A Cognitive Grammar Motivated Ontology For Processing With Term Cycles and Non-Classical Negation

Daniel T. Heinze, Ph.D.
A-Life Medical, Inc.
6055 Lusk Boulevard – Suite 200
San Diego, CA 92121
dheinze@alifemedical.com

Abstract

Circular definitions and hypotheses about the possible implications of counterfactual assertions are common in natural language. Most logic and ontology systems, however, do not provide meaningful results on definitions with term cycles and, because of limitations in the handling of negation, result in an undefined condition due to counterfactual assertions. L-space is an adaptation of the Scott lattice to computationally express the semantic space metaphor of natural language as described in Langacker's cognitive grammar. L-space is here presented in terms of its knowledge and truth ordering, showing how it, in fulfilling the modeling requirements of cognitive grammar, provides a solution to the persistent problems associated with term cycles and negation in logic based ontologies. We show that in L-space ontologies, fixpoint calculations on definitions that include the term cycles terminate at the intuitively expected solution. Beyond dealing with term cycles this feature is shown to be of importance for term categorization problems that are faced in representing ontologies for use in natural language processing. Negation in L-space is presented as a fundamental alternative to negation by failure and is compared to classical and strong negation. A method is introduced for using L-space to reason about conditional and counterfactual statements in terms of possible and impossible worlds.

Introduction

A particular advantage of L-space for use in natural language processing is its use of unrestricted roles. This feature, however, is dependent on other more basic capabilities that include the ability to reason intuitively about definitions that include term cycles and the ability to represent symmetrical, though non-classical, negation, which capabilities are the focus of this paper.

The L-space model of cognitive grammar has proven effective for ontology representation and reasoning in a major commercial system for information extraction from clinical medicine reports (Heinze, et al., 2001).

L-space (Heinze, 1994; Davenport, 1994; Heinze, et al., 1994; Heinze and Davenport 1995) was originally derived from Scott's mathematical theory of computation (Scott Lattices) (Scott, 1970a; Scott, 1970b) in fulfillment of the modeling requirements of cognitive grammar (Langacker, 1987; Langacker, 1991). Over the intervening decade, ontologies for use in natural language processing have continued largely to use description logic representations. Though useful, logic based ontologies have perennially struggled with the problems of an inability to deal elegantly and consistently with the term cycles that are common in natural language and the inability to achieve a symmetrical representation with non-classical negation. Among other things, these limitations make the use of unrestricted roles impractical. In this paper, we map a path from description logic to L-space demonstrating the nature of the noted problems for logic systems and the solution presented by L-space which by way of reviewing and updating the treatment of term cycles in L-space lays the foundation for introducing recent work on handling negation. This allows a consistent computation of symmetric non-classical negation and provides the intuitively expected fix-point computation of definitions that contain term cycles. Further, the interpretational problems due to conditionals and counterfactuals, will be introduced as representations of possible and impossible worlds in L-space as grounded in its treatment of negation.

Term Cycles, Ambiguity and Negation

Cycles in logic and description systems come in a variety of forms. Simple cycles such as "a human is an animal whose parents are human" are relatively easy to spot and eliminate if they are not desired in a particular ontological representation. It is argued that at least some cycles may be indicative of problems in the model (Horrocks and Sattler, 2003; Rector, 2002). Cycles, however, may be difficult to locate, are frequently a part of common sense definitions and are not necessarily incorrect, per the example above. It is, therefore, desirable to have a system in which calculations over term cycles have a natural termination and a proper interpretation.

Ambiguity is a related problem in defining ontologies. Medical terminology is fraught with ambiguities that are resolved by context. For example, in medicine, the shoulder region is variously considered part of the back and part of the upper extremity. Further, only part of the shoulder region is really considered part of the back. Hence, the ontology should both represent the ambiguity and provide a principled means for resolution.

Negation, or more generally certainty, is the third problem area. Medical parlance is highly conditioned: explicitly so ("possible pneumonia"); with the use of conditionals ("if the patient has pneumonia"); and with counterfactuals ("if the patient had not skipped her medication, she would not have become dizzy."). This last area is a rich and relatively little explored problem area, but it at least suggests the value of a system for negation that is non-classical in that it does not support the law of the excluded middle, is stronger than default negation in that it allows for representable and symmetrical negation and which can compute the feasibility and distance of some possible or impossible world as compared to the currently salient propositions.

Term Cycle and Negation Problems in Logic

Semantic representation schemes such as predicate logic, default logic, autoepistemic logic, description logic, is-a hierarchies, semantic nets and production systems have fundamental similarities that can be abstractly modeled in terms of lattices. For purposes of investigating the problems of terminological cycles, (Nebel, 1991) presents a small terminological formalism $\mathcal{TLN}$ that is a subset of the above noted systems. $\mathcal{TLN}$ is sufficiently abstract and has sufficient coverage to make it useful for analysis and comparison purposes as a starting point. Note that several logic-based approaches to the term cycle problem have been developed (Horrocks and Sattler, 2003) but are, by nature, workarounds.

$\mathcal{TLN}$ is composed of a set **R** of atomic roles R and a set **A** of atomic concepts A and B along with the elements TOP ($\top$), that denotes everything, and BOTTOM ($\bot$) that denotes nothing. Set **D** consisting of concept descriptions C and D is defined by the rules:

$C, D \rightarrow \mathbf{A}$	*atomic concept*	|
C & D	*concept conjunction*	|
$\forall R: C$	*value restriction*	|
$\exists^{\geqslant n} R$	*minimum restriction*	|
$\exists^{\leqslant n} R$	*maximum restriction*	

A description written in this language is intended to categorize all *objects* that fulfill the description. An interpretation $I = <\delta, [[\cdot]]^I>$, where δ is the domain and $[[\cdot]]^I$ is the interpretation function that maps atomic concepts to a subset of δ and atomic roles to the total functions from δ to 2^{δ}. A terminology T is a total function T: $\mathbf{A} \rightarrow \mathbf{D}$, where $T(A)$ is the concept description defining the meaning of A or, if A is primitive in the terminology, $T(A) = A$.

An interpretation I is a model of T *iff* $[[A]]^I = [[T(A)]]^I$ $\forall A \in \mathbf{A}$.

C is subsumed by D in the terminology $T(C \leqslant_T D)$ *iff* $[[C]]^I \subseteq [[D]]^I$ *for all models I of T.*

C is equivalent to D in the terminology $T(C =_T D)$ *iff* $[[C]]^I = [[D]]^I$ *for all models I of T.*

C is incoherent in terminology T *iff* $C =_T \bot$.

As a basis for understanding L-space, note the manner in which concepts of progressively greater specificity are formed by means of conjunction and possibly also restriction. For example, *woman* and *mother-of-daughter* can be defined as:

T(*woman*) = HUMAN & FEMALE

T(*mother-of-daughter*) = WOMAN & $\exists^{\geqslant 1}$ CHILD & $\forall$CHILD: *woman*

As concepts become more specific, the movement is from $\top$ towards $\bot$.

Given the terminology T with fixed **A** and **R**, the set of interpretations that have the same initial partial part f (i.e. are identical in terms of roles and primitive concepts) are denoted $\mathbf{\Psi}_f$. $\mathbf{\Gamma}$ is a function mapping interpretations to interpretations. A fixed point of $\mathbf{\Psi}$, that is an interpretation I with the property $\mathbf{\Psi}(I) = I$, is an admissible model of terminology T. For such a system, there exists both a least fixpoint (lfp) and a greatest fixpoint (gfp) for $\mathbf{\Psi}(I) = I$.

Relative to ontology definitions, there are two particular problems in $\mathcal{TLN}$: categorization with inheritance and definitions with term cycles. The tenacity of these problems is evident in the variety of logics that grapple with these problems using varieties of nonmonotonicity and restrictions on cycles.

To enable a comparison to L-space, we recast $\mathcal{TLN}$ as a lattice and will refer to the lattice representation of $\mathcal{TLN}$ as $\mathcal{TLN}$-space. The term ENTITY is be assigned to $\bot$, which is the root of the hierarchy of all THINGS and RELATIONS. In $\mathcal{TLN}$-space, $\bot$ contains nothing (i.e. ENTITY is defined by no concepts, either atomic or derived). A set of atomic (i.e. self-defining) concepts is also necessarily defined else it is otherwise impossible to build a lattice beyond $\bot = \top$ where $\bot = \varnothing$.

To illustrate the problem of categorization with inheritance, add the atomic concept LIVING. Without being too concerned about the exact semantics, we derive the concepts T(*plant*) = THING & LIVING and T(*mineral*) = THING & *exists*$^{\leqslant 0}$ LIVING. Add the atomic concept IG (ingestible) meaning is suitable for eating. If, at this point, term definitions are to be introduced which use the atomic concept IG, there is a question of what strategy to use because only some THINGs, some *plant*s and some *mineral*s should be marked as IG. Either the entire lattice can be divided into IG *thing*s and those not so marked, or the introduction of IG can be delayed. With the introduction of an increasing number of markers, the first strategy degenerates from categorization to listing. The second strategy sacrifices the semantic that some THINGs, some *plant*s and some *mineral*s are IG. This problem can be solved using restrictions, but this sacrifices monotonicity (Daelemans, et al., 1992).

A term cycle exists when within some terminology T, there is a concept A in which $T(A)$ is defined using A. To illustrate, consider the $\mathcal{TLN}$-space definition T(*human*) = *animal* & $\forall$PARENT: *human.* For use in L-space, computing the lfp is the most desirable solution because it corresponds to the partial function that gives results for terminating computations and is undefined for nonterminating computations. However, in the presence of term cycles, the lfp is a problem as seen in (Nebel, 1991) who proves that in such propositions, A (e.g. *human*) evaluates to $\overset{lfp}{_T}\bot$, i.e. nothing, hence A and *human*s are incoherent definitions, i.e. they do not exist.

Finally, by way of introducing the logic problems to be addressed, is negation. In partial valuation (or three-valued) logics "the space $\{\bot$, *false, true*$\}$ is given a knowledge ordering $\bot <_k$ *false*, with $\boldsymbol{x} <_k \boldsymbol{y}$ not holding in any other case." (Fitting, 2002). This provides the lower semi-lattice, with $\bot$ evaluating to undefined, that is neither *true* nor *false,* that is common for many knowledge representation schemes. In such schemes, negation is negation by failure (default negation), and the ordering of concepts is with less specific concepts closer to $\bot$ and specificity being built by adding semantic markers (e.g. roles and concepts) thus moving up the semi-lattice or by restricting (negating) inheritance. Such constructs are non-monotonic, and ***P*** ← *not* ***P*** evaluates to $\bot$ (neither *true* nor *false*).

Belnap logic (Belnap, 1977) is a four-valued logic in which the lattice is complete with $\top$ being both *true* and *false*. This provides two natural orderings, knowledge ordering which increases from $\bot$ as in three-valued logic, and truth ordering which increases from left (*false*) to right (*true*). The *lfp* of ***P***← *not* ***P*** evaluates to $\bot$ (neither *true* nor *false*), and the *gfp* of ***P***← *not* ***P*** evaluates to $\top$ (both *true* and *false*).

L-space

As compared to Belnap logic, in L-space the relation between the knowledge ordering and the means by which specificity is increased is inverted. For purposes of convention, L-space maintains the notion of knowledge (specificity) increasing from $\bot$ to $\top$ but inverts the method of increasing specificity (from this point the term specificity will be used) such that in the truth lattice, $\bot$ is both *true* and *false*, and $\top$ is neither *true* nor *false*. A lattice of this nature is described by (Scott, 1970a) and (Scott, 1970b) as a means of proving a consistent and correct framework in support of the notion of unrestricted functions. Five axioms from Scott provide the framework for constructing L-space. In reviewing the axioms, we preserve Scott's use of the term "data-types" as the correspondence to logic is obvious.

Given $x, y \in D$ are two elements of a data type, we say that y is a better version of what x is trying to approximate. This is represented as $x \sqsubseteq y$, meaning y is consistent with x and (possibly) more accurate than x. To avoid confusion, it can also be stated that the intent is to imply that x and y are both approximations of the same thing, but y gives more information about that thing than does x. Consistent data types should always be represented in this manner. Further, $\sqsubseteq$ is reflexive, transitive and antisymmetric.

Axiom 1: A data type is a partially ordered set.

Given data types D and D' with partial orderings $\sqsubseteq$ and $\sqsubseteq'$, and $f\colon D \rightarrow D'$ is a reasonable mapping, then for $x, y \in D$ and $x \sqsubseteq y$ then $f(x) \sqsubseteq' f(y)$. That is: $x \sqsubseteq y \Rightarrow f(x) \sqsubseteq' f(y)$, or f is monotonic with respect to partial orderings.

Axiom 2: Mappings between data types are monotonic.

Given the partiality of the data this has the effect that functions also become partial and hence the values that they produce are also. An infinite sequence of approximations of the form $x_0 \sqsubseteq x_1 \sqsubseteq \ldots x_n \sqsubseteq x_{n+1} \sqsubseteq \ldots$ tends toward a limit, say $y = \bigsqcup_{n=0}^{\infty} x_n$. As such, the limit is the least upper bound (*lub*). Since the successive terms of the sequence provide more and more information, the limit represents a *join* of the separate contributions. Because the existence of the *lub* implies the existence of the greatest lower bound (*glb*), a complete lattice is formed.

Axiom 3: A data type is a complete lattice under its partial ordering.

If $x, y \in D,$ there exists a *lub* or *join* $x \sqcup y \in D,$ and all D has a *lub* called $\top$. In this sense, $\top$ is an over-determined element. The *lub* of the empty subset of D is the element $\bot \in D,$ the most under-determined element. The *lub* of the set of all lower bounds of a subset $X \subseteq D,$ is the *glb* $\sqcap X \in D$.

$X \in D$ is directed if every finite subset $\{x_0, x_1, \ldots, x_{n+1}\} \subseteq X$ has an upper bound $y \in X$ such that $x_0 \sqcup x_1 \ldots \sqcup x_{n+1} \sqsubseteq y$. The limit of the directed set is the *lub* $\sqcup X$. For some specified finite amount of information about $\sqcup X$, by directedness, all of it is contained in at least one single element of X.

Given a monotone function $f\colon D \rightarrow D'$ and limit $\sqcup X$ of a directed subset $X \subseteq D$, we will say function $f(\sqcup X)$ requests a finite amount of information about $\sqcup X$. Since only a single element $x \in X$ is needed, $f(x)$ returns what is needed. Further, all the information about $f(\sqcup X)$ is the limit of its finite parts, hence $f(\sqcup X) = \sqcup\{f(x)\colon x \in X\}$. As such, the mapping preserves the limit, which implies that it is continuous.

Axiom 4: Mappings between data types are continuous.

A data type D satisfying Axioms 1 and 3 can be regarded as a topological space. In this space, a *basis* (say $E \subseteq D$) is a dense subset of the space in terms of which all the other elements can be found as limits. To make data types "physical", E must be "known" in the sense that it is at most countably infinite and enumerable. Given such an E:

Axiom 5: A data type has an effectively given basis.

So then, it is possible to give effectively better and better approximations to x, which converge to x in the limit. Although there may be uncountably many elements in D, there can only be countably many computable elements.

In addition to these axioms, we note from Scott three important constructs for obtaining new data types from the given (atomic) ones: $(D \times D')$, $(D + D')$, $(D \rightarrow D')$.

The Cartesian product $D \times D'$ has as elements pairs $\langle x, x'\rangle$ where $x \in D$ and $x' \in D'$, and for which $\langle x, x'\rangle \sqsubseteq \langle y, y'\rangle$ *iff* $x \sqsubseteq y$ *and* $x' \sqsubseteq y'$.

The sum $D + D'$ is defined as a disjoint union D and D' in which $\bot = \bot$ and $\top = \top'$.

The function space $D \rightarrow D'$ has as elements all the continuous mappings from D into D' for which $f \sqsubseteq g$ *iff* $f(x) \sqsubseteq' g(x)\ \forall x \in D$.

As an example, consider the real number lattice in Figure 1. $\bot$ is the interval of all real numbers. It is maximally under-specified and as such contains no information. $\top$ is the empty interval. It is maximally over-specified and as such contains too much information. Moving from $\bot$ to $\top$ one encounters elements with increasing specificity. A truth lattice so formed corresponds to the semi-inverted Belnap logic discussed above.

The axioms and constructs discussed here are sufficient for the semantics of Langacker's cognitive linguistics model ((Langacker, 1987) and (Langacker, 1991)) or see (Heinze, 1994) for a concise summary).

In cognitive linguistics, instead of atomic markers, there are basic domains. Each basic domain is a space, and in L-space, abstract domains are spaces formed either by the Cartesian product of or by the mapping of one domain into another. In either case, the original domains may be basic and the result will always be abstract. $\bot$ is maximally under-specified and as such is information-free by maximum entropy. In the real number lattice, $\bot$ contains all real numbers. It is, therefore, the maximally under specified approximation of all real numbers and so contains no information. Information is introduced as intervals are created that more closely approximate the real numbers, which are themselves degenerate intervals. Increasing specification or narrowing of the intervals, then, introduces increasing information. The degenerate intervals, which perfectly approximate the real numbers, are perfectly specified and contain perfect information. Continued specification beyond this level results in over-specification. $\top$ is said to be maximally over-specified or, in other words, there is too much information. The notion is the same for L-space, except, instead of intervals, the more general notion of subspaces is employed.

With the lattice in this orientation, computation from $\bot$ to $\top$ progresses in the direction of increasing conceptual information. By way of interest, it is noted that in the L-space, by duality, computation from $\top$ is defined as perceptual. In this regard, the lattice is not isomorphic in that $\bot$ is now empty and, as such, is perceptually over-specified whereas $\top$ is the space of all perception and is perceptually under-specified. From this perspective, $\top$ is the perfect carrier signal.

The basic domain lattices of L-space are convex spaces, and their combination into abstract domains via Cartesian products or mappings forms convex spaces. This property of convexity is maintained by $\sqcap$ to the end that the result of the perception is over-specified as compared to the objective reality that provides the stimulus. By duality, the resulting conception(s) formed by this process are under-specified or more general than the realities they represent. As concepts are coded into language, the process is in the direction of $\bot$ with the result that linguistic events virtually always under-specify the concepts from which they are coded. This is exactly Langacker's claim and is really just a principled way of saying that language is ambiguous.

In the real number lattice, the basis consists of the intervals with rational end points plus the element $\bot$. Because L-space is intended as a computer implementation, it is finite and the space is discrete. Hence, the basis consists of the subspaces with discrete end points. Due to finiteness, this implicitly includes $\bot$. (Davenport and Heinze, 1994) and (Davenport, 1994) present a proof that the discrete representation maintains the quality of monotonicity and approximates the continuity of the continuous representation in a way that preserves its qualities.

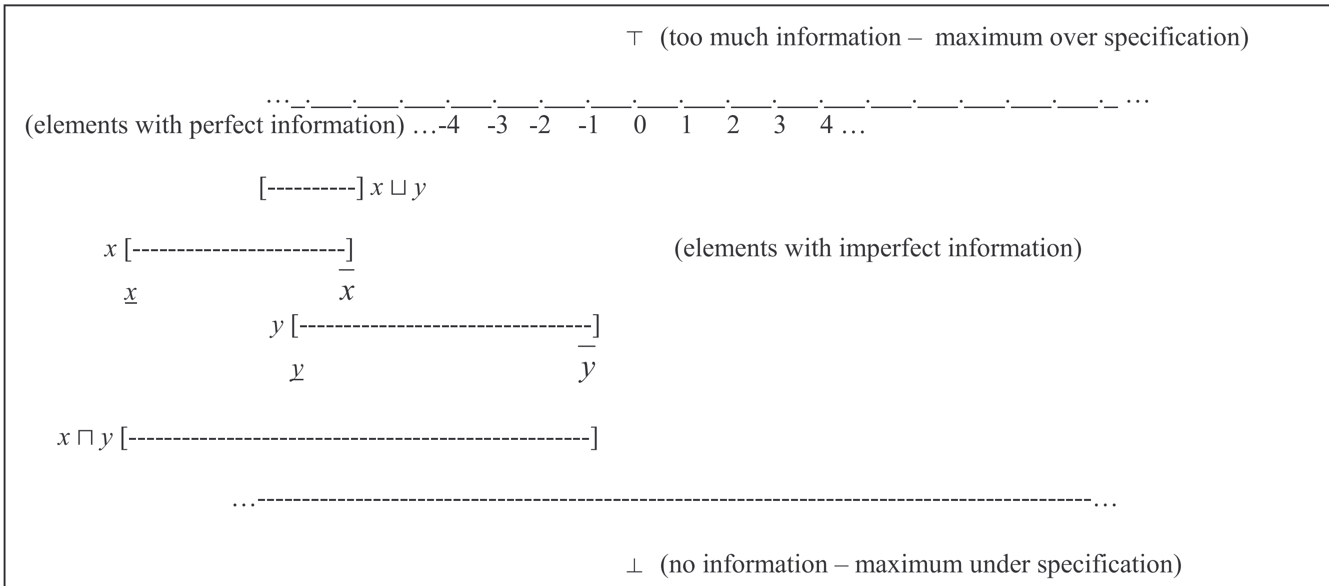

Figure 1: The real number Scott lattice

With this lattice, we can define Langacker's conceptual relations (Langacker, 1987):

1. *Inclusion*: [A IN B] or [B INCLUDE A]
2. *Separation* (noncoincidence): [A OUT B] or [B OUT A]
3. *Identity* (coincidence): [A ID B] or [B ID A]
4. *Association* (location within a neighborhood): [A ASSOC B] or [B ASSOC A]

Given three subspaces $[\underline{x}, \overline{x}]$, $[\underline{y}, \overline{y}]$ and $[\underline{z}, \overline{z}]$, we can define:

1. *Inclusion*: $[[\underline{x}, \overline{x}]$ IN $[\underline{y}, \overline{y}]]$ *iff* $\underline{x} \geqslant \underline{y}$ and $\overline{x} \leqslant \overline{y}$
2. *Separation*: $[[\underline{x}, \overline{x}]$ OUT $[\underline{y}, \overline{y}]]$ *iff* $\underline{x} \leqslant \underline{y}$ and $\overline{x} \geqslant \overline{y}$
3. *Identity*: $[[\underline{x}, \overline{x}]$ ID $[\underline{y}, \overline{y}]]$ *iff* $\underline{x} = \underline{y}$ and $\overline{x} = \overline{y}$
4. *Association*: $[[\underline{x}, \overline{x}]$ ASSOC $[\underline{y}, \overline{y}]]$ *iff* $[[\underline{x}, \overline{x}]$ IN $[\underline{z}, \overline{z}]$ and $[\underline{x}, \overline{x}]$ NOT IN $[\underline{y}, \overline{y}]]$

Finally:

1. *Sanctioning*: $[\underline{y}, \overline{y}]$ sanctions $[\underline{x}, \overline{x}]$ *iff* $([[\underline{x}, \overline{x}]$ IN $[\underline{y}, \overline{y}]]$ or $[[\underline{x}, \overline{x}]$ ID $[\underline{y}, \overline{y}]])$
2. *Partial Sanctioning*: $[\underline{y}, \overline{y}]$ sanctions $[\underline{x}, \overline{x}]$ *iff* $(((\underline{x} \leqslant \underline{y})$ and $(\underline{y} \leqslant \overline{x} \geqslant \overline{y}))$ or $((\underline{y} \leqslant \underline{x} \geqslant \overline{y})$ and $(\overline{x} \geqslant \underline{y})))$
3. *Non-sanctioning*: $[\underline{y}, \overline{y}]$ does not sanction $[\underline{x}, \overline{x}]$ *iff* $[[\underline{x}, \overline{x}]$ NOT IN $[\underline{y}, \overline{y}]]$
4. *Classification*: $[\underline{z}, \overline{z}]$ is a super-class of $[\underline{x}, \overline{x}]$ and $[\underline{y}, \overline{y}]$ *iff* $([[\underline{x}, \overline{x}]$ ASSOC $[\underline{y}, \overline{y}]]$ or $[\underline{y}, \overline{y}]$ sanctions $[\underline{x}, \overline{x}])$ and $([[\underline{x}, \overline{x}]$ IN $[\underline{z}, \overline{z}]]$ and $[[\underline{y}, \overline{y}]$ IN $[\underline{z}, \overline{z}]]))$
5. *Salience*: *if* S is a lattice of salience metric 0 or 1 and D is a lattice, *then* $S \times D = E$ where the salience of any point with respect to any interval will by 1 if the point is within the interval an 0 otherwise. (More precisely, S is a metric from 0 to 1 and the Cartesian product allows for a metric that indicates the level salience of an intensional element to the whole concept.)

Term Cycles in L-space

The L-space lattice provides for consistent fixpoint calculations. This is true even when the lattice contains term cycles. Fixpoint calculations converge to the correct, not to mention intuitively obvious, bound. This bound is the interval that accommodates all aspects of the term being defined while including no unnecessary information. The interval or space nature of the lattice representation directly accomplishes the goals of computing partial sanctioning, and because the lattice is monotone, fix-point calculations are direct and principled.

The term cycle example given earlier, T(*human*) = *animal* & ∀PARENT: *human,* converges as follows:

1. Insert ⊥ in the place of "human" in the right hand side of the definition.
2. Apply the PARENT restriction: *f*(PARENT, ⊥) = ⊥ (this would also be ⊥ in $\mathcal{TLN}$, et al.)
3. Take the intersection with *animal*: *animal intersect* ⊥ = *animal* (this would, however, again be ⊥ in $\mathcal{TLN}$, et al.)
4. Iterate, inserting *animal* in the right hand side of the definition. (at this point, an endless loop begins in $\mathcal{TLN}$, et al. with ⊥ as the solution even if the loop is blocked.)
5. Apply the PARENT restriction: *f*(PARENT, *animal*) = *animal*
6. Take the intersection with *human*: *human intersect animal = human*

Term Categorization in L-space

To properly understand categorization in L-space, it must be recognized that whatever intensional elements are salient in any instantiation will also be salient to some degree in the extensional parent(s) of that instantiation. More properly, the intensional definition of any sense is the normalized sum of the intensional definitions of all the extensions of that sense. Hence, categorization is not dependent on conjunction with delayed restriction/negation of intensional elements, thus introducing non-monotonicity. Rather, all extensions are subject to restriction/negation in a manner that maintains monotonicity. Further, by this method, ambiguity is represented both extensionally and intensionally, as will be exemplified below.

The interpretation of the L-space lattice in information terms is thus different from logic constructs, but more accurate in terms of information theory because ⊥ is information-free due to maximum entropy rather than due lack of semantic markers. ⊤ is also information-free in that it is the perfect carrier signal. Information exists between the bounds of ⊥ and ⊤ to the extent that there is some degree of entropy with restraints.

Although L-space in some ways reflects the structure of a semantic net, there are important differences. Membership of a sense in a class is determined by its existence as a subspace of the space (schema) defining the class. This allows for very fuzzy definitions of classes and class inclusion. In a semantic net, class inclusion is determined by the presence or absence of a set of features (markers) constituting the class. A second difference is that L-space allows partial sanctioning of one sense as another. This is vitally important to proper cognition of novel uses in natural language.

Considering the initial example of a term categorization problem, namely that the shoulder region is ambiguously defined as part of both the back and upper extremity, L-space allows for a representation as follows. As with is-a relations, meronomy is transitive and can also be used as the extensional axis for ontology, e.g. the regional or systemic ontology of anatomy. Remember also that in L-space, any intension of the child/part is an intension of the parent/whole but at a likely lower salience. As such, the shoulder region could be instantiated as a part of both the back and upper extremities and each would have an intensional definition that included the intensional definition of the shoulder region. This would be acceptable except that the ambiguity in these terms is deeper. The concept represented by "back" sometimes includes the shoulder region and sometimes does not with the distinction being inferred from the medical condition for which "back" is the site. So then, both "shoulder" and "back" are ambiguous such that "shoulder" when spoken of as part of the "back" is in reality only part of the full shoulder region, and "back" may or may not include the intensional elements of the restricted sense of "shoulder". This situation is represented in L-space by making "shoulder" a peer of "back" and "upper extremity" and using two spaces for "back", one of which has the restricted sense of "back" without "shoulder" as a part and one of which has the expanded sense of "back" with "shoulder" as a part. Intensionally, each will be sanctioned only by the particular medical conditions with which they are appropriate. The ambiguity problem with only part of the shoulder region being part of the expanded sense of "back" is handled similarly. In this way, both the upper and lower bounds are correct and so computation from either direction is well behaved. Note also, that in logic based ontologies the ambiguity inherent in both term abstraction and polysemy is expressed extensionally but is not properly captured intensionally.

There are obvious compute space considerations with the L-space representation; however, there are implementation shortcuts that stem from the theory and that provide for computational tractability in practice.

Negation in L-space

Negation by failure is an obvious possibility in L-space in that a search for a non-instantiated space will evaluate to $\top$. As such, the certainty of negation by failure in L-space is higher than is often the case with negation by failure in logic systems where exhaustive search might not be employed.

Classical negation ($\neg$) is not possible in L-space because the complement of the set of recursively enumerable relations is not closed under complementation (Fitting, 2002). Further, classical negation satisfies the law of the excluded middle thus forcing a *true* or *false* decision with no room for uncertainty (Alferes, et al., 1998). However, on the positive side, classical negation is symmetrical whereas default negation is not.

Negation in L-space is obviously non-classical, does not satisfy the law of the excluded middle, but is symmetrical and, by *separation*, represents negation in a strong sense (although the semantics as presented above are different from "strong negation" (Alferes, et al., 1998) in terms of the *meet* and *join* of true and false.) See also (Rondogiannis and Wadge, 2002) for another description of an infinite-valued negation.

Again drawing on a common issue in the parlance of clinical medicine, we have the issue of degrees of certainty and the presentation of hypotheses as counterfactuals. The following brief discussion of this topic illustrates a conceptualization in terms of L-space negation. It is presented as an introduction to work in progress. Counterfactuals are hypotheses that are the inverse of some actual fact. These types of constructs are common in natural language in situations where, for example, medical practitioners hypothesize that if X had not occurred then Y would not have occurred (when in fact, X and Y did occur). The result of such possible worlds hypotheses is a potentially alternative set of pragmatics under which understanding and reasoning take place across the scope of the hypothesis. Logic or ontology systems in which (a | b) is undefined when b = false (which implies (a | b) = (a | c) where b = false and c = false for all b and c) do not yield a useful possible world semantic. Following is an initial proposal for handling the situation in L-space.

If X and Y are actual and the hypothesis is "if not X, then Y", compute the *meet* of the salient concept spaces in which X is not evident (this will be some threshold value on the epistemic value of X) and select the epistemic value Y'. The resultant space that Y' defines is the extent of the effect of the negation of X and can be checked, as a possible world, for consistency with the observed world.

Given "if not X, then not Y", compute the *meet* of the salient concept spaces in which X is not evident and take 1 minus the resultant epistemic value for Y.

If performing an impossible worlds type scenario such as "if X = not X, then Y = not Y", then compute the *meet* of the salient concept spaces in which X is entails Y into S_x. Then abstract S_x (move it further in the direction of under-specification, i.e. $\perp$) far enough that the epistemic value of X falls below some threshold T_x. Finally, evaluate this space for the epistemic value of Y (call this E_y). The truth-value (V) of the original statement is:

$$\text{If } E_y > T_x \text{ then } V = 1 - (E_y - T_x).$$

$$\text{If } E_y \leqslant T_x \text{ then } V = 1$$

The distance between S_x and the space in which E_x falls to T_x is an indication of just how much and where the observed world conception must be modified to make the statement true. With $\perp$ taken to be maximally under-specified, the original statement is true for all X and for all Y at $\perp$, i.e. in the limit, because X and Y take on all epistemic values at $\perp$. If a greatest lower bound is found somewhere above $\perp$, it is an indication that the counter-factual can be accommodated without sacrificing one's entire conception of the world.

This method relies on an interpretation of conditionals consistent with Adams' (Adams, 1966; Adams 1974; Bamber, 1994; Bamber, 2000). In prepositional logic, the statement *if A then B* is expressed using the material conditional symbol as $A \supset B$. However, because this statement is equivalent to $\neg A \vee B$, Adams proposes that it is not equivalent to the natural language intent of *if A then B,* which he instead construes as $\Pr(B \mid A)$ *is close to one*.

This is consistent with the notion in L-space that the salience, and hence the probability, of any observation or statement is one only at $\perp$ and zero only at $\top$. This definition allows us to compute some interior point even for a statement such as "if X = not X, then Y = not Y" without converging at either $\perp$ or $\top$.

Conclusion

L-space provides a natural language motivated solution to several persistent ontology construction problems that are faced in dealing with definitions that contain cycles and ambiguity. With regard to negation, L-space possesses the desirable qualities of both classical and non-classical negation while overcoming the limitations of each. In this regard, L-space also holds promise of being useful for reasoning about hypothetical situations as presented by several commonly used conditional constructs. Further investigation in the area of conditional and counterfactual constructs seems warranted, particularly because this area has received minimal treatment from the natural language processing community.

References

Adams, E. W. 1966. Probability and the Logic of Conditionals. In *Aspects of Inductive Logic*. J. Hintikka and P. Suppes, eds. 265-316. Amsterdam: North Holland Publishing.

Adams, E. W. 1974 The Logic of "Almost All". *Journal of Philosophical Logic* 3:3-17.

Alferes, J.; Pereira L. M.; and Przymusinski, T. C. 1988. "Classical" Negation in Non-Monotonic Reasoning and Logic Programming. *Journal of Automated Reasoning* 20.

Bamber, D. 1994. Probabilistic Entailment of Conditionals by Conditionals. *IEEE Transactions on Systems, Man and Cybernetics* 24:1714-1723.

Bamber, D. 2000. Entailment with Near Surety of Scaled Assertions of High Conditional Probability. *Journal of Philosophical Logic* 29:1-74.

Belnap Jr., N. D. A 1977. Useful Four-Valued Logic. *Modern Uses of Multiple-Valued Logic*, J. M. Dunn and G. Epstein eds. Reidel.

Daelemans, W.; De Smedt K.; and Gazdar, G. 1992. Inheritance in Natural Language Processing. *Computational Linguistics* 18(2).

Davenport, D. M. 1994. Basal Graphs over Finite Alphabets. CIIP Technical Report, The Pennsylvania State University.

Davenport, D. M.; and Heinze, D. T. 1994. Consistent Least Fixpoint Computation for Natural Language Processing. CIIP Technical Report, The Pennsylvania State University.

Fitting, M. 2002. Fixpoint Semantics for Logic Programming: A Survey. *Theoretical Computer Science* 278(1-2).

Heinze, D. T. 1994. Computational Cognitive Linguistics. Ph.D. diss., Department of Industrial and Management Systems Engineering, The Pennsylvania State University.

Heinze, D. T.; Davenport, D. M.; Kumara, S. R. T.; and Crowder, W. 1994. Crisis Action Message Analyzer. *Proceedings: DoD 1994 Government/Industry ILS Exchange Conference.* Society of Logistics Engineers.

Heinze, D. T.; Morsch, M.; Sheffer, R.; Jimmink, M.; Jennings, M.; Morris, W.; and Morsch, A. 2001. LifeCode: A Deployed Application for Automated Medical Coding. *AI Magazine* 22(2):76-88.

Horrocks, I.; and Sattler, U. 2003. The Effect of Adding Complex Role Inclusion Axioms in Description Logics. *Proc. of the 18th Int. Joint Conf. on Artificial Intelligence (IJCAI 2003)*. Morgan Kaufmann.

Langacker, R. W. 1987. *Foundations of Cognitive Grammar: Vol I – Theoretical Prerequisites*. Stanford University Press.

Langacker, R. W. 1991. *Foundations of Cognitive Grammar: Vol II – Descriptive Application.* Stanford University Press.

Nebel, B. 1991 Terminological Cycles: Semantics and Computational Properties. In *Principles of Semantic Networks,* J. F. Sowa ed. Morgan Kaufmann Publishers, Inc.

Rector, A. 2002. Analysis of Propagation Along Transitive Roles: Formalization of the Galen Experience with Medical Ontologies. *Proceedings of DL 2002*, CEUR-WS.

Rondogiannis, P.; and Wadge, W. W. 2002. An Infinite-Valued Semantics for Logic Programs with Negation. *Proceedings of the 8th European Conference of Logics in Artificial Intelligence.* Cosenza, Italy.

Scott, D. 1970a. Outline of a Mathematical Theory of Computation. *Proceedings of the Fourth Annual Princeton Conference on Information Sciences and Systems*. Princeton University.

Scott, D. 1970b. *Outline of a Mathematical Theory of Computation*. Technical Monograph PRG-2. Oxford University Computing Laboratory, Programming Research Group.