HARRY BUNT AND LEEN KIEVIT

AGENT-DEPENDENT METONYMY IN A CONTEXT-CHANGE MODEL OF COMMUNICATION

1. Metonymy

A well-known problem in the semantic interpretation of natural language is presented by the use of referring expressions to not directly point at their intended referents, but at some associated object. Examples are:

(1) I'm reading Shakespeare.

(2) The London office called.

(3) John works for this newspaper

(4) Turn up the radio.

(5) The buses are on strike.

The phenomenon illustrated by (1) - (5) is known as *metonymy*, which is defined in the Encyclopedia Brittanica (Brittanica, 2000) as follows:

figure of speech in which the name of an object or concept is replaced with a word closely related to or suggested by the original

Metonymic expressions often have a predicate applied to a different type of argument then it expects. The examples (1) - (5) all illustrate this: one doesn't read a person, but something written by that person; an office cannot make phone calls, only someone in the office, and so on.

Hobbs (2001) gives a slightly different definition of , namely as:

the linguistic device by which an entity is referred to by referring to a functionally related entity.

By requiring a 'functional relation' rather than being 'closely related to or suggested by', Hobbs ties the relation between intended and indicated referents closer, and requires that the relation always delivers a uniquely determined intended referent.

The idea of a functional relation between intended and indicated referent can be expressed formally as follows. We represent the application of a predicate P to an argument a as $P(a)$, and the 'functional relation' that constitutes a bridge over the type mismatch between predicate and argument as a function f. The modification of an expression E, in which a type mismatch occurs, to form an expression E' in which the type mismatch is resolved, we denote by $E \rightsquigarrow E'$. The bridging relation f thus has the effect (6):

(6) $P(a) \rightsquigarrow P(f(a))$

The construction of a bridge for dealing with a type mismatch can be expressed in terms of types as follows. Let the predicate P be applicable to objects of type β; P thus has the type of a function from β to the type *truthvalue*: $\mathsf{Type}(P) = (\beta \rightarrow \mathit{truthvalue})$. Let the argument a have type α. The function f then forms a bridge between β and α if it turns objects of type α into objects of type β, i.e. if $\mathsf{Type}(f) = (\alpha \rightarrow \beta)$. In terms of types, (6) corresponds to (7).

(7) $(\beta \rightarrow tv)(\alpha) \rightsquigarrow (\beta \rightarrow tv)((\alpha \rightarrow \beta)(\alpha))$

The phenomenon that the interpretation of an expression forces a part of it to be interpreted in a specific way in order to fit the context, is also known as *coercion*, and a function like f in (6) is called a *coercive function*. It may be noted that the representation of coercion in (6) can be read in two ways, since the right-hand side can be read as either (8a) or (8b):

(8) a. $P(\,(f(a)))$

b. $(P \circ f)(a)$

The first of these expresses the view that the argument a is coerced into the argument $f(a)$, to which the predicate P is applied. By contrast, (8b) expresses that the coercive function is used to construct a new predicate $(P \circ f)$, leaving the original argument unaffected. Nunberg (1995) calls the latter form of coercion *predicate transfer*, and suggests that most cases of metonymy should be analysed in that way, rather than as referring to a different argument. He argues that coercion of the argument only occurs when a demonstrative is used, as when a speaker is holding up his car keys and says (9):

(9) This is parked out back

Nunberg provides several arguments supporting the view that (9) is not a case of predicate transfer, such as the strangeness of continuing (9) with (10a), in contrast with the continuation (10b):

(10) a. *It only fits the front door.

b. It may not start at once.

Nunberg (1979) speaks of 'deferred ostentation' in cases like (9); perhaps *'argument transfer'* would be a more appealing terminology, parallelling 'predicate transfer'. We will use the latter terminology.

The claim that most cases of metonymy should be viewed as instances of predicate transfer is supported by the examples (1) - (5) which, different from (9), can all be continued by referring to the *indicated* arguments, as the (a) sentences show, rather than to the *intended* referents, as the (b) sentences witness. (See also Copestake and Briscoe, 1995 for arguments supporting this view.)

(11) I'm reading Shakespeare.

a. Did you see the house where he was born when you were in Stratford?

b. *He is in rather small print, though, which makes the reading less enjoyable.

(12) The London office called.

a. It's the one in Fleet Street.

b. *It didn't say a name.

(13) John works for a newspaper.

a. It's the one that you like to read at breakfast.

b. *It was founded in 1945.

(14) Turn up the radio.

a. It's in the corner behind that plant.

b/ *It's too low for me to hear what they say.

(15) The buses are on strike.

a. They are otherwise quite comfortable.

b. *They want better payment and more security.

We noted, in connection with (6), that the introduction of a coercive function in the semantic representation in itself may not be sensitive to the distinction between predicate transfer and argument transfer; however, in a DRT-style representation of a discourse like the one formed by (9) and (10b), the interpretation of the pronoun *It* in the second sentence may be problematic on a predicate-transfer analysis, where the first sentence would have introduced a discourse referent of the type *key* which would not fit the pronoun in the second sentence.

Although metonymy often involves type mismatches between a predicate and an argument, this is not necessarily the case. The examples (16) and (17) illustrate this.

(16) Aimez-vous Brahms?

(17) The train is getting more expensive next month.

Sentence (16) can be a question about the addressee's amorous feelings toward Mr Brahms or about her appreciation of Brahms' music, depending on the context. The 'product-of' reading may seem more plausible, but if the context is for instance that of a novel about Johannes Brahms and his love affairs, then the 'direct' reading may be more plausible. Similarly, (17) may occur in a conversation where one of the participants is considering to buy a train, and the direct interpretation may be the intended one; in more mundane contexts, where traveling by train is in focus rather than buying a train, the interpretation that train *tickets* are going up is more plausible. In these cases, an interpreter may have to construct a bridge for getting over the type mismatch between a predicate and the *intended*, rather than the *indicated* argument.

The most influential and elaborate computational treatment of metonymy has been provided by Pustejovsky (1993; 1995). This treatment assumes that entries in a lexicon may contain *qualia structures*, a set of entailments associated with the lexical entry. Pustejovsky has proposed four basic *qualia roles*:

1. *constitutive,* relating an object to its parts;
2. *telic,* connecting an object to its function and purpose;
3. *formal,* distinguishing an object within a larger domain;
4. *agentive,* relating an object to objects involved in its creation.

For instance, the lexical entry for *newspaper* might be like in (18), which says that a newspaper is a daily magazine that is created by a company who publishes it:

(18) dailymagazine(x)
  agentive: company(z) & publishes(z,x)

Qualia roles can be exploited when during interpretation a type mismatch is detected between the selectional restrictions imposed by a predicate, and one of its arguments.

Suppose for instance that in (3) the predicate *work for* is interpreted as employs(a,b), with the restrictions that a is a company and b is a person. Then dailymagazine(a) is not acceptable as interpretation of *newspaper*, since it is not a company. Following Pustejovsky's approach, we should then use one of the objects associated with the direct referent through some qualia role. In this case we could use company(a), which leads to a logical form like (19):

(19) ∃c newspaper(c) & ∃a company(a) & publishes(a,c) & employs(a,john)

This approach has been shown to be able to handle a wide variety of cases, in particular of instances of what Pustejosvsky has called *logical metonymy,* where additional meaning seems to arise for noun-verb combinations or adjective-noun combinations in a systematic way, as illustrated by (20) and (21):

(20) a. I've finished the book.

  b. Philippa enjoyed the book.

(21) a. Richard likes a fast car.

  b. Einstein was a passionate violinist.

The general phenomenon that is illustrated by (20) is that a verb that semantically takes an eventuality is combined with an object involved in an eventuality; similarly, (21) illustrates the application of an adjective which semantically applies to eventualities (*fast* to *go*, *passionate* to *play*) to an object involved. In both types of examples it would seem inadequate to postulate different word senses for the verbs and adjectives involved, as Pustejovsky has argued. The use of qualia structures in a single lexical meaning provides a more adequate treatment. There are counterexamples for this approach, however. For instance, as Lascarides and Copestake (1998) have noted, if Philippa in (20b) is a goat, then the intended interpretation of *enjoyed eating the book* cannot be accounted for. (See also Briscoe et al., 1990). Similarly, if Richard in (21a) is an employee in a car demolition company, then the intended interpretation *car that can be demolished quickly* will be unaccounted for. It clearly seems to be the case, as Lascarides and Copestake argue, that context considerations can override the considerations expressed in lexical qualia structures, and an adequate treatment of metonymy should have both a lexical and a context-dependent pragmatic component, a conclusion which is supported by the corpus study reported by Verspoor (1997).

Nunberg (1979) already argued that a purely linguistic account of metonymy, based on syntactic and semantic rules and lexical structures, is inadequate because metonyny in general requires taking nonlinguistic world knowledge into account. One of his most convincing examples to demonstrate this is

(22) The ham sandwich is waiting for his check.

which would seem possible only in a restaurant context when both speaker and addressee are waiters or waitresses. It is hard to see how the coercion that occurs in this example could be accounted for along the lines of Pustejovsky's approach, since a lexicon will not contain a qualia role relating ham sandwiches to the people who ordered them. What we need instead is an approach where the possible coercions depend on context and on the communicating agents that operate in the context.

Another phenomenon that points at the need of a more general interpretation of coercion is that coercive functions sometimes exhibit anaphoric behavior. Consider, for instance, the following pair of sentences:

(23) a. Try listening at that door.

b. Now try the other one.

In the second sentence *try* is used with an argument whose semantic type is not an event but an object. The only coercion that seems plausible in this case is one in which we map a door to the activity of listening at that door. This coercive function thus stands in an anaphoric relation to the linguistic context.

Hobbs (2001) has suggested a treatment of both predicate- and argument-transfer analyses of metonymy as well as of other forms of coercion on the basis of his 'interpretation-as-abduction' framework (Hobbs *et al.*, 1993). His treatment integrates syntactic, semantic and pragmatic analysis and uses weighted abduction to prove the logical form of an utterance, using the most salient axioms and other information, and making the minimal number of consistent and plausible assumptions to make the proof go through. These assumptions may include bridging functions for dealing with type mismatches. Syntax and semantics come together in the logical form as follows. A predicate Syn is introduced with seven arguments:

(24) $\mathsf{Syn(w, e, f, x, a, y, b)}$

where w is a string of words, e is the eventuality described by the string, f is the syntactic category of the head of the phrase w, x and y are the logical subject and object of the head of w if w contains them, and a and b are their respective syntactic categories; if the head of w does not contain its logical subject and/or object, then x and/or y and their categories are the empty symbol "-". A typical lexical axiom is the following:

(25) $\mathsf{read(e, x, y)} \wedge \mathsf{person(x)} \wedge \mathsf{text(y)} \rightarrow \mathsf{Syn(}\text{“}\textit{read}\text{”}\mathsf{, e, V, x, N, y, N)}$

To account for metonymy, Hobbs uses separate axioms for predicate transfer and argument transfer. For example, the axiom for predicate transfer in subject position is as follows:

(26) $\mathsf{Syn(w, e, P, x, a, y, b)} \wedge \mathsf{coerce\_rel(x, x')} \rightarrow \mathsf{Syn(w, e, P, x', a, y, b)}$

The effect of this axiom is that the predicate P is coerced into a predicate $\mathsf{P} \circ \mathsf{F}$, where F is a functional relation sanctioned by implying coerce_rel.[1] Similarly for coercion in object position.

[1] Hobbs uses a relation named '*rel*' for this purpose, rather than '*coerce_rel*'. We use '*coerce_rel*' for the sake of clarity.

This approach is clearly more powerful than Pustejovsky's lexically-based approach, but it lacks an explicit representation of contexts and a modelling of knowing, communicating and interpreting agents, which we believe to be essential for dealing with examples like *The ham sandwich is waiting for his check.*

In this chapter we will show how a formal model of communication between agents with explicitly represented information states leads to a notion of agent- and context-dependent coercion. The general problem of context-dependent interpretation and the use of CTT models of context is also discussed in Kievit (1998), Ahn (2000), and Bunt (2000).

2. Representing context and utterance meaning

The phenomenon of metonymy and related coercion phenomena belong to the boundary area of pragmatics, semantics, and knowledge representation and reasoning. Semantics is involved because the meanings of the words are the starting points of the interpretation of the metonymic expressions. Pragmatics comes in because the interpretations that we are trying to obtain depend heavily on the context in which the expressions are used; the context determines on the one hand which interpretations are relevant and plausible, and on the other hand provides the information that has to be combined with purely semantic information. The combination of various pieces of information from the context and from the lexical and phrasal meanings is in fact a form of *reasoning*; reasoning is also what we do when we decide which available contextual information we choose in order to arrive at a sensible interpretation. And finally, knowledge representation is needed to represent context information.

In the multimodal dialogue project DenK (see Bunt et al., 1998) we have developed and applied a powerful formalism for knowledge representation, semantics and reasoning, called Constructive Type Theory (CTT). Research in the DenK-project has shown the great potential of type-theoretical contexts as formalizations of context information, provided the expressive capabilities and proof methods of standard type theory are extended in order to model states of information and intention of agents participating in a dialogue (see e.g. Borghuis, 1994; Ahn & Borghuis, 1997; Kievit, 1998; Ahn, 2000; Kievit, Piwek, Beun & Bunt, 2001).

We briefly consider semantic representation in CTT and the use of CTT contexts for modelling the knowledge of communicating agents before outlining our CTT-based model of communication and applying it to the interpretation of metonymic utterances.

2.1. *Meaning representation in CTT*

CTT is a form of typed lambda calculus which, unlike traditional meaning representation formalisms, is not so much concerned with truth but with the representation of an agent's knowledge (or beliefs) and its dynamics, and with reasoning in the sense of effective construction of proofs on the basis of a given state of knowledge. Typed lambda calculi have been applied in foundational mathematical research, in the mathematical study of programming languages, and in the semantics of natural laguage. Barendregt (1991) noticed that many existing systems of typed lambda calculus can be uniformly represented using the format of *Pure Type Systems*. Formalisms witin this class include *Automath* (De Bruijn, 1980; Nederpelt et al., 1994) and the *Calculus of Constructions* (Coquand and Huet, 1988); closely related is also Martin-Löfs *Intuitionistic Type Theory* (Martin-Löf, 1984).

The language of typed lambda calculus consists of statements of the general form $\mathsf{V} : \mathsf{T}$, which should be read: 'V is an object of type T', or 'V is an *inhabitant* of T'. For example, an agent's recognition that Fido is a dog can be represented by: $\mathsf{fido} : \mathsf{dog}$, where dog is a type, and fido an inhabitant of that type. The subexpressions at the left-hand side of an introduction are variables or complex types (see below), those at the right-hand side are types. Statements where the left-hand side is a variable are called 'introductions'. Introductions can be grouped into sequences, which in type theory are called *contexts.* Contexts are the type-theoretical representations of an agent's knowledge.

Types are atomic or complex. Atomic types come in two varieties: constants and variables. A variable x may be used as a type, provided that it has been introduced in the context in which occurs; constant atomic types may be used at any point in a context, and are also called *'sorts'*. Among the sorts is *prop*, for propositions (more about propositions below), and *obj*, for objects of any other kind.

The ordering of statements in a context creates a dependency between them; for instance, type theory allows the statement $\mathsf{fido} : \mathsf{dog}$ to be formed only if the current context already contains $\mathsf{dog} : obj$, introducing the type dog. More generally, the type of a statement must be present or constructible in the context as an inhabitant. This relation of constructibility between a context Γ and a statement $A : T$ is written as $\Gamma \vdash A : T$.

A crucial and peculiar feature of type theory is that *propositions are treated as types* and inhabitants of a proposition as proofs of that proposition. Propositions thus occur in type-theoretical expressions in the same way as individual concepts, and proofs occur as individual objects. For instance, when an agent observes that it is raining, we can represent this by a statement like (27):

(27) x_0 : $\mathsf{itsraining}$

The term x_0 here stands for the agent's reason for believing that it is raining. Again, the type $\mathsf{itsraining}$ itself must have a type as well. Just as object concepts have the sort *obj* as their type, propositions are inhabitants of another supertype: *prop*(the type of all propositions).

The difference between proofs of propositions, such as x_0, and objects like fido is that we can express fido in natural language. Most objects in the world, like chairs, tables, and cars do not have a name; to represent information about them, an agent may use an expression like c_0 : car to record the knowledge of the existence of some specific nameless car.

Proofs and other objects may be combined to form new proofs and other objects. The inference rules of the type system describe the ways in which this can be done, and guarantee that all reasoning is sound. Type theory thus takes proofs to be first-class citizens: proofs are considered as abstract objects, have representations in the language, and are fully integrated within the formalism. This means that in type theory we can represent not only *what* an agent believes, but also *how* he comes to believe it, by having explicit representations of the proofs that justify his beliefs. The framework is thus *constructivist* in the sense that effective constructability of a proof is central, rather than truth.

The most important kind of complex type in type theory is that of the function type. Function types are defined as follows. If t_i and t_j are types and x is a variable then $\Pi\mathsf{x}{:}\mathsf{t}_i.\mathsf{t}_j(\mathsf{x})$ is a function type. If f is a function of this type, and y is an inhabitant of type t_i, then $\mathsf{f}(\mathsf{y})$ is an inhabitant of the type $\mathsf{t}_j(\mathsf{y})$. Since the resulting type $\mathsf{t}_j(\mathsf{y})$ depends on y, such types are called *dependent function types.* A special case is that where the variable x does not occur in $\mathsf{t}_j(\mathsf{y})$, in that case we use the simpler notation $\mathsf{t}_i \rightarrow \mathsf{t}_j$, instead of $\Pi\mathsf{x}{:}\mathsf{t}_i.\mathsf{t}_j(\mathsf{x})$.

Functions can be constructed in CTT by means of lambda-abstraction; when x is a variable of type t_i and E an expression of type t_j, then

$\lambda x : t_i.E$ is an inhabitant of the type $\mathsf{t}_i \rightarrow \mathsf{t}_j$ (or, more generally of the type $\Pi \mathsf{x}{:}\mathsf{t}_i.\mathsf{t}_j(\mathsf{x})$).

Predicates are represented by functions from types like car and dog to propositions. The knowledge that dogs bark may for instance be represented using the predicate barks:

(28) $\mathsf{barks} : \mathsf{dog} \rightarrow \mathit{prop}$

If an agent knows Fido, i.e. $\mathsf{fido} : \mathsf{dog}$ is in his context, then with this predicate he can construct the statement (29):[2]

(29) $\mathsf{barks(fido)} : \mathit{prop}$

Note that this does not mean that the agent believes that Fido barks, but merely that 'Fido barks' is a proposition, and as such may be true or false.

A predicate introduced as in (29) is applicable only to dogs. It is for example not applicable to cats. For predicates with a broader domain of application, it is convenient to introduce a notion of *inheritance* for types (indicated by '<'). So, instead of using two predicates for making growling sounds, one for dogs and one for cats, we can then introduce a single one, defined on a higher type:

(30) $\mathsf{animal} : \mathit{obj}$
$\mathsf{dog} < \mathsf{animal}$
$\mathsf{cat} < \mathsf{animal}$
$\mathsf{growls} : \mathsf{animal} \rightarrow \mathit{prop}$

This predicate 'growl' can be applied both to Fido and to Garfield.

Functions can also be used to construct individual objects. Since every dog's mother is a dog, we can have 'dogmother' as a function from the type dog to the type dog, i.e. $\mathsf{dogmother} : \mathsf{dog} \rightarrow \mathsf{dog}$. Such a function is called an *accessor function.*

Suppose we would like to represent an agent's belief that *all dogs bark.* This can be done by introducing a function f which, given a dog x, returns a proof that x barks. In other words, $\mathsf{f}(x)$ should be a proof of $\mathsf{barks}(x)$. Note that the type of the application of the function depends on the argument it is applied to. The Π operator can be used to construct such a function:

[2] In order to make CTT formulas better readable for readers who are not familiar with type theory, we use a notation that differs in some respects from the standard type-theoretical notation, and we occasionally simplify slightly.

(31) $\mathsf{f} : \Pi x : \mathsf{dog} \,.\, \mathsf{barks}(x)$

The Π operator thus functions as the type-theoretical counterpart of the universal quantifier in predicate calculus. Using the function (31), the agent can derive the proof $\mathsf{f(fido)} : \mathsf{barks(fido)}$, i.e. Fido barks.

Just as we have universal and existential quantification in predicate calculus, we also have two operators in type theory. Besides Π-types, corresponding to universal quantifiers, CTT has Σ-types, which corresponds to existential quantification. The notation of Σ-types is similar to that of Π-types. For example, *Some dog barks* is represented as

(32) $\mathsf{h} : \Sigma x : \mathsf{dog} \,.\, \mathsf{barks}(x)$

An inhabitant of this type is a pair consisting of a dog and a proof that that dog barks, such as our dog fido and the proof $\mathsf{f(fido)}$. We can use *projection* functions π_1 and π_2 to access the members of the pair, for instance, $\pi_1(\mathsf{h})$ has type dog. The standard notation for pairs using ordinary or sharp brackets is adequate when the second element in a pair does not depend on the variable introduced with the Σ operator; in such cases we will write (A, B) instead of $\Sigma x : \mathsf{A.B}$.

Dependent function types also allow us to represent utterances which are difficult to represent in predicate calculus. For instance, donkey sentences (known from the DRT literature) have a rather straightforward representation in CTT because we can make the type of the consequent dependent on the objects introduced in the antecedent. As an example, suppose that we have the following additional statements available:

(33) $\mathsf{man} : \mathit{obj}$
$\mathsf{own} : (\mathsf{man}, \mathsf{dog}) \rightarrow \mathit{prop})$
$\mathsf{walk} : (\mathsf{man}, \mathsf{dog}) \rightarrow \mathit{prop})$

The donkey sentence *If [a man]^i owns [a dog]^j [he]$_i$ walks [it]$_j$*, with the indicated links between the pronouns and the noun phrases, can be represented by the following statement:

(34) $\mathsf{z} : \Pi i : \mathsf{man} \,.\, \Pi j : \mathsf{dog} \,.\, (\mathsf{own}(i, j) \rightarrow \mathsf{walk}(i, j)$

This says that if we have a man i, and a dog j and a proof that i owns j, then we can construct a proof of the proposition that i walks j. For a detailed investigation into the correspondence between CTT-representations and DRSs, the reader is referred to Ahn and Kolb

(1990), where it is shown that CTT can be seen as a generalization of DRT.

2.2. *Communication in CTT*

Conventionalized cases of metonymy, such as *I like Shakespeare*, can be accounted for compositionally by assuming that 'Shakespeare' has conventionally acquired the additional meaning 'Shakespeare's works'. In general, however, metonymy is one of those linguistic phenomena where an intended meaning is expressed that cannot be computed compositionally from constituent meanings, but that an interpreter is supposed to infer with the help of context information. Examples like *The ham sandwich is getting restless* illustrate this. A general treatment of metonymy is therefore impossible in a classical compositional semantic framework, but requires a framework with communicating agents, context modeling, and inference.

When agents communicate, they try to convey something that they know or believe (or want, or hope,..). This means that they have to express elements from their internal states in a way that is observable and interpretable for the other agent.

Using CTT contexts to represent the information states of two agents, we can set up a formal model of communication (see Ahn and Borghuis, 1997; Ahn, 2000; Bunt, 2000) in which elements of CTT contexts can be converted into a common language of communication; when interpreting an utterance in the communication language, an interpreting agent constructs a representation of its meaning in CTT and uses this to update his (CTT-) context. To make this possible, the communicating agents must must both know how to express elements from their respective CTT contexts in the common language (and vice versa). For instance, if agent A uses the term dog_a in his context to represent knowledge about dogs, he may use the word *dog* to express this in the common language. Similarly, B may express the term dog_b from his knowledge state as *dog*. Note that the objects in each of the agents' knowledge states are known to the respective agents only, so each agent links his *own* concept *'dog'* to the word *"dog"*. We model this by assuming that for each agent there is a partial mapping between CTT terms and words in the language of communication.

Suppose, for example, that A wants to communicate to B that Fido barks. Let the knowledge of A be modelled by the CTT context on the left in (35a) and that of B by the context on the left in (35b), and let A and B have the partial mappings from CTT to the language of communication ('NL') indicated in these representations at the right.

(35) a.

A	CTT-NL mapping
dog_a : *obj*	dog_a $\rightsquigarrow$ *dog*
fido_a : dog_a	fido_a $\rightsquigarrow$ *Fido*
barks_a : dog_a $\rightarrow$ *prop*	barks_a $\rightsquigarrow$ *barks*
p_a : barks_a(fido_a)	

b.

B	CTT-NL mapping
dog_b : *obj*	dog_b $\rightsquigarrow$ *dog*
fido_b : dog_b	fido_b $\rightsquigarrow$ *Fido*
barks_b : dog_b $\rightarrow$ *prop*	barks_b $\rightsquigarrow$ *barks*

In general, more information is communicated by a NL utterance than a single CTT statement, so we assume that the first thing an agent does when communicating, is selecting a sub-sequence of statements from his context to be expressed in NL. Such a sequence is called a *segment.* In this case, A selects from his context the segment [p_a : barks_a(fido)].

Since proof terms typically have no natural language counterparts, A's CTT-NL mapping is not defined for p_a. However, the *existence* of a proof for a proposition may be communicated by uttering a linguistic realization of the proposition itself. So A tries to find a translation from barks_a(fido_a) into NL. The elements in this complex term both have a NL phrase assigned to them, and by combining these A can form the phrase

(36) Fido barks

Upon receiving and interpreting the utterance, B has to do something with it. Communicating agents typically assume that utterance are *meaningful*, that is, the information conveyed by the utterance can be integrated with the knowledge that B already has. What this means can be formalized in CTT as follows.

– An utterance is *meaningful* to an agent X if it can be converted into an *extending segment* of X's context.

– A segment $[A_1 : T_1, A_2 : T_2, \ldots, A_n : T_n]$ is an *extending segment* of a context Γ if there exists a type T such that $\Gamma \vdash T_1 : T$ and $[A_2 : T_2, \ldots, A_n : T_n]$ is an extending segment of $\Gamma, A_1 : T_1$ (that is, of the the context formed by Γ extended with the statement $A_1 : T_1$).
  A one-element segment $[A_1 : T_1]$ is an *extending segment* of a context Γ if $\exists T\ \Gamma \vdash T_1 : T$.

Following this assumption, and using the CTT - NL mapping in the direction from NL to CTT, B can interpret (36) to mean:

(37) $[x : \mathsf{barks\_b}(\mathsf{fido\_b})]$,

where x is a fresh proof term. If B believes what A is telling him, he can accept the segment and extend his context with it. For utterances with other communicative functions than statements, B will have to update his context in different ways rather than simply extending it this way (see Bunt, 1989; 2000).

Note that, if B did not have the mapping $\mathsf{fido\_b} \rightsquigarrow \textit{Fido}$ in his context, A could not communicate the message to him, since it would not be meaningful to B. Clearly, in order to communicate with B in a meaningful way, it is essential for A to know what B already knows. We will assume that each agent has not just a representation of his own knowledge, but also a representation of the information that he believes to be *shared*. We will call this the agent's *common context*. An agent can only use those segments in the communication with another agent that are an extending segment of his common context.

3. Meaningfulness and coercion

When an agent tries to interpret a metonymical utterance, he may end up with segments in which predicates are applied to arguments to which they cannot be applied in a meaningful way, due to a type mismatch. In terms of the communication framework outlined above, this means that the utterance does not correspond to an extending segment of the common context, and is thus not meaningful to the receiver.

Consider, for instance, the case where an agent A is presented with the utterance (13), repeated here for convenience as (38):

(38) John works for a newspaper

Assume that A has the knowledge represented in the context in the left part of (39), and uses the CTT $\rightarrow$ NL mapping represented in the right part. This context represents that A knows the concepts ‘publication’, ‘newspaper’, ‘company’ and ‘person’, and that A knows that a newspaper is a publication; that a publication is published by a company; that companies employ persons, and that John is a person.

(39)

A	CTT - NL mapping
$\mathsf{publication} : \mathit{obj}$	$\mathsf{publication} \rightsquigarrow \mathit{publication}$
$\mathsf{newspaper} < \mathsf{publication}$	$\mathsf{newspaper} \rightsquigarrow \mathit{newspaper}$
$\mathsf{company} : \mathit{obj}$	$\mathsf{company} \rightsquigarrow \mathit{company}$
$\mathsf{person} : \mathit{obj}$	$\mathsf{person} \rightsquigarrow \mathit{person}$
$\mathsf{publisher} : \mathsf{publication} \rightarrow \mathsf{company}$	$\mathsf{publisher} \rightsquigarrow \mathit{published\ by}$
$\mathsf{employs} : (\mathsf{company}, \mathsf{person}) \rightarrow \mathit{prop}$	$\mathsf{employs} \rightsquigarrow \mathit{works\ for}$
$\mathsf{john} : \mathsf{person}$	$\mathsf{john} \rightsquigarrow \mathit{John}$

When interpreting the utterance (38), A will initially construct the segment (40).

(40) $[n : \mathsf{newspaper}, p : \mathsf{employs}(n, \mathsf{john})]$

This is not an extending segment of A’s context, since the type of n is newspaper, while the predicate employs requires something of type company. The agent can try to remedy this by constructing a ‘type bridge’ which connects the types newspaper and company. Such a bridge can indeed be constructed by using the accessor function publisher. What is the mechanism that A applies when doing so?

Following the assumption that agents intend to communicate meaningful utterances, A, as a cooperative communicating agent, may be expected to try to make an adjustment to the segment (40) that would make it an extending segment of his context, and presumably he should not make unnecessary adjustments. Denoting A’s context, as represented in (39), by Γ_A, the first element of the segment does not cause a problem for the segment to be an extending one; $[n : \mathsf{newspaper}]$ is a one-element extending segment since $\Gamma_A \vdash \mathsf{newspaper} : \mathit{obj}$. But there is no type T such that $\Gamma_A \vdash \mathsf{employs}(n, \mathsf{john}) : T$; a mechanism to adjust the segment (40) minimally should therefore operate on the element $p : \mathsf{employs}(n, \mathsf{john})$.

To find the adjustment that we're looking for, a limited form of abduction can be applied. While abduction is usually defined as trying to prove that a formula is *true*, adding assumptions where necessary, we use abduction for trying to prove that a representation is *well-formed*, again making assumptions where necessary. In particular, we allow assumptions that consist of the insertion of accessor functions that form a bridge between the type of an indicated argument and that of an intended argument. In our agent's context we have such a function: publisher, even though its domain type is publication, because the type newspaper inherits from that. So a possible extending segment is:

(41) $[n : \mathsf{newspaper}, p : \mathsf{employs}(n, \mathsf{publisher}(\mathsf{john}))]$

Note that this bridging construction can be viewed both as an instance of argument coercion and as a case of predicate coercion, depending on whether one 'parses' the expression $\mathsf{employs}(n, \mathsf{publisher}(\mathsf{john}))$ semantically as (42a) or as (42b):

(42) a. *function* : employs,
argument : $(n, \mathsf{publisher}(\mathsf{john}))$

b. *function* : $\lambda X : \mathsf{employs}(\pi_1(X), \mathsf{publisher}(\pi_2(X)),$
argument : (n, john)

The mechanism illustrated in this example would allow *any* accessor function to be inserted for interpreting a metonymic expression. Nunberg (1995) goes at some length arguing that not just *any* function or relation may be used to construct or interpret a metonymic expression, but only one that says something 'noteworthy' about the argument to which a predicate is applied metonymically. For instance, we can say *the shoes were tied*, although literally only shoe*laces* can be tied, but we cannot say *the shoes were frayed.* Nunberg's explanation for this is that a shoe acquires a noteworthy property by its laces being tied, but not by its laces being frayed. Hobbs (2001) explicitly distinguishes those functions and relations that may be used for coercion from those that may not, by introducing an separate axiom for each coercive relation. For instance, to explain that *John read Proust* can be interpreted as *John read something written by Proust*, the axiom (43) is introduced, which says that writing as coercion relation:

(43) $\mathsf{write(e,\ x,\ y)} \rightarrow \mathsf{coerce\_rel(y,\ z)}$

In the abductive reasoning, this is combined, among other things, with the lexical axiom $Proust(e, z)$. We could do something similar in our approach and stipulate in an agent's knowledge state which functions can be used coercively.

An alternative view, however, is that when metonymy arises due to a predicate-argument type mismatch, the argument that is used serves as a way to identify the *intended* argument. If this view is correct, then *any* function that is effective to identify the intended argument would be a candidate for coercion. Nunberg's requirement of 'noteworthiness' can be interpreted as saying that a coercive function should be sufficiently 'noteworthy' to be effective for identifying the intended argument. Indeed, taking Nunberg's example of *tied/frayed shoes*, in a context where everyone's shoes are neatly tied, but one shoe stands out because its lace is frayed, it might make sense to identify that shoe by *the shoe that is frayed* or even by *the frayed lace*, by analogy with *the ham sandwich*.

As an example that supports this view on metonymy as *effective contextual identification*, consider a university classroom situation with a teacher who does not know the names of the students. If someone in the audience is signalling to ask a question, the teacher might react to this by using any of the following ways to identify the student in question:

(44) a. The blue sweater has a question.

b. The thick glasses has(!) a question.

c. The yellow Chevrolet has a question.

d. The wheelchair has a question.

e. The lighthouse has a question.

f. The baby face has a question.

g. The fatso has a question.

Some of these ways to identify someone may be infelicitous for social or political reasons, but even in those cases, if e.g. (44d) would be used, it could certainly be interpreted metonymically as identifying the intended person. On this view, there is no need to stipulate which functions or relations can be used for coercion, but this is strongly context-dependent. The context-dependence is for instance clear in the case of (44c), which is only possible if the speaker knows that a certain student in the class room has a yellow Chevrolet, and that the other students also know this (and that there's only one student with a yellow Chevrolet). Also, if the teacher is in a remote teaching situation, looking at the class room via a black and white monitor, and someone present in the class room (who is unaware that the teacher is looking at a black and white monitor) is using sentences from (44) to draw the teacher's attention to the fact that a student wants to ask a question, then the utterances (44a) and (44e) would not work.

So perhaps Nunberg's insistence that only 'noteworthy' properties of arguments can be used in metonymy could be construed as insisting that only properties can be used that are effective in a given context to identify an intended referent. Indeed, what is a 'noteworthy' property seems to be a context-dependent issue.

The assumption that the possible forms of metonymy are strongly context-dependent also allows us to explain how a sentence such as (45):

(45) The ham sandwich is getting restless

can be interpreted, and this is where our account differs fundamentally from lexically-based accounts, like Pustejovsky's.

If our agent is in fact a waitress in a restaurant, her knowledge context can be taken to contain the following information:

(46)

A	CTT - NL mapping
person : obj	person $\rightsquigarrow$ *person*
order : obj	order $\rightsquigarrow$ *order*
ham_sandwich $<$ order	ham_sandwich $\rightsquigarrow$ *ham sandwich*
orders : (person, order) $\to$ $prop$	orders $\rightsquigarrow$ *orders*
get_restless : person $\to$ $prop$	get_restless $\rightsquigarrow$ *gets restless*
orderax : Πx : order . Σy : person . orders(y, x)	

The last entry in (46) is an axiom stating that every order is ordered by someone. It will be crucial for our example, as it allows the construction of a bridge to coercively resolve the type mismatch in the agent's segment for the ham sandwich sentence, which would be (47):

(47) $[h : \mathsf{ham\_sandwich}, p : \mathsf{get\_restless}(h)]$

The type mismatch here is between the type of argument that get_restless expects and the argument type that is found. The context does not contain an accessor function mapping ham_sandwich to person, but we can construct one. The axiom in the last entry is a function which maps each object x of type order onto an object of type $\Sigma y : \mathsf{person}\ .\ \mathsf{orders}(y, x)$. As we have seen, inhabitants of such Σ-types are pairs. The first element in this pair will be an object of type person, being the person that ordered the menu item. In the CTT context at hand we can construct a function that maps order onto person, since the CTT type inference rules ensure that:

(48) $\Gamma_A \vdash \lambda r : \mathsf{order}\ .\ \pi_1(\mathsf{orderax}(r)) : \mathsf{order} \to \mathsf{person}$

Let us call this accessor function orderer.[3] Using this function, our waitress can turn the segment (47) into a well-formed one by applying it to the argument of the get_restless predicate:

(49) $[h : \mathsf{ham\_sandwich}, p : \mathsf{get\_restless}(\mathsf{orderer}(h))]$

Other examples, such as (2), *The London office called,* can be handled in the same way.

Like most treatments of metonymy, we have so far concentrated on those cases where a type mismatch occurs between a predicate and an argument. However, as already noted, metonymy may also arise without a type mismatch; examples are:

(50) a. Aimez-vous Brahms?

b. That ham sandwich looks nice.

c. The ham sandwich has fallen.

d. That's right beside André Previn.

(Where *André Previn* in (50d) may refer not only to the man himself, but for instance to a bust of André Previn, to a record with music by André Previn, or to a book containing the score of an André

[3] CTT has a definition device for formally introducing abbreviations. Using this device, the function orderer can be introduced by the statement $\mathsf{orderer} = \lambda r : \mathsf{order}\ .\ \pi_1(\mathsf{orderax}(r)) : \mathsf{order} \to \mathsf{person}$.

Previn composition.) Whether utterances like those in (50) should be interpreted metonymically is entirely context-dependent, and in some contexts these utterances are ambiguous between a non-coercive reading and various possible coercive ones.

A communication-based approach to metonymy seems the way to go in such cases even more than in the type-mismatch cases that we have discussed, since here an interpreter's task is clearly to determine the *intended* interpretation of the utterance, and this involves taking into account an interpreter's model of the speaker's intentions. It would take us beyond the scope of the present chapter to add the representation of intentions to type-theoretical contexts (see Ahn, 2000 and Bunt, 2000 for steps in this direction); here we merely outline how we may account for the ambiguity of such utterances.

The ambiguity between a metonymic and a nonmetonymic reading can be viewed in two ways, depending on whether the predicate involved is viewed as ambiguous or not, and this leads to slightly different analyses. This may be illustrated with the examples (50b) and (50c).

Consider example (50b) in a context, similiar to the previous one, but with additional elements for representing (in CTT) and expressing (in NL) that something or someone looks nice. If we assume that *looking nice* for a person means that this person seems pleasant, kind, while a nice-looking sandwich is one that seems tasty, i.e. that *looking nice* is in fact an ambiguous NL predicate, then an adequate way to treat this example is by having two different predicates in C]T (seem_pleasant and seem_tasty), which both map into the NL predicate *looks nice*. This is represented in (51).

(51)

A	CTT - NL mapping
order : *obj*	order ⇝ *order*
ham_sandwich < order	ham_sandwich ⇝
person : *obj*	*sandwich*
orders : (person, order) $\rightarrow$ *prop*	person ⇝ *person*
seem_pleasant : person $\rightarrow$ *prop*	orders ⇝ *orders*
seem_tasty : ham_sandwich $\rightarrow$	seem_pleasant ⇝ *looks nice*
prop	seem_tasty ⇝ *looks nice*
get_restless : person $\rightarrow$ *prop*	get_restless ⇝ *gets restless*
orderax : Πx : order . Σy : person . orders(y, x)	

When interpreting the utterance, A constructs two alternative CTT segments due to the lexical ambiguity of *look nice*:

(52) a. $[h : \mathsf{ham\_sandwich}, p : \mathsf{seem\_tasty}(h)]$

b. $[h : \mathsf{ham\_sandwich}, p : \mathsf{seem\_pleasant}(h)]$

The first of these is an extending segment of A's context, since A is familiar with the concept of a ham sandwich and with the predicate of seeming tasty, and these are such that A knows that ham sandwiches can appear tasty. Moreover, through his CTT – NL mapping A knows that one talks about ham sandwiches with a tasty appearance as *looking nice*. The second segment has a predicate-argument type mismatch.

A naive interpretation algorithm would in all likelihood discard the second interpretation because of the type conflict, given that there is an interpretation without type conflicts. This is clearly too simple. In a more sophisticated algorithm that also considers the metonymic interpretation of the utterance, we can apply the same mechanism as in the type-mismatch cases. As before, A can construct the function orderer which may be used to identify a person through his order; to this person the predicate seem_pleasant may apply, and so on.

Example (50c) is different, since the predicate *has fallen* has the same meaning irrespective of whether it is applied to a person or to a sandwich. We therefore assume a single CTT predicate fallen, that can apply to anything material, such as people and sandwiches. An agent interpreting this utterance will construct a single meaning representation (53):

(53) $[h : \mathsf{ham\_sandwich}, p : \mathsf{fallen}(h)]$

The knowledge state of the interpreting agent A now includes his knowledge of the predicate fallen. This is represented in (54).

(54)

A	CTT - NL mapping
material : *obj*	ham_sandwich $\leadsto$
ham_sandwich $<$ material	*ham sandwich*
person $<$ material	person $\leadsto$*person*
order : *obj*	order $\leadsto$*order*
content : ham_sandwich $\rightarrow$ order	content $\leadsto$*for a*
places : (person, order) $\rightarrow$ *prop*	orders $\leadsto$*places*
fallen : material $\rightarrow$ *prop*	fallen $\leadsto$*has fallen*
get_restless : person $\rightarrow$ *prop*	get_restless $\leadsto$ *gets restless*
orderax : Πx : order . Σy : person . orders(y, x)	

(It may be noted that we have made some additional changes in the context that used here, compared to the previous one. We have now introduced ham_sandwich as a subtype of material, in order to introduce the predicate fallen adequately. Keeping ham_sandwich also as a subtype of order would give rise to technical complications. Moreover, once we view a sandwich as a kind of material object, it no longer seems consistent to view it as a type of order; rather, it seems conceptually preferable to view a sandwich as the *content* of an order.)

To obtain the metonymic interpretation of (53), we can again use the same abductive mechanism as before except that now the mechanism is no longer used only to make adjustments for making the meaning representation well-formed and meaningful, but also to make adjustments consisting of the arepresentation well-formed and meaningful, but also to make adjustments consisting of the addition of accessor functions that *preserve* well-formedness.

Applying the same method again, using the accessor function orderer, A can construct the alternative representation (55).

(55) $[h : \mathsf{ham\_sandwich}, p : \mathsf{fallen}(\mathsf{orderer}(h))]$

In this way we can account both for those ambiguities between metonymic and nonmetonymic interpretations that may arise due to the ambiguity of natural language predicates and for those that are caused by polymorphic predicates in an agent's knowledge representation.

This approach can also handle examples of coercive functions with anaphoric behaviour, like (23) *Try listening at that door. Now try the* *other one.*, as well as so-called *sortal crossings,* where one part of an utterance should be interpreted metonymically and an anaphoric reference in another part should be interpreted nonmetonymically, or vice versa, as illustrated by (56):

(56) a. John works for the newspaper that you're reading.

b. Die blauwe Peugeot heeft haast; die moet vrijdag klaar zijn. (That blue Peugeot is in a hurry; it must be ready on Friday.)

c. My Willis gives the same printing problem, maybe we should contact him.[4]

4. Conclusions

Many cases of metonymy can be handled by using rich semantic lexical items, as Pustejovsky's use of qualia structures shows; this applies in particular to examples of logical metonymy and other conventionalized cases. But a general theory of metonymy must also have a pragmatic component that takes world knowledge into account, as Lascarides and Copestake, and several other authors have argued.

We have outlined an approach to metonymy where both an agent's world knowledge and his knowledge of the language are explicitly represented. We have used Constructive Type Theory to represent an agent's world knowledge, and his linguistic knowledge we have represented in an extremely primitive fashion as a partial mapping between CTT and NL terms. It should be noted that the particular choice of CTT as a representation formalism for world knowledge is not really crucial to our approach, however. What is crucial is that world knowledge and linguistic knowledge are both explicitly represented in a form that allows the application of inference techniques such as abductive reasoning.

We believe that a fully adequate theory of metonymy should not only take linguistic knowledge and world knowledge into account, but calls for a model of communicating agents, each with their states of world

[4] One of us (HB) actually used sentence (56c) when talking to this book's co-editor ET about a persisting problem in printing the postscript file which corresponds to Willis' and Manandhar's chapter in this book. Willis being a less famous author than Shakespeare or Proust, the coercive use of his name for referring to something written by Willis presents a real-life example of strongly agent- and context-dependent metonymy.

knowledge, which includes taking into account how the agents' knowledge states may change as a result of communication, as well as the way in which the terms in these knowledge states relate to expressions in the language of communication. Such a treatment can account for the context- and agent dependence of nonconventionalized metonymic expressions like the *ham sandwich* and *Willis printing* sentences.

ACKNOWLEDGEMENTS

I would like to thank Alex Lascarides for detailed comments on an earlier version of this chapter.

References

Ahn, R. (2000) *Agents, Object and Events. A computational approach to knowledge, observation and communication.* Ph.D. Thesis, Eindhoven University of Technology.

Ahn, R. and Borghuis, T. (1997) Communication Modelling and Context-Dependent Interpretation: An Integrated Approach. In A. Benz and G. Jäger (eds) *Proceedings of Mundial'97*, Munich, 1997.

Ahn, R. & Kolb, H.P. (1990) Discourse Representation Meets Constructive Mathematics. In: L. Kálmán & L. Pólos (eds.) *Papers from the Second Symposium on Logic and Language*, Budapest: Akadémiai Kiadó, 105–124.

Barendregt, H. (1991) Introduction to Generalized Type Systems. *Journal of Functional Programming, 1(2).* 125-154.

Borghuis, T. (1994) *Coming to terms with modal logic: on the interpretation of modalities in typed λ-calculus.* Ph.D. Thesis, Eindhoven University of Technology.

Briscoe, E., Copestake, A. and Boguraev, B. (1990) Enjoy the paper: lexical semantics via lexicology. In *Proc. of the 13th International Conference on Computational Linguistics* COLING-90,q 42–47.

Brittanica, Encyclopedia (2000) `http://www.brittanica.com/seo/metonymy`

De Bruijn, N.G. (1980) A survey of the project Automath. In: Seldin & Hindley (eds.) *To H.B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalisms.* Academic Press. 579-606.

Bunt, H.C. (1989) Information dialogues as communicative action in relation to partner modeling and information processing. In: M.M. Taylor, F. Néel & D.G. Bouwhuis (eds.) *The Structure of Multimodal Dialogue.* North-Holland, Amsterdam. 47-73.

Bunt, H.C. (2000) Dialogue pragmatics and context specification. In: H.C. Bunt & W.J. Black (eds.) *Abduction, Belief and Context in Dialogue, Studies in Computational Pragmatics*, Amsterdam: Benjamins, 81–150.

Bunt, H.C. Ahn, R., Kievit, L.A., Piwek, P., Verlinden, M., Beun, R.J., Borghuis, T. & Overveld, C. van (1998) Multimodal cooperation with the DenK System. In: H.C. Bunt, R.J. Beun, & T. Borghuis (eds.) *Multimodal Human-Computer Communication.* Berlin: Springer, 39–67.

Copestake, A. and Briscoe, E. (1995) Semi-productive polysemy and sense extension. *Journal of Semantics 12(1),* 15–67.

Coquand, T. and Huet, G. (1988) The calculus of constructions. *Information and Computation 76,* 95–120.

Hobbs, J.R., Stickel, M., Appelt, D. & Martin, P. (1993) Interpretation as Abduction. *Artificial Intelligence 63*, 69–142.

Hobbs, J.R. (2001) Syntax and Metonymy. In: P. Bouillon & F. Busa (eds.) *The Language of Word Meaning.* Cambridge, UK: Cambridge University Press, 290–311. Also available at `http://www.ai.sri.com/ hobbs/metsyn/metsyn.html`.

Kievit, L. (1998) *Context-driven Natural Language Interpretation.* Ph.D. Thesis, Tilburg University, 1998.

Kievit, L., P. Piwek, R.J. Beun & H.C. Bunt (2001) Multimodal Cooperative Resolution of Referential Expressions in the DENK system. In H. Bunt & R.J. Beun (eds.) *Multimodal Generation, Interpretation and Cooperation.* Lecture Notes in Artificial Intelligence 2155. Berlin: Springer, 202–220.

Lascarides, A. and A. Copestake (1998) Pragmatics and Word Meaning. *Journal of Linguistics 34,* 387–414.

Martin-Löf P. (1984) *Intuitionistic Type Theory*. Naples: Bibliopolis.

Nederpelt, R., Geuvers, H. and de Vrijer, R. (1994) (eds.) *Selected Papers on Automath.* Studies in Logic and the Foundations of Mathematics, Vol. 133, Amsterdam: North-Holland.

Nunberg, G. (1979) The Non-Uniqueness of Semantic Solutions: Polysemy. *Linguistics and Philosophy 3(1),* 143–184.

Nunberg, G. (1995) Transfers of Meaning. *Journal of Semantics 12,* 109-133.

Pustejovsky, J. (1993) Type Coercion and Lexical Selection. In J. Pustejovsky (ed.) *Semantics and the Lexicon*, Studies in Linguistics and Philosophy 49, 73 – 94, Dordrecht: Kluwer.

Pustejovsky, J. (1995) The Generative Lexicon. Cambridge, MA: MIT Press.

Verspoor, C. (1997) Conventionality-Governed Logical Metonymy. In H. Bunt and R. Muskens (eds) *Proceedings of the Second International Workshop on Computational Semantics,* Tilburg 1997, 300–312.