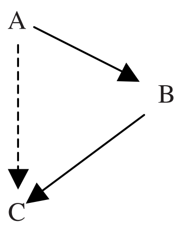
On the other hand, in term logic, which dates back at least to Aristotle and his notion of the syllogism, the basic element is a subject-Predicate statement, denotable

```
A → B
```

where → denotes a notion of inheritance or specialization. Logical inferences take the form of "syllogistic rules," which give patterns for combining statements with matching terms. (We don't use the → notation much in PLN, because it's not sufficiently precise for PLN purposes, since PLN introduces many varieties of inheritance; but we will use the → notation in this section because here we are speaking about inheritance in term logic in general rather than about PLN in particular.)

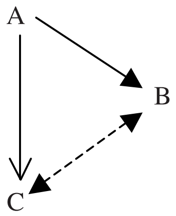
Examples are the deduction, induction, and abduction rules:

```
A → B
B → C
|-
A→ C
```



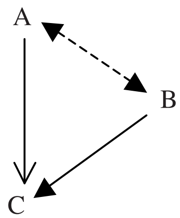
Deduction

```
A → B
A → C
|-
B → C
```



Induction

```
A → C
B → C
|-
A → B
```



Abduction

When we get to defining the truth-value formulas corresponding to these inference rules, we will observe that deduction is infallible in the case of absolutely certain premises, but uncertain in the case of probabilistic premises; while abduction and induction are always fallible, even given certain premises. In fact we will derive abduction and induction from the combination of deduction with a simple rule called inversion

```
A → B
|-
B → A
```

whose truth-value formula derives from Bayes rule.

Predicate logic has proved to deal more easily with deduction than with induction, abduction, and other uncertain, fallible inference rules. On the other hand, term logic can deal quite elegantly and simply with all forms of inference. Furthermore, the predicate logic formulation of deduction proves less amenable to "probabilization" than the term logic formulation. It is for these reasons, among others, that the foundation of PLN is drawn from term logic rather than from predicate logic. PLN begins with a term logic foundation, and then adds on elements of probabilistic and combinatory logic, as well as some aspects of predicate logic, to form a complete inference system, tailored for easy integration with software components embodying other (not explicitly logical) aspects of intelligence.

Sommers and Engelbretsen (2000) have given an excellent defense of the value of term logic for crisp logical inference, demonstrating that many pragmatic inferences are far simpler in term logic formalism than they are in predicate logic formalism. On the other hand, the pioneer in the domain of uncertain term logic is Pei Wang (Wang 1996), to whose NARS uncertain term logic based reasoning system PLN owes a considerable debt. To frame the issue in terms of our above discussion of PLN's relation to traditional probabilistic logic approaches, we may say we have found that the formulation of appropriate heuristics to guide probabilistic inference in cases where adequate dependency information is not available, and appropriate methods to extend first-order extensional inference rules and formulas to handle other sorts of inference, are both significantly easier in a term logic rather than predicate logic context. In these respects, the use of term logic in PLN is roughly a probabilization of the use of term logic in NARS; but of course, there are many deep conceptual and mathematical differences between PLN and NARS, so that the correspondence between the two theories in the end is more historical and theory-structural, rather than a precise correspondence on the level of content.

1.3 PLN Knowledge Representation and Inference Rules

In the next few sections of this Introduction, we review the main topics covered in the book, giving an assemblage of hints as to the material to come. First, Chapter 2 describes the knowledge representation underlying PLN, without yet saying anything specific about the management of numbers quantifying uncertainties. A few tricky issues occur here, meriting conceptual discussion. Even though PLN knowledge representation is not to do with uncertain inference per se, we have found that without getting the knowledge representation right, it is very difficult to define uncertain inference rules in an intuitive way. The biggest influence on PLN's knowledge representation has been Wang's NARS framework, but there are also some significant deviations from Wang's approach.

PLN knowledge representation is conveniently understood according to two dichotomies: extensional vs. intensional, and first-order vs. higher-order. The former is a conceptual (philosophical/cognitive) distinction between logical relationships that treat concepts according to their members versus those that treat concepts according to their properties. In PLN extensional knowledge is treated as more basic, and intensional knowledge is defined in terms of extensional knowledge via the addition of a specific mathematics of intension (somewhat related to information theory). This is different from the standard probabilistic approach, which contains no specific methods for handling intension and also differs from, e.g., Wang's approach in which intension and extension are treated as completely symmetric, with neither of them being more basic or derivable from the other.

The first-order versus higher-order distinction, on the other hand, is essentially a mathematical one. First-order, extensional PLN is a variant of standard term logic, as originally introduced by Aristotle in his Logic and recently elaborated by theorists such as Wang (1996) and Sommers and Engelbretsen (2000). First-order PLN involves logical relationships between terms representing concepts, such as

```
Inheritance cat animal

ExtensionalInheritance Pixel_444 Contour_7565
```

(where the notation is used that R A B denotes a logical relationship of type R between arguments A and B). A typical first-order PLN inference rule is the standard term-logic deduction rule

```
A → B
B → C
|-
A → C
```

which in PLN looks like

```
ExtensionalInheritance A B
ExtensionalInheritance B C
|-
ExtensionalInheritance A C
```

As well as purely logical relationships, first-order PLN also includes a fuzzy set membership relationship, and specifically addresses the relationship between fuzzy set membership and logical inheritance, which is closely tied to the PLN concept of intension. In the following text we will sometimes use the acronym FOI to refer to PLN First Order Inference.

Higher-order PLN, on the other hand (sometimes called HOI, for Higher Order Inference), has to do with functions and their arguments. Much of higher-order PLN is structurally parallel to first-order PLN; for instance, implication between statements is largely parallel to inheritance between terms. However, a key difference is that most of higher-order PLN involves either variables or higher-order functions (functions taking functions as their arguments). So for instance one might have

```
ExtensionalImplication
    Inheritance $X cat
    Evaluation eat ($X, mice)
```

(using the notation that

```
R
    A
    B
```

denotes the logical relationship R applied to the arguments A and B). Here Evaluation is a relationship that holds between a predicate and its argument-list, so that, e.g.,

```
Evaluation eat (Sylvester, mice)
```

means that the list *(Sylvester, mice)* lies within the set of ordered pairs characterizing the *eat* relationship. The parallel of the first-order extensional deduction rule given above would be a rule

```
ExtensionalImplication A B
ExtensionalImplication B C
|-
ExtensionalImplication A C
```

where the difference is that in the higher-order inference case the tokens A, B, and C denote either variable-bearing expressions or higher-order functions. Some higher-order inference rules involve universal or existential quantifiers as well.

While first-order PLN adheres closely to the term logic framework, higher-order PLN is better described as a mix of term logic, predicate logic, and combinatory logic. The knowledge representation is kept flexible, as this seems to lead to the simplest and most straightforward set of inference rules.

1.4 Truth-value Formulas

We have cited above the conceptual reasons why we have made PLN a probabilistic inference framework, rather than using one of the other approaches to uncertainty quantification available in the literature. However, though we believe in the value of probabilities we do not believe that the conventional way of using probabilities to represent the truth-values of propositions is adequate for pragmatic computational purposes. One of the less conventional aspects of PLN is the quantification of uncertainty using truth-values that contain at least two components, and usually more (in distinction from the typical truth-value used in probability theory, which is a single number: a probability). Our approach here is related to earlier multi-component truth-value approaches due to Keynes (2004), Wang (2006), Walley (1991), and others, but is unique in its particulars.

The simplest kind of PLN truth-value, called a SimpleTruthValue, consists of a pair of numbers <*s*,*w*> called a strength and a confidence. The strength value is a probability; the confidence value is a measure of the amount of certainty attached to the strength value. Confidence values are normalized into [0,1].

For instance <.6,1> means a probability of .6 known with absolute certainty. <.6,.2> means a probability of .6 known with a very low degree of certainty. <.6,0> means a probability of .6 known with a zero degree of certainty, which indicates a meaningless strength value, and is equivalent to <*x*,0> for any other probability value *x*.

Another type of truth-value, more commonly used as the default within PLN, is the IndefiniteTruthValue. We introduce the mathematical and philosophical foundations of IndefiniteTruthValues in Chapter 4. Essentially a hybridization of Walley's imprecise probabilities and Bayesian credible intervals, indefinite probabilities quantify truth-values in terms of four numbers <*L*, *U*, *b*, *k*>: an interval [*L*,*U*], a credibility level *b*, and an integer *k* called the "lookahead." IndefiniteTruthValues provide a natural and general method for calculating the "weight-of-evidence" underlying the conclusions of uncertain inferences. We ardently believe that this approach to uncertainty quantification may be adequate to serve as an ingredient of powerful artificial general intelligence.

Beyond the SimpleTruthValues and IndefiniteTruthValues mentioned above, more advanced types of PLN truth-value also exist, principally "distributional truth-values" in which the strength value is replaced by a matrix approximation to an entire probability. Note that this, then, provides for three different granularities of approximations to an entire probability distribution. A distribution can be most simply approximated by a single number, somewhat better approximated by a probability interval, and even better approximated by an entire matrix.

Chapter 5 takes the various inference rules defined in Chapter 2, and associates a "strength value formula" with each of them (a formula determining the strength of the conclusion based on the strengths of the premises). For example, the deduction rule mentioned above is associated with two strength formulas, one based on an independence assumption and the other based on a different "concept geometry" based assumption. The independence-assumption-based deduction strength formula looks like

```
B <sB>
C <sC>
ExtensionalInheritance A B <sAB>
ExtensionalInheritance B C <sBC>
|-
ExtensionalInheritance A C <sAC>
```

$s_{AC} = s_{AB}\ s_{BC} + (1-s_{AB})\ (s_C - s_B\ s_{BC}) / (1- s_B)$

This particular rule is a straightforward consequence of elementary probability theory. Some of the other formulas are equally straightforward, but some are subtler and require heuristic reasoning beyond standard probabilistic tools like independence assumptions. Since simple truth-values are the simplest and least informative of our truth-value types; they provide quick, but less accurate, assessments of the resulting strength and confidence values.

We reconsider these strength formulas again in Chapter 6, extending the rules to IndefiniteTruthValues. We also illustrate in detail how indefinite truth-values provide a natural approach to measuring weight-of-evidence. IndefiniteTruthValues can be thought of as approximations to entire distributions, and so provide an intermediate level of accuracy of strength and confidence.

As shown in Chapter 7, PLN inference formulas may also be modified to handle entire distributional truth-values. Distributional truth-values provide more information than the other truth-value types. As a result, they may also be used to yield even more accurate assessments of strength and confidence.

The sensitivity to error of several inference rule formulas for various parameter values is explored in Chapter 8. There we provide a fairly detailed mathematical and graphical examination of error magnification. We also study the possibility of deterministic chaos arising from PLN inference.

We introduce higher-order inference (HOI) in Chapter 10, where we describe the basic HOI rules and strength formulas for both simple truth-values and indefinite truth-values. We consider both crisp and fuzzy quantifiers, using indefinite probabilities, in Chapter 11; treat intensional inference in Chapter 12; and inference control in Chapter 13. Finally, we tackle the topics of temporal and causal inference in Chapter 14.

1.5 Implementing and Applying PLN

The goal underlying the theoretical development of PLN has been the creation of practical software systems carrying out complex, useful inferences based on uncertain knowledge and drawing uncertain conclusions. Toward that end we have implemented most of the PLN theory described in this book as will briefly be described in Chapter 13, and used this implementation to carry out simple inference experiments involving integration with external software components such as a natural language comprehension engine and a 3D simulation world.

Chapter 14 reviews some extensions made to basic PLN in the context of these practical applications, which enable PLN to handle reasoning about temporal and causal implications. Causal inference in particular turns out to be conceptually interesting, and the approach we have conceived relates causality and intension in a satisfying way.

By far the most difficult aspect of designing a PLN implementation is inference control, which we discuss in Chapter 13. This is really a foundational conceptual issue rather than an implementational matter per se. The PLN framework just tells you what inferences can be drawn; it doesn't tell you what order to draw them in, in which contexts. Our PLN implementation utilizes the standard modalities of forward-chaining and backward-chaining inference control. However, the vivid presence of uncertainty throughout the PLN system makes these algorithms more challenging to use than in a standard crisp inference context. Put simply, the search trees expand unacceptably fast, so one is almost immediately faced with the need to use clever, experience-based heuristics to perform pruning.

The issue of inference control leads into deep cognitive science issues that we briefly mention here but do not fully explore, because that would lead too far afield from the focus of the book, which is PLN in itself. One key conceptual point that we seek to communicate, however, is that uncertain inference rules and formulas, on their own, do not compose a comprehensive approach to artificial intelligence. To achieve the latter, sophisticated inference control is also required, and controlling uncertain inference is difficult – in practice, we have found, requiring ideas that go beyond the domain of uncertain inference itself. In principle, one could take a purely probability-theoretic approach to inference control – choosing inference steps based on the ones that are most likely to yield successful conclusions based on probabilistic integration of all the available evidence. However, in practice this does not seem feasible given the current state of development of applied probability theory. Instead, in our work with PLN so far, we have taken a heuristic and integrative approach, using other non-explicitly-probabilistic algorithms to help prune the search trees implicit in PLN inference control.

As for applications, we have applied PLN to the output of a natural language processing subsystem, using it to combine premises extracted from different biomedical research abstracts to form conclusions embodying medical knowledge not contained in any of the component abstracts. We have also used PLN to learn rules controlling the behavior of a humanoid agent in a 3D simulation world; for instance, PLN learns to play “fetch” based on simple reinforcement learning stimuli.

Our current research involves extending PLN's performance in both these areas, and bringing the two areas together by using PLN to help the Novamente Cognition Engine carry out complex simulation-world tasks involving a combination of physical activity and linguistic communication. Quite probably this ongoing research will involve various improvements to be made to the PLN framework itself. Our goal in articulating PLN has not been to present an ultimate itself. Our goal in articulating PLN has not been to present an ultimate and final approach to uncertain inference, but rather to present a workable approach that is suitable for carrying out uncertain inference comprehensively and reasonably well in practical contexts. As probability theory and allied branches of mathematics develop, and as more experience is gained applying PLN in practical contexts, we expect the theory to evolve and improve.

1.6 Relationship of PLN to Other Approaches to Uncertain Inference

Finally, having sketched the broad contours of PLN theory and related it to more traditional approaches to probabilistic logic, we now briefly discuss the relationship between PLN and other approaches to logical inference. First, the debt of PLN to various standard frameworks for crisp logical inference is clear. PLN's knowledge representation, as will be made clear in Chapter 2, is an opportunistically assembled amalgam of formalisms chosen from term logic, predicate logic and combinatory logic. Rather than seeking a pure and minimal formalism, we have thought more like programming language designers and sought a logical formalism that allows maximally compact and comprehensible representation of a wide variety of useful logical structures.

Regarding uncertainty, as noted above, as well as explicit approaches to the problem of unifying probability and logic the scientific literature contains a number of other relevant ideas, including different ways to quantify uncertainty and to manipulate uncertainty once quantified. There are non-probabilistic methods like fuzzy logic, possibility theory, and NARS. And there is a variety of probabilistic approaches to knowledge representation and reasoning that fall short of being full-on "probabilistic logics," including the currently popular Bayes nets, which will be discussed in more depth below, and Walley's theory of imprecise probabilities (Walley 1991), which has led to a significant literature (ISIPTA 2001, 2003, 2005, 2007), and has had a significant inspirational value in the design of PLN's approach to confidence estimation, as will be reviewed in detail in Chapters 4, 6, and 10.

Overall, regarding the representation of uncertainty, PLN owes the most to Pei Wang's NARS approach and Walley's theory of imprecise probabilities. Fuzzy set theory ideas are also utilized in the specific context of the PLN Member relationship. However, we have not found any of these prior approaches to uncertainty quantification to be fully adequate, and so the PLN approach draws from them ample inspiration but not very many mathematical details.

We now review the relationship of PLN to a few specific approaches to uncertainty quantification and probabilistic inference in a little more detail. In all cases the comments given here are high-level and preliminary, and the ideas discussed will be much clearer to the reader after they have read the later chapters of this book and understand PLN more fully.

1.6.1 PLN and Fuzzy Set Theory

Fuzzy set theory has proved a pragmatically useful approach to quantifying many kinds of relationships (Zadeh 1965, 1978), but we believe that its utility is fundamentally limited. Ultimately, we suggest, the fuzzy set membership degree is not a way of quantifying uncertainty – it is quantifying something else: it is quantifying partial membership. Fuzzy set membership is used in PLN as the semantics of the truth-values of special logical relationship types called Member relationships. These fuzzy Member relationships may be used within PLN inference, but they are not considered the same as logical relationships such as Inheritance or Similarity relationships whose truth-values quantify degrees of uncertainty.

Some (though nowhere near all) of the fuzzy set literature appears to us to be semantically confused regarding the difference between uncertainty and partial membership. In PLN we clearly distinguish between

- Jim belongs to degree .6 to the fuzzy set of tall people. (MemberLink semantics)
- Jim shares .6 of the properties shared by people belonging to the set of tall people (where the different properties may be weighted). (IntensionalInheritanceLink semantics)
- Jim has a .6 chance of being judged as belonging to the set of tall people, once more information about Jim is obtained (where this may be weighted as to the degree of membership that is expected to be estimated once the additional information is obtained). (IntensionalInheritanceLink, aka Subset Link, semantics)
- Jim has an overall .6 amount of tallness, defined as a weighted average of extensional and intensional information. (Inheritance Link semantics)

We suggest that the fuzzy, MemberLink semantics is not that often useful, but do recognize there are cases where it is valuable; e.g., if one wishes to declare that a stepparent and stepchild are family members with fuzzy degree .8 rather than 1.

In terms of the above discussion of the foundations of probability theory we note that partial membership assignments need not obey Cox's axioms and need not be probabilities – which is fine, as they are doing something different, but also limits the facility with which they can be manipulated. In PLN, intensional probabilities are used for many of the purposes commonly associated with fuzzy membership values, and this has the advantage of keeping more things within a probabilistic framework.

1.6.2 PLN and NARS

Pei Wang's NARS approach has already been discussed above and will pop up again here and there throughout the text; furthermore, Appendix A1 presents a comparison of some of the first-order PLN truth-value formulas with corresponding NARS formulas. As already noted, there is a long historical relationship between PLN and NARS; PLN began as part of a collaboration with NARS's creator Pei Wang as an attempt to create a probabilistic analogue to NARS. PLN long ago diverged from its roots in NARS and has grown in a very different direction, but there remain many similarities. Beneath all the detailed similarities and differences, however, there is a deep and significant difference between the two, which is semantic: PLN's semantics is probabilistic, whereas NARS's semantics is intentionally and definitively not.

PLN and NARS have a similar division into first-order versus higher-order inference, and have first-order components that are strictly based on term logic. However, PLN's higher-order inference introduces predicate and combinatory logic ideas, whereas NARS's higher-order inference is also purely term logic based. Both PLN and NARS include induction, deduction, and abduction in their first-order components, with identical graphical structures; in PLN, however, induction and abduction are derived from deduction via Bayes rule, whereas in NARS they have their own completely independent truth-value functions. Both PLN and NARS utilize multi-component truth-values, but the semantics of each component is subtly different, as will be reviewed in appropriate points in the text to follow.

1.6.3 PLN and Bayes Nets

Bayes nets are perhaps the most popular contemporary approach to uncertain inference. Because of this, we here offer a few more detailed comments on the general relationship between PLN and Bayes nets. Of course, the actual relationship is somewhat subtle and will be clear to the reader only after completing the exposition of PLN.

Traditional Bayesian nets assume a tree structure for events, which is unrealistic in general, but in recent years there has been a batch of work on "loopy Bayesian networks" in which standard Bayesian net information propagation is applied to potentially cyclic graphs of conditional probability. Some interesting alternatives to the loopy Bayesian approach have also been proposed, including one that uses a more advanced optimization algorithm within the Bayesian net framework.

Bayes nets don't really contain anything comparable to the generality of PLN higher-order inference. However, in the grand scheme of things, first-order PLN is not all that tremendously different from loopy Bayesian nets and related schemes. In both cases one is dealing with graphs whose relationships denote conditional probabilities, and in both cases one is using a kind of iterative relaxation method to arrive at a meaningful overall network state.

If one took a forest of loopy Bayes nets with imprecise probabilities, and then added some formalism to interface it with fuzzy, predicate, and combinatory logic, then one might wind up with something reasonably similar to PLN. We have not taken such an approach but have rather followed the path that seemed to us more natural, which was to explicitly shape a probabilistic inference framework based on the requirements that we found important for our work on integrative AI.

There are many ways of embodying probability theory in a set of data structures and algorithms. Bayes nets are just one approach. PLN is another approach and has been designed for a different purpose: to allow basic probabilistic inference to interact with other kinds of inference such as intensional inference, fuzzy inference, and higher-order inference using quantifiers, variables, and combinators. We have found that for the purpose of interfacing basic probabilistic inference with these other sorts of inference, the PLN approach is a lot more convenient than Bayes nets or other more conventional approaches.

Another key conceptual difference has to do with a PLN parameter called the "context." In terms of probability theory, one can think of a context as a universe of discourse. Rather than attempting to determine a (possibly non-existent) universal probability distribution that has desired properties within each local domain, PLN creates local probability distributions based on local contexts. The context parameter can be set to Universal (everything the system has ever seen), Local (only the information directly involved in a given inference), or many levels in between.

Yet another major conceptual difference is that PLN handles multivariable truth-values. Its minimal truth-value object has two components: strength and weight of evidence. Alternatively, it can use probability distributions (or discrete approximations thereof) as truth-values. This makes a large difference in the handling of various realistic inference situations. For instance, the treatment of "weight of evidence" in PLN is not a purely mathematical issue, but reflects a basic conceptual issue, which is that (unlike most probabilistic methods) PLN does not assume that all probabilities are estimated from the same sample space. It makes this assumption provisionally in some cases, but it doesn't make it axiomatically and comprehensively.

With the context set to Universal, and with attention restricted to the strength component of truth-values, what we have in PLN-FOI is – speaking conceptually rather than mathematically – a different way of doing the same thing that loopy Bayes networks (BN) and its competitors are trying to do. PLN, loopy BN, and other related methods are all viewable as optimization algorithms trying to relax into a condition giving the "correct probability distribution," and at some risk of settling into local optima instead. But the ability to use more flexible truth-values, and to use local contexts as appropriate, makes a very substantial difference in practice. This is the kind of difference that becomes extremely apparent when one seeks to integrate probabilistic inference with other cognitive processes. And it's the kind of difference that is important when trying to extend one's reasoning system from simple inferences to extremely general higher-order inference – an extension that has succeeded within PLN, but has not been successfully carried out within these other frameworks.

1.7 Toward Pragmatic Probabilistic Inference

Perhaps the best way to sum up the differences between PLN and prior approaches to (crisp or uncertain) inference is to refer back to the list of requirements given toward the start of this Introduction. These requirements are basically oriented toward the need for an approach to uncertain inference that is adequate to serve as the core of a general-purpose cognition process – an approach that can handle *any kind of inference* effectively, efficiently, and in an uncertainty-savvy way.

Existing approaches to crisp inference are not satisfactory for the purposes of general, pragmatic, real-world cognition, because they don't handle uncertainty efficiently and gracefully. Of course, one can represent uncertainties in predicate logic – one can represent *anything* in predicate logic – but representing them in a way that leads to usefully rapid and convenient inference incorporating uncertainties intelligently is another matter.

On the other hand, prior approaches to uncertain inference have universally failed the test of comprehensiveness. Some approaches, such as Bayes nets and fuzzy set theory, are good at what they do but carry out only very limited functions compared to what is necessary to fulfill the inference needs of a general-purpose cognitive engine. Others, such as imprecise probability theory, are elegant and rigorous but are so complex that the mathematics needed to apply them in practical situations has not yet been resolved. Others, such as NARS and Dempster-Shafer theory, appear to us to have fundamental conceptual flaws in spite of their interesting properties. And still others, such as traditional probabilistic logic as summarized by Halpern and Hailperin, fail to provide techniques able to deal with the scale, incompleteness, and erroneousness typifying real-world inference situations.

In sum, we do not propose PLN as an ultimate and perfect uncertain inference framework, only as an adequate one – but we do suggest that, in its adequacy, PLN distinguishes itself from the alternatives currently available. As noted above, we suspect that the particulars of the PLN framework will evolve considerably as PLN is utilized for more and more pragmatic inference tasks, both on its own and within integrative AI systems.

Chapter 2: Knowledge Representation

Abstract In chapter 2, we review the basic formalism of PLN knowledge representation in a way that is relatively independent of the particularities of PLN truth-value manipulation. Much of this material has nothing explicitly to do with probability theory or uncertainty management; it merely describes a set of conventions for representing logical knowledge. However, we also define some of the elements of PLN truth-value calculation here, insofar as is necessary to define the essential meanings of some of the basic PLN constructs.

2.1 Basic Terminology and Notation

The basic players in PLN knowledge representation are entities called *terms* and *relationships* (atomic formulae). The term *Atom* will refer to any element of the set containing both terms and relationships. The hierarchy of PLN Atoms begins with a finite set *S* of elementary terms. (In an AI context, these may be taken as referring to atomic perceptions or actions, and mathematical structures.) The set of ordered and unordered subsets of *S* is then constructed, and its elements are also considered as terms. Relationships are then defined as tuples of terms, and higher-order relationships are defined as predicates or functions acting on terms or relationships.

Atoms are associated with various data items, including

- Labels indicating type; e.g., a term may be a Concept term or a Number term; a relationship may be an Inheritance relationship or a Member relationship
- Packages of numbers representing "truth-value" (more on that later)
- In some cases, Atom-type-specific data (e.g., Number terms are associated with numbers; Word terms are associated with character strings)

We will sometimes refer to uncertain truth-values here in a completely abstract way, via notation such as <*t*>. However, we will also use some specific truth-value types in a concrete way:

- "strength" truth-values, which consist of single numbers; e.g., <*s*> or <.8>. Usually strength values denote probabilities but this is not always the case. The letter *s* will be habitually used to denote strength values.
- SimpleTruthValues, which consist of pairs of numbers. These pairs come in two forms:
  - the default, <*s,w*>, where s is a strength and *w* is a "weight of evidence" – the latter being a number in [0,1] that tells you,

qualitatively, how much you should believe the strength estimate. The letter *w* will habitually be used to denote weight of evidence values.

  - <s,N>, where N is a "count" – a positive number telling you, qualitatively, the total amount of evidence that was evaluated in order to assess s. There is a heuristic formula interrelating w and N, w=N/(N+k) where k is an adjustable parameter. The letter N will habitually be used to denote count. If the count version rather than the weight of evidence version is being used, this will be explicitly indicated, as the former version is the default.

- IndefiniteTruthValues, which quantify truth-values in terms of four numbers <[*L*,*U*],*b*,*k*>, an interval [*L*,*U*], a credibility level *b*, and an integer *k* called the lookahead. While the semantics of IndefiniteTruthValues are fairly complex, roughly speaking they quantify the idea that after *k* more observations there is a probability *b* that the conclusion of the inference will appear to lie in the final interval [*L*,*U*]. The value of the integer *k* will often be considered a system-wide constant. In this case, IndefiniteTruthValues will be characterized more simply via the three numbers <[*L*,*U*], *b*>.

- DistributionalTruthValues, which are discretized approximations to entire probability distributions. When using DistributionalTruthValues, PLN deduction reduces simply to matrix multiplication, and PLN inversion reduces to matrix inversion.[1]

The semantics of these truth-values will be reviewed in more depth in later chapters, but the basic gist may be intuitable from the above brief comments.

PLN inference rules are associated with particular types of terms and relationships; for example, the deduction rule mentioned in the Introduction is associated with ExtensionalInheritance and Inheritance relationships. At the highest level we may divide the set of PLN relationships into the following categories, each of which corresponds to a set of different particular relationship types:

- Fuzzy membership (the Member relationship)
- First-order logical relationships
- Higher-order logical relationships
- Containers (lists and sets)
- Function execution (the ExecutionOutput relationship)

To denote a relationship of type R, between Atoms *A* and *B*, with truth-value *t*, we write

```
R A B <t>
```

If *A* and *B* have long names, we may use the alternate notation

---

[1] We have so far developed two flavors of DistributionalTruthValues, namely StepFunctionTruthValues and PolynomialTruthValues.

```
R <t>
    A
    B
```

which lends itself to visually comprehensible nesting; e.g.,

```
R <t>
    A
    R1
          C
          D
```

Similarly, to denote a term *A* with truth-value *t*, we write

```
    A <t>
```

For example, to say that *A* inherits from *B* with probability .8, we write

```
Inheritance A B <.8>
```

To say that *A* inherits from *B* with IndefiniteTruthValue represented by <[.8,.9], .95>, we write

```
Inheritance A B <[.8,.9],.95>
```

(roughly, as noted above, the [.8, .9] interval represents an interval probability and the .95 represents a credibility level).

We will also sometimes use object-field notation for truth-value elements, obtaining, for example, the strength value object associated with an Atom

```
(Inheritance A B).strength = [.8,.9]
```

or the entire truth-value, using .tv

```
(Inheritance A B).tv = <[.8,.9], .9, 20>.
```

Finally, we will sometimes use a semi-natural-language notation, which will be introduced a little later on, when we first get into constructs of sufficient complexity to require such a notation.

2.2 Context

PLN TruthValues are defined relative to a Context. The default Context is the entire universe, but this is not usually a very useful Context to consider. For instance, many terms may be thought of as denoting categories; in this case, the strength of a term in a Context denotes the probability that an arbitrary entity in the Context is a member of the category denoted by the term.