A Dynamic Semantics for Sense Extension

Carl Vogel
O'Reilly Institute
Trinity College
University of Dublin
Dublin 2
Ireland

Catherine Collin
Centre for Cognitive Science
University of Edinburgh
2 Buccleuch Place
Edinburgh EH8 9LW
UK

Ulrike Hahn
Department of Psychology
University of Warwick
Coventry CV4 7AL
UK

{vogel,cati,ueh}@cogsci.ed.ac.uk

Summary: Sense extension, specifically as required in the generation of metaphor, presents a problem for formalization of a linguistic domain as it requires dynamic interpretation of predicates. We extend a classical model of formalisation to incorporate generation of metaphor. This involves complicating I, the interpretation function, by threading it through each clause where it could potentially be extended by a sense extension of the object interpreted. Further extensions include incorporating predication constructs of the copula *is* in the language and maintaining monotonicity in entailments for literal predicates. This work has application in other areas of linguistic analysis that involve sense extension.

Areas: Representational Formalisms, Formalizability, Sense Generation, Open Texture, Metaphor Generation.

Contents

1 Introduction

This paper is about the problem of sense extension. "Sense" can be understood as its non-technical manifestation — the sense or meaning of a word. Specifically, this paper deals with sense extension as applied to non-literal meaning, i.e. the generation of metaphor.[1] Sense generation becomes a problem when one addresses the issue of formalization of a linguistic domain in which it appears. Typically, to construct a formalization, one uses tools provided by logic — we define a language and interpret its semantics compositionally in relation to a world. We may consider a dynamic interpretation, that is allowing interpretation of the sentence itself to have an effect on aspects of the semantics. Previous approaches to dynamic semantics have focused on just the variable assignment functions. However, the problem of sense extension seems to require more complex treatment, as it's not just the variables that have dynamic interpretation, but predicates as well. In this paper, we develop an analysis of sense extension in terms of a novel dynamic interpretation of predicates in a formal logic. It is important to provide such a model if one wants, generally, to arrive at a deeper understanding of the semantics of natural language and specifically, for projects such as the formalization of law (see Hahn & Vogel, 1995). The model is described as classical because we examine sense generation in the context of a standard first order logic with the usual semantics. The classical model clarifies how one can go about accommodating sense generation in a formal system of restrictive expressive capacity. The model behaves appropriately with respect to a range of examples phenomena associated with sense extension. We present the system incrementally, and discuss its advantages and limitations.

2 A Classical Approach

We aim to make use of the insights into formalization achieved primarily during the first half of this century (for example: Turing, 1936; Church, 1936; Post, 1936). In particular, that there is an equivalence between definability/deducibility in certain logics and extant models of effective computation, where effective computability includes notions of strong limits to computability (along with the thesis that any other model of effective computation will also be equivalent), allows us to conclude that definability in a logic with appropriate expressive capacity yields as adequate a sense of formalization as is possible, modulo issues of aesthetics and perspicuity. That is, if it's not formalizable in a logical language, then it isn't formalizable. Note that there are logics which are in some sense more powerful than first-order logic. But they are also less powerful in the sense that they do not guarantee the same provability properties for all sentences expressible in the language thus provided. Thus, we begin with a classical first order language.

2.1 The Classical Tools

To keep the discussion self-contained we provide a description of the usual compositional semantics for first order predicate calculus.[2] This is done with an abstraction of the

[1] It is also the essence of the problem of open texture in legal theory. This problem can be loosely characterized as follows: we have laws which might be formulated as (suitably restricted) universally quantified conditionals; we also have individual cases to which the law may or may not apply because it may or may not fall under the extension of the restricting predicate; such cases require extending the literal interpretation of the restricting predicate to include the case at hand.

[2] The recursive presentation of the syntax we assume is immediate in the interpretation clauses given: there are no other ways of forming a legal expression except via combinations of those possibilities of combination of

world as a nonempty domain D of entities, and with an interpretation function I that maps constants to entities and predicates to elements of the domain. The semantics is thus extensional in that what a predicate *means* is the set of entities it is true of.[3] Because it is a first order system, we also require functions g for the variables that pick out entities ranged over. We assume one I, but any number of g.

- $I(c) \in D$ for each constant in the language L.
- $g(x) \in D$ for each variable in L.
- $I(P^n) \subseteq D^n, n > 0$
- $I(P) = D$ if P is understood as a true proposition letter in L, and $\emptyset$ otherwise.

It is standard to define relative to each assignment function g a related function $g_{[x/d]}$ which is just like g in assignments of elements in the domain to all of the other variables in L apart from x; the value assigned to x by the related function is d. Now it is possible to define a function ($[\![\ ]\!]$) which maps well-formed expressions in L to their meanings.

- $[\![c]\!] = I(c), \forall c \in L$
- $[\![x]\!]^g = g(x), \forall x \in L$
- $[\![P]\!] = I(P)$
- $[\![P^n(t_1, ...t_n)]\!]^g = D$ if $\langle [\![t_1]\!]^g, ...[\![t_n]\!]^g \rangle \in I(P^n), \emptyset$ otherwise.
- $[\![\neg P^n]\!]^g = D$ if $[\![P^n]\!]^g = \emptyset$, and $\emptyset$ otherwise.
- $[\![P^n \vee Q^m]\!]^g = D$ if $[\![P^n]\!]^g = D$ or if $[\![Q^m]\!]^g = D$, and $\emptyset$ otherwise.
- $[\![P^n \wedge Q^m]\!]^g = D$ if $[\![P^n]\!]^g = D$ and $[\![Q^m]\!]^g = D$, and $\emptyset$ otherwise.
- $[\![P^n \Rightarrow Q^m]\!]^g = D$ if $[\![P^n]\!]^g = \emptyset$ or if $[\![Q^m]\!]^g = D$, and $\emptyset$ otherwise.
- $[\![\forall x P^n]\!]^g = D$ if for all $d \in D$ $[\![P^n]\!]^{g[x/d]} = D$, and $\emptyset$ otherwise.
- $[\![\exists x P^n]\!]^g = D$ if for some $d \in D$ $[\![P^n]\!]^{g[x/d]} = D$, and $\emptyset$ otherwise.

Note that the mapping from expression to meaning is relative to the domain and the interpretation function. It also depends on the assignment functions that map variables to the domain. For a case like $[\![P^n \vee Q^m]\!]^g$, if there are no free variables in either P^n or Q^m, whatever arity n and m are set to, then the choice of variable assignments doesn't matter at all. If there is a free variable, then the truth depends on truth holding under the assignment. Sentences are just those expressions without free variables, and the quantifiers are used to bind them. It is in the case of quantified sentences that we need

atomic expressions (those in $\mathcal{R}$) and nonatomic expressions, given semantic interpretation.

[3]We hope that it does not cause too much confusion in this paper that there are two senses of *extension* in use: the first is exactly the one just mentioned – the extension of a predicate is the set of tuples it's true of; the other is the nominalization of the verb whose meaning we're trying to capture here (sense extension). Our solution to sense extension is a dynamic semantics which adds tuples to the extensions of the concerned predicates.

to consider other possible ways of assigning elements of the domain to the variables. It is nonetheless sufficient to consider only those assignment functions that are just like g, with the exception of the assignment to the quantified variable (as notated). In the quantified cases, for the expression to be true it just has to be that all (some) possible ways of assigning elements in the domain to the quantified variable make the statement true. The revised assignment functions must all remain constant for those variables that remain free in the expression. Thus, the truth of a sentence is not relative to any particular assignment function as it has no leftover unbound variables to be interpreted.

2.2 Dynamic Variable Assignment

This classical picture of the semantics of first order logic has been extended to provide dynamic variable assignment as a way of formalizing the inaccessibility of noun phrases in certain embedded contexts (like the consequent of a conditional, or inside the scope of negation) as antecedents for subsequent anaphors in natural language discourse (Kamp & Reyle, 1993). In such a framework, one would worry about the variable assignments in a more complicated way. The interpretation of a sentence has a dynamic effect on the availability of assignments to variables, so expressions are interpreted relative to input and output states (states consisting of sets of assignment functions, for instance). To illustrate, one might define: ${}^{gi}[\![\neg P^n]\!]^{go} = D$ iff $gi = go$ and no gm exists with ${}^{gm}[\![P^n]\!]^{go} = D$ (Groenendijk & Stokhof, 1991). The effect, when given a similarly structured interpretation to conjunction and the other connectives, is that there will be no way to link the assignment function that binds a free variable introduced in the scope of negation to the binding of a variable in a conjoined statement. If pronouns are modeled in logic as free variables, this has the correct effect of ruling out an anaphoric link between the indefinite NP embedded under the negation and a subsequent pronoun in a discourse such as (1) unless the second sentence is in an elaboration discourse relation to the first (thus subordinating it to the first sentence's VP context) as is more clearly illustrated by the contrasting discourses in (2-4).

(1) Sandy doesn't have a bike. She washes it.

(2) Sandy doesn't have a bike. She sold it.

(3) #Sandy doesn't have a bike and she sold it.

(4) Sandy doesn't have a bike because she sold it.

Modifications to the interpretation of a first order language like that described above have been described as dynamic semantics as they give attention to the state of information before and after the interpretation of a sentence. While this does seem to provide the right set of tools for analyzing aspects of meaning associated with anaphoric reference, it does not directly provide the means to interpret sense extension as happens in metaphor generation. We emphasize that we are not focused in this paper on formalizing the recognition of an existing metaphor, but on that which enables a new metaphor to come into being. It is a puzzle for traditional approaches to semantics because when we use idealized languages like the first order logic described above (under either interpretation), we still have the fixed L and D. If sense extension were about adding new symbols to the language, then we could offer a trivial formalization which says that if L

models the original language, then what we have is L' such that $L \subset L'$ (in particular L' would come equipped with a larger set of constants and predicates), $D \subset D'$, $I \sqsubset I'$ (the latter constraint entailing that the extended interpretation agrees with the old interpretation on all of the atomic expressions expressible in L). Additional constraints are required to take care of particular *theories* stated in L and L'. That is, adding terms to the language is a different matter from adding terms to a language and incorporating them into a set of sentences stated in the original language. For instance, one would presumably want to be aware of the fact that if given a set of sentences in a language, and a larger set of sentences in an extended language, the larger set of sentences could be inconsistent with the smaller set (hence with itself) even if the only new sentences are those containing the new expressions in the language. As an example, consider (a very basic!) language L whose only proposition letter is P, and a set of sentences Γ composed from that language: $\{\neg P\}$. Now, it is possible to extend L to L' with just an additional propositional letter Q. Additionally, form a larger set Γ' of sentences in the expanded language: $\{\neg P, Q, Q \Rightarrow P\}$. This larger language is clearly inconsistent, but both Γ and $\Gamma' - \Gamma$ are consistent. However, consistency maintenance is its own very large literature (see for a start: Alchourrón, Gärdenfors, & Makinson, 1985), and is not the direct topic of the current paper.

2.3 Dynamic Interpretation

We wish instead to keep both L and D fixed. This satisfies the intuition about sense extension that it involves an existing expression in the language, just a novel sort of usage. Keeping D fixed maintains a conservative sort of realism in which we presume we're modeling linguistic interactions with the world that is determined already, one which is not determined by interactions in the world. We are also interested in developing a the nonclassical approach to sense extension that is slightly less conservative on precisely this point. Our task now is to provide a classical sort of model in which it is possible to make novel uses of expressions already in the language. Our approach is inspired by the dynamic logic treatment of variable assignments sketched above (Groenendijk & Stokhof, 1991). However, instead of complicating the assignment functions, we complicate I, the interpretation function.

We begin with a different characterization of I than the one we initially presented in §2.1. Instead of statements like $I(c) \in D$ *for each constant in the language* L, we give the function as tuples that comprises I, maintaining a functional relation: $\forall c \in C, \exists d \in D : \langle c, d\rangle \in I$.

(5) $g(x) \in D$ for each variable in L.

(6) $\forall c \in C, \exists d \in D : \langle c, d\rangle \in I$.

(7) $\forall P^n \in \mathcal{R}, n > 0, \exists \delta \in \mathcal{P}(D^n), [|\delta| > 0 \Leftrightarrow \forall \tau \in \delta : \langle P^n, D, \tau\rangle \in I] \wedge [|\delta| = 0 \Leftrightarrow \langle P^n, D, \delta\rangle \in I]$.

(8) $\forall P^n \in \mathcal{R}, n = 0, \langle P, D\rangle \in I$ iff P is true.

(9) $[\![c]\!] = I(c), \forall c \in L$

(10) $[\![x]\!]^g = g(x), \forall x \in L$

(11) $[\![P^0]\!] = I(P)$

(12) $[\![P^n(t_1, ...t_n)]\!]^g = D$ if $\langle [\![t_1]\!]^g, ...[\![t_n]\!]^g \rangle \in I(P^n, D), \emptyset$ otherwise.

(13) ${}^I[\![\neg P^n]\!]^{Ig} = D$ if ${}^I[\![P^n]\!]^{Ig} = \emptyset$, and $\emptyset$ otherwise.

(14) ${}^I[\![P^n \vee Q^m]\!]^{Ig} = D$ if ${}^I[\![P^n]\!]^{Ig} = D$ or if ${}^I[\![Q^m]\!]^{Ig} = D$, and $\emptyset$ otherwise.

(15) ${}^I[\![P^n \wedge Q^m]\!]^{Ig} = D$ if ${}^I[\![P^n]\!]^{Ig} = D$ and ${}^I[\![Q^m]\!]^{Ig} = D$, and $\emptyset$ otherwise.

(16) ${}^I[\![P^n \Rightarrow Q^m]\!]^{Ig} = D$ if ${}^I[\![P^n]\!]^{Ig} = \emptyset$ or if ${}^I[\![Q^m]\!]^{Ig} = D$, and $\emptyset$ otherwise.

(17) ${}^I[\![\forall x P^n]\!]^{Ig} = D$ if for all $d \in D$ ${}^I[\![P^n]\!]^{Ig[x/d]} = D$, and $\emptyset$ otherwise.

(18) ${}^I[\![\exists x P^n]\!]^{Ig} = D$ if for some $d \in D$ ${}^I[\![P^n]\!]^{Ig[x/d]} = D$, and $\emptyset$ otherwise.

The presentation in (5–18) is actually equivalent to that given in §2.1. What is different is that the interpretation function is threaded through each clause where the interpretation function could potentially be extended by a sense extension of the object interpreted. This means we assume that sense extension does not apply to constants, variables or propositions (10,9,11). Under this formulation, nothing in the language so far is dynamic; however, it does set the stage for what follows

We now extend the language to include an English-like predication construct. In fact, we'll use two forms of the copula *is*, $\mathbf{is}_{\text{LIT}+}$ and $\mathbf{is}_{\text{LIT}-}$. Essentially, this yields two ways of predicating instead of the one given above. For $\sigma \in (C \cup V)^n, n > 0$ and $P^n \in \mathcal{R}, n > 0$, we can now form additional sentences σ $\mathbf{is}_{\text{LIT}+}$ P^n and σ $\mathbf{is}_{\text{LIT}-}$ P^n. The corresponding interpretations are as follows:

(19) ${}^I[\![\sigma \ \mathbf{is}_{\text{LIT}+} \ P^n]\!]^{Ig} = {}^I [\![P^n(\sigma)]\!]^{Ig}$

(20) ${}^I[\![\sigma \ \mathbf{is}_{\text{LIT}-} \ P^n]\!]^{I \cup \{\langle P^n, D, [\![\sigma]\!]^g \rangle\} g} = D$ iff ${}^I[\![P^n(\sigma)]\!]^I = \emptyset$

There are important implications of this definition. First, we presume for the present that the predications in this initial language involve only atomic predications ($P^n \in \mathcal{R}$). Nothing interesting happens for $\mathbf{is}_{\text{LIT}+}$. The effect of the definition for $\mathbf{is}_{\text{LIT}-}$ is to add the subject[4] to the extension of the predicate, as (by hypothesis) it is not there in the initial interpretation. Sense extension is modeled by increasing the extension of the predicate involved. We present this simple formulation to illustrate the essence of our solution to sense extension. Instead of making truth relative to a domain and interpretation function, we allow for the interpretation of a sentence to extend the interpretation function at stake. It is a dynamic semantics in that it uses the interpretation function as the input and output states of processing the sentence. Literal sentences do not extend the interpretation function at all. The use of a new metaphor, on the other hand, has the effect of extending the extension of the metaphorical predicate to include the entity (tuple) under predication. Note that the extension of the literally intended predicate is untouched. In the following section we give a simple example of how this works, pointing out the extreme limitations on expressivity, before providing a more comprehensive treatment in the framework.

[4] Actually, this is the tuple comprising all of the arguments—we do not assume that only unary relations may be involved in sense extensions of the sort modeled.

2.4 An Example

Let $I = \{\langle \text{stapler}, D, a\rangle, \langle \text{stapler}, D, b\rangle, \langle \text{spiral}, D, a\rangle, \langle \text{spiral}, D, b\rangle, \langle \text{hat}, D, b\rangle\}$, and let $C = \{a, b, c, d\}, \mathcal{R} = \{\text{stapler}, \text{spiral}, \text{hat}\}$. Then, ${}^{I}[\![d \ \mathbf{is}_{\text{LIT}} \ \_ \ \text{stapler}]\!]^{I \cup \{\langle \text{stapler}, D, d\rangle\}g} = D$. Also, ${}^{I}[\![a \ \mathbf{is}_{\text{LIT}} \ \_ \ \text{hat}]\!]^{I \cup \{\langle \text{hat}, D, a\rangle\}g} = D$. Note that extending the interpretation function has an strong effect on the set of truths in the system. In I, $\forall x$ stapler(x) $\Leftrightarrow$ spiral(x), but in $I \cup \{\langle \text{stapler}, D, d\rangle\}$ only $\forall x$ spiral$(x) \Rightarrow$ stapler(x) holds. In $I \cup \{\langle \text{hat}, D, a\rangle\}$ $\forall x$ hat$(x) \Leftrightarrow$ stapler(x), although this did not hold in I.

Also note that apart from the predication involved being required to be basic, there is a real expressive limitation in that nonliteral predications cannot be used in complex predicates. This follows from the equivalence of the semantics with the threaded interpretations functions to the semantics presented in §2.1: in each of the threaded clauses the input is required to be identical to the output interpretation. This will prohibit a metaphorical usage from being part of any complex predication, which is clearly limiting.

2.5 Extending the model

The first step is to correct (20) to maintain monotonicity in entailments for literal predicates in interpretation functions involved in sense extensions. We introduce a new symbol, ν, into the language which corresponds to a non-literal extension of a predicate.[5]

(21) ${}^{I}[\![\sigma \ \mathbf{is}_{\text{LIT}} \ \_ \ P^n]\!]^{I \cup \{\langle P^n, [\![\nu]\!], D, [\![\sigma]\!]^g\rangle\}g} = D$ iff ${}^{I}[\![P^n(\sigma)]\!]^{I} = \emptyset$

Of course, this does not preserve monotonicity over all predicates, as clearly when more information is added to the interpretation function, the entailments involving extended predicates will fluctuate. However, we do not at yet have a mechanism for accumulating additions to the interpretation function as at the present we do not have a way to embed a metaphorical sentence in a more complex construction (like a coordination) in which the interpretation of a later conjunct is affected by the augmented interpretation function. Below we give the modified clauses just for those cases in which modification is required.

(22) ${}^{I}[\![P^n(t_1, ..., t_n)]\!]^{Ig} = D$ if

a. $\langle [\![t_1]\!]^g, ..., [\![t_n]\!]^g\rangle \in I(P^n, D)$,

b. $\langle [\![t_1]\!]^g, ..., [\![t_n]\!]^g\rangle \in I(P^n, [\![\nu]\!], D), \emptyset$ otherwise.

(23) ${}^{I}[\![P^n \vee Q^m]\!]^{Og} = D$ if $I \subseteq O$, and

a. ${}^{I}[\![P^n]\!]^{Og} = D$ or if

b. ${}^{I}[\![Q^m]\!]^{Og} = D$, and $\emptyset$ otherwise.

(24) ${}^{I}[\![P^n \wedge Q^m]\!]^{Og} = D$ if $\exists M, I \subseteq M \subseteq O, {}^{I}[\![P^n]\!]^{Mg} = D$ and ${}^{M}[\![Q^m]\!]^{Og} = D$, and $\emptyset$ otherwise.

(25) ${}^{I}[\![P^n \Rightarrow Q^m]\!]^{Og} = D$ if

a. $\exists M, I \subseteq M \subseteq O, [{}^{I}[\![P^n]\!]^{Mg} = D] \wedge [{}^{M}[\![Q^m]\!]^{Og} = D]$, or if

b. $I = O \wedge [{}^{I}[\![P^n]\!]^{Ig} = \emptyset] \wedge [{}^{I}[\![Q^m]\!]^{Ig} = D]$, and $\emptyset$ otherwise.

[5] We will assume that $[\![\nu]\!] = D$, and for a predicate P^n, we obtain a new predicate P^{n+1}, however the effect of ν is just to increase the arity of the predicate P^n, thus preserving the extension of the original.

(26) ${}^{I}[\![\forall x P^n]\!]^{Og} = D$ if

a. $I = O$, and for all $d \in D$, ${}^{I}[\![P^n]\!]^{Ig[x/d]} = D$, or
b. $\exists O : I \subseteq O$, and for all $d \in D$, ${}^{I}[\![P^n]\!]^{Og[x/d]} = D$, and $\emptyset$ otherwise.

(27) ${}^{I}[\![\exists x P^n]\!]^{Og} = D$ if

a. $I = O$, and there is some $d \in D$, ${}^{I}[\![P^n]\!]^{Ig[x/d]} = D$, or
b. $\exists O : I \subseteq O$, and there is some $d \in D$, ${}^{I}[\![P^n]\!]^{Og[x/d]} = D$, and $\emptyset$ otherwise.

While this looks fairly complicated, the idea is actually very simple; the complication is just a propagation of cases that the approach creates. First, basic predication (22) is true just if it is true under static interpretation in the literal extension of the predicate (a), or in the static interpretation of the extended sense of the predicate (b). Consider disjunction (23). We thread the interpretation function through the interpretation of a disjoined formula. The effect of interpreting the formula can be to extend the interpretation function if it turns out that a metaphorical predication has been used with an atomic predicate inside one of the disjuncts. The metaphorical predication is the only one which stipulates the way in which the interpretation function can be extended. We assume, in fact, that this is the only way for the function to be extended. So, while we take an unspecified O, such that $I \subseteq O$ is the extended interpretation function after interpreting the disjunction, it's not that any O that contains I will do, only those that arise by construction from an embedded sense extension. The case of conjunction (24) is slightly more interesting as it take the output interpretation function (M) from the first conjunct, and makes that the input interpretation function for the second. The interpretation clause for implication (25) is given in two cases: (a) threads an intermediate interpretation function from the antecedent into the consequent of the conditional, if the antecedent is true, and yields the output interpretation function of the consequent as the output of the whole; (b) is the case of vacuous truth for the implication, and here we have stipulated that sense extensions will not propagate from vacuously true implications (both antecedent and consequent are static). The quantifier cases (26 and 27) are the most interesting. Here we have structured the system so that a quantified expression is true if it is true under quantification under the literal or extended (static) interpretation of the predicate (a) or if interpreting the quantified formula itself creates a sense extension (b). The first two cases keep the interpretation function static and look to one or other of the forms for the predicate. The final case accommodates the possibility that a nonliteral predication could be used within the quantified formula.

2.6 Discussion

The extensions of predicates in the initial interpretation function are untouched throughout the dynamics of sentence interpretation. Entailments that hold in the initial interpretation are not affected by nonliteral extension (cf. §2.4). However, this is not the case for non-literal extensions (necessarily). This accords with the intuition that a closed system (in terms of elements of the domain and basic expressions in the language) which still admits sense extension has triviality as its result in the limit: for each predicate in the language it is possible to assert its nonliteral extension using a universal quantifier, making each predicate true of all elements in the domain. Nothing prevents this.[6] The

[6] Just as nothing prevents one from uttering $a \wedge \neg a$.

intuition is that if everything is meant nonliterally, then nothing nontrivial can actually be meant at all.

The revised model of §2.5 is interesting, in part, because it allows nonliteral predications "σ $\mathbf{is}_{\text{LIT}\,-}$ P^n" to occur in nonatomic expressions. It maintains the restriction from the original system that the predication P^n itself be atomic ($P^n \in \mathcal{R}$). This predicts (for instance) that it is not possible to *generate* a metaphor of the form:

(28) k $\mathbf{is}_{\text{LIT}\,-}$ *a stapler and a wedge.*

It is possible to state:

(29) k $\mathbf{is}_{\text{LIT}\,-}$ *a stapler.* k $\mathbf{is}_{\text{LIT}\,-}$ *a wedge.* k $\mathbf{is}_{\text{LIT}\,+}$ *a stapler and a wedge.*

In the latter case (29) the complex predication is used *literally* but with respect to a previously extended sense of the predicate, whose extended interpretation is available as input to the interpretation of the final expression in the complex formula. This accords with intuitions about the distinction between expressive limits at work during sense extension as opposed to those at work when a previously extended expression is used (i.e. metaphor generation vs. recognition).

This does rely on our presumption that there is indeed legitimate reason for considering $\mathbf{is}_{\text{LIT}\,-}$ and $\mathbf{is}_{\text{LIT}\,+}$ as distinct forms of the copula. We feel this to be the case, on evidence that (pretheoretically at any rate) irony and other cues of nonliteral intended meaning are perceptible. Note that our model does not preclude nonliteral meanings from being interpreted somewhat literally. That is what the last example demonstrated. This follows because the meaning of a predication using the sentence $\mathbf{is}_{\text{LIT}\,+}$ can make use of a literal or extended denotation, because of (19) and (22). However, the interpretation of sentences using $\mathbf{is}_{\text{LIT}\,-}$ succeeds unless the predication was literally true to start with, and extends the predicate. Thus, we make use of signals of 'irony' as essential to sense extension, but as inessential to second nonliteral use (assuming that the extended interpretation function is available by the compositional threading that we outlined). The 'ambiguity' of $\mathbf{is}_{\text{LIT}\,+}$ implies that we don't in the current formulation have a mechanism for constraining interpretation to either the literal or to the extended senses. However, such a parameterization could be accommodated. For simplicity in the current presentation we leave things as they are $P(x)$ is true if it is literally true or if it is true according to an accessible extended sense.

While the system does render certain extensions inaccessible, by virtue of the threading mechanism (for instance extensions made in the scope of a negation or in a vacuously true conditional). There is no mechanism for making inaccessible the denotation of the literal predicate that might have been used instead of the sense-extending nonliteral predication. Consider examples of the following form:

(30) x $\mathbf{is}_{\text{LIT}\,-}$ a stapler. They have property Ψ.

In the example, *they* can refer to the set of literal staplers or to the extended set of nonliteral staplers, but *they* cannot refer to the set of entities that would have sufficed using a literal predicate instead of a nonliteral one (maybe *x is a professional negotiator*).

(31) x $\mathbf{is}_{\text{LIT}\,-}$ a stapler. They attach things.

(32) x $\mathbf{is}_{\text{LIT}}$ _ a stapler. #They bring conflicting parties together.

Examples (31) and (32) demonstrate this more clearly. The model would require an additional sort of facility in order to render the intended literal meaning (rather than the unextended literal meaning, which would be strictly false) inaccessible to subsequent anaphoric reference. It is not clear to us at the present how to go about formulating such a modification to the system.[7]

3 Final Remarks

We have presented a dynamic first order semantics for nonliteral predication yielding sense extension. The method, which we believe to be novel, applies the technique from dynamic predicate logic of threading assignment functions through the semantic interpretation clauses to the interpretation function instead. What DPL is able to achieve for anaphoric reference with pronouns, we are able to achieve for the nonlogical constants in a language. This can serve as the foundation of a model theoretic semantics for metaphor generation since metaphor generation has as a necessary component the extension of a sense of an already existing expression in the language to a denotation that is already present in the world. We emphasize that we do not claim to have given a theory of metaphor recognition (see Veale & Keane, 1992). Nor do we claim to have given a theory of preconditions to metaphor (Indurkhya, 1987). It remains, in fact, to explore how our semantic analysis integrates with more heterogeneous formulations of processes at stake in systems that do address those issues. It would be useful, for instance, to explore the limits of expressive facility in the dynamic first order treatment of sense extension that we have articulated here with the formulation based on conditional/default logic presented by Copestake and Briscoe (1996). A reason to pursue this is that our approach keeps the interpretation to a less expressive class of logic. We would like to explore the complications induced by introducing the DPL-style threading of variable assignment functions in tandem with the interpretation function to yield a single system for both sense extension and anaphora. We also intend to apply the model to other areas of linguistic analysis that rely on a formalization of sense extension—such as the problem of open texture in legal reasoning.

References

Alchourrón, C. E., Gärdenfors, P., & Makinson, D. (1985). On the Logic of Theory Change: Partial Meet Contraction and Revision Functions. *The Journal of Symbolic Logic*, *50*, 510–30.

Church, A. (1936). An Unsolvable Problem of Elementary Number Theory. *American Journal of Mathematics*, *58*, 345–63.

Copestake, A. & Briscoe, T. (1996). Semi-productive Polysemy and Sense Extension. In Pustejovsky, J. & Boguraev, B. (Eds.), *Lexical Semantics: The Problem of Polysemy*, pp. 15–67. Cambridge Unviersity Press.

[7] While the reader will recall that we did add a symbol to the language ν, we made no use of the symbol in the syntax. Rather, we made use of its denotation, which was just the entire domain, as an arity-increaser on the nonliteral extension of a predicate. This means that we discriminate in the semantics between nonliteral and literal extensions, but we have not explored the possibility of exploiting this in the syntax.

Groenendijk, J. & Stokhof, M. (1991). Dynamic Predicate Logic. *Linguistics and Philosophy*, *14*, 39–100.

Hahn, U. & Vogel, C. (1995). Notes on Lawgicisim: Formalization, Computation, and Law. Presented at the *17th IVR '95 World Congress*, Bologna, Italy, June 16-20, 1995.

Indurkhya, B. (1987). Approximate Semantic Transference: A Computational Theory of Metaphors and Analogies. *Cognitive Science*, *11*(4), 445–80.

Kamp, H. & Reyle, U. (1993). *From Discourse to Logic.* Kluwer.

Post, E. L. (1936). Finite Combinatory Processes. Formulation I. *Journal of Symbolic Logic*, *1*, 103–5.

Turing, A. (1936). On Computable Numbers, with an Application to the *Entscheidungsproblem*. *Proceedings of the London Mathematical Society*, *2*(42-43), 230–65,544–6.

Veale, T. & Keane, M. (1992). Conceptual Scaffolding: A Spatially Founded Meaning Representation for Metaphor Comprehension. *Computational Intelligence*, *8*(3), 494–519.