AKIKO YOSHIMURA

EXPLICATURE AND IMPLICATURE FORMATION IN THE MODELING OF METAPHOR AND METONYMY*

In a recent study, Gabbay and Kempson (1992a) envisage an approach to natural language interpretation based on proof-theoretic techniques, the procedural character of which leads to significant insights into various syntactic problems. In this paper we propose to explore applications of the same approach to the interpretation of such rhetorical tropes as METAPHOR, e.g., (1) and (2), and METONYMY, e.g., (3) and (4).

(1) John is a donkey.

(2) You are the cream in my coffee.

(3) Mary is reading Shakespeare.

(4) The printers' union wants to buy that newspaper. (Green 1989:49)

It has often been said that the phenomenon which announces the presence of a metaphor is a selectional restriction violation. However, though both (5) and (6) violate a selectional restriction, it is next to impossible to interpret (6) as a metaphor, while (7) may have a metaphorical reading in spite of the lack of any selectional restriction violation.

(5) Juliet is the sun.

(6) Sally is a prime number between 17 and 23. (Searle 1979:92)

(7) I have climbed to the top of the greasy pole. (ibid.:97, orig. Disraeli)

How, then, can hearers recognize when an utterance is an instance of metaphor or of metonymy?

Since the time when people regarded semantics as treating truth-conditional meanings and pragmatics as dealing with the meanings of utterances minus their truth-conditional meaning, metaphor and metonymy have posed a big problem for pragmatics. The implication conveyed by an utterance has been characterized as what is drawn by the interaction of its propositional content with other propositions or as what should be supplemented if we assume that interlocutors observe Gricean (1975) conversational principles. However, a major premise underlying these observations was that the propositional content of the utterance was true. From this point of view, the *falsity* of the literal meanings of metaphors and metonymical usages causes a problem. After a proposition is abandoned as false in semantics, how can it revive and take part in inference in pragmatics? Traditional theories have difficulty explaining this problem.

How do we recognize metaphor? How does a *false* proposition participate in inference? Any plausible theory of utterance interpretation should give appropriate answers to these questions. In this paper, we would like to show that a formal logical framework of utterance interpretation called LABELED DEDUCTIVE SYSTEMS (LDSs), proposed by Gabbay and Kempson (1992a), makes it possible to answer these questions and to articulate clearly the similarities and differences between the processing procedures of metaphor and metonymy. We shall conclude that both depend on abduction resolved through consulting lexically linked databases, but that they are invoked in separate components. Before we enter into the discussion, we will situate our procedural approach in a theoretical perspective, and in a subsequent section we will confirm that utterance interpretation consists of two essentially parallel tasks.

*This is a revised version of a paper which I presented at the 41st Machikaneyama Kotoba no Kai, held at Osaka University on January 14, 1993. I would like to take this opportunity to express my thanks to Seisaku Kawakami and Yukio Oba for their help and encouragement while I was writing this paper. I also wish to thank Ruth M. Kempson, Keiko Tanaka, Michael T. Wescoat, and Deirdre Wilson for their valuable comments and suggestions on earlier versions of this work. Thanks are also due to Stephen W. Horn, Daniel Long, Everdyn A. Wescoat, and M. T. Wescoat for reviewing this text and providing many stylistic suggestions. Of course the responsibility for any remaining deficiencies is mine alone.

S. Kawakami & M. T. Wescoat (eds.), Osaka University Papers in English Linguistics, 1, 1993, 175–184.

1 Relevance Theory and the Procedure of Interpretation

Our proposals about utterance interpretation are procedural in nature, and it is therefore interesting to consider, as a preliminary step, a theory which delves into the relationship between processing and communication.

In RELEVANCE THEORY, proposed by Sperber and Wilson (1986), the speaker's intention in communication is to modify the hearer's COGNITIVE ENVIRONMENT. Such modifications are called CONTEXTUAL EFFECTS. RELEVANCE is a function of contextual effects and the processing effort which is needed to produce them. Other things being equal, the more contextual effects the information has, the more relevant it is; and the more processing effort it requires, the less relevant it becomes. Sperber and Wilson propose the principle of relevance, which states that the speaker guarantees the utterance to be relevant for the hearer to process; their principle is reproduced below in (8) along with some auxiliary definitions in (9).

(8) PRINCIPLE OF RELEVANCE

Every act of ostensive communication communicates the presumption of its own optimal relevance.

(9) PRESUMPTION OF OPTIMAL RELEVANCE

a. The set of assumptions {*I*} [*sic*] which the communicator intends to make manifest to the addressee is relevant enough to make it worth the addressee's while to process the ostensive stimulus.

b. The ostensive stimulus is the most relevant one the communicator could have used to communicate {*I*} [*sic*]. (Sperber and Wilson 1986:158)

Sperber and Wilson go on to argue that the hearer processes the utterance so as to extract the optimal relevance.

It is worth underscoring the basically procedural point of view implicit in relevance theory. Were one to consider the problem of communication from a strictly declarative perspective, one would come to the erroneous conclusion that all possible contextual effects of a given utterance ought to be equally accessible simultaneously. This is clearly not so. On the relevance-theoretic view, the hearer undertakes a procedure to build up contextual effects. This is a search which could be open-ended, but the hearer clearly chooses a point at which she halts when she feels that she has probably come to the point of diminishing returns in her efforts to find contextual effects. The procedural character of this search and the possibility of suspending it at any propitious-seeming point is crucial to explaining why an utterance which may bear a special metaphorical or metonymical interpretation on one occasion might not be perceived as doing so in another context. Metaphors and metonymical usages are inferences that require more processing effort than literal readings and might not be realized, if the hearer finds satisfactory relevance in simpler, more straightforward interpretations. Furthermore, the notion that the speaker guarantees the relevance of an utterance spurs the hearer on to search for relevance, even if simple, literal interpretations do not seem to yield any particular contextual effects. These rather abstract comments will be given more concrete content throughout the course of the subsequent discussion. With this principle of relevance in hand, we may proceed with our description of the utterance interpretation procedure.

2 Two Tasks of Utterance Interpretation and Their Parallelism

Utterance interpretation comprises two tasks: the formation of propositional forms as explicatures and the formation of implicatures. Consider the following examples:

(10) Peter: Would you like a cup of coffee?

Mary: [ɪtkipsmi:əweɨk] 'It keeps me awake.'

(11) a. It keeps me awake.

b. Coffee keeps Mary awake. EXPLICATURE

c. Mary doesn't want to have coffee. IMPLICATURE

The sound stimulus associated with Mary's utterance is assigned the logical form (11a) in the input system, and in the central system, it is developed into the propositional form (11b) through reference assignment, disambiguation, and enrichment, thus forming an explicature. An explicature is an explicitly communicated assumption which contains a logical form for the utterance. Now Peter, remembering that Mary will have to get up early the next morning for her job, adds the assumption that she wants to go to bed early to his cognitive environment. This, in turn, interacts with (11b) and gives us (11c), which is an implicature. An implicature is any proposition added to the cognitive environment in order for the utterance to attain relevance.

Here we would like to show that the formation processes of explicatures and implicatures are parallel and that they employ abduction, i.e., the inference rule whereby assumptions are adopted in order to save what would otherwise be an absurdity. The formation processes for both explicatures and implicatures use premises in the following three ways:

1. Reuse of Immediately Preceding Premises

   The examples below show that the immediately preceding premise may be employed for performing pronoun resolution in explicature formation (12) and for deriving logical deduction in implicature formation (13).

   (12) John hit Mary$_i$. She$_i$ cried.

   (13) a. *A* says: If John is coming to the party, Sue won't come.
   b. *B* says: John *is* coming to the party.
   c. *A* realizes: Sue won't come.

   In (12), the referent of *she* is recovered through association with *Mary*, which is included in the preceding utterance. In (13), on hearing (13b), *A* will combine it with (13a), draw an implicature, and realize (13c).

2. Use of Premises from Lexically Linked Databases

   In the explicature formation process for (14), the referent of *the bishop* is recovered from the database linked to *Norwich*.

   (14) a. Jane visited Norwich. The bishop was friendly.
   b. $\Delta_{Norwich} = \{\ldots,$ Norwich has a bishop, $\ldots\}$

   For example, the database connected to *Norwich*, which we designate as $\Delta_{Norwich}$, might contain the proposition 'Norwich has a bishop.' A similar phenomenon is observable in implicature formation. In (15), suppose that John and Sue parted recently, that she is afraid of seeing him, and that everyone knows that if John is coming to the party, Sue won't come.

   (15) a. John's coming to the party.
   b. $\Delta_{John} = \{\ldots,$ If John is coming to the party, Sue won't come, $\ldots\}$

   The utterance (15a) leads the hearer to combine (15a) with the database entry written out in the database in (15b), which is linked to *John*, and to draw the implicature 'Sue won't come.' Both (14) and (15) use premises linked to certain concepts. In fact, the process of drawing on premises from lexically linked databases underlies metaphor and metonymy, as we will see later.

3. Creation of Necessary Premises (One Form of Abduction)

   First, consider explicature formation. In (16), suppose that on entering her house, *A* says (16a).

   (16) a. *A* says: The wall tapestry is missing! Call the police.
   b. *B* abduces: The house used to have wall tapestry.

   Even if *B* is visiting *A*'s house for the first time, she would create (16b) and thereby recover the referent of *the wall tapestry*. This type of abduction is also relevant for implicature formation:

(17) a. *A* says: Would you like to have (eat) an apple?
b. *B* says: I don't eat South African apples.
c. *A* abduces: The apples are South African.

In (17), *A* would create (17c) and draw the implicature '*B* won't eat the apples.' Examples (16) and (17) both create necessary premises to form an explicature and an implicature, respectively.

In selecting additional premises, it would be reasonable to assume the rule in (18).

(18) LEAST-EFFORT ENRICHMENT RULE (LER)

In order for a linguistic input φ to attain relevance, any assumption ψ may be added, provided that if there is more than one way to add ψ, the one which needs the least effort to produce a certain contextual effect should be selected.

This is a natural consequence from the relevance-theoretic point of view.

Now we can see a significant parallelism between the processes of explicature and implicature formation, which have heretofore been regarded as two quite different phenomena. In both processes what is needed is given by *(a)* immediately preceding premises, *(b)* premises in some linked database, or *(c)* creation and expansion of databases. The difference between the two processes is that what is sought is a representation for an entity in the case of explicature formation, and an assumption in the case of implicature formation.

So far, we have been talking about the processes of explicature and implicature formation in fairly abstract terms. We shall now put forward some more concrete proposals about how this interpretive process should be formally modeled. The next section outlines the LDS framework of Gabbay and Kempson and surveys how it develops a string of words into a propositional form.

3 LABELED DEDUCTIVE SYSTEMS AND BRIDGING CROSS-REFERENCE

As mentioned above, Gabbay and Kempson use a formal logical framework called LABELED DEDUCTIVE SYSTEMS (LDSs). These systems associate labels with logical forms in such a way that the labels indicate how information is combined and manipulated through the reasoning procedure. Gabbay and Kempson claim that the two components of utterance interpretation each have their own LDS. Explicature formation ensures the translation of natural language inputs into well-formed formulae (wffs) of predicate logic. Implicature formation involves drawing contextual implications in the manner prescribed in relevance theory.

The LDS for the explicature formation component employs a logic called L_0, which has two atomic sentences, e and t, corresponding to atoms in logical type theory, and the connective $\rightarrow$. Associated with formulae of L_0 are labels in the form of expressions from a predicate logic called L_1. The syntax for such labeled formulae is $\alpha : \varphi$, where φ stands for the L_0 formula and α for the L_1 label. The translation of natural language inputs into wffs of L_1 takes place in the form of an L_0 proof.

One starts with a set of initial assumptions drawn from the lexical material that constitutes the natural language input. In the most basic cases the lexically derived assumptions take the form of an L_0 expression indicating the logical type of the relevant lexical item, to which is added a label consisting of the L_1 translation appropriate for that item. Thus, *Norwich* might give rise to the assumption **Norwich** : e, and *visit* might yield **visit** : $e \rightarrow [e \rightarrow t]$. In the latter formula $e \rightarrow [e \rightarrow t]$ is the L_0 rendering for the complex logical type $\langle e, \langle e, t\rangle\rangle$.

With the lexically derived assumptions in hand, one constructs a proof where the goal is to use all the lexical assumptions exactly once in order to derive a labeled formula $\alpha : t$, where t is the logical type of sentences, and the label α is a placeholder for some L_1 expression. Whatever expression fills the position marked by α will be the L_1 translation of the sentence. The flow of proofs in L_0 mimics type manipulations in typical categorial grammar frameworks. For the purposes of this paper we will be mostly concerned with the logical inference rule MODUS PONENDO PONENS (MPP), which in standard logic allows one to derive ψ from $\varphi \rightarrow \psi$ and φ. In the case of L_0, combining $e \rightarrow [e \rightarrow t]$ with e yields $e \rightarrow t$ by MPP; this derivational step echoes the combination of an expression of the logical type $\langle e, \langle e, t\rangle\rangle$ with something of type e to yield an

expression of type $\langle e, t\rangle$. However, logical inference rules in an LDS must take into account the labels associated with formulae. In the LDS employed for translating natural language inputs into L_1 logical forms, the rule of MPP will say that combining the labeled formulae $\alpha : \varphi \rightarrow \psi$ and $\beta : \varphi$ yields $\alpha(\beta) : \psi$. The labels from the two premise formulae are algebraically combined into a new label $\alpha(\beta)$. The new label displays the usual notation for function application, which is the basis of the syntax of L_1. Consequently, combining **visit** : $e \rightarrow [e \rightarrow t]$ and **Norwich** : e would yield **visit**(**Norwich**) : $e \rightarrow t$, and the label **visit**(**Norwich**) would be the proper L_1 translation for the sentence fragment *visit Norwich.* Note further that if an assumption, say **Jane** : e, were made available by the lexical items composing the input sentence, then that assumption could be combined with **visit**(**Norwich**) : $e \rightarrow t$ by MPP in order to get **visit**(**Norwich**)(**Jane**) : t. Thus the goal of the proof, a labeled formula $\alpha : t$, would have been reached, with the L_1 translation for the sentence *Jane visited Norwich* as a result.

The LDS framework also imposes a box discipline on logical proofs. Nested boxes are used to implement locality constraints on the sharing of information in the course of a proof. In the linguistic application of LDS theory, individual boxes generally correspond to clauses, and the nesting of the boxes mimics syntactico-semantic embedding. This treatment enables one to implement various clause-based locality effects that have become well-known in the linguistic literature. This paper will not actually take advantage of this capability, however. In the LDS framework, inference is regarded as a goal-directed process, and each box is assigned a goal. When the goal is attained, the process exits the box. Boxes are also given labels, which are written with the same infixed-colon notation as was employed for labeled formulae. Box labels can be repositories of important information, but they will not play any crucial role in this paper.

Particularly important for our proposal is the treatment of anaphoric expressions, e.g., pronouns and definite descriptions in the LDS framework. The highly context-dependent referents of such forms are determined through Skolemization. This associates variables with values typically determined by lexical look-up techniques. For instance, in (14) above, the referent of *the bishop* was determined by referring to a database associated with *Norwich.* The LDS framework handles (14) as follows:

(19) Jane visited Norwich. The bishop was friendly. [=(14)]

s_0 :

goal: $\{\mathbf{Jane} : e, \ldots, \mathbf{friendly} : e \rightarrow t\} \vdash \alpha : t + t$	
s_1 : goal: $\{\mathbf{Jane} : e, \ldots, \mathbf{Norwich} : e\} \vdash \beta : t$	
1. **Jane** : e	assume
2. **visit** : $e \rightarrow [e \rightarrow t]$	assume
3. **Norwich** : e	assume
4. **visit**(**Norwich**) : $e \rightarrow t$	MPP; 2, 3
5. **visit**(**Norwich**)(**Jane**) : t	MPP; 4, 1
s_2 : goal: $\{u : e, \ldots, \mathbf{friendly} : e \rightarrow t\} \vdash \gamma : t$	
6. $\langle u, \{\mathbf{bishop}(\Theta(u)),\ \Theta(u) \in s_i,\ s_i < s_2\}\rangle : e$	assume
add $c : e$ and **bishop**$(c) : t$ to s_1	$\Delta_{Norwich}$
choose $u = c$ for c in s_1	
7. **friendly** : $e \rightarrow t$	assume
8. **friendly**$(c) : t$	MPP; 7, 6

Box s_0 has as its goal the combined translation of the two sentences in the input. Each sentence is handled in its own nested box. Box s_1 displays the inference already described in the foregoing discussion. Notice that box s_1 attains a wff of type t at line 5, which means that the components of the first utterance have been shown to constitute a sentence. Furthermore, the label in line 5, **visit**(**Norwich**)(**Jane**), shows the translation for that utterance.

A little reflection will reveal that the order in which assumptions are employed in the flow of the proof is crucial to getting the correct translation. In other words, nothing said so far would prevent one from reaching the conclusion **visit**(**Jane**)(**Norwich**) : t with a label meaning something like 'Norwich visited Jane.' The LDS framework does provide a means for dealing with this problem,

which has been put to use in other LDS-based linguistic studies. The labels employed in LDSs are intended to control the flow of inference and may contain any sort of relevant information. Thus instructions like 'use me first,' 'use me last,' etc., could be encoded in the labels during the lexical scanning process that collects the initial assumptions from which the proof proceeds. Having noted that a suitable solution to the order-of-use problem exists, we shall simply ignore the details of its implementation for the sake of simplicity.

Box s_2 displays some as yet undiscussed features. Line 6 and the accompanying instructions handle the definite description *the bishop*. This form is regarded as an anaphoric element, assigned a Skolem variable, and resolved later. The symbol Θ represents a meta-function which instantiates the Skolem variable. Line 6 features a complex label that contains not only an L_1 expression, the variable u, but also a set of conditions on the instantiation of that variable. The constraints state that the referent of *the bishop* must satisfy the predicate **bishop** and that it was introduced in a box preceding s_2. In other words, the definite description is understood as an instruction to 'look for an appropriate value with this condition.' However, if no suitable value can be found, a necessary assumption is added as a datum linked to the most accessible label, **Norwich**. This process is expressed in the first instruction under line 6. In the next instruction the referent of *the bishop* is identified as the bishop of Norwich and is resolved. Finally, the goal of box s_2 is attained at line 8, and this leads to the attainment of the goal of box s_0, $t + t$.

This processing procedure for handling bridging cross-reference, as in the case of *the bishop*, is fundamentally similar to that for dealing with metaphor/metonymy. The crucial difference is that the reference assignment of a definite description is lexically imposed and is to be resolved at the L_0 level in the first attempt at explicature formation. Otherwise, the process cannot exit the box. On the other hand, in the case of metaphor/metonymy the reference assignment is not lexically imposed, and the process exits the box with a semantic anomaly in the resulting logical form. The next section applies the LDS approach to metaphor and shows that its processing characteristics are properly captured.

4 Metaphor

The treatment of metaphor takes place in the implicature formation component. This component is modeled by an LDS based on the logical language L_1, which was the language into which the previous LDS, formed around L_0, translated input utterances. This component is used to make inferences employing the logical translations of the natural language inputs provided by the explicature formation component. The implicature formation component works only after a wff of type t has been attained in the explicature formation component. The labels employed with formulae at this level will simply record the derivational history of a given stage of the proof and will be of no particular significance to the analysis. L_1 is a predicate logic with a richer proof apparatus than that of L_0. However, the inference techniques employed here will be familiar enough from standard logic to pass without special comment.

In the interpretation of metaphor, an initial failure in the implicature component is involved, and the manner in which this comes about can best be seen through an example. Consider again (1), *John is a donkey*. The explicature component would translate this input into **donkey(John)**. However, this logical form is clearly untenable in the implicature formation component if John is a human being. The principle of relevance and the rule of abduction lead us to search for some other usable implicature. This will involve a proof-theoretic procedure which first takes the logical translation of (1) as an assumption, and then seeks implicata by combining that with assumptions recovered from databases linked to lexical items in the input.

Here, Δ is a database consisting of the set of assumptions shared by the speaker and hearer, and $\Delta?Q$ stands for 'Is $\Delta \vdash Q$ (Δ logically entails Q) true?' If the hearer answers the question 'Is John nice?' in (20a) in the manner set out in (20b), the processing procedure will be something like the following:

(20) a. question: Δ?**nice(John)** 'Is John nice?'

b. answer: John is a donkey. [=(1)]

s_0 :

goal: $\{\mathbf{John} : e, \mathbf{donkey} : e \rightarrow t\} \vdash \alpha : t$	
1. $\mathbf{John} : e$	assume
2. $\mathbf{donkey} : e \rightarrow t$	assume
3. $\mathbf{donkey(John)} : t$	MPP; 2, 1

goal: one of $\Delta \cup \{a_0 : \mathbf{donkey(John)}\} \vdash \left\{ \begin{array}{l} \alpha : \mathbf{nice(John)} \\ \beta : \neg\mathbf{nice(John)} \end{array} \right\}$

M_1 :

goal: $\Delta \cup \{a_0 : \mathbf{donkey(John)}\} \vdash \alpha : \mathbf{nice(John)}$

$\vdots$

M_0 :

M_2 :

goal: $\Delta \cup \{a_0 : \mathbf{donkey(John)}\} \vdash \beta : \neg\mathbf{nice(John)}$	
4. $a_0 : \mathbf{donkey(John)}$	assume
5. $a_1 : \forall x[\mathbf{donkey}(x) \rightarrow \mathbf{obstinate}(x)]$	Δ_{donkey}
6. $a_2 : \mathbf{donkey(John)} \rightarrow \mathbf{obstinate(John)}$	UI; 5
7. $a_2a_0 : \mathbf{obstinate(John)}$	MPP; 6, 4
8. $a_3 : \forall x[\mathbf{obstinate}(x) \rightarrow \neg\mathbf{nice}(x)]$	$\Delta_{obstinate}$
9. $a_4 : \mathbf{obstinate(John)} \rightarrow \neg\mathbf{nice(John)}$	UI; 8
10. $a_4a_2a_0 : \neg\mathbf{nice(John)}$	MPP; 9, 7
11. $a_5 : \forall x[\mathbf{donkey}(x) \rightarrow \mathbf{slow}(x)]$	Δ_{donkey}
12. $a_6 : \mathbf{donkey(John)} \rightarrow \mathbf{slow(John)}$	UI; 11
13. $a_6a_0 : \mathbf{slow(John)}$	MPP; 12, 4
$\vdots$	$\vdots$

The process attains a wff of type t at line 3 and exits box s_0. Since the LDS approach adopts a proof theoretic process and its goal in the LDS comprising L_0 is simply to derive the conclusion t, regardless of the truth or falsity of the L_1 expression built up in the labels, the 'falsity' of the literal meaning of metaphors causes no problems at this level.

Upon exiting box s_0, one has an L_1 translation for (20b). Now one turns to the matter of determining how the response bears on the question. To do this, two goals are considered: one is to prove **nice(John)**, and the other goal is to demonstrate **¬nice(John)**. The goal of box M_2 is the latter. Since John is not a donkey, the explicature of the reply contradicts the content of one's cognitive environment and leads to an absurdity. Consequently, the explicature is not entered into the cognitive environment. Then the principle of relevance and the rule of abduction lead us to search for some implicature, using **donkey(John)**.

Since one of the most important rules of inference for drawing implicatures is modus ponens, one obvious technique to try would be to employ **donkey(John)** as the minor premise for that rule. It follows that the major premise takes the form of **donkey(John)** $\rightarrow \varphi$. However, since nothing of the form **donkey(John)** $\rightarrow \varphi$ is present in our cognitive environment, the major premise must be derived from something more abstract like $\forall x[\mathbf{donkey}(x) \rightarrow \varphi]$ by UNIVERSAL INSTANTIATION (UI). This premise is recovered from the database linked to the most accessible concept, *donkey*. This procedure takes place in lines 5 and 6. Then MPP derives line 7, **obstinate(John)**, which is later combined with **obstinate(John)** $\rightarrow$ **¬nice(John)** to draw **¬nice(John)**. The goal of M_2 is thus attained.

In the process, other assumptions which attain sufficient relevance to offset the needed processing effort are drawn from the database Δ_{donkey}—**slow(John)** or **stupid(John)**, for example. These implicatures are entered into the cognitive environment along with **obstinate(John)**. Therefore, since *John is a donkey* modifies the hearer's cognitive environment significantly, in spite of the fact that much more effort is needed than for a simple answer *no*, the input sentence succeeds in attaining relevance. This is typical of metaphor.

The process in lines 8–10, after the drawing of line 7, **obstinate(John)**, is common to all forms of indirect speech. Consequently, if we search for some defining characteristic of metaphor in this procedure, which is not shared by any other indirect speech form, it would probably be the fact

that the explicature of the metaphorical utterance is untenable and therefore is not entered into the cognitive environment. Only the implicatures of metaphor are accepted as new information to retain. Thus, the communicative goal of metaphor is attained solely through implicatures.

What makes this process possible is the proof theoretic approach which the LDS framework adopts. Since inferences proceed, not on the meaning of the premise, but on its form, a propositional form can be used as a premise to initiate a search for relevant implicatures, even if it is literally false.

This process similarly applies to example (7), which was pointed out above, because it displayed no selectional restriction violation and was therefore a challenge to identify as a metaphor. Example (7), *I have climbed to the top of the greasy pole* as an answer to a question such as *Have you performed any challenging physical stunts lately?* is not regarded as a metaphor. The utterance is literally interpreted, its explicature is entered into the cognitive environment, and the question is taken to have been answered in the affirmative.

On the other hand, example (7) as an answer to *Has your political career been smooth so far?* is regarded as a metaphor. In this case also, a wff of type t is attained without difficulty in the L_0-based LDS that handles initial translation, but the explicature doesn't form an answer to the question. Then implicature formation begins, based on the form of the explicature, as in the case of (20b). The questioner draws the answer *no* and finds implicatures such as 'He has had great difficulty so far' or 'He may have nearly fallen many times' and so on. The hearer finds relevance by entering these assumptions into her cognitive environment. Unlike the previous, non-metaphoric case, the explicature of the utterance is not itself inducted into the cognitive environment, because it functions only as a premise to draw relevant implicatures, and the degree of the hearer's confidence in its literal content is next to nothing.

5 Metonymy

Consider (3) again, repeated in (21b). Suppose concretely that the questioner asks *Is Mary reading* Gone with the Wind*?* and that the hearer answers *Mary is reading Shakespeare.* The processing procedure of the answer will be as follows (let $\mathcal{S}$ abbreviate **Shakespeare**, and $\mathcal{G}$, **Gone_with_the_Wind**):

(21) a. question: $\Delta ?\mathbf{read}(\mathcal{G})(\mathbf{Mary})$ 'Is Mary reading *Gone with the Wind*?'

b. answer: Mary is reading Shakespeare. [=(3)]

s_0 :

goal: $\{\mathbf{Mary} : e, \ldots, \mathcal{S} : e\} \vdash \alpha : t$	
1. $\mathbf{Mary} : e$	assume
2. $\mathbf{read} : e \rightarrow [e \rightarrow t]$	assume
3. $\mathcal{S} : e$	assume
4. $\mathbf{read}(\mathcal{S}) : e \rightarrow t$	MPP; 2, 3
5. $\mathbf{read}(\mathcal{S})(\mathbf{Mary}) : t$	MPP; 4, 1

M_0 :

goal: one of $\Delta \cup \{a_0 : \mathbf{read}(\mathcal{S})(\mathbf{Mary})\} \vdash \left\{ \begin{array}{l} \alpha : \mathbf{read}(\mathcal{G})(\mathbf{Mary}) \\ \beta : \neg\mathbf{read}(\mathcal{G})(\mathbf{Mary}) \end{array} \right\}$

M_1 :

goal: $\Delta \cup \{a_0 : \mathbf{read}(\mathcal{S})(\mathbf{Mary})\} \vdash \alpha : \mathbf{read}(\mathcal{G})(\mathbf{Mary})$

$\vdots$

M_2 :

goal: $\Delta \cup \{a_0 : \mathbf{read}(\mathcal{S})(\mathbf{Mary})\} \vdash \beta : \neg\mathbf{read}(\mathcal{G})(\mathbf{Mary})$

6. $a_0 : \mathbf{read}(\mathcal{S})(\mathbf{Mary})$ assume

$\vdots$

The explicatum is absurd because things which are read must be written texts. Also, no metaphorical reading can be determined. Go back to s_0, imposing the constraint that the first argument of **read** must satisfy the predicate **writings**.

s_0 :

as before up to line 2	
$3'.\ \langle u, \{\mathbf{writings}(\Theta(u)),\ \Theta(u) \in s_0\}\rangle : e$	assume
add $c : e$ and $\mathbf{writings}(c) : t$ to s_0	$\Delta_{Shakespeare}$
choose $u = c$ for c in s_0	
$4'.\ \mathbf{read}(c) : e \rightarrow t$	MPP; 2, 3′
$5'.\ \mathbf{read}(c)(\mathbf{Mary}) : t$	MPP; 4′, 1

M_2 :

goal: $\Delta \cup \{a_0 : \mathbf{read}(c)(\mathbf{Mary})\} \vdash \beta : \neg\mathbf{read}(\mathcal{G})(\mathbf{Mary})$	
$6'.\ a_0 : \mathbf{read}(c)(\mathbf{Mary})$	assume
$7'.\ a_1 : \forall x, y, z[[\mathbf{read}(y)(x) \wedge y \neq z] \rightarrow \neg\mathbf{read}(z)(x)]$	$\Delta_{World\ Knowledge}$
$8'.\ a_2 : [\mathbf{read}(c)(\mathbf{Mary}) \wedge c \neq \mathcal{G}] \rightarrow \neg\mathbf{read}(\mathcal{G})(\mathbf{Mary})$	UI; 7′
$9'.\ a_3 : c \neq \mathcal{G}$	$\Delta_{Shakespeare}$
$10'.\ a_3a_2a_0 : \neg\mathbf{read}(\mathcal{G})(\mathbf{Mary})$	MPP; 8′, 6′, 9′

A wff of type t is attained in line 5 of s_0, and the process enters M_0. The wff $\mathbf{read}(\mathcal{S})(\mathbf{Mary})$ can be paraphrased as 'Mary reads the person William Shakespeare.' This proposition contradicts pre-existing assumptions like 'for all x and y, if x reads y, then y is a written document.' Consequently the proposition is prevented from being entered into the cognitive environment. The implicature formation process begins, motivated by the principle of relevance and the rule of abduction. The procedure so far is completely the same as in the case of metaphor. However, in the case of metonymy, the implicature formation component recognizes this statement as anomalous, after failing to find any relevant implicatures. We propose that the processing procedure backs up at this point to the explicature formation component in order to formulate a new logical translation for (21b). This backtracking is again motivated by the principle of relevance and the rule of abduction. As a part of this step, the assumption that anything read is a written document is introduced as a constraint on *Shakespeare*. At this point, the explicature formation component may employ the skolemization procedure alluded to above, in order to seek out a more plausible referent for the expression *Shakespeare*.

The procedure in 3′ is the same process as in the case of *the bishop* discussed above. It recovers written documents from the database linked to *Shakespeare*. At line 5′ a wff of type t is derived, and the assumption 'Mary is reading the writings of Shakespeare' enters M_2 and is accepted as a relevant statement to be inducted into the cognitive environment. Then, since we can read only one book at one time, we eventually draw the conclusion 'Mary is not reading *Gone with the Wind*' and thereby attain the goal of M_2. Lines 6′–10′ reflect a process common to all indirect speech acts. However, just as in the case of metaphor, there is a distinguishing characteristic here which makes this procedure recognizable as metonymy. Failing to find relevance in either the explicatures or implicatures of the first, literal reading of the metonymical utterance, the procedure backs up from the implicature formation component and re-enters the level at which explicatures are derived. However, on this round a new constraint is added, and through the skolemization process a second explicature is constructed, one which can be accepted as relevant when the process passes back into the implicature formation component. In this way, metonymy attains its relevance through the reconstruction of explicatures.

The procedure of reference assignment associated with metonymy is fundamentally similar to the case of the definite description *the bishop*. The crucial difference is the presence/absence of the lexical requirement. It the case of *the bishop*, the noun phrase with the definite article requires the reference assignment lexically. On the other hand, in the case of metonymy there is no such lexical requirement. Rather, the failure of the literal interpretation to achieve relevance initiates backtracking so that references may be reassigned in accordance with added constraints.

6 Conclusion

Now we can answer the two questions we posed at the beginning. First, 'How do we recognize utterances as instances of metaphor or metonymy?' We recognize these two rhetorical tropes by their characteristic processing procedures. In neither case is the literal explicature able to achieve

relevance, and so the interpretation procedure, motivated by the principle of relevance and the rule of abduction, continues to search for some way to make sense of the utterances. However, the strategies employed to recover from the initial failure of the literal explicature differ between metaphor and metonymy. Metaphor attains its relevance by implicature formation through lexical look-up for premises to use for inference. Metonymy attains its relevance by explicature reconstruction through lexical look-up for representation of entities. These processes are both pursued through databases linked to lexical items included in the utterance itself. However, the levels at which the recovery takes place are different—the implicature formation component in the case of metaphor and the explicature formation component in the case of metonymy.

Now recall our second question, 'How do "false" propositions take part in inference?' In the LDS framework, the 'falsity' of propositions doesn't cause any problems, because it adopts a proof-theoretic procedure and inferences are pursued based on forms. The falsity of the input merely causes the proposition to be expelled from the cognitive environment once it has served as an assumption to derive some relevant implicatum.

Let us therefore say in conclusion that the LDS framework, proposed by Gabbay and Kempson (1992a), brings to light the clear parallelism and difference between the interpretation processes of metaphor and metonymy, while solving the problem associated with the 'falsity' of propositions that are expressed by metaphorical or metonymical usages.