In: Matthias Weisgerber (ed.) 2003: *Proceedings of the Conference "sub7 – Sinn und Bedeutung". Arbeitspapier Nr. 114, FB Sprachwissenschaft, Universität Konstanz, Germany.* `http://ling.uni-konstanz.de/pages/conferences/sub7/`

# Interpreting 'Be'*

Ronnie Cann
University of Edinburgh

R.Cann@ed.ac.uk

Abstract

This paper proposes an approach to the interpretation of copular sentences in English that analyses be as a one-place predicate with underspecified semantic content. The interpretation of a copula clause is determined through the interaction of syntactic and pragmatic processes and depends on the properties of expressions collocated with be and general contextual factors, both local and non-local. Analyses are provided for predicational, specificational and equative sentences using the framework of Dynamic Syntax.

## 1 Introduction

The copula verb be can appear in range of constructions apparently involving complements of different sorts and a wide variety of interpretations. For example, in English we find be inducing an interpretation of identity with a noun phrase complement in equatives (1-a); as doing little more than hosting tense and agreement information with adjective, prepositional and nominal phrases in predicatives (1-b); giving rise to existential interpretation in construction with there (1-c); as some sort of presentational marker with an expletive subject (1-d); as part of a construction determining focus in cleft (1-e), and pseudo-cleft (1-f) constructions; and (rarely) as providing 'existential focus' in certain intransitive constructions (2):[1]

(1)
- a. John is the teacher.
- b. Kim is happy.
- c. There is a riot on Princes Street.
- d. It's me.
- e. It is John who is the teacher.
- f. What I want is a good night's sleep.

(2) Neuroses just are (they don't need a cause)

The variability in the interpretation of be in (1) is further compounded by the subtle differences in meaning apparently exhibited by very similar sentences. For example, copular clauses involving a definite noun phrase give rise to slightly different interpretations according to the order in which the noun phrases appear and are often divided into two

*I am grateful to Ruth Kempson, Caroline Heycock, Lutz Marten, Masayuki Otsuka, and Dan Wedgwood for discussions of the ideas that appear in this paper. I am also grateful to the people at the Existence conference in Nancy in September 2002 who commented on an earlier version of this paper and to the members of the Edinburgh Syntax and Semantics Research Group for putting up with presentations that last considerably longer than an hour. The usual caveats apply.

[1] I leave on one side the 'grammaticalised' constructions of passive and progressive in English.

classes: equative clauses, as in (3-a) where the post-copular definite appears to be fully referential, and specificational clauses, as in (3-b) where the initial definite appears to provide a description of an unknown entity, rather than to pick out some specific object.

(3) a. John is the culprit.
 b. The culprit is John.

There have been many attempts to reconcile these different interpretations of copular constructions and so reduce the aparent homonymy associated with be. For example, predicative and equative interpretations which seems to require at least two homonyms of the verb be, one more or less empty of content and the other specifying an identity relation. Semantic attempts to resolve this ambiguity, such as those in Williams (1983) and Partee (1986) favour treating the copula as 'essentially' predicative. For example, Partee's account provides the copula with a single semantic type $(e \rightarrow t) \rightarrow (e \rightarrow t)$ with the semantic structure proposed in Montague (1973), i.e. $\lambda P \lambda x.P(x)$[2]. The difference between the two readings is derived through a type shifting operation (Ident) on a postcopular term to turn it into an identity predicate. The details are not important here but one of the things that such an analysis fails to account for is the existential effect of be exhibited not only in the there be construction in (1-c) but also in the intransitive usage in (2).

Although examples of this existential focus construction are not common and somewhat marginal, they mark a significant difference between the copula and modal auxiliaries in English. Be but not the modals allows construal of existence in a null context as illustrated in (4) whereas the modals may and can do not license interpretations where possibility and ability in general are ascribed to the subject, but can only be interpreted elliptically. This indicates that be is being treated as a one-place predicate, rather than as providing a null complement that requires reconstruction from context.

(4) a. Neuroses just are. (= Neuroses exist)
 b. Neuroses just may. ($\neq$ Neuroses are possible)
 c. The students just can. ($\neq$ The students are able)

Interestingly, only definites and generics appear felicitously in this construction. So, (5-a) can only be interpreted elliptically, while (5-b) must be interpreted elliptically or generically.

(5) a. Every woman just is, OK?
 b. A woman just is, OK?

Furthermore, while an existential interpretation of the copula is found also with there as in (1-c), it is not found in all occurences with this string. So, in presentational uses (typically involving a definite associate as in There's the student you wanted to see), existence is rather presupposed than asserted. Interestingly in such examples, the interpretation of the whole sentence depends on non-local context[3]. So, for example, I might be telling the hearer that some relevant student is here or reminding the hearer about her afternoon appointments and so on. The fact that the interpretation of a clause containing be may alter according to the expressions with which it appears, indicates that it is dependent on

[2]Partee, in fact, allows a variable type and analysis with the arguments of the expression appearing in either order.

[3]Non-local context being interpreted here as context that is not provided by the linguistic string so far parsed.

context for its meaning. Thus, the interpretation of a there be sentences as presentational or existential seems to be attributable to the definiteness of the post-copular associate, existence is not (necessarily) predicated of definite associates (see also Mikkelsen 2002, Geist 2002, inter al.). Since the opposite seems to be true in the case of the existential focus construction (or at least that 'true' indefinites seem not to give rise to an existential interpretation) further indicates that the form of the whole clause (and its prosody) contributes to the interpretation. In other words, the interpretation of copular clauses depend on inference in context and should be analysed pragmatically rather than semantically.

This paper develops this idea and pursues an analysis of predicative, specificational and equative constructions in English that treats be as providing an underspecified one-place predicate. The content of this predicate is provided by the interaction of syntactic and pragmatic (inferential) processes, and the interpretation of these copular clauses is shown to be determined by the properties of collocated expressions, non-local context and the parsing process itself.

## 2 Dynamic Syntax

The framework to be used is that of Dynamic Syntax (Kempson et al 2001) according to which the process of natural language understanding is a monotonic tree growth process defined over the left-right sequence of words, with the goal of establishing some propositional formula as interpretation. To model the process of establishing such a structure as interpretation, all nodes in the semantic trees constructed during a parse are introduced with requirements to be fulfilled, reflecting the idea that the tree is underspecified with respect to some property that needs to be specified. Requirements may be to specify values for any of the labels that decorate a node, but the principal drivers of the parsing process are requirements to establish nodes decorated with formulae of certain types, starting from the initial (universal) requirement to build a representation of the propositional content expressed by a string in context: $?Ty(t)$, an instruction to build a tree rooted in $Ty(t)$, the type of a proposition.

To satisfy such requirements, a parse relies on information from various sources. In the first place, there are general processes of construction which give templates for building trees that may be universally available or specific to a language. A pair of such construction rules determine that a tree rooted in $?Ty(Y)$ may be expanded to one with argument daughter $?Ty(X)$ and functor daughter $?Ty(X \rightarrow Y)$. Thus, the initial unfolding of a requirement $?Ty(t)$ may be to establish subgoals $?Ty(e)$ and $?Ty(e \rightarrow t)$, requirements to build the subject and predicate nodes, respectively, as in Figure 1.

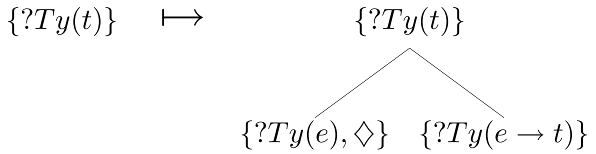


Figure 1: An initial expansion of $?Ty(t)$

Information about tree building also comes from packages of actions encoded in lexical entries which are accessed as words are parsed. An entry for a word contains conditional information initiated by a trigger (the condition that provides the context under which subsequent development takes place), a sequence of actions (possibly involving the building of nodes and/or the annotation of a node with type and formula information) and a failure statement (commonly an instruction to abort the parsing sequence) if the conditional action fails. For example, parsing the word John is associated with the lexical information in (6) which induces an annotation of the current node with formula and type values just in case that node has a requirement of type $e$ and otherwise fails. Parsing a verb like upset, on the other hand, gives rise to a more complex set of actions that build and annotate nodes and impose an additional requirement to construct a representation of the content of an object DP as illustrated in Figure 2. [4]

| | | | | |
|---|---|---|---|---|
| | | IF | $Ty(e)$ | trigger |
| (6) | John | THEN | $put(Ty(e), Fo(John), [\downarrow]\bot)$ | actions |
| | | ELSE | ABORT | failure |

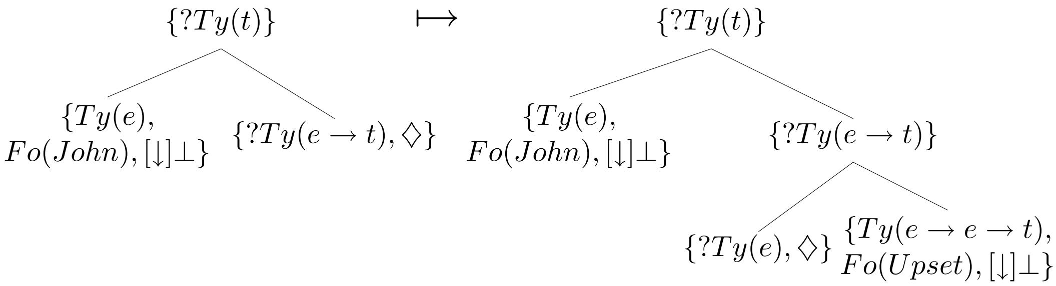


Figure 2: Parsing John upset

The parse continues just in case there the next word has a trigger of the appropriate type, i.e. $?Ty(e)$. A string like John upset Mary thus gives rise to a tree with all terminal nodes type and formula complete. The remaining requirements on the predicate and propositional nodes are satisfied through the compilation of the tree which is obtained by applying functional application over types to yield the completed tree in Figure 3.

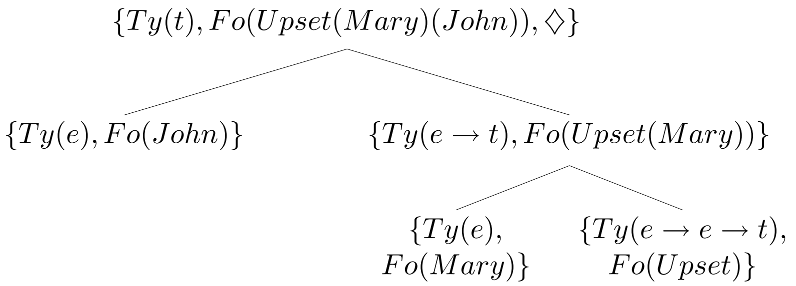


Figure 3: Completing a parse of John loves Mary

As noted above the driving force of the parsing process is the need to resolve requirements to specify underspecified information, of which the most important is the requirement to construct a formula value with a particular type. However, any predicate used to decorate tree nodes may be associated with a requirement and this will drive the parsing process in different ways. One such requirement is to find a fixed position within a tree. Every node in a tree is associated with an address which is encoded as a value of the treenode predicate, $Tn$. The topnode of a tree has an address $Tn(0)$ from which other addresses are constructed regularly: the functor daughter of a node with address $Tn(n)$ has an address

[4] See Kempson et al. (2001) for more details. Here and below, all tense information is ignored as not germane to the current discussion.

$Tn(n1)$ while the argument daughter has an address $Tn(n0)$. In Figure 3, for example, the node labelled by $Fo(John)$ has an address of $Tn(00)$, the predicate node has address $Tn(01)$ and the node decorated with $Fo(Upset)$ has address $Tn(011)$ and so on.

This method of defining treenode addresses is related to one of the principal descriptive mechanisms of Dynamic Syntax: the Logic of Finite Trees (LOFT, Blackburn and Meyer-Viol 1994). This modal logic provides a means of referring to arbitrary nodes in a tree using the following modal operators (amongst others): $\langle\downarrow\rangle$ the general daughter relation; $\langle\downarrow_0\rangle$ and $\langle\downarrow_1\rangle$ the argument and functor daughter relations, respectively; $\langle\downarrow_*\rangle$ the dominance relation (the reflexive, transitive closure of the daughter relation); and the inverses of these using the mother relation, $\uparrow$. The modalities $\langle\uparrow_*\rangle$ and $\langle\downarrow_*\rangle$ provide a means of characterising dislocated expressions. When an expression is parsed, it need not be associated with a fixed position within a tree but may be underspecified in position with respect to some dominating node, $\alpha$, thus having a modality $\langle\uparrow_*\rangle\alpha$, with a requirement to find a fixed position within the tree $?\exists\mathbf{x}.Tn(\mathbf{x})$. Such positional underspecification is used to account for long distance dependencies which are analysed in terms of initially unfixed nodes whose position in the emergent tree structure is fixed at some later stage in the parsing process. A construction rule of *Adjunction[5] introduces unfixed nodes, defining a transition from an incomplete tree of $Ty(t)$ with only a single node to a tree that contains in addition a node characterised as dominated by a tree node $a$ with requirements to identify the address of the unfixed node and to construct a type $e$ decoration, as shown in Figure 4.[6]

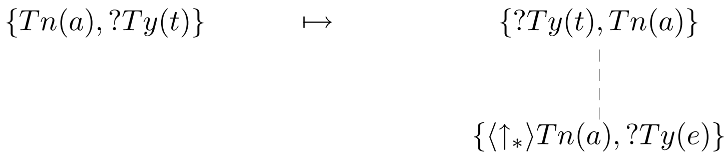


Figure 4: Introducing an unfixed node

Analysing the string Mary, John dislikes in these terms is illustrated in Figure 2 with an initially projected unfixed node and the pointer at the object position. At the point

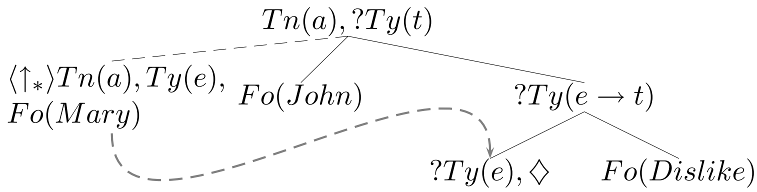


Figure 5: Parsing Mary, John dislikes

in the parse at which all words in the string have been processed, two requirements remain outstanding: to find a fixed position for the unfixed node and to construct a node of type $e$. In this environment, a process of merge (indicated by a dashed grey line) may take place which unifies the unfixed treenode with the current node. In this process, the information on both nodes is combined and the merge is successful just in

[5]Formally,

$$\frac{\{\{Tn(a), \ldots, ?Ty(t), \Diamond\}\}}{\{\{Tn(a), \ldots, ?Ty(t)\}, \quad \{\langle\uparrow_*\rangle Tn(a), \ldots, ?Ty(e), ?\exists x.Tn(x), \Diamond\}\}}$$

[6]The modality $\langle\uparrow_*\rangle$ is defined as: $\langle\uparrow_*\rangle\alpha \rightarrow \langle\uparrow\rangle\alpha \vee \langle\uparrow\rangle\langle\uparrow_*\rangle\alpha$.

case no contradictory decorations result. The merge process satisfies both outstanding requirements: the unfixed node provides the necessary type and formula decorations, while the fixed node provides the appropriate treenode address for the unfixed tree. Ultimately, completion of the tree yields a $Ty(t)$ Formula value, $Dislike(Mary)(John)$ decorating the topnode, with all requirements fulfilled.

In Dynamic Syntax it is the interaction of computational, lexical and pragmatic processes which determines the interpretation of a string. A wellformed string is one for which at least one logical form can be constructed from the words in sequence within the context of a given class of computational and pragmatic actions with no requirements outstanding. In consequence, as we shall see, the imposition of requirements and their subsequent satisfaction are central to explanations of given phenomena.

## 3 Content Underspecification

Within Dynamic Syntax, noun phrases are analysed as projecting expressions of type $e$, a move that is made possible for indefinites and (certain) quantified expressions by the use of the epsilon calculus of Hilbert and Bernays (1939). Indefinite noun phrases, for example, project epsilon terms that denote arbitrary witnesses for the set denoted by the common noun (see also Egli and von Heusinger 1995, Kempson et al 2001, Meyer-Viol 1995). Despite being of type $e$, the tree structures that represent the content of such quantified terms in DS is complex, containing two nodes of Type $e$, that of the top node and one embedded within the structure that hosts the variable bound by the quantifier. A quantified term thus consists of a triple: a quantifier, a variable, and a restrictor contraining an instance of the variable determined by the content of the common noun. It is not necessary at this point to go into details, but what is important to note for the purposes of this paper is that indefinites project fully specified tree structures as shown in Figure 6 which illustrates the structure projected on parsing the indefinite noun phrase a student.

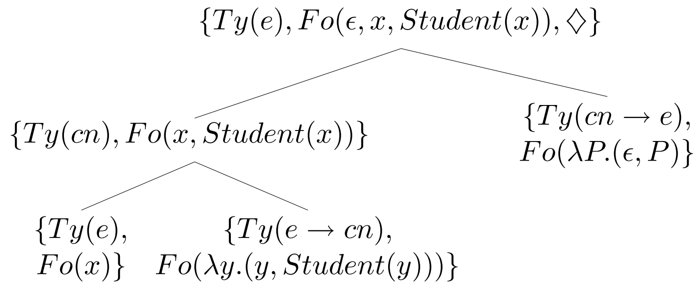


Figure 6: Parsing a student

While indefinites are treated as projecting epsilon terms with full tree structure, pronouns are treated as providing placeholders for such terms. This phenomenon of content underspecification involves lexical projection of a metavariable which is to be replaced by some selected term during the construction process. Such replacement is associated with a substitution process that is pragmatic, and system-external, restricted only in so far as locality considerations distinguishing individual anaphoric expressions preclude certain formulae as putative values of the projected metavariable (i.e. analogues of the Binding Principles, Chomsky 1981, etc.).

(7) Q: Who upset Mary?
Ans: John upset her.

In processing the pronoun in (7), the object node is first decorated with a metavariable U, with an associated requirement, $?\exists\mathbf{x}.Fo(\mathbf{x})$ to find a contentful value for the formula label. Construed in the context provided, substitution will determine that the formula $\mathbf{U}$ is replaced by $Mary$:

| | | | |
|---|---|---|---|
| | | IF | $?Ty(e)$ |
| (8) | her | THEN | $put(Fo(\mathbf{U}_{FEMALE}), Ty(e), ?\exists\mathbf{x}.Fo(\mathbf{x}), [\downarrow]\bot)$ |
| | | ELSE | ABORT |

Note the restriction on the metavariable indicated by the subscript. This provides the 'presupposition' associated with the pronoun that the term to be substituted is female or has some property that licenses the use of the descriptive property female. The substitution of Mary rather than (say) Bill for the metavariable in (7) is supported by the fact that Mary is assumed generally to be a name for a female while Bill is not. The fact that the pronoun her could be used to refer to Bill (or some other male) in a different context (e.g. because Bill is dressed as a woman) does not undermine the use of the pronoun to identify a relevant term (e.g. by identifying a term picking out something that is dressed as a woman). The property of being female may not, therefore, cash out truth conditionally as a property of whatever term is substituted for the metavariable in all circumstances: the presupposition is a constraint on a pragmatic process, not an assertion that some property holds of some particular term.

A similar approach can be taken with respect to definites. The definite article projects a metavariable and the associated common noun phrase provides the pressupposition[7]. The effect of the metavariable is to force some inferential effort to satisfy the associated requirement to find a formula value. This process involves the identification of some relevant term constructed from the local context which may be some name, actual or arbitrary, or an epsilon term constructed from information already provided within the discourse. Consider the small text in (9).

(9) Mary's PDA was stolen. The culprit got clean away.

Here, the first sentence provides the context for interpreting the definite in the second. So we have (something like) $\exists x.Stole(PDA)(x)$ as the formula value for the former and parsing the definite NP as $\mathbf{U}_{CULPRIT}$ in the latter requires the identification of some contextually salient term that also satisfies the property of being a culprit. Since a culprit is someone/thing that does something nefarious like stealing, the most obvious choice of term in this context is to take the existential formula provided by preceding sentence and select the arbitrary individual who stole Mary's PDA, i.e. $(\epsilon, x, Steal(PDA)(x))$, as substituend. The second clause is thus given a formula value in (10), the presupposition being satisfied by the lexical semantics of steal.

(10) $Get - Away((\epsilon, x, Stole(PDA)(x))_{CULPRIT})$.

The concept of content underspecification as represented by a metavariable provides the basis for the current analysis of the copula, be. As discussed in section 1, the variability in interpretation of the copula in the sentences in (1) and (3) provide some evidence for treating be as inducing pragmatic inference, rather than projecting content directly. Since metavariables induce inference because of their associated requirement to identify content,

[7]Formally defined through a LINK relation, see Cann 2002.

be may be treated as projecting a metavariable of some type. The various constructions in (1) imply a number of different types that could be adcribed to the copula amongst which are $e \rightarrow e \rightarrow t$ (equative), $(e \rightarrow t) \rightarrow (e \rightarrow t)$ (predicative), and $e \rightarrow t$ (existential focus). Here, I adopt the hypothesis that it is the latter type, that of a simple one-place predicate, that is appropriate, partly for the reason noted earlier that the existential focus construction indicates that there is no necessary 'complement position' which permits only an elliptical reading when be appears intransitively, unlike with the modals. Thus, I propose that forms of be project a predicate metavariable which I represent as **BE**, rather than **U** in order to highlight its type and provenance. [8]

| | | | |
|---|---|---|---|
| | | IF | $?Ty(e \rightarrow t)$ |
| (11) | be | THEN | $\mathtt{put}(Ty(e \rightarrow t), Fo(\lambda x.\mathbf{BE}(x)), ?\exists\mathbf{x}.Fo(\mathbf{x}))$ |
| | | ELSE | ABORT |

4 Predicative Constructions

The analysis of copula constructions developed here utilises underspecification of both formula value and position within a tree and takes as its starting point the analysis of expletives sketched in Cann (2001) and Cann et al. (2002). Pronouns in English typically do not permit substitution by an unfixed expression and so we find that the use of resumptive pronouns with topic constructions and WH questions is marginal or excluded.[9]

(12) a. ??Many types of beans, I like them, but much meat, I don't like it.
b. *Who did you see them?

However, there are pronouns that are systematically associated with material that occurs elsewhere in a string, amongst which is the expletive pronoun it in English. One way of characterising this pronoun is in terms of the bottom restriction, $[\downarrow]\bot$, the constraint associated with contentive expressions that they annotate a terminal node in the tree by requiring any node they dominate to have no properties. If an expression fails to annotate its node with the bottom restriction, it will allow further development of that node. In other words, parsing the expression may provide a placeholder that can be developed later on in the parse process, the essential characteristic of an expletive.

Consider the pronoun, it in extraposition constructions such as It seems that I am wrong. This example may be analysed as involving the annotation by the pronoun of a (propositional) node in subject position with a metavariable U and associated requirement, $?\exists\mathbf{x}.Fo(\mathbf{x})$, to find a contentful formula value. Hence, in parsing the example, the tree

[8] Only stative predicates that are associated with non-verbal expressions can be associated with copular be.
i. *Kim knows the answer and Lou is, too.
ii. *Kim is knows the answer.
Maienborn (2002) argues for a differentiation between Davidsonian states (or D-states) and states that she refers to as K-states following Kim (1969, 1976)'s notion of temporally bounded property exemplifications. She suggests that such states are not eventualities but form a separate class of abstract object (Asher 1993) somewhere between world bound facts and spatio-temporally defined eventualities. This idea can be implemented here by annotating the proposition derived from applying **BE** to some term with a label $s_K$, restricting the output only to K-states, i.e. $\lambda x.s_K : \mathbf{BE}(x)$, but this refinement is ignored in what follows.

[9] See Cann, Kempson and Kaplan 2002 for some discussion of resumptive pronouns in topic and relative clause constructions.

unfolds with requirements for nodes of types $t$ and $t \to t$. The word it provides a placeholder for the type requirement of the subject node and the verb is parsed to project a formula $\lambda p.Seem(p)$ of type $t \to t$. The partial tree may at this point be completed to yield a tree rooted in $Ty(t)$ with an incomplete formula value: $Fo(Seem(\mathbf{U}))$. A general construction rule, Final *Adjunction[10], may then be invoked which, given a complete propositional structure, allows the construction of an unfixed node of arbitrary type.

An application of this rule permits the construction of an unfixed propositional node that allows the parsing of the string final clause. As illustrated in Figure 7, the only node with which the unfixed node can merge coherently is that decorated by the metavariable. This is so, because only the subject node lacks the bottom restriction and only its formula value is consistent with that decorating the unfixed propositional tree. Merging the unfixed tree with the subject node thus yields a tree with a formula value $Fo(Seem(Wrong(Ronnie)))$, as required.

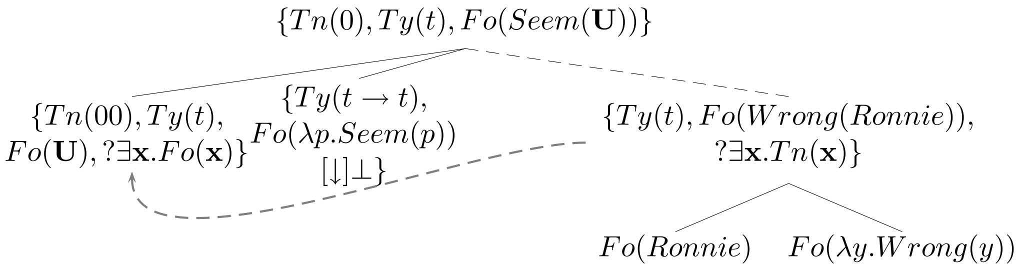


Figure 7: Parsing It seems that I am wrong

This combination of an unfixed final expression plus merge with some fixed position provides the general characterisation of copular clauses. Since, by hypothesis, the copula provides a one-place predicate, it can combine with the subject to give a type-complete tree in the same as any other intransitive construction. However, like the expletive construction sketched above, this propositional tree is not formula complete, requiring some predicate formula to be determined. In predicative constructions, this formula is provided directly by the syntactic mechanisms already introduced, i.e. a right unfixed node plus merge, except that the final unfixed tree is not of type $e$ but of type $e \to t$ and the merge site is thus the node decorated by the copula. This is possible because be does not annotate the predicate node with a bottom restriction, giving it the properties of an expletive whose content is determined within the string being parsed.

As an example, consider the parse of Kim is happy. The first two words are taken to project a structure which compiles to yield the propositional $Fo(\mathbf{BE}(Kim))$. The rule of Final*Adjunction allows the projection of an unfixed predicate node which permits the parse of any one-place predicate, in this case the simple adjective happy. The node decorated by the adjective then merges with the underspecified main predicate expression, satisfying both the requirement of the unfixed node to find a fixed position within the tree and the requirement that **BE** be replaced by some contentful concept. This process is illustrated in Figure 8 which yields a final formula value $Happy(Kim)$.

[10]Formally,

$$\frac{\{\{Tn(a), \ldots Ty(t), \diamondsuit\}\}}{\{\{Tn(a), \ldots, Ty(t)\}, \quad \{\langle\uparrow_*\rangle Tn(a), \ldots, ?Ty(X), \diamondsuit\}\}}$$

See Cann et al. 2001 for a justification of this rule.

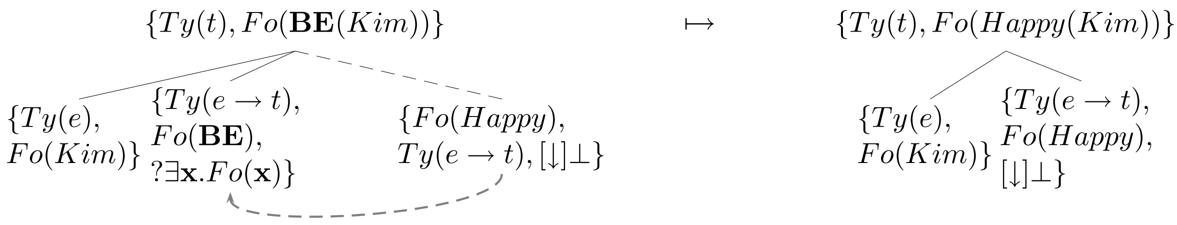


Figure 8: Parsing Kim is happy.

Note that it is possible for **BE** to be replaced through the process of pragmatic substitution, as with any metavariable when the pointer is at the appropriate node. This is the analysis of 'elliptical' constructions with be such as Kim is where some appropriate predicate is selected from context. For example, in a context provided by the question Who's in the kitchen, this elliptical expression will be given propositional content as $\lambda x.In(x, (\epsilon, y, Kitchen(y)))(Kim)$. However, if pragmatic substitution occurs when there is a following predicate, the parse will fail because the unfixed node will not be able to be fixed and yield a coherent interpretation, since the formula values on the two predicate nodes will not (in general) be compatible.

Prepositional predicates may be treated in the same way, under the (natural) assumption that such expressions may be of predicate type. So, a sentence like that in (13-a) gets the formula value in (13-b).

(13) a. My partner is on a train.
b. $\lambda x.(On(\epsilon, y, Train(y))(x))(Robert)$.

For common noun predicates in English, the story is complicated by the appearance of the indefinite article with singulars, but the basic analysis holds under the (normal) assumption that such phrases should be analysed as one-place predicates rather than terms.

## 5 Non-predicative Constructions

Interestingly, equatives and specificational sentences can be analysed in a similar fashion to the predicative and expletive constructions already discussed. Both types of clause necessarily involve a definite noun phrases, either before or after the copula (or both). A copular clause without a definite cannot be easily interpreted as an equative (e.g. (14-d).

(14) a. John is the teacher.
b. That student over there is the best in the class.
c. A PhD student is the lecturer for this course.
d. A plant is a gift for life.

As discussed in Section 2, in Dynamic Syntax definite noun phrases are analysed as projecting metavariables with associated presuppositions whose value must be sought in context. It is this property that is used here to account for the equative and specificational constructions without having to postulate an ambiguous argument structure for the copula.

### 5.1 Specificational Clauses

Specificational sentences have a definite noun phrase in subject position which allows an analysis very like that for pure expletives. Consider the short text in (15):

(15) A: Who drank the last of the milk?
B: The culprit is John.

Here, B provides information to A by producing a definite noun phrase whose referent A does not know and supplies that referent by the postcopular noun phrase. As noted in section 2, definite noun phrases are treated as projecting a metavariable with an associated presupposition, so the culprit is represented as $\mathbf{U}_{CULPRIT}$. Substitution is optional (like all rules in DS) and so the content of the metavariable does not have to be identified as soon as the definite is parsed. In this case, because A does not know what term to substitute, he can simply choose to proceed with parsing the next word, be. As we have seen, parsing the copula completes the type requirements for a propositional tree and so the propositional structure is compiled to yield a tree with formula value $\mathbf{BE}(\mathbf{U}_{CULPRIT})$, i.e. one that is formula-incomplete on both subject and predicate nodes. An application of Final *Adjunction yields an unfixed tree of type $e$ which permits the parsing of John. The final unfixed node has to find a position in the tree and so merges with the subject node, providing the value for the metavariable in the process as shown in Figure 9. The formula value after merge has applied is thus $\mathbf{BE}(John_{CULPRIT})$.

The value of the metavariable, **BE**, is still to be established, however,. This, like all other values for metavariables, may be freely identified in context. However, pragmatic constraints on accessibility and parsimony operate on this selection process and, in particular, in conformity with the Principle of Relevance, the most accessible predicate in the context that yields inferential effects should be chosen as value (Sperber and Wilson 1986/1995, Carston 2002). I cannot here go into details about accessibility, but clearly the context must provide amongst the set of accessible concepts also those concepts that are introduced as part of the parsing process, i.e. through the lexical items that appear in some string, all such concepts being more accessible than anything that is part of the external 'context'[11]. Assuming this, there will be an effect of strictly local, linguistic context on the substitution process. In the current case, the predicate concept provided by the presupposition associated with the definite noun phrase is thus the most accessible in the domain. It is also highly informative because it has not already been 'used' (to identify some referent) and because it provides the answer to A's question. By the tenets of Relevance Theory, this must be chosen as the substituend for the predicate metavariable projected by be, to yield $Culprit(John)$ as the final interpretation.

The formula that results from this sort of specificational sentence is, of course, exactly what results from parsing the similar predicative sentence (John is a culprit in this case). However, the process of establishing this content differs in both cases, yielding different informational effects. Thus, in parsing John is a culprit, we identify the subject term *John* and predicate the property of being a culprit to this. With the specificational sentence The culprit is John, however, the unfixed node provides the value for the metavariable in subject position and the relevant property that should be predicated of this term is derived inferentially from the information contained in the presupposition associated with the node. The effect is that in the latter construction the value of some initial

[11]Which may be characterised as, $\Delta$, a database of propositions and terms reflecting information assumed to be accessible (mutually manifest) to the interlocutors

underspecified term is provided , while the former simply predicates a property of some identified element. It is thus the process whereby such strings are parsed that gives rise to the different interpretations, not the propositional content of the string itself.

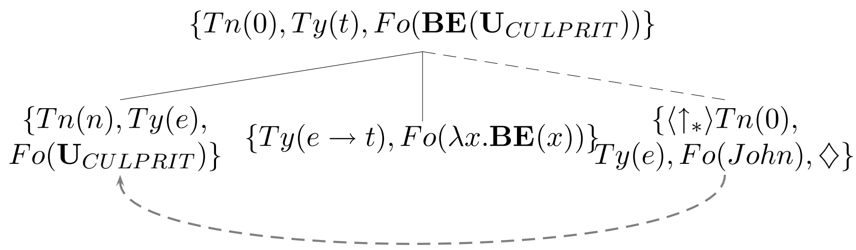


Figure 9: Completing The culprit is John

5.2 Equative Clauses

The analysis of equatives, although similar, is more complex for two reasons: firstly, the content of the definite is not provided through the merge process, but gives rise to a reading of identity between two terms; and secondly, the content of the underspecified predicate has to be determined through non-local context. To see how the analysis works consider how a sentence like (1-a) is parsed in the context provided in (16).

(16) A: I hear John has finally got a lecturing job:
B: Yeah, I'm taking a course in Applied Knitting at Langshanks College and John is the teacher.

In interpreting the relevant clause, the hearer, A, has access to John as the topic of the conversation, infers the existence of a teacher of Applied Knitting from the typical properties of courses taught in Colleges and then identifies John with the inferred teacher. The information so gained is then used to infer that John has indeed got a lecturing job.

The parsing process proceeds as follows. The first two words to give rise to an initial type-complete propositional structure which is compiled to give a tree with formula value $Fo(\mathbf{BE}(John))$. Final *Adjunction allows the postulation of an unfixed node of type $e$ which allows the post-verbal noun phrase to be parsed, as illustrated in Figure 10. The post-copular noun phrase has, as formula value, a metavariable with a presupposition that whatever substitutes for the metavariable must be a teacher: $Fo(\mathbf{U}_{TEACHER})$. Unlike the specificational case, a referent for this definite can be reconstructed from context by the hearer, since there is an already mentioned teacher in the context: the person teaching Applied Knitting at Langshanks College. A thus substitutes the epsilon term that picks out whatever entity is teaching this course, i.e. $(\epsilon, x, Teach(c)(x))$ where $c$ stands for the

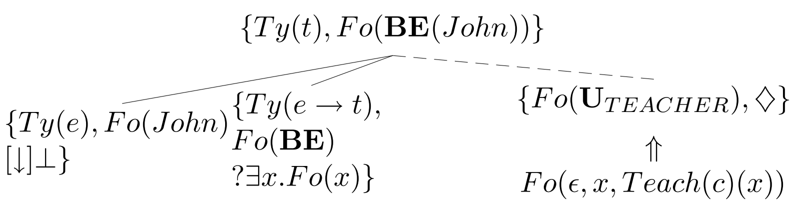


Figure 10: Parsing John is the teacher

content of a course in Applied Knitting at Langshanks College, which, by the relation between the verb teach and the noun teacher, satisfies the presupposition.

The problem at this point is that merging the right unfixed node with the subject yields two distinct formula values decorating the same node, i.e. $Fo(\epsilon, x, Teach(c)(x))$ and $Fo(John)$. Multiple formulae on a single node are not permitted unless one of the formula values decorating a node is less informative than the other (e.g. because it is a metavariable), i.e. just in case $Fo(\alpha) \leq Fo(\beta)$. However, we could modify this view and adopt the position that a node may host multiple formula values just in case they can be assigned an identical denotation. With respect to the combination of a metavariable plus some other value of appropriate type, this reduces just to the value of the second formula. However, with two contentful formulae, some new combined formula should be derived, one that denotes a single object. Although such a semantic condition might seem to be at odds with the representationalist spirit of Dynamic Syntax, we may exploit the properties of the epsilon calculus to provide a straightforward way of incorporating this idea into the representation system. Thus, if a node is decorated with two distinct epsilon terms, $\epsilon, x, P(x)$ and $\epsilon, x, Q(x)$, the assumption that they must both denote the same entity (whatever that may be) means that $Q(\epsilon, x, Q(x))$ and $Q(\epsilon, x, P(x))$ both have the same truth value, i.e. the witness for $P$ is a witness for $Q$ (and vice versa). A proposition $Q(\epsilon, x, P(x))$ licenses the construction of a term $(\epsilon, x, P(x) \wedge Q(x))$ to pick out the witness of the two predicates $P$ and $Q$, thus allowing us to resolve the two formulae into one. This step allows a node which contains two epsilon terms to be resolved into a single term, picking out the witness for both sets, which is necessarily a single object.

If we now adopt the position that proper names are in fact disguised epsilon terms (therefore being interpreted as named John), we may provide a coherent interpretation for the node decorated with the two terms, $Fo(\epsilon, x, John(x))$ and $Fo(\epsilon, x, Teach(c)(x))$, i.e. $Fo(\epsilon, x, John(x) \wedge Teach(c)(x))$. With this substitution for **U** in Figure 10, the resulting propositional formula for the tree thus becomes:

(17) $\mathbf{BE}((\epsilon, x, John(x) \wedge Teach(c)(x))_{TEACHER})$.

Because of the pro-predicate nature of **BE** there must be some salient property associated with John which may be substituted for it. The most accessible predicate, as discussed above, is that provided by the presupposition of the definite noun phrase, i.e. $Teacher$. In this case, however, this is not informative to A, as it was used as the condition for identifying the term substituting for the metavariable. Some other term must, therefore, be found in context that is relatively accessible and yields suffiecient information. There is another predicate that is recoverable and informative in (16): the predicate representing the content of has a new lecturing job, $\lambda x.Have(\epsilon, y, New - Job(y))(x)$. The final output of the computation is thus as in (18)

(18) $Have(\epsilon, y, New - Job(y))(\epsilon, x, John(x) \wedge Teach(c)(x))$

Notice that because indefinites construct full trees with true terminal nodes decorated with contentive formulae, they will not license specificational or equative analyses of this form because there will be no node in the subject tree with which a final unfixed tree could be coherently merged. Thus, even though an indefinite may be represented as an epsilon term, the fact that it does this explicitly and not through substitution for a metavariable (which is not associated with a bottom restriction), equative readings are never available for indefinite subjects. Examples like A culprit is John must therefore be analysed as a real inverted copular sentence, a position that is supported by the agreement facts, where the verb agrees with the postcopular expression, not the initial one.

(19) a. A genius are you.
b. *A genius is you.
c. A fool am I.
d. *A fool is I.
e. ?A fool is me.

It may be objected that a theory of equatives that depends on a distinct representation of the content of definite and indefinite noun phrases is not desirable, given the fact that many languages do not have articles that encode definiteness and that definite and indefinite noun phrases have more or less identical syntactic properties. In dynamic-type semantic theories such as Discourse Representation Theory (Kamp and Reyle 1993) and Dynamic Predicate Logic (Chierchia 1995), it is commonplace to treat definites the same as indefinites except for the way the bound variable/discourse referent is treated with respect to its relation to others: indefinites are not identified with discourse salient referents, while definites are. Similarly, Egli and von Heusinger (1995) treat all definite and indefinite noun phrases as projecting epsilon terms, the differences being ascribed to the choice functions that map the relevant sets onto actual objects taken to be appropriate witnesses: the choice functions associated with definites map sets onto known (salient) objects, whereas indefinites do not. However, it is precisely the representational difference posited here that gives rise to the difference in interpretation between definites and indefinites. The fact that the former project metavariables means that inferential processes are invoked to construct or identify and appropriate substituend from the current context (which includes the common ground shared by interlocutors). This necessarily gives rise to the ‘familiarity condition’ (Heim 1982) associated with these expressions. On the other hand, indefinites simply project a quantificational structure where the restrictor, and thus the epsilon term picking out its witness, is defined by the parsing of lexical expressions. The requirement within Dynamic syntax that such expressions introduce fresh variables (Kempson et al. 2001:238), inevitably gives rise to the ‘novelty condition’ of indefinites. The constraints on the interpretation of these two types of expression thus fall out naturally from the different forms of representation given to them within a theory that permits inferential processes to interact with syntactic ones.

## 6 Conclusion

In this paper, I have presented an analysis of the English copular verb, be, that treats it uniformly as a one-place predicate with underspecified content. Within the framework of Dynamic Syntax, this underspecification is represented as the projection of a metavariable whose actual value must be derived pragmatically from the context in which the copular clause is uttered. This context involves both external and local linguistic content and the latter determines to a large degree whether the copular clause is interpreted as a predicative, equative or specificational. No type differences either for the copula or its associates are invoked to distinguish these constructions and the only additional assumption that is required is that merge may be licit if it gives rise to a node decorated with two distinct formulae that can be combined to create a formula with a single denotation. This assumption is, however, a natural one with potential for analysing certain relative clause constructions. The success of this style of analysis supports the dynamic view of utterance interpretation and the benefits of move away from static models of autonomous syntax.

## 7 References

Asher, N (1993) Reference to Abstract Objects in Discourse. Dordrecht, Kluwer.
Blackburn, P and W Meyer-Viol (1994) 'Linguistics, logic and finite trees'. Bulletin of Interest Group of Pure and applied Logics 2:2-39.
Cann, R. (2001) How pronominal are expletive pronouns? In Dekker, P. and van Rooy, R. (eds.) Proceedings of the 13th Amsterdam Colloquium, 55-60.
Cann, R (2002) Intransitive 'be' in the grammar of English. Paper presented at the workshop Existence: Semantics and Syntax, Université de Nancy 2, September 2002. (http://www.ling.ed.ac.uk/ ronnie)
Cann, R, T Kaplan and R Kempson (2002) Data at the Grammar-Pragmatics Interface: the case of resumptive pronouns in English. MS University of Edinburgh, Blackwell Publishers, Kings College London.
Cann, R., Kempson, R. and Otsuka, M. (2002) 'On left and right dislocation: a dynamic perspective'. MS University of Edinburgh/Kings College London.
Carston, R. (2002) Thoughts and Utterances: the pragmatics of explicit communication. Oxford, Basil Blackwell.
Chierchia, G (1995) Dynamics of Meaning. Chicago, Chicago University Press.
Chomsky, N (1981) Lectures on Government and Binding. Dordrecht, Foris Publications.
Egli, U and K von Heusinger (1995) 'The epsilon operator and E-type pronouns'. In U egli et al. (eds.) Lexical Knowledge in the Organization of Language. Amsterdam, Benjamins: 121-141.
Geist, L (2002) Predication and Equation: Two BE's or not two BE's. Paper presented at the workshop Existence: Semantics and Syntax, Université de Nancy 2, September 2002.
Heim, I (1982) The semantics of Definite and Indefinite Noun Phrases. New York, Garland Press.
Hilbert, D and Bernays, P (1939) Grundlagen der Mathematik II. Berlin, Julius Springer.
Kamp, H and U Reyle (1993) From Discourse to Logic. Dordrecht, Kluwer.
Kempson, R, W Meyer-Viol and D Gabbay (2001) Dynamic Syntax. Oxford, Basil Blackwell.
Kim, J (1969) 'Events and their descriptions: some considerations'.In N Rescher et al. (eds.) Essays in Honor of Carl G. Hempel. Dordrecht, Reidel: 198-215.
Kim, J (1976) 'Events as property exemplifications'. In M Brand and D Walton (eds.) Action theory. Dordrecht, Reidel: 159-177.
Maienborn, C (2002) 'Against a davidsonian analysis of copula sentences'. Paper presented at the workshop Existence: Semantics and Syntax, Université de Nancy 2, September 2002.
Meyer-Viol, W (1995) Instantial Logic. Doctoral dissertation, University of Utrecht.
Mikkelsen, L (2002)
Montague, R (1973)
Partee, B H (1986) Ambiguous pseudoclefts with unambiguous be. NELS 16: 354-366.
Sperber, D and D Wilson (1989/1995) Relevance. Oxford, Basil Blackwell.
Williams, E (1983) Semantic vs. syntactic categories. Linguistics and Philosophy 6: 423-446.