Semantic underspecification and the pragmatic interpretation of be

Semantic Underspecification and the Interpretation of Copular Clauses in English

Ronnie Cann
University of Edinburgh

July 2004

Abstract

This paper presents an analysis of basic predicative, equative and specificational constructions in English in which the interpretation of *be* is taken to depend on the properties of the expressions with which it combines. Construing the copula as projecting underspecified semantic content within the framework of Dynamic Syntax, provides the basis of an account of these constructions in a combination of pragmatic and syntactic processes interact to determine the interpretive content of the copula in the context in which it appears.

1 Introduction

The problem with analysing the verb *be* in English (and the copular verb in many other languages) is that it appears in a wide range of constructions which apparently involve complements of different sorts and which show a variety of interpretations.[1] The content of *be* itself appears to vary from apparently nothing, through concepts of identity and specification, to existential and locative interpretations. These differences in interpretation depend crucially on the expressions with which the verb appears. Thus, whenever the postcopular expression is predicative, the content of *be* appears to yield little more than providing tense information, (1).[2]

(1) a. John was foolish.
b. That student is a violinist.
c. Every pet is in the house.

[1]I am grateful to many discussions with Ruth Kempson, with whom many of the ideas in this paper were worked through; to Caroline Heycock for inspiring me to pursue the topic; and to conversations with Lutz Marten, Virve Vihman, Dan Wedgwood, Yicheng Wu, and Stavros Assimakopoulos. I am also grateful to the Edinburgh Syntax and Semantics Research Group, the King College Dynamic Syntax Group and the audiences at the *Existence* workshop in Nancy and the conference *Where Semantics Meets Pragmatics* at the University of Michigan for comments on earlier talks that covered some of the material presented in this paper.

[2]This also appears to be the case with the grammaticalised uses of *be* in passive and progressive constructions which are not considered in this paper.

Where both subject and postcopular expressions are 'referential' DPs, the interpretation is either equative (2a) when both are construed as referential or specificational where the subject DP is an ordinary definite noun phrase that can be construed as a description of an unknown entity, rather than as picking out some specific object.[3] (2b) or it may be ambiguous between these (2c).

(2) a. Mary is the dancer.
 b. The culprit is John.
 c. The murderer is the person who had opportunity.

With the expletive *there* and a weak NP associate, the verb appears to provide an existential reading (3b) while with a definite NP associate[4] there tends to be a presentational or locative reading (3b).

(3) a. There is a riot on Princes Street.
 b. There's the student you wanted to see.

The existential interpretation appears also in certain uses of the copular without any apparent complement. So we have examples of what may be termed the *existential focus* construction illustrated in (4a), where the verb simply seems to assert the existence of the subject. Such interpretations, though rare, are seen in a number of famous expressions such as those in (4b,c).

(4) a. Neuroses just ARE (they don't need a cause)
 b. I think therefore I am.
 c. To be or not to be.

There have been many attempts to reconcile these different interpretations of copular constructions and so reduce the apparent homonymy associated with *be*. However, the fact that the interpretation of a clause containing *be* may alter according to the expressions with which it appears, indicates that it is crucially dependent on context for its meaning. Thus, the interpretation of *there be* sentences as presentational or existential seems to be attributable to the definiteness of the post-copular associate, as 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), it must be the case that the form of the whole clause contributes to the interpretation. In other words, the interpretation of copular clauses depends on inference in context and should be analysed pragmatically rather than semantically. This is the approach to be taken in this paper in which an essentially anaphoric/expletive story of the copula is provided which relies on a process of pragmatic strengthening, as well as syntactic processes, to account for the different readings of the verb *be* in predicative, equative and specificational clauses.

[3]See Heycock 1994, Heycock and Kroch 1999, Mikkelsen 2002, etc.
[4]A term often used for the postcopular noun phrase in a *there be* construction.

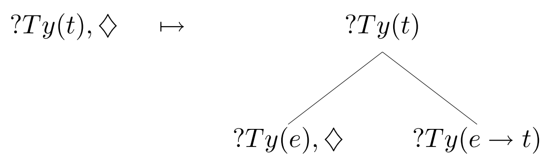
2 Dynamic Syntax

The framework to be used is that of *Dynamic Syntax* (Kempson et al 2001) which models the process of natural language understanding as the monotonic growth of trees representing the semantic content of some string of words uttered in context. The process is goal-driven, beginning with the initial, universal requirement to establish propositional content for some utterance. Such content is represented in terms of binary trees establishing the argument structure of a proposition as it is built up incrementally through general construction rules, information provided by the words in some string, and pragmatic processes of enrichment. Intrinsic to this process of building up content are concepts of underspecification whose resolution is driven by requirements (goals and subgoals). For the purposes of this paper, a central role is given to the underspecification of semantic content and of the argument status of some element within an emerging propositional structure.

Nodes in trees are decorated with labels specifying (amongst other things) the type of the node (label Ty[5]), its semantic content shown as a lambda expression (Fo) and an address specifying where in the tree the node is (Tn, see below for details). 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 of certain types, starting from $?Ty(t)$, 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 are (by assumption) universal (although language specific conditions may be imposed on such rules). 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 $?Ty(X \to Y)$. Thus, the initial unfolding of a requirement $?Ty(t)$ may be to establish subgoals $?Ty(e)$ and $?Ty(e \to t)$, requirements to build the subject and predicate nodes, respectively, as shown in (5).[6] The diamond, $\Diamond$, in the tree diagrams indicates which node is under development.

(5) An initial expansion of $?Ty(t)$



Information about tree building also comes from the 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

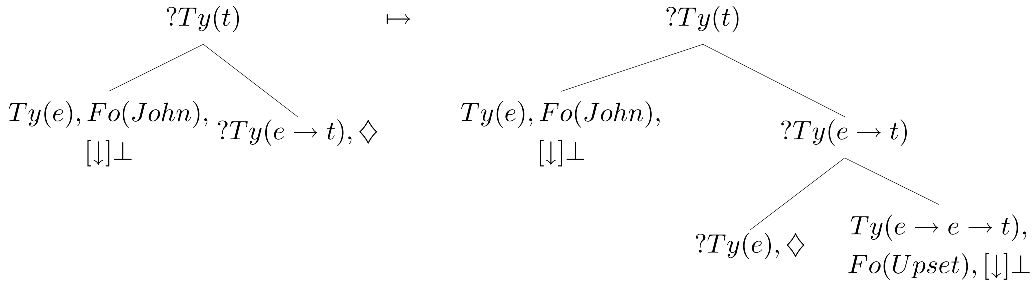
[5]DS uses only a restricted set of types: e the type of a term, t the type of a proposition, cn the type of a common noun, $e \to t$, the type of a (one-place) predicate and higher arities of predicates. The theory eschews the use of type-altering operations.

[6]To simplify the exposition, I do not give the formal definitions of the rules in this paper. See Kempson et al. (2001:ch. 3) and Kempson, Cann and Marten fcmg ch. 2 and passim for details.

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* gives rise to the set of actions in (6) which annotate the current node with formula and type values[7]. Parsing the verb *upset*, on the other hand, gives rise to a more complex set of actions that build and annotate nodes and the imposition of an additional requirement to construct a representation of the content of an object DP as illustrated in (7).[8]

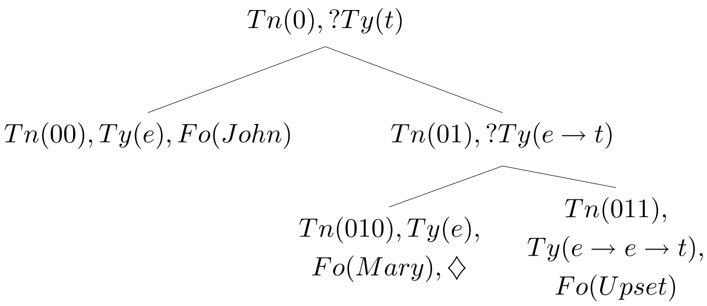
		IF	$Ty(e)$	Trigger
(6)	*John*	THEN	$put(Ty(e), Fo(John), [\downarrow]\bot)$	Actions
		ELSE	ABORT	Failure

(7) Parsing *John upset*



The parse of a string will continue just in case the next word has a trigger of the appropriate type, i.e. $?Ty(e)$. A string like *John upset Mary* will thus give rise to the tree in (9) with all terminal nodes now type and formula complete. The remaining type 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 (9).

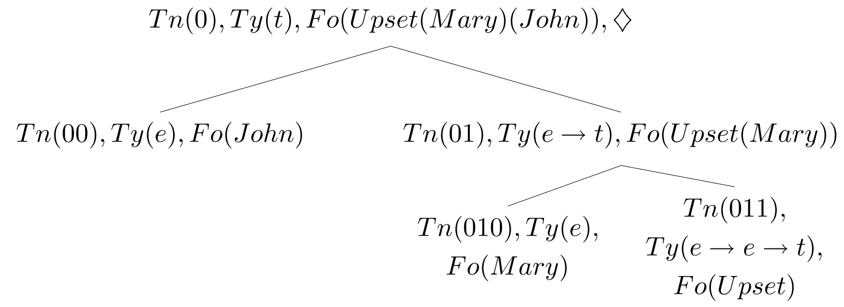
(8) Parsing *John upset Mary*



[7]Although see below for a revised view of the actions associated with proper names.

[8]Here and below, all tense information is ignored as not germane to the current discussion. See section 2.2 for discussion of the modality $[\downarrow]\bot$, which marks a node as terminal.

(9) Completing *John upset Mary*



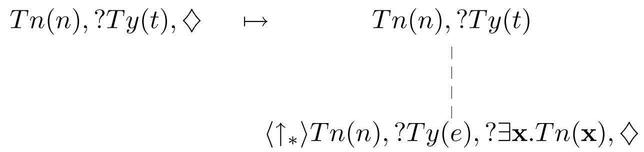
2.1 Left Dislocation

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 the requirement to find a fixed position within a tree. Every node in a tree is associated with an ADDRESS which is encoded as a value to 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 $Tn(n1)$ while the argument daughter has an address $Tn(n0)$. In (9), for example, the node labelled by $Fo(John)$ has an address of $Tn(00)$ while that decorated with $Fo(Upset)$ has $Tn(011)$ and so on.

Annotations of nodes derived through construction rules or lexical actions are expressed using the Logic of Finite Trees (LOFT, Blackburn and Meyer-Viol 1994) which 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, $\uparrow$. This logical apparatus allows an expression to project structures that are within some tree but not yet assigned fixed position within it. Instead, a node is annotated as having an underspecified dominance relation with respect to some other node, shown by the modality $\langle\uparrow_*\rangle$, with a requirement to find a fixed position within the tree, represented as $?\exists\mathbf{x}.Tn(\mathbf{x})$. 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 introduces unfixed nodes, defining a transition from an incomplete tree rooted in $?Ty(t)$ 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.[9] The transition induced by this rule is illustrated in (10).

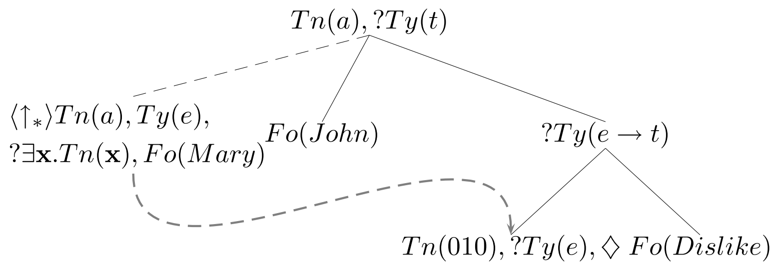
[9]The modality $\langle\uparrow_*\rangle$ is defined as: $\langle\uparrow_*\rangle\alpha =_{def} \langle\uparrow\rangle\alpha \vee \langle\uparrow\rangle\langle\uparrow_*\rangle\alpha$.

(10) *Adjunction



Analysing the string *Mary, John dislikes* in these terms is illustrated in (11) with an initially projected unfixed node and the pointer at the object position. At the point in the parse at which all words in the string have been processed, there remains outstanding an unfixed node and a requirement to construct a node of type e. In this environment, a process of MERGE 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 case no contradictory decorations result.[10] The MERGE 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. The Merge process is indicated by the dashed line in (11) and below.

(11) Parsing *Mary, John dislikes*



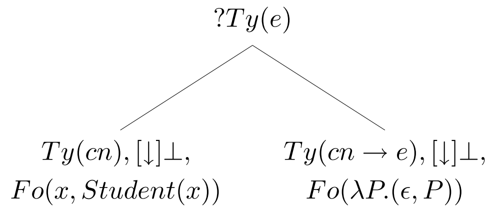
2.2 Analysing Basic Noun Phrases

In *Dynamic Syntax*, all noun phrases translate into expressions of type e. This is made possible in part by the use of the epsilon calculus of Hilbert and Bernays (1939) where indefinite noun phrases, for example, project epsilon terms, expressions that denote arbitrary witnesses for some property (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 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 containing 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 (12) shows the structure projected on parsing the indefinite noun phrase *a student* which yields a formula

[10]Well-formed treenode descriptions are thus rather like the categories of Generalised Phrase Structure Grammar which are defined as partial functions from attributes to values (Gazdar et al. 1985).

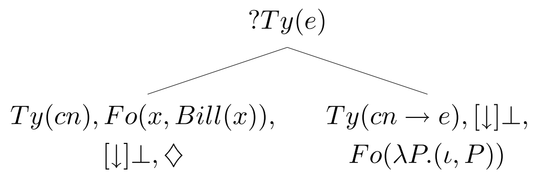
$(\epsilon, x, Student(x))$ when compiled and completed.[11]

(12) Parsing *a student*



Like indefinites, proper names may be treated as projecting full structure, in this case as iota terms, where an iota term is construed here as an epsilon term with an associated unique choice function that picks out only that object identified by the name. The result of parsing a name like *Bill* is given in (13) (although in the discussion below this structure will not in general be shown).

(13) Parsing *Bill*



Unlike indefinites and proper names, which project full tree structures with fully specified content, pronouns in Dynamic Syntax, while still projecting an expression of type e, provide only underspecified content, reflecting the fact that the processing of anaphors is context dependent. Within DS, such underspecification of content is analysed by means of the projection of a METAVARIABLE, a placeholder for a formula that requires to be replaced by some selected term during the parsing process. Such replacement is associated with a substitution process that is pragmatic, and system-external, restricted by locality considerations (such as analogues of the Binding Principles, Chomsky 1981, etc.) and by lexical presuppositions (such as gender).

(14) Q: Who upset John?
Ans: Mary upset him.

In processing the pronoun *him* in (14), the object node is first decorated with a metavariable $\mathbf{U}$, and an associated requirement, $?\exists \mathbf{x}.Fo(\mathbf{x})$, which can only be fulfilled by the identification of some contentful value of the formula label. The relevant actions are shown in (15): on the trigger of a requirement for an expression of type e, a node is annotated with a metavariable, a $Ty(e)$ label, the 'bottom restriction' ($[\downarrow]\bot$) and a requirement to find the content of the formula.[12]

[11] In fact, there is further structure under the $Ty(cn)$ node that I have omitted for expository purposes. See Kempson et al. 2001 ch. 4 for details.

[12] Other information is also projected such as positional restrictions determined by case and by an analogue of Principle B on 'binding'. This is omitted as irrelevant to the discussion.

(15) *him*

IF	$?Ty(e)$	
THEN	$put(Fo(\mathbf{U}_{Male(\mathbf{U})}),$	Metavariable
	$Ty(e),$	Type
	$?\exists\mathbf{x}.Fo(\mathbf{x}),$	Formula requirement
	$[\downarrow]\bot)$	'Bottom restriction'
ELSE	ABORT	

The modality, $[\downarrow]\bot$, has an important function within Dynamic Syntax. A node so annotated may not dominate any other material since the modality requires that no properties hold of any node below it (i.e. 'necessarily below this node nothing holds'). It thus prevents further elaboration of that node, ensuring that pronouns behave, in English, like contentive expressions in that they must decorate a 'terminal node' on a tree. This has an effect in preventing dislocated expressions from being associated with a position labelled with a pronoun by the process of MERGE, hence the ungrammaticality of the examples in (16).

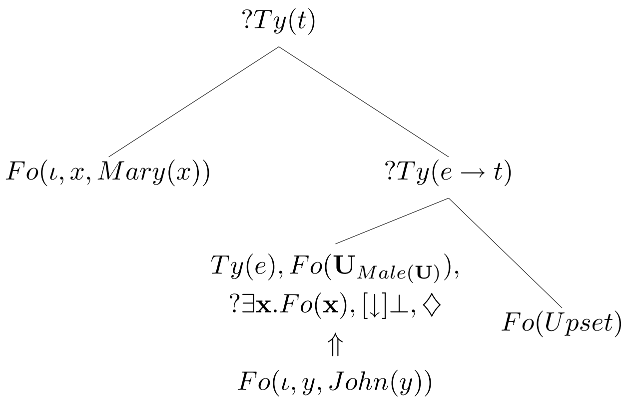
(16) a. *Much beer, I like it, but many fizzy drinks, I detest them.
(cf. Much beer, I like, but many fizzy drinks, I detest.)

b. *What did you see it?
(cf. What did you see?)

Pronouns may also come with restrictions on the content of expressions that replace them. Thus, *him* requires to be identified with a referent that is male. Kempson et al. (2001) display such presuppositions as annotations on a metavariable, yielding such formula representations for pronouns like *him* as $Fo(\mathbf{U}_{Male(\mathbf{U})})$. The function of such 'presuppositions' is to act as a constraint on the process of SUBSTITUTION: the property associated with a metavariable guides the hearer towards a relevant choice of term as substituend. The substitution of $Fo(John)$ rather than (say) $Fo(Jane)$ for the metavariable in (14) is supported by the fact that John is assumed generally to be a name for a male while Jane is not. The fact that the pronoun *him* could be used to refer to Jane (or some other female) in a different context (e.g. because Jane is dressed as a man) 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 male would not, in such circumstances, cash out truth conditionally as a property of whatever term is substituted for the metavariable: the presupposition is a constraint on a pragmatic process, not an assertion that some property holds of some particular term. The result of parsing *Mary upset him* in the context provided in (14) is shown in (18b) with substitution shown by the symbol $\Uparrow$. After substitution the information that the string contained a pronoun is entirely lost in the representation, yielding the final propositional formula in (18a).

(17) $Fo(Upset(\iota, y, John(y))(\iota, x, Mary(x)))$.

(18) Parsing *Mary upset him* - substitution and completion



Definite noun phrases are treated analogously to pronouns in Dynamic Syntax in projecting underspecified content which requires to be enriched. However, the presuppositional content of such expressions is not projected from the lexicon, as part of the actions associated with parsing *the*, but comes from the information contained in the common noun phrase associated with the definite article. Thus, the formula projected by a phrase like *the man* can be represented as $Fo(\mathbf{U}_{Man(\mathbf{U})})$ while that associated with *the student with red hair* may be represented as (something like) $Fo(\mathbf{U}_{Student(\mathbf{U}) \wedge With\text{-}Red\text{-}Hair(\mathbf{U})})$. It is possible to provide a compositional and monotonic account of definite noun phrases using the DS concept of LINKed structures (see Kempson et al. 2001:110-120), but as the formal analysis is not germane to the discussion of copular constructions, the full analysis is not provided (see Cann to appear, for details). The analysis involves the definite article as projecting a metavariable like a pronoun[13], but additionally induces the construction of a presupposition from the content of the common noun phrase. Like pronouns in english, the definite article also projects the bottom restriction in order to disallow such strings as **Who did you think Jane saw the man?*.

As has already been seen with respect to pronouns, the effect of a 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 (19).

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

Here, the first sentence provides the context for interpreting the definite NP in the second. So we have (something like) $\exists x.Stole(PDA)(x)$ as the formula value for the former. Parsing the definite NP in the latter yields the underspecified formula $Fo(\mathbf{U}_{Culprit(\mathbf{U})})$ which requires the identification of some contextually salient term that also satisfies the property of being a culprit. There are two possible choices of substituend at this point (assuming no other contextually salient terms): the term $Mary$ and the epsilon term signifying the arbitrary individual who stole Mary's PDA, i.e. $(\epsilon, x, Steal(PDA)(x))$. Since

[13] A reflection of the diachronic development of the definite article from a demonstrative pronoun.

someone who steals may be described as a culprit (stealing being a form of wrongdoing and culprits being wrongdoers), and Mary is (in normal circumstances) not likely to have stolen her own PDA, the only relevant choice of term in this context is the epsilon term which is duly substituted for the metavariable to give rise to the formula value in (20a). Since the presupposition is satisfied through the lexical semantics of *steal* over its subject argument, it can be cashed out as an entailment as in (20b) but, because the presupposition has been fully discharged, the informational content is just that in (20c).[14]

(20) a. $\textit{Get-Away}(\epsilon, x, Stole(PDA)(x))_{Culprit(\epsilon,x,Stole(PDA)(x))}$.
b. $\textit{Get-Away}(\epsilon, x, Stole(PDA)(x)) \wedge Culprit(\epsilon, x, Stole(PDA)(x))$.
c. $\textit{Get-Away}(\epsilon, x, Stole(PDA)(x))$.

2.3 Expletives in Dynamic Syntax

Although pronouns in English are typically associated with the bottom restriction that prevents them from being directly substituted by the content of some dislocated term, there are pronouns that are systematically associated with material that occurs elsewhere in a string. Amongst these is the expletive pronoun *it* in English. In the *it-extraposition* construction in English, illustrated in (21), for example, a subject expletive is associated with a postverbal finite clause.

(21) a. It is likely that I will resign.
b. It was announced that the dean had resigned.

The pronoun *it* in (21b) is not 'referential', taking its value from the context in which the string is uttered, but expletive in that it takes its content from the postverbal expression. Expletive *it* thus appears to provide a placeholder that is subsequently replaced by some propositional formula.

In non-pro-drop languages such as the Germanic languages, lexicalised expletives are essential in such constructions. Without them, the parsing sequence breaks down, because the pointer cannot move on from the subject node without lexical input of the appropriate type. This follows because the trigger for (lexical) verbs is a predicate type requirement ($?Ty(e \rightarrow t)$), not a propositional one, and the verb does not annotate the subject node in any way.[15]

(22) *Is likely that I am confused.

The function of an expletive use of a pronoun, accordingly, is to keep the parsing process alive: it first provides a metavariable as an interim value to some type requirement associated with one node. In addition, the effect of parsing an expletive is also to move the pointer on to another node (the predicate node in this case) in order to preclude substitution, which may only occur at a node when it hosts the pointer. As the pointer is moved on as part of the actions determined by parsing the expletive pronoun, no

[14] I leave to one side a fuller discussion of the theory of presupposition adopted here, but it resembles in many ways that proposed for DRT in Kamp (2001).

[15] Unlike in pro-drop languages where verbs are analysed as supplying a placeholder for their subject term directly. See Kempson, Cann and Marten (fcmg) for discussion.

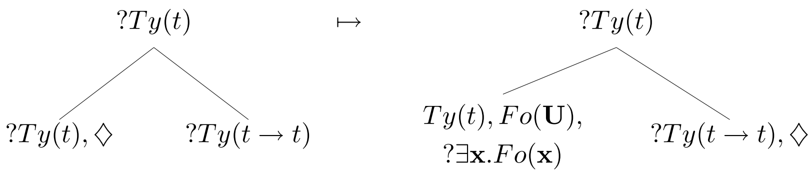
substitution can take place and an open formula requirement necessarily remains on the node decorated by the metavariable. Finally, the expletive pronoun fails to project the bottom restriction, thus permitting later growth of the tree below the node it decorates, structure which is projected by a right dislocated expression.

The definition of these actions is shown in (23) which assumes that $Ty(t)$ can decorate a subject node, and that certain predicates project a formula of type $t \to t$.[16] The effect of these lexical actions is to license the transition in (24).

(23) *It*

IF	$?Ty(t)$	
THEN	IF	$\langle\uparrow\rangle\bot$
	THEN	ABORT
	ELSE	$\mathtt{put}(Fo(\mathbf{U}), Ty(t), ?\exists x Fo(x))$,
		$\mathtt{go}(\langle\uparrow_0\rangle\langle\downarrow_1\rangle)$
ELSE	ABORT	

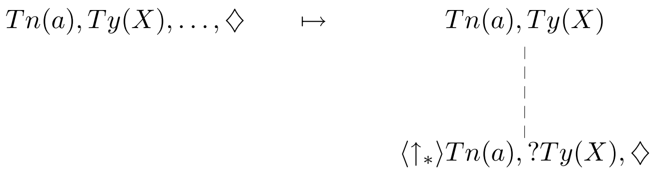
(24) Parsing *it*



Once the verb has been parsed and the predicate node decorated, the pointer moves to the mother node in order to complete the propositional type requirement. However, because the subject node still carries an unsatisfied formula requirement no evaluation can proceed and the pointer must move back down to the subject daughter in order to complete the requirements on this node. Since the node is type-complete, however, it looks as if the parse is doomed to failure. But it is at this point that a variant of *Adjunction, "Late *Adjunction", applies to provide an unfixed node with an open type requirement, allowing the parse of new material to take place.[17]

Unlike the version of *Adjunction briefly presented in Section 2.1, Late *Adjunction projects an unfixed node with a requirement for the same type as the node from which it is projected. Since no further direct development of the fixed node is possible, this version of *Adjunction defines directly the structural context to which Merge applies, i.e. the unfixed node and the fixed node from which it is projected. The effect of such a rule is shown in (25).

(25) Late*Adjunction

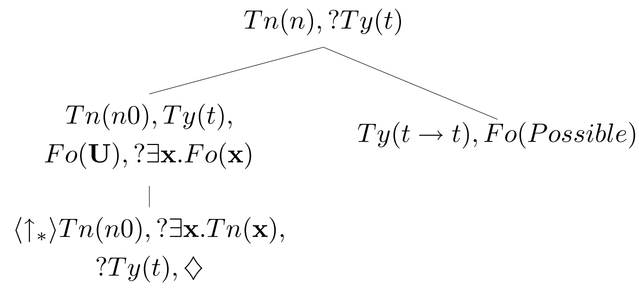


[16]Note the extra condition, $\langle\uparrow\rangle\bot$, which checks whether the current node is the topnode in a tree and aborts the parse if it is, thus preventing *it* from being the sole expression in a sentence.

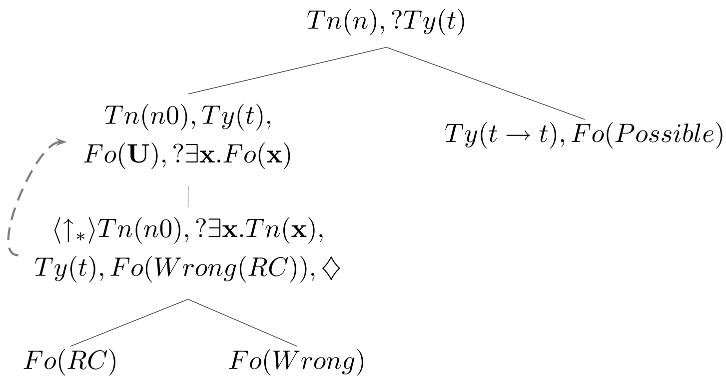
[17]This form of *Adjunction is used in Kempson, Cann and Marten fcmg to provide an account of Right Node Raising which is based on an earlier analysis in Cann, Kempson and Otsuka (2003).

Applying Late*Adjunction to the subject node in a parse of *It is possible* yields the configuration in (26).[18] This permits the parse of the post-verbal string and the completion of the unfixed propositional tree immediately feeds an application of Merge, as shown in (27), which yields a complete subject node and a final formula value $Fo(Possible(Wrong(RC)))$ as desired.[19]

(26) Parsing *It is possible*



(27) Parsing *It is possible that I am wrong*



3 Copula Clauses

In section 1, I suggested (as have many others) that the interpretation of *be* is dependent on context and particularly on the properties of the post-copular expression: when there is a definite noun phrase, we have an equative or specificational reading; when there is a predicative expression, a predicative interpretation; when there is no complement, an existential or elliptical interpretation follows. This context dependence points to the hypothesis that *be* projects underspecified content, the value of which is provided from the context in which it appears. Such underspecified content when associated with a pronoun is represented by a metavariable of type e, but metavariables may be postulated for any type and so it seems reasonable to hypothesize that the copula projects a *predicate*

[18]Ignoring the contribution of the copula (and tense).

[19]I treat the complementizer *that* as fully expletive in this context, i.e. as not providing any significant update to the emerging tree. Whether this is generally appropriate is a problem for another time.

metavariable of some sort, with an associated requirement to identify content, possibly through pragmatic inference.

Granted that *be* projects an underspecified predicate, the question still remains as to its arity. Unlike any other auxiliary (or main) verbs in English, *be* appears with 'complements' of every (non-finite) syntactic category apart from a full clause[20] and a bare verb phrase.

(28) a. Mary is a friend of mine — (predicative) NP
b. John is the teacher — (definite) NP
c. Lou will be happy one day — AP
d. A rabbit is in the garden — PP
e. The kids were playing football — (progressive) VP
f. Kim is disliked by Hannibal — (passive) VP
g. *I am play cricket — (bare) VP
h. I am to play cricket — (to) VP
i. *There was John to be in the bathroom — S

This flexibility of complement type is not matched by other auxiliary verbs, where complement categories are restricted to bare VPs, as illustrated with *can* in (29).

(29) a. *Mary can a friend of mine — NP
b. *John can the teacher — (definite) NP
c. *Lou can happy one day — AP
d. %A rabbit can in the garden — PP (ellipsis only)
e. *The kids can playing football — (progressive) VP
f. *Kim can disliked by Hannibal — (passive) VP
g. I can play cricket — (bare) VP
h. *I can to play cricket — (to) VP
i. *There can John to be in the bathroom. — S

A further difference between the copula and the modals is that the former allows construal of existence in a null context as illustrated in (30) while the latter, such as *may* and *can*, do not license interpretations where the general modality, such as possibility and ability, are ascribed to the subject. Without a complement VP, modals can only be interpreted elliptically, whereas, as we have already seen, *be* can give rise to a non-elliptical interpretation of existence in intransitive contexts.

(30) a. Neuroses just ARE. (= Neuroses exist)
b. Neuroses just MAY. ($\neq$ Neuroses are possible)
c. The students just CAN. ($\neq$ The students are able)

[20] Unless one analyses the associate plus coda existential constructions, such as *There was a man being sick outside* as small clauses.

These differences from the auxiliaries, the variability in apparent complement type and non-elliptical interpretation in intransitive contexts, is most easily accommodated by hypothesizing that *no* complement is required or indeed licensed by the copula and that *be* is uniformly a one-place predicate of type $e \rightarrow t$.[21]

3.1 The copula as an expletive

The assumptions that *be* is uniformly intransitive and projects underspecified content can be analysed in DS by having the copula project a metavariable of type $e \rightarrow t$ (shown as $Fo(\mathbf{BE})$), a predicate proform, parallel to pronouns.[22] Under this hypothesis, the machinery set up in the previous section, which is required to analyse constructions not involving the copula, is sufficient to provide a uniform account of *be* in predicative, equative and specificational contexts.

As *be* can have its content established directly within the same clause in predicative constructions, it appears that it has the characteristics of an expletive, as indeed it is treated in many frameworks, at least in its purely auxiliary function. The analysis of expletives presented in the last section rests on three properties:

- The projection of a metavariable to satisfy the type requirement;
- The lack of a bottom restriction, licensing merge with an unfixed node;
- The movement of the pointer away from the trigger node.

Treating *be* as an expletive gives rise to set of lexical actions given in (31) for all forms of the verb: triggered by a predicate requirement, the predicate node is annotated with the metavariable **BE** and a requirement for a formula value, and the pointer is moved up to the mother node.[23]

		IF	$?Ty(e \rightarrow t)$
(31)	*be*	THEN	$\mathtt{put}(Ty(e \rightarrow t), Fo(\mathbf{BE}), ?\exists \mathbf{x}.Fo(\mathbf{x})), \mathtt{go}(\langle\uparrow_1\rangle)$
		ELSE	ABORT

On parsing the copula, then, the value of the metavariable, **BE**, that it projects must be subsequently established, which, like all other values for metavariables, may be freely identified in context. This gives a direct way to account for ellipsis involving the copula, as illustrated in (32), the copula effectively acting as a free proform.

(32) a. John's really happy, John is.

[21] Lamarche 2003 comes to essentially the same conclusion, though for different reasons.

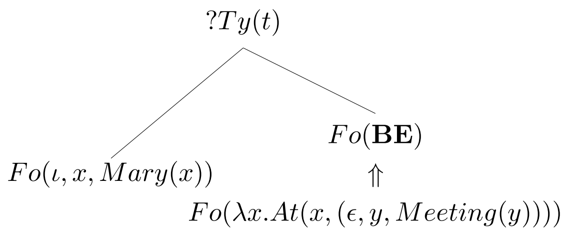
[22] It may be that the type of the copula has to be modified to allow for propositional and property subjects as exemplified in (i) and (ii):
i. That he will be here soon is highly unlikely.
ii. Honest is honest.
I do not explore these constructions here, but they do not undermine the essence of the current analysis. The important point here is that *be* does not project an internal argument, whatever the properties of its subject argument may be.

[23] As elsewhere, tense and agreement information is omitted for expository reasons. The latter can be given as a condition on the subject node to be third singular, but the technicalities introduce concepts that are orthogonal to current concerns.

b. A. Who was at the meeting?
   B. Mary was.

Under the assumption that *be* projects a metavariable, the elliptical utterances in (32) will be well-formed because the preceding utterance includes an accessible (and relevant) one place predicate which can substitute for the metavariable in the normal way. The situation resulting from parsing the second clause in (32b) for example is shown in (33) up to the point of substitution.[24] The resulting formula is, as required, $Fo(At(Mary, (\epsilon, y, Meeting(y))))$.[25]

(33) Parsing *Mary was*



Interestingly enough, this analysis also directly accounts for the possible interpretation of *be* as existential in the existential focus constructions illustrated in (4a) repeated below:

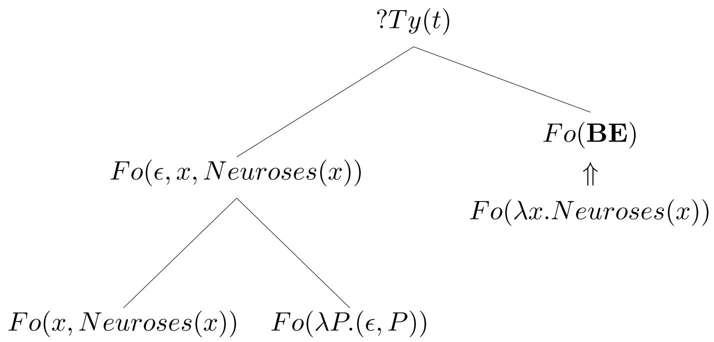
(4) a Neuroses just ARE.

In identifying the potential substituends for the predicate metavariable **BE**, the context also includes predicates derivable from the tree currently under construction. Thus, instead of identifying a predicate from the previous discourse, a hearer may construct one from the immediate context (the tree currently under construction) and substitute that for the predicate metavariable. In the tree constructed to parse (4a), the only available predicate is that derived from the common noun in the subject position, as illustrated in (34).

[24]From now on, trees will be simplified through the omission of completed types and irrelevant information. It should be stressed that while the trees that follow have nodes that are decorated only by formulae, this is not technically the case.

[25]It is not the case that just any predicate can associate with *be*, of course, but only stative predicates that are associated with non-verbal expressions.
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 (in the sense of Asher 1993) somewhere between world bound facts and spatio-temporally defined eventualities. This restriction may be achieved in DS by showing this as a presuppositional condition on substitution. However, I ignore the consequences of this move (which are significant for the interpretation of the progressive) in this paper.

(34) Parsing *Neuroses (just)* ARE



Making this substitution gives rise to the output formula in (35a) which, by the established equivalence in the epsilon calculus shown in (35b), gives rise to the existential statement in (35c).

(35) a. $Fo(Neuroses(\epsilon, x, Neuroses(x)))$

b. $F(\epsilon, x, F(x)) \leftrightarrow \exists x.F(x)$

c. $\exists x.Neuroses(x)$

While more needs to be said about the existential focus construction, especially with respect to the possibility of quantified subjects and the interaction with tense, it should be clear from this discussion that the treatment of *be* as projecting semantically underspecified content that may be pragmatically enriched provides a basis of a unified account of both ellipsis in copula clauses and existential focus readings, an unexpected result.

3.2 Predicative constructions

The analysis of *be* as a predicate expletive allows us to tackle the bewildering variety of copular constructions in English in a uniform manner, the burden of explanation shifting from considerations of the core 'meaning' of *be* as denoting existence, predication or identity to an account of inference in context that derives the expected interpretations of sentences.

In the elliptical case discussed above, the update for the predicate metavariable projected by *be* is determined through pragmatic substitution. However, there is a construction in which the appropriate predicate is supplied syntactically without the intervention of pragmatics. This is the basic predicative construction where a non-verbal predicate appears in postcopular position and *be* appears to be entirely meaningless (apart possibly for constraints on event type, see Rothstein 2001, Maienborn 2002, inter alia).

(36) a. John is happy.

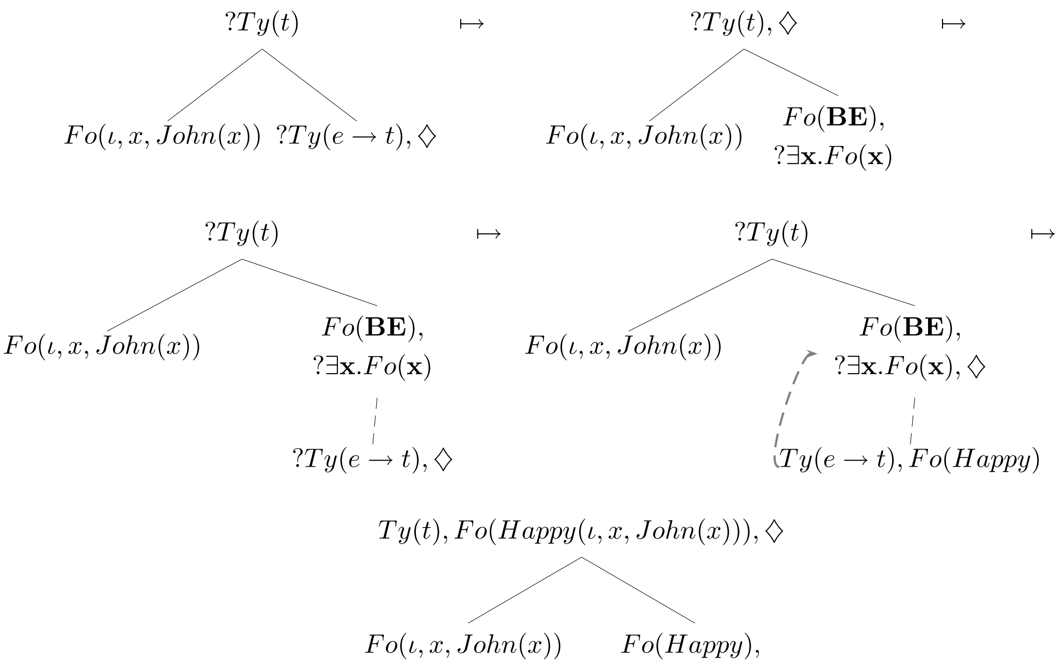
b. Robert was on a train.

c. Mary is a teacher.

We already have the machinery to analyse this construction straightforwardly. Note that the lexical entry for the copula in (31) does not write a bottom restriction to the predicate node, giving it one of the characteristic properties of an expletive. This allows us to use the same mechanism, Late*Adjunction, that we used to account for it-extraposition above, except that the unfixed node is projected from the predicate, and not the subject, node.

To see how the analysis works, consider the parse of (36a). The first two words project a subject-predicate structure in the normal way and the pointer is on the top node. At this point, all type requirements are fulfilled but there remains an outstanding formula requirement on the predicate node which prevents the tree from being completed. The pointer thus must move back to the incomplete predicate node, permitting an application of Late*Adjunction which provides an unfixed node with type requirement $?Ty(e \rightarrow t)$ (the rule is free with respect to the type of node). This 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 (37), from the parse of the initial word *John*, through the parsing of the copula, the unfolding of the unfixed node and the parse of the predicate to give the result in the final tree.

(37) Parsing *John is happy.*



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 (38a) gets the formula value in (38b).

(38) a. Robert was on a train.

b. $\lambda x.(On(\epsilon, y, Train(y))(x))(\iota, x, Robert(x))$.

For common noun predicates, in some languages such as Classical (and Modern) Greek, the nominal predicate may be treated directly as a predicate just like an adjective or a prepositional phrase and be analysed accordingly, the expression in (39a) giving rise to the formula value in (39b) through merge of the nominal predicate with the metavariable projected by the copula.

(39) a.

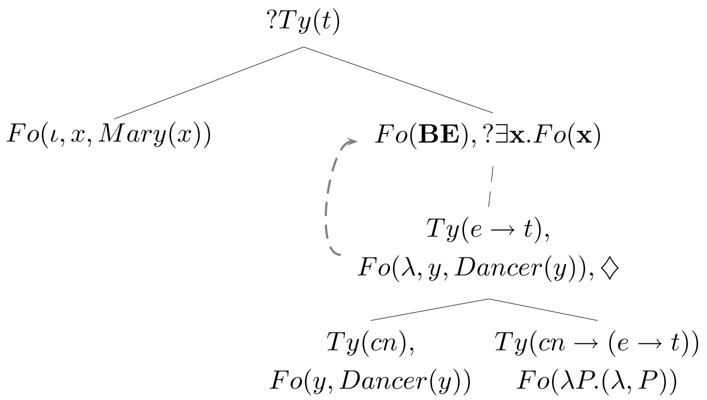
ho	*sōkratēs*	*ēn*	*philosophos.*
the.nom.sg	Socrates.nom.sg	be.3.sg.impf	philosopher.nom.sg

Socrates was a philosopher

b. $Philosopher(Socrates)$.

For nominal predicates in English, a slightly more complicated story needs to be told because of the appearance of the indefinite article in singular constructions: *The student is *(a) genius.*[26] A trivial way of analysing this construction is to treat the indefinite article as being ambiguous between something that constructs an epsilon term in the context of a requirement to construct an expression of type e and one that makes a common noun into a one-place predicate in a context in which such an expression is required. A sentence like *Mary is a dancer* may then be parsed in the same way as other post-copular predicate constructions, as illustrated in (40). The indefinite article then provides some binder for the distinguished variable in the common noun, an epsilon operator in the context of a requirement for a term and a lambda operator in the context of the requirement for a one-place predicate. The output formula is simply $Fo(Dancer(Mary))$ as required.[27]

(40) Parsing *Mary is a dancer*



3.3 Equative Clauses

Equative clauses are typically described as involving the identification of the referents of two definite, referential, noun phrases that appear with the copula. This equating function

[26] Plural nominal predicates do not pose a problem and can be analysed as predicates that merge with the main predicate node. (i) thus gets interpreted as (ii) directly through MERGE.

i. Those students are fools.

ii. $Fool(\epsilon, x, Student(x) \wedge Plural(x))$.

[27] A more interesting story might be told by treating the postcopular noun phrase as projecting an epsilon term and merging this with the terms provided by the subject, along the lines of the equative construction discussed in the next section. However, the fact that both subject and predicative term project full structures precludes a straightforward adaptation of this analysis and so I leave this possibility to one side, but see Kempson, Cann and Marten (fcmg) ch. 8 for some discussion.

may be viewed as deriving from the copula (Montague 1973, Dowty, Wall and Peters 1981, inter alia) or through some operation on the term expressed by the postcopular noun phrase (e.g. Partee 1986, Williams 1983). In both cases the effect is the same: the copula is treated as a two-place predicate of some sort and the output is a statement of identification between the two terms, $\alpha = \beta$. If the assumption put forward above is correct, that the copula is an expletive one-place predicate without explicit semantic content, the question arises as to whether and how a relation of identity relation can be derived.

An obvious way of accounting for equatives would be to adapt the approach of Partee (1986) which treats the copula as essentially predicative (of type $(e \rightarrow t) \rightarrow (e \rightarrow t)$ with the semantic structure proposed in Montague (1973) (although with lower typing), i.e. $\lambda P \lambda x.P(x)$[28]) with a type shifting operation (**Ident**) on a postcopular term to turn it into an identity predicate. Within the current framework one could allow referential noun phrases to be homonymous between a term and an identity predicate founded on an epsilon term constructed from the common noun phrase (e.g. *the teacher* could be realised either as $\mathbf{U}_{Teacher(\mathbf{U})}$ in a context requiring a term or $\lambda x.x = \epsilon, x, Teacher(x)$ in a predicate context). The analysis could then follow that given for predicative expressions above, deriving equative expressions directly.

A more interesting approach suggests itself, however, that exploits the machinery of Dynamic Syntax presented above and derives the equative interpretation without recourse to assuming that either the copular or the definite article (or other definite determiners) are homonymous.[29] As we have seen, equative and specificational clauses necessarily involve a definite noun phrase, either before or after the copula (or both) and a copular clause without a definite cannot be easily interpreted as equative or specificational, as illustrated in (41).

(41) a. John is the teacher. *Equative*
b. That student over there is the best in the class. *Equative*
c. The best in the class is that student over there. *Specificational/Equative*
d. The culprit is John. *Specificational*
e. A PhD student is the lecturer for this course. *Equative?*
f. A plant is a gift for life. *Predicative*

There must something specific to definite expressions which allows equative or speci-

[28]Partee, in fact, allows a variable type and analysis with the arguments of the expression appearing in either order, i.e. $\lambda x \lambda P.P(x) : e \rightarrow ((e \rightarrow t) \rightarrow t)$.

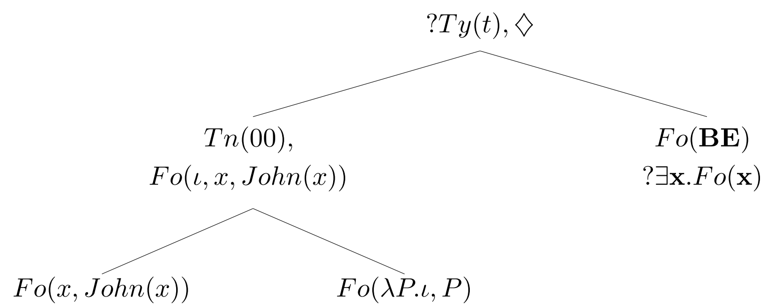
[29]It may be objected that, since I have treated the indefinite article as homonymous in the previous subsection, there is no a priori reason to reject such an analysis for the definites. However, there are two reasons for eschewing homonymy for definites in this case. Firstly, there is only one indefinite article involved in the predicative construction, *a*. Even plural *some* does not give rise to a predicative reading (*John and Mary are some teachers* is not synonymous with *John and Mary are teachers*). For definites, it would have to be assumed that all definite determiners, including demonstratives, are homonymous, so that the homonymy is not lexically restricted. Secondly, the analysis proposed for definite noun phrases in terms of a metavariable plus 'presupposition' seems not to be easily relatable to an operation that turns an epsilon term into an identify predicate. There would, therefore, be no obvious explanation for why it is definites in particular that are subject to this particular interpretation.

ficational readings that is not available to other types of noun phrase. To see what this might be, consider the short text in (42). In interpreting the equative clause in B's utterance, the hearer, A, assumes the existence of someone who drank the last of the milk and then identifies this person with John through the semantics of the concept *Culprit*.

(42) **A**: Oh no, someone has drunk the last of the milk again.
B: John is the culprit.

Analysing *John is the culprit*, we begin by establishing the structure in (43) through the parsing of the first two words (ignoring tense as usual).

(43) Parsing *John is*

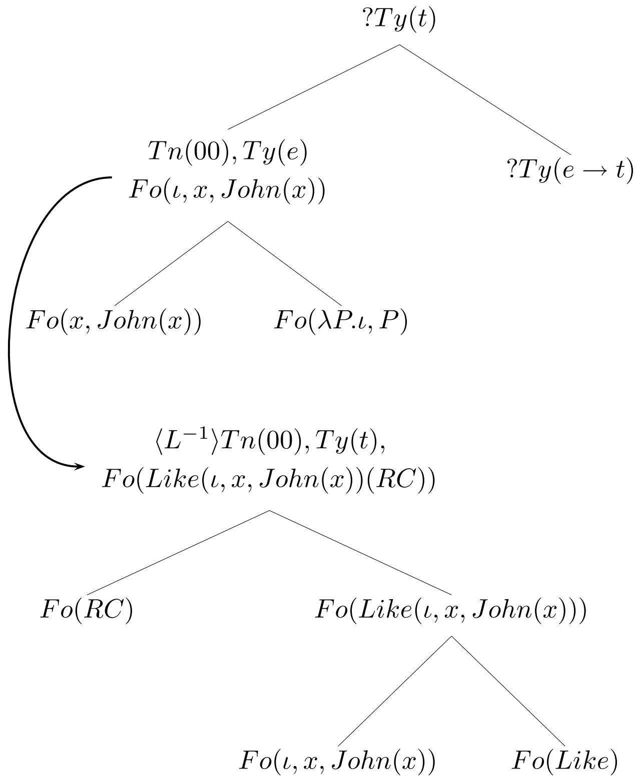


At this point the predicate node remains incomplete and so the pointer must move down the tree from the topnode. One possibility, of course, is that it moves to the subject node, a move that is permitted even though the node is complete. Further development of this node, on the other hand, is possible only if restrictions associated with the node are not violated (such as the bottom restriction $[\downarrow]\bot$) and the resulting structure does not give rise to contradictory information holding of dominated nodes. Clearly with respect to (43) any development by Late*Adjunction is likely to lead to an ill-formed outcome, as any growth from the subject node will contain information that clashes with the formula (and possible other) information holding at that node: for example, the structure induced from a parse of *the culprit* will be incompatible with that projected by *John* because the former will project a bottom restriction which is contradicted by the daughters projected by the latter. This is not to say that *no* development of the tree from the subject node is disallowed, but any further update must not result in the node dominating any other node.

One of the innovative aspects of Dynamic Syntax is that it allows for the building of structures in tandem, constructing first one partial structure, and then another which uses the first as its context. A characteristic property of such "linked" structures is that they typically share a common term, and furthermore, the process of inducing the second of such a pair of structures involves a transition from one tree to the other which itself imposes a constraint for a second occurrence of the term to be shared in that second tree. LINK structures have their clearest application in characterising relative clauses, where from a completed $Ty(e)$ node decorated by a term $Fo(\alpha)$ a new propositional tree is projected which is required to contain a copy of $Fo(\alpha)$.

The full details of the DS analysis of Relative Clauses are not important here (for which see Kempson et al. 2001, Kempson 2002), but as an illustration the figure in (44) shows the structure induced by the parse of the first four words of the sentence *John, who I like, smokes*, where the thick black line indicates the LINK relation and the first person pronoun is taken to pick out the author.

(44) Parsing *John, who I like,*



The modality that connects the LINKed propositional tree to its host node is a novel one: $\langle L^{-1}\rangle$ (and its inverse $\langle L\rangle$). This modality is independent of the mother/daughter dominance relations ($\uparrow$ and $\downarrow$) which do not carry over from one tree to a LINKed tree. So while $\langle\downarrow\rangle Fo(x, John(x))$ holds of node $Tn(00)$, $\langle\downarrow\rangle Fo(Like(\iota, x, John(x))(RC))$ does not. Conversely, $\langle L\rangle Fo(Like(\iota, x, John(x))(RC))$ holds of node $Tn(00)$, while $\langle L\rangle Fo(x, John(x))$ does not. The LINK mechanism therefore provides a means of developing structure from a node without violating the bottom restriction.

Although the LINK mechanism has its clearest application in the analysis of relative clauses, it may be used for other types of modification as well. In particular, it may be used to account for the stacking of noun phrases in apposition constructions, as illustrated in (45).[30]

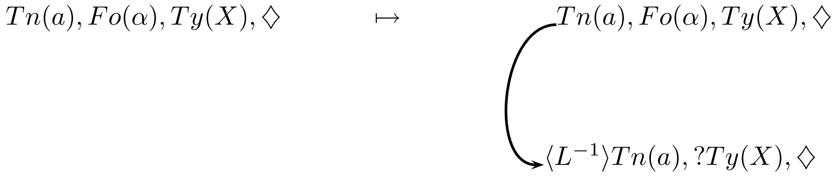
(45) Ruth, a colleague from London, a Leverhulme research professor, is giving a talk next week.

Such constructions can be analysed straightforwardly through LINK and we can posit a general rule that induces a LINKed structure with a requirement for a formula as the same type as the node from which the structure is projected. The effect of this rule is

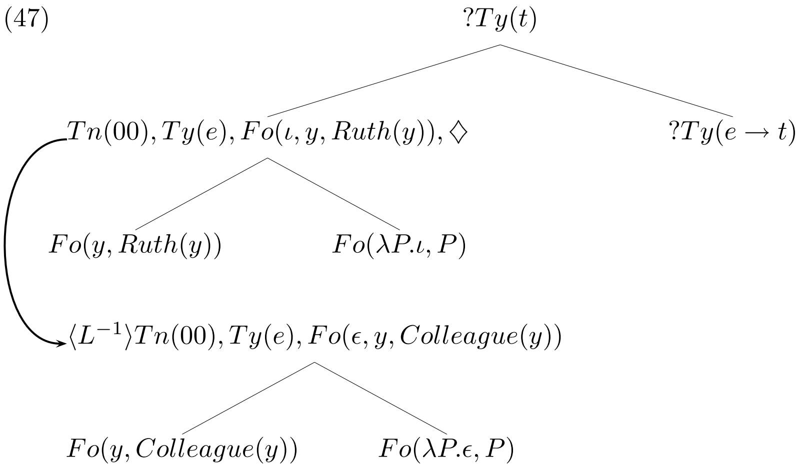
[30]See also Swinburne 1999 for further uses of the LINK modality to analyse modifier constructions.

shown in (46), where again the thick black arrow shows the LINK relation, and the tree analysing the initial three words of the string *Ruth, a colleague (from London), is giving a talk next week* is given in (47).

(46) Appositive LINK Adjunction

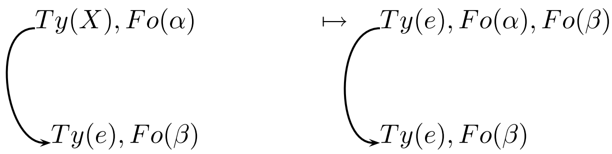


(47)



Note that this rule does not impose a requirement that the LINKed tree share a term with the host. In this structure, the interpretive interdependence of the two trees is established through an evaluation rule that combines the content of the LINKed tree with that of its host node. This it does by simply copying the formula value from the rootnode of the LINKed tree onto the host node, as shown in (48)[31]

(48) Appositive LINK Evaluation



Applying this to the structure in (47) gives rise to a decoration on the subject node of two distinct formula values, i.e. $Fo(\iota, x, Ruth(x))$ and $Fo(\epsilon, x, Colleague(x))$. Decorations on nodes must, however, be coherent, i.e. not involve incompatible values for the same label, where compatibility is defined as identity or subsumption: $\alpha \leq \beta$. Since it is not the case that $Fo(\iota, x, Ruth(x)) \leq Fo(\epsilon, x, Colleague(x))$, this result should be

[31] For more discussion of LINK evaluation rules and their application see Kempson 2002 and Kempson, Cann and Marten fcmg.

ill-formed. However, we may exploit the identity part of the subsumption ($\leq$) relation over informativeness and allow two formula values to decorate the same node just in case they are as informative as each other, the interpretation of the node thus not involving contradictory information. From a semantic point of view this will be (extensionally) satisfied as long as the two formulae have identical denotations. Although such a semantic condition might seem to be at odds with the representationalist spirit of Dynamic Syntax, the properties of the epsilon calculus may be exploited to provide a straightforward way of incorporating this idea into the representation system. If two distinct epsilon terms, $\epsilon, x, P(x)$ and $\epsilon, x, Q(x)$, both denote the same entity (whatever that may be), then $Q(\epsilon, x, Q(x))$ and $Q(\epsilon, x, P(x))$ must 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, $Q(\epsilon, x, Q(x))$, $Q(\epsilon, x, P(x))$, and $Q(\epsilon, x, P(x) \wedge Q(x))$ are all truth conditionally equivalent under this assumption so we may substitute $(\epsilon, x, P(x) \wedge Q(x))$ for $(\epsilon, x, Q(x))$ salva veritate. This then allows a node which contains two epsilon terms to be resolved into a single term which picks out the witness for both restrictors.

Using this result, it is possible to give a representational slant to the idea that the merge of two formulae is permissible just in case they are denotationally identical, by adopting a resolution rule that combines two epsilon terms into a single term, if these decorate the same node. The rule is shown as an inference rule in (49) and the result of applying it to the subject node in (47) is shown in (50), a formula which picks out the (unique) entity that is Ruth and a colleague.[32]

(49) Term Resolution:

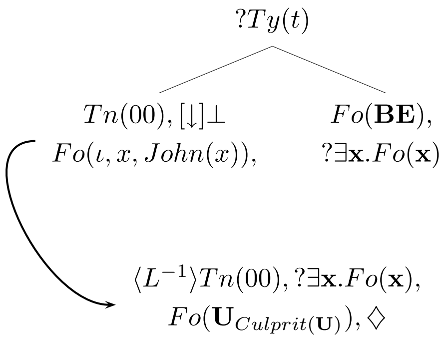
$$\frac{Fo(\epsilon, x, P(x)), Fo(\epsilon, x, Q(x)),}{Fo(\epsilon, x, (P(x) \wedge Q(x))}$$

(50) $Fo(\iota, x, Ruth(x) \wedge Colleague(x))$.

So, there is now the possibility of developing the subject node in (43) by applying the Appositive LINK Adjunction rule to project a LINK structure with a type e requirement. This permits the parse of a post-copular noun phrase such as the definite noun phrase *the teacher*, as shown in (51).

[32] The formula retains the iota operator as defining a more restricted type of epsilon term, one with a unique value in any context.

(51) Parsing *John is the culprit* via Appositive LINK Adjunction



Next the value of the metavariable projected by the definite article may be established. Substitution of this is constrained by the presuppositional information that the substituting term must be describable as a culprit. Such a substituend is provided by the content of A's utterance by taking the epsilon term that picks out the witness for the act of drinking the last of the milk, i.e. $(\epsilon, x, Drink(m)(x))$ (where m is a term denoting the last of the milk that was drunk). Under the assumptions that someone who drinks the last of the milk (without getting more) is guilty of wrongdoing and that someone who is guilty of wrongdoing is describable as a culprit, the substitution of this term for the metavariable projected by the definite is licensed (through abduction on the semantics of culprit) and the presupposition discharged. These steps of reasoning are summarised in (52).

(52) a. **Given Context**: $\exists x.Drink(m)(x)$.

b. **Arbitrary term satisfying a**: $(\epsilon, x, Drink(m)(x))$.

c. **Assumption**: $Drink(m)(\alpha) \vdash Wrongdoer(\alpha)$

d. **Semantics of Culprit**: $Culprit(\alpha) \vdash Wrongdoer(\alpha)$

e. **Substitution**: $Fo(\mathbf{U}_{Culprit(\mathbf{U})}) \Leftarrow Fo((\epsilon, x, Drink(m)(x))$

Appositive LINK Evaluation and Term Resolution then apply to the resulting structure to give the annotation on the subject node shown in (53).[33]

(53) **Term resolution** : $Ty(e), Fo((\iota, x, John(x) \wedge Drink(m)(x)_{Culprit(\epsilon,x,Drink(m)(x))})$

Notice that Term Resolution in (49) effectively determines the identity of two terms by picking out the witness for both internal predicates: it captures identity without the need for any type-raising operation or even the use of the identity relation. Hence, the equative reading is neither particular to the content of the copula nor derived from some operation on the denotation of a noun phrase. Instead, the identity of the two terms in an equative clause is derived through grammatical operations (Appositive LINK Adjunction

[33]The formula retains the iota operator as defining a more restricted type of epsilon term, one with a unique value in any context.

and Evaluation) that are themselves independently required to account for certain types of modification.

The analysis has not yet finished, however, as the value for the predicate still needs to be identified. In keeping with the assumptions of Dynamic Syntax, I adopt a general Relevance Theoretic perspective on pragmatic processes such as substitution whereby there is a tradeoff between processing cost and information gained (see Wilson and Sperber 1995, Carston 2002 for proper discussion of the theory). Optimal Relevance is determined as a trade-off between cognitive effort and informativeness (the more effort required to access an interpretation the more informative it should be). A hearer will thus take as substituend the most accessible formula that is likely to yield significant inferential effects. The pragmatic process of substitution occurs *within* the construction of a propositional representation, however, and so will tend to prefer substituends which are provided by the immediate discourse because the domain over which other inferences are to be carried out may not yet be complete. In substituting for the predicate metavariable in (51), the context given in (42) provides the three candidate predicates in (54), the most informative of which should be chosen as the substituend.

(54) **Possible Predicate Substituends**:

a. From *the culprit*: $\lambda y.Culprit(y)$.

b. From substituend (and main predicate of A's utterance): $\lambda y.Drink(m)(y)$

c. From *the (last of the) milk*: $\lambda x.Milk(x)$

Of the predicates in (54), (54c), picking out the property of being milk, is least likely to be chosen because of the (likely) processing cost needed to derive useful inferential effect from the proposition that *John is milk*. Of the remaining two predicates, that of being a culprit has been used to identify the appropriateness of substituting $\epsilon, x, Drink(m)(x)$ for $\mathbf{U}_{Culprit(\mathbf{U})}$ and so is less informative than (54b), the property of drinking the last of the milk, leaving the latter as the most informative potential substituend in the context.[34]

The result in (55a) of choosing (54b) as the substituend for **BE** is shown in (55b). Given the equivalence in (55c) (which has already been used above to account for the interpretation of certain existential focus constructions), the output content of parsing *John is the culprit* in the context given is that in (55d), a statement asserting that someone did steal the last of the milk (a confirmation of A's initial assertion in (42)) and that that someone is John, as required.

(55) a. **Substitution**: $Fo(\mathbf{BE}) \Leftarrow Fo(\lambda y.Drink(m)(y))$.

b. $Fo(Drink(m)(\iota, x, John(x) \wedge Drink(m)(x)))$

c. $F(\epsilon, x, F(x)) \equiv \exists x.F(x)$

d. $\exists x.John(x) \wedge Drink(m)(x)$

There exists someone who drank the last of the milk and that person is John.

[34]Note that this result is arrived at without actually undertaking any further inference at this point: milk not being a property semantically predicable of a human being and culprit (under assumptions already made) subsumes drinking the last of the milk.

By exploiting underspecification of various sorts and the inferential process of substitution, I have provided an account of the interpretation of equative clauses without the use of the logical identity operator. The only additional requirement I have had to make is that two epsilon terms decorating the same node may be resolved into one complex one. Such a resolution is motivated by the general and independent requirement that information on nodes not be contradictory, rather than by any specific consideration of the equative construction itself. This resolution does not involve type changing or indeed the use of identity at all[35] but simply from the notion of content underspecification as applied to the copula.

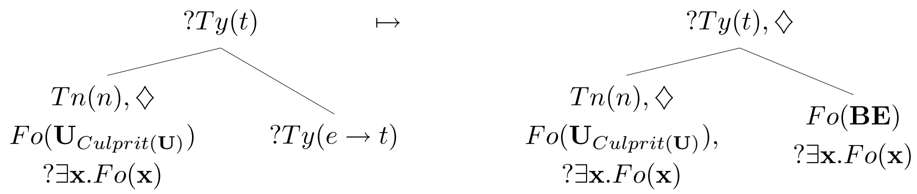
3.4 Specificational Clauses

We turn now to specificational clauses, which as mentioned above, like equatives, involve two apparently referential noun phrases. However, unlike true equatives specificational sentences involve a definite in pre-copular position which can be construed as a description rather than as picking out some definite entity. In other words, the subject definite provides a description whose referent is assumed to be unknown to the hearer, and whose value is supplied by a referential post-copular noun phrase. An analysis of such clauses is straightforward under the assumptions made here. Consider the text in (56).

(56) **A**: Where are my socks?
B: The culprit is John.

The analysis of B's utterance begins with the parse of the definite noun phrase projecting a metavariable with presupposition as before and shown in the first tree in (57). The current context, however, does not provide provide an obvious substituend for the metavariable and, substitution being an optional process, the pointer moves to the predicate node, licensing the parse of the copula to give the second tree in (57).

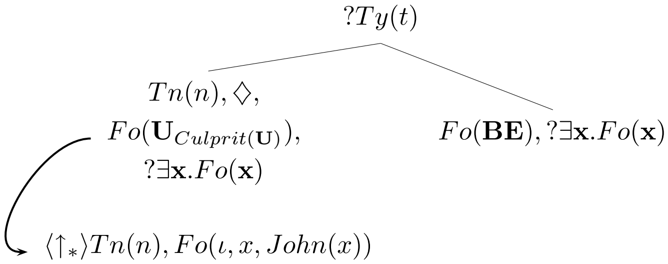
(57) Parsing *The culprit is*



The pointer returns to the incomplete subject node. As this is decorated with the bottom restriction, Late*Adjunction cannot apply, but as with equatives discussed in the last section appositive LINK Adjunction can apply. This creates a LINK structure and provides an open type e requirement permitting the parse of *John*, as illustrated in (58).

[35] This leaves open the intriguing possibility that the logical relation of identity does not form part of the semantic representation system of natural languages, although it does form part of the interpretation language of that system.

(58) Parsing *The culprit is John*



Appositive LINK Evaluation then applies to copy the formula projected by the proper name onto that decorated by the metavariable to give the decoration on the subject node shown in (59a). Notice, however, that Term Resolution does not apply here. Indeed, no resolution rule is required at all. This is because metavariables subsume all other formula values ($Fo(\mathbf{U}) \leq Fo(\alpha)$ for all α) and so by copying the value of the LINKed structure onto the host the value for the metavariable is established and no further inference is required. The output value for the subject node in (58) is thus just (59b) and this gives rise to the interpretation of specificational clauses as providing a description referent whose identity is determined by the postcopular noun phrase.

(59) a. $Ty(e), Fo(\mathbf{U}_{Culprit(\mathbf{U})}), Fo(\iota, x, John(x)), [\downarrow]\bot$
b. $Fo(\iota, x, John(x)_{Culprit(\mathbf{U})})$

As before, we need a value for **BE**. In the current context, the most accessible predicate is that of being a culprit: $\lambda x.Culprit(x)$. This has not been used to identify any substituend and there is no other accessible predicate which it subsumes. It, therefore, must be chosen as substituend as shown in (60a) with the formula in (60b) resulting as that for the propositional tree. From which some inference must be made between John's culpability and A's inability to find socks.

(60) a. $\mathbf{BE} \Leftarrow \lambda x.Culprit(x)$
b. $Fo(Culprit(\iota, x, John(x)))$ (= John is a culprit).

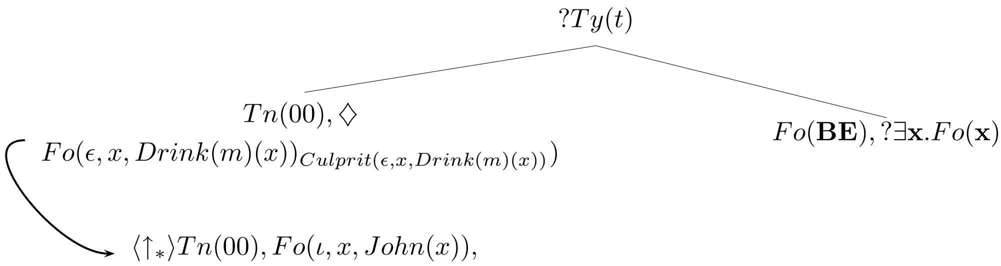
Specificational clauses thus may end up being truth-conditionally equivalent to predicative clauses, but notice that the process by which the interpretation is obtained is distinct. In parsing *John is a culprit*, a term is identified, $\iota, x, John(x)$, and a property, $\lambda x.Culprit(x)$, is predicated of this without need of inference. In parsing *The culprit is John*, on the other hand, the possibility there is someone who is a culprit is presented to the hearer and this someone is identified as John through a process of Appositive LINK Evaluation. The informational effects are thus distinct, with the latter providing focus on *John*, and at the same time (in the current context) providing information that may enable the hearer to find their socks by indicating that somebody may be responsible for wrongdoing with respect to those socks. Process is central to Dynamic Syntax and forms part of the procedural 'meaning' of an utterance without the need to define different representations or layers of information to specifically encode differences in meaning between different constructions.

There is another way in which copular clauses with definite subjects can be analysed given the assumptions of this paper, through substitution of the subject metavariable. Consider in this instance (61).

(61) A: Who's drunk the last of the milk?
B: The culprit is John.

As before, the definite NP is parsed to yield a metavariable with associated presupposition decorating the subject node. In this case, however, a substitution can be made from context: the epsilon term picking out the arbitrary object that is assumed to have drunk the last of the milk. Substitution duly occurs and the copula is parsed. The pointer again goes to the (complete) subject node but as with the equative clause in the last section, Appositive LINK Adjunction applies and the postcopular term *John* is parsed to give the tree in (62).

(62) Parsing *The culprit is John*



Appositive LINK Evaluation decorates the subject node with the formula on the rootnode of the LINKed tree and the two epsilon terms are resolved to one by Term Resolution, exactly as in the equative construction, to give the complex term in (63a). Substitution for **BE** may as before be by the predicates $\lambda x.Drink(m)(x)$ or $\lambda x.Culprit(x)$. In this instance, it appears that either predicate may be a substituend, since the presuppositional information about being a culprit need not necessarily have been used. The results of the two substitutions is shown in (63), yielding slightly different interpretations.

(63) a. $Fo((\epsilon, x, Drink(m)(x) \wedge John(x))_{Culprit(\epsilon,x,Drink(m)(x))})$
b. $Drink(m)(\epsilon, x, Drink(m)(x) \wedge John(x))_{Culprit(\epsilon,x,Drink(m)(x))})$
c. $\exists x.Drink(m)(x) \wedge John(x) \wedge Culprit(x)$
There exists someone who is a culprit and who drank the last of the milk and that person is John.
d. $Culprit(\epsilon, x, Drink(m)(x) \wedge John(x)))_{Culprit(\epsilon,x,Drink(m)(x)\wedge John(x)))}$
The person who drank the last of the milk and who is John is a culprit.

Notice that the result may either be like an equative or specificational (predicative) reading. The difference in the former case from *John is the culprit* is again a matter of information structure: *John* being final is necessarily in focus in *The culprit is John* where it is not in the straightforward equative. In this way, I account for the fact that equative and specificational constructions may give rise to quite different interpretations in different contexts (see Heycock and Kroch 1999) but without needing to assume that subtly different readings give rise to (or result from) different representations.

4 Conclusion

In this paper, I have presented a theory of the copula that treats it as providing underspecified semantic content which requires enrichment for interpretation to occur. This enrichment may be provided directly through the parse of expressions that follow the copula or through pragmatic inference over predicates provided by local context. The different analyses of definite and indefinite noun phrases have been shown to affect the way that pragmatic inference may be driven while differences in the way content is derived have been argued to give rise to differences in information content and inferential effect.

There are, of course, many consequences of this approach that remain to be explored. Not least is the way the substitution process works with respect to modal and negative contexts. Clearly in the discourse in (64), although the first conjunct appears to provide the accessible predicate $\lambda x.\neg Happy(x)$ this cannot be substituted for the metavariable projected by *be*.

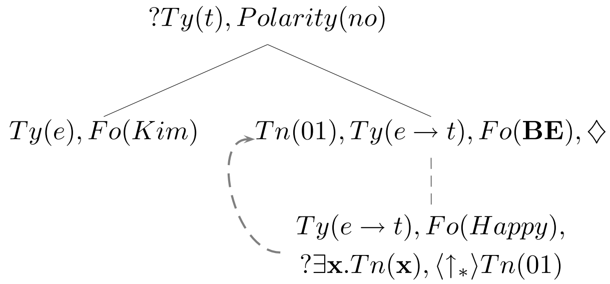
(64) Sandy isn't happy, but Kim is.
$\vdash$ Kim is happy.
$\nvdash$ Kim isn't happy.

The conclusion must be that *isn't* does NOT project a 'negative predicate' $\lambda x.\neg\mathbf{BE}(x)$ and that *is* must never be associated with a (non-syntactic[36]) negation. There is no accepted theory of negation in Dynamic Syntax (as yet), but the current concerns point to an account like that of Situation Semantics where negation (and affirmation) is treated in terms of a polarity marking (Barwise and Perry 1983). Thus, we may assume that a VP negative introduces a polarity label on the topnode as in (65) which gives rise to analyses such as that in (66) and interpretations such as that in (67).

(65) *isn't*

IF	$?Ty(e \rightarrow t)$
THEN	$\mathtt{go}(\langle\uparrow^1_*\rangle), \mathtt{put}(Polarity(no)), \mathtt{go}(\langle\downarrow^1_*\rangle)$
	$\mathtt{put}(Ty(e \rightarrow t), Fo(\mathbf{BE}), ?\exists\mathbf{x}.Fo(\mathbf{x}))$
ELSE	ABORT

(66) Parsing *Kim isn't*



[36] Because of the need to interpret the second conjunct in the following text as *Kim is unhappy* not *Kim is happy*.
i. Sandy isn't unhappy, but Kim is.

(67) $< Happy(Kim), no > \vdash \neg Happy(Kim)$

More needs to be said in justification of this approach and the way negation interacts with scope and other aspects of interpretation, but in principle negation (and one would hope, modality) does not undermine the current account of *be*. I take the success in which a uniform view of the copula within Dynamic Syntax leads to successful analyses of the copula in elliptical utterances, existential focus, predicatives, equatives and specificational clauses to support the use of underspecification, both syntactic and semantic, and concepts of pragmatic enrichment as a tool in analysing natural language.

5 References

Asher, N (1993) *Reference to Abstract Objects in Discourse.* Dordrecht, Kluwer.

Barwise, J and Perry, J (1983) *Situations and Attitudes.* Cambridge, Massachusetts, MIT Press.

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 (to appear) Towards a dynamic account of 'be' in English. In Comorowski, I and von Heusinger, K (eds.) *Existence.* Dordrecht, Kluwer.

Cann, R, R Kempson, and M Otsuka (2003) 'On left and right dislocation: a dynamic perspective'. MS University of Edinburgh/King's College London.

Carston, R. 2002. *Thoughts and Utterances: the pragmatics of explicit communication.* Oxford, Basil Blackwell.

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.

Gazdar, G, E Klein, G K Pullum and I A Sag (1985) *Generalized Phrase Structure Grammar.* Oxford, Basil Blackwell.

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.

von Heusinger, K (2002) Choice functions and the anaphoric semantics of definite NPs. In K. von Heusinger, R. Kempson and W. Meyer-Viol (eds.) *Proceedings of the workshop "Choice Functions and Natural Language Semantics"* Arbeitspapier 110. FB Sprachwissenschaft, Universität Konstanz:63-83.

Heycock, C (1994) *Layers of Predication.* New York, Garland Press.

Heycock, C and Kroch, A (1999) Pseudocleft connectedness: implications for the LF interface. *Linguistic Inquiry* 30:365-397.

Hilbert, D and Bernays, P (1939) *Grundlagen der Mathematik II.* Berlin, Julius Springer.

Kamp, H (2001) The importance of presupposition. In C. Rohrer and A. Roschdeutscher (eds.) *Linguistic Form and its Computation.* University of Chicago Press, CSLI.

Kempson, R. (2002) Relative clause construal: the dynamics of terms under construction.

MS, Kings College London.

Kempson, R, Cann, R and Marten, L (fcmg) *The Dynamics of Language*. Elsevier.

Kempson, R, W Meyer-Viol and D Gabbay (2001) *Dynamic Syntax*. Oxford, Basil Blackwell.

Kempson, R, Kiaer, J and Cann, R (2004) Topic, focus and the structural dynamics of language. MS Kings College London/University of Edinburgh.

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.

Lamarche, J (2003) ‘*Be* the one’. In M Weissberger (ed.) *Proceedings of Sinn und Bedeutung 7*, University of Konstanz.

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) Specificational clauses and semantic types. Paper presented at the workshop *Existence: Semantics and Syntax*, Université de Nancy 2, September 2002.

Montague, R (1973) The proper treatment of quantification in ordinary English. In J. Hintikka, J. Moravcsik and P. Suppes (eds.) *Approaches to Natural Language*. Dordrecht, Reidel.

Partee, B H (1986) Ambiguous pseudoclefts with unambiguous be. *NELS* 16: 354-366.

Rothstein, S (2001) *Predicates and Their Subjects*. Dordrecht, Kluwer.

Sperber, D and D Wilson (1989/1995) *Relevance*. Oxford, Basil Blackwell.

Swinburne, D (1999) LINK and the dynamics of utterance interpretation. Unpublished PhD dissertation, University of London.

Williams, E (1983) Semantic vs. syntactic categories. *Linguistics and Philosophy* 6: 423-446.