Auxiliary Verbs:
A Dynamic Syntax Account

Christine Howes
christine.howes@kcl.ac.uk

Philosophy Department, King's College London

In partial completion of the MSc in
Computational Linguistics and Formal Grammar

September 2007

Contents

List of Tables

List of Figures

Auxiliary Verbs:
A Dynamic Syntax Account

Abstract

Dynamic Syntax is a grammar formalism that seeks to combine insights from studies in syntax, semantics, pragmatics and psycholinguistics into a coherent model of communication. It is based on the idea of monotonic tree growth, whereby interpretations are built up in an incremental manner; word by word, as each is encountered in a string. The theoretical underpinnings of Dynamic Syntax are briefly explored in section 1.

English auxiliary verbs are a well-studied group of words in many different formalisms, chiefly because they are a small group with highly idiosyncratic properties. The group of auxiliaries is described in section 2, and previous accounts of them in different frameworks, as well as attempts to explain the data from a diachronic perspective, are outlined in section 3.

Section 4 introduces Dynamic Syntax's formal tools, with examples to illustrate how the key notions interact to produce semantic parse trees for grammatical strings. Various possible ways of using these tools to analyse the complex problems presented by the English auxiliary verbs are explored in section 5.

Full details of how the key notions of LINKed structures and underspecification can account for the syntactic peculiarities of English auxiliary verbs, and also accommodate their semantic idiosyncracies are given in section 6. Specific attention is paid to the tricky interaction between the auxiliary system and *not*, and other negative words (notably *never*), and the wh-question words. The analysis is extended to account for exceptional items, and its implications for infinitival-*to* are discussed. Conclusions and questions for further research are offered in section 7.

1 Introduction

Dynamic Syntax is a grammar formalism which is based on the idea of monotonic tree growth. The theoretical notions upon which it depends are outlined below, with the formal tools laid out in detail in section 4.

Briefly, the foundations of Dynamic Syntax are based in the recognition of the fact that what are usually considered independent features of language; syntax, semantics and pragmatics, are in fact mutually dependent features of human communication. Parsing and processing are taken to be two sides of the same coin, thus doing away with the (usually implicitly accepted) Chomskyan distinction between competence and performance (see section 1.1). Additionally, words are analysed in the order in which they occur in a string, thus taking how an interpretation is built up to have a central role. Further, complete trees in Dynamic Syntax have no representation of word order, and are thus more appropriately seen as analogous to semantic trees. Dynamic Syntax rejects the notion that a separate descriptive level is required for syntax, postulating instead that phenomena usually described as syntactic can be explained and described by the state of a (partial) parse tree.

Like in many theories of semantics (for example, lambda calculus as in Carpenter (1997)), language is seen as compositional, with meanings built up as words are added to the parse tree. The principle of compositionality holds that, in natural languages, the basic meaning of an utterance or sentence can be built up by the respective meanings of its parts. This principle has strong intuitive arguments in its favour. These are, briefly, productivity and systematicity. Systematicity refers to the fact that there are definite patterns that are apparent in natural languages; for example, if we know that someone understands the sentence “John loves Mary”, we would expect them to also understand “Mary loves John”. Productivity refers to the human ability to understand and be able to produce an infinite number of novel utterances. This suggests that we are able to combine words that we already know and their meanings in a systematic way to allow us to interpret or form strings which we could not have previously encountered. Although these intuitive arguments towards a compositional theory of meaning are compelling, an acceptance of the principle is not to suggest that a complete semantic system can be based purely on syntactic considerations. Even simple sentences can have different meanings in different contexts (as the mere existence of sarcasm demonstrates), however, it seems that compositionality has an important role to play in any theory of grammar that seeks to incorporate meaning, as Dynamic Syntax does.

Dynamic Syntax also seeks to incorporate notions of context into the theory, thus formalising ideas that are usually consigned to the “pragmatic wastebasket” (Bar-Hillel (1971)). Dynamic Syntax asserts that, contrary to popular belief, pragmatics is an integral part of a theory of grammar. Although differing slightly in detail, the basic theoretical standpoint for incorporating contextual factors into a theory of grammar is that of Relevance Theory, as espoused by Sperber & Wilson (1995), and outlined below in section 1.2. Like Relevance Theory, Dynamic Syntax takes the act of communication to be of fundamental importance to any linguistic theory. Communication is viewed as a goal-driven process, with the aim of recovering intended meanings. Because of this, the onus is on the hearer in any communicative situation, and interpretation (i.e. parsing), not language production is seen as primary. This view is supported by data from language acquisition studies, and our intuitions that people (especially children and second language learners, who do not yet have a full grasp of the language) can understand linguistic inputs of greater complexity than they can spontaneously produce. Like Relevance Theory, too, the philosophical underpinnings of Dynamic Syntax share the commitment to a Fodorian representational theory of mind (see e.g Fodor (1998), for philosophical arguments and discussion). This does not mean that compositionality is lost, merely that the compositional semantics and syntax interact with contextual and pragmatic inferences as interpretations are built, in order to uncover the speakers intended meaning.

A further key feature of the Dynamic Syntax approach is that it is not only to be seen as a ‘snapshot’ of language, fixed both in time and place. It takes on board the challenges presented by both cross-linguistic and diachronic data, and, while offering formal tools which generalise to different languages and linguistic environments, also tries to incorporate its notions of pragmatic constraints to explain how processing pressures, including concepts such as routinisation, can lead to language change. Routinisation, as a phenomenon that occurs at a scale between participants in a dialogue, is supported by psycholinguistic studies, such as Pickering & Garrod (2004). What Dynamic Syntax tries to formalise is the other level at which routinisation occurs; within subjects, whereby sets of linguistic actions which are called up frequently in similar situations can form a routine. An example relates to word order, which is known to have been much freer in Old English than it is in modern English. A Dynamic Syntax explanation of this might point to the fact that certain actions (originally associated with specific lexical items) always led to the initial actions in the interpretation of a sentence being the same, which, over time became encoded as a general rule. Over successive uses, such a routine can become calcified, and its origins in processing pressures may be lost. In this way, while it is clear, for example, that the auxiliary system in English represents an idiosyncratic language specific lexical group, with highly specialised syntactic and semantic properties which children must simply learn (in a similar way to the clitic phenomena in Spanish, as studied in Bouzouita & Kempson (2006), and supported by studies in children's language acquisition, such as Richards (1990)), Dynamic Syntax offers us a way to hypothesise about where these idiosyncracies might have originated. It therefore seems able to capture the linguistic facts suggesting the gradual nature of linguistic change, as it can show how certain sets of commonly used actions could become routinised and preferred, before becoming calcified in a grammar, whilst alternatives are still, in principle, possible. This contrasts to the notion of parametric variables in language which must be set either one way or the other. We will see how using a Dynamic Syntax type of analysis to explain how such calcifications of actions may have arisen from processing and pragmatic preferences can help to account for negative contracted auxiliary forms (e.g. *can't*, section 6.3.4) and negative words like *never* (section 6.3.3), later.

1.1 Psycholinguistic Support for Incremental Processing

Historically, the fields of theoretical linguistics and psycholinguistics have not had much impact on one another. The reasons for this are clear when we consider the adherence to the Chomskyan notions of competence and performance. In these terms, competence is the 'perfect' grammatical knowledge people are supposed to have in their heads, and performance involves a 'mangled' imperfect version of this. This division was meant to account for performance errors (describing what happens when, for example, we produce an ungrammatical or incomplete string, or how we can understand one) without compromising the fact that, as we know when strings are ungrammatical, we must have some accurate grammatical template in our heads (competence). Although Dynamic Syntax does not license strictly ungrammatical strings (as the parse will abort, see section 4.1, for details), a partial tree will have been generated up to the point where the parse fails, which would potentially be available to be updated from context. Nevertheless, the distinction has been largely respected by both theoretical linguists and psycholinguists, the former concentrating on investigating competence grammar, whilst the latter were concerned with performance effects, such as memory constraints.

More recently, however, researchers on both sides of the divide have begun to appreciate that the distinction may not be as principled as first thought, as evidenced by a growing body of literature (see e.g. Hawkins (2004)). Indeed, Chomsky's appeal to our intuitive ability to distinguish grammatical from ungrammatical sentences seems unable to account for our sense that some grammatical strings are somehow 'less' grammatical than others. In Dynamic Syntax terms, we might speculate that, given two sentences and all other things being equal, more complicated steps would be required to reach a complete parse tree in the less 'acceptable' sentence. Note, however, that such a notion of *complexity* is not well defined in Dynamic Syntax, and therefore no testable hypotheses based on complexity as a factor in acceptability can currently be formulated.

In fact, studies on cross-linguistic and diachronic data, seem to show that performance 'facts' can explain and affect syntactic ones and vice versa. With this in mind, syntactic theories that can easily be used or translated as parsing strategies which explain the psycholinguistic data should be preferred to those which have to postulate entirely separate machinery to account for 'performance' data.

Psycholinguistic studies on head-final languages like Japanese and Korean (exemplified by Kamide & Mitchell (1999)), and also languages like Dutch and German which have head-final constituents seem to show that, in parsing, incrementality is of vital importance. If a parse were head-driven, as, for example, in Pritchett (1991), who claims that a "node cannot be projected before the occurrence of the head, since the relevant features which determine its categorial identity and license both its own and its arguments' attachment are theretofore undetermined", then there would have to be multiple unattached constituents, in even simple Japanese sentences (e.g. (1)), thus placing demands on short-term memory:

(1) *Chika-ga* *Kayo-ni* *koneko-o* *ageta*
Chika$_{NOM}$ Kayo$_{DAT}$ kitten$_{ACC}$ gave
'Chika gave Kayo a kitten.'

Contrarily, incremental accounts posit fewer constraints on the starting point for computing structural relations. They suggest that representational features can be postulated before 'head' words appear (for example, case markers on nouns could indicate structure to be built up prior to the verb being encountered). Arguments could therefore be assigned to an as-yet-unprocessed verb, incrementally, as they are encountered. This is indeed what a Dynamic Syntax account of head final languages proposes. Although the three noun phrases are in some sense unfixed before the verbs appearance, constructive use of case markers allows the building up of structure, meaning that, instead of three separate noun elements waiting for the verb, there is one noun group, with the relationships between them already determined. This type of approach is more in line with speaker intuitions, as Steedman & Baldridge (2003) note, “Dutch, German and Japanese speakers greet with hilarity the suggestion that their languages prohibit any analysis until the verb group . . . has been processed”.

1.2 Relevance Theory

Following Grice (1975), Relevance Theory sees communication as an act of cooperation between speaker and hearer. The Gricean program, which differentiates between the truth-conditions of a sentence (which was taken to be the realm of semantics) and what is actually communicated on any given occasion, via implicatures (taken to be the realm of pragmatics) treats what is said as quantifiably different from what is meant, whilst also making the assumption that certain aspects of utterance meanings, for example, conventional implicatures, can be systematically defined from their literal meanings. At its heart is Grice’s formulation of the ‘folk-linguistic’ idea that “as speakers, we expect what we say to be accepted as true, [and] as hearers, we expect what is said to us to be true” (Wilson & Sperber (2002)), the *Principle of Quality*. However, as Wilson & Sperber (2002) note, “The relevance-theoretic account is based on another of Grice’s central claims: that utterances automatically create expectations which guide the hearer towards the speaker’s meaning.” Relevance Theorists therefore believe that the fundamental expectation of a hearer is that an utterance will be relevant to the discourse, and not that they will necessarily be told something true.

Sperber & Wilson (1995, 2002) define two principles of relevance, based on a cost / benefit model whereby processing effort is seen to be the cost and positive cognitive effects are the benefits.

(2) The First, or Cognitive, Principle of Relevance
The human cognitive system tends towards processing the most relevant inputs available

(3) The Second, or Communicative Principle of Relevance
Every utterance conveys a presumption of its own optimal relevance (Sperber & Wilson (2002)

With these two principles in place, Relevance Theorists believe they can show how speakers and hearers can arrive at a shared meaning, via presumptions about what the other takes as relevant in any given context, knowing that the other will be following a path of least effort to arrive at a plausible hypothesis.

Relevance Theorists do not deny that an utterance can have literal meaning, but claim that all utterances, not just figurative or loose uses of language, are approached with expectations of relevance. They also reject the idea that figurative meanings can be systematically derived from their literal counterparts, or that there can be contextually determined standards of precision, another deviation from the Gricean program. In effect, Relevance Theorists, and the formalism of Dynamic Syntax, believe that decoding alone is not sufficient for intended meaning, even where literal meanings are relevant in a discourse situation. This, of course, links to the notion that context is crucial, (where context could just refer to the discourse participants shared world knowledge), a notion which Dynamic Syntax aims to formalise.

2 Auxiliary Verbs

Auxiliary verbs in modern English are an extremely well studied group of words, from the earliest linguistic analyses to the present day. The reasons for this are manifold. In English, auxiliary verbs represent a small number of items which are highly distinctive both syntactically and semantically. Although certain elements cause disagreements[1] amongst linguists, there is a basic set of auxiliary elements upon which most syntacticians agree. This is comprised of the *modal* auxiliaries (including *can*, *may*, *must*, *will* and *shall*) and certain forms of *have*, *do* and *be*. The modal auxiliaries could be described as having a 'defective' verbal paradigm in English, as they have no non-finite forms at all, and no third person marking (see 4, below).

(4) a. I might pick grapes
b. He might pick grapes
c. *He mights pick grapes
d. *I am mighting pick grapes
e. *I have mighted pick grapes

In addition, although historically the majority of modals had two tensed forms, present and past, (e.g. *could* is derived from the past tense form of *can*), there is no agreement in the literature as to whether treating the modals as tensed pairs remains a legitimate distinction (Sag et al. (2003) do away with it altogether, treating the forms can and could for example as completely separate, whilst others e.g. Langacker (1978) and Lewis (1986) reformulate the connection between the linked pairs as being one of distance, though not necessarily temporal distance). In contrast, *be*, *do* and *have* all have inflecting and tensed forms, and the auxiliary forms of *be* and *have* can appear in non-finite forms (see table 1 below). It should also be noted that the non-modal auxiliaries have other uses where the verb contributes semantic information which it does not in its auxiliary uses. This may be because the auxiliary use does not contribute any semantic information (as in the case of *do*), or it could simply be different semantic information, as in the case of

[1] A well-documented example being the status of *need* and *dare*, discussed in section 2.2.1, below

have which can be possessive in its non-auxiliary use, but represents the perfect tense in its auxiliary use. The fact that the lexical forms of these verbs sometimes behave syntactically like auxiliaries (see the discussion of the NICE properties, in section 2.1 below), but sometimes do not, further muddies the distinction between the categories.

A basic consensus is possible because auxiliary verbs in English seem to act as functional elements, meaning that semantically they contribute notions of time, possibility, obligation and necessity, though not any concrete or conceptual meaning, and they have syntactic peculiarities, not shared by the *lexical* verbs. In addition, the roots of auxiliaries can be traced diachronically, to earlier stages in the development of English when they were not distinct from lexical verbs, although there is little agreement as to why or how such changes took place (see e.g. Denison (1993)). From a syntactic point of view, there are strict restrictions on when and where auxiliary verbs can occur in modern English sentences. They are always optional (in declarative sentences) and precede a lexical verb[2], whose form is determined by the preceding auxiliary, but which is always a non-finite form; an infinitive (either a *base*-infinitive or a *to*-infinitive), a present participle (or *-ing* form, though there is some confusion in the terminology with gerunds, I shall refer to them as *-ing* forms, following Palmer (1988)) or a past participle (or *-en* form, as before). In addition, whilst auxiliaries can co-occur, there are strict restrictions on the possible grammatical orders available (see tables 1 and 2, adapted from Palmer (1988)).

finite	-en	-ing	-en
take(s)/took			
is/was		taking	
has/had	taken		
has/had	been	taking	
is/was			taken
is/was		being	taken
has/had	been		taken
has/had	been	being	taken

Table 1: Primary paradigms for co-occurrence in auxiliaries

2.1 The NICE properties of English Auxiliaries

The main syntactic features of auxiliary verbs, upon which this paper will focus, are those known as the NICE properties[3]. These are **N**egation, **I**nversion, **C**ode (otherwise known as post-verbal or verb phrase ellipsis) and **E**mphatic affirmation and although some syntacticians, e.g. Sag et al. (2003) use the acronym for **N**egation, **I**nversion, **C**ontraction (of negative forms by the enclitic n't - to be dealt with in the current paper alongside

[2] or another auxiliary, but always with a lexical verb as the final verb in the sequence, except in cases of ellipsis (see section 2.1.3)

[3] an acronym coined by Huddleston (1976), which renamed features already documented by e.g. Palmer (1965), among others.

modal	infinitive	-en	-ing	-en
can	take			
can	be		taking	
can	have	taken		
can	have	been	taking	
can	be			taken
can	be		being	taken
can	have	been		taken
can	have	been	being	taken

Table 2: Primary paradigms for co-occurrence in auxiliaries, including modals

negation) and **E**llipsis (an alternative term for *Code*, as above), there is broad agreement on the syntactic facts.

2.1.1 Negation

In English, negation can be achieved by the placement of the negative particle *not*, or its cliticised form *n't* after an auxiliary verb. The same is not the case for lexical verbs, as can be seen in examples 5 to 7.

(5) a. She could eat cheese.
b. She could not eat cheese.[4]
c. She couldn't eat cheese.

(6) a. She has eaten cheese.
b. She has not eaten cheese.
c. She hasn't eaten cheese.

(7) a. She eats cheese.
b. *She eats not cheese.
c. *She eatn't cheese.

[4]This example is ambiguous between whether it is the auxiliary or the main verb being negated. It could be argued that certain lexical verbs can also be negated in this way, however, in this case the negation can only apply to the following verb ('eat') and the whole clause can also be negated. A cliticised version is also not possible. See Palmer (1988), for detailed discussion.

(i) I prefer not to eat cheese.
(ii) I don't prefer not to eat cheese.
(iii) *I prefern't to eat cheese.

In order to negate a modern English sentence which does not contain an auxiliary (as in 7a), it is necessary to supply one. This is the function of *do-support*, whereby a form of do is added to the sentence, to support the negative particle.

(8) She doesn't eat cheese.

The auxiliary *do* differs from the modal auxiliaries, however, in that, like lexical verbs and the auxiliary forms of *have* and *be*, it has third person forms, as well as a genuine past form.

(9) *She don't eat cheese.[5]

In uses of negation (and inversion (section 2.1.2) and emphasis (section 2.1.4)), the auxiliary *do* is usually considered to be semantically empty, and is sometimes, therefore, referred to as *dummy*-do.

While the above description of negation holds true for most auxiliary verbs, there are idiosyncracies, especially where the clitic form of the verb is used, for example, most commentators agree that *mayn't* is unacceptable (however, this is disputable) and it is certainly the case that there is no negative cliticised form of *am not* (although in certain contexts, e.g. tag questions, *aren't* fills the role, and in certain dialects *ain't* is acceptable). In addition, the cliticised negative of *will* (*won't*) bears little phonetic similarity with its positive counterpart, which is also true of *shall* (*shan't*). Although the usual interpretation of these negative auxiliaries is as the auxiliary + *enclitic not*, these idiosyncracies have led some commentators to analyse them as separate inflectional forms (e.g. Zwicky & Pullum (1983); see section 6.3.4 for discussion).

2.1.2 Inversion

A similar pattern in modern English can be seen in cases of inversion. In various constructions, the most common being different types of questions, auxiliary verbs can *invert* with the subject (thus preceding it), whilst lexical verbs cannot. Examples of inversion in different contexts, including with dummy-*do* insertion are shown in examples 10 - 14, below.

(10) a. She will like him.
b. Will she like him?

(11) a. She likes him.

[5] This is acceptable in certain dialects suggesting that in these dialects *do* is more aligned with the modal auxiliaries than in Standard British English of the type discussed here.

b. *Likes she him?

c. Does she like him?[6]

(12) a. If I had known you were late, I would have waited.

b. Had I known you were late, I would have waited. *(Conditional)*

(13) You are being facetious, aren't you? *(Tag question)*

(14) What have you done? *(Non-subject interrogative)*

2.1.3 Code

Code[7], or ellipsis, refers to sentences where a full verbal phrase can be effectively picked up, thereby avoiding needless repetition. In English, this function is carried by the auxiliary verbs, for example in 15, below, the second half of the sentence is understood as meaning "Julia can run very fast, too", where the auxiliary *can* stands in for the whole of the complex verbal phrase explicitly stated in the first half of the sentence. In 16 we can see that again, as with cases of negation and inversion, dummy-*do* can step in as the code for lexical verbs which cannot themselves support VP ellipsis, as shown in examples 17 and 19. It is important to notice, however, that while this is true of all lexical verbs, when verbs take a verbal complement with a to-infinitive, VP-ellipsis *is* possible, provided the *to-* infinitival marker is included, suggesting that this element in some way licenses the ellipsis (see the example petulant answers in 18). Further common examples can be seen in questions and answers, as in examples 20 and 21.

(15) Alice can run very fast, and Julia can, too.

(16) Jenny wants to go to the zoo but Martin doesn't.

(17) *Lisa wears pretty clothes, and Isabel wears, too.

(18) a. I'm sorry, we can't go to the zoo.

b. *But I want!

c. But I want to!

(19) Did you see the race last night? *Yes, I saw.

---

[6]Properly, this could be seen as the inversion of "She does like him", but this is usually considered to be semantically synonymous with 11a.

[7]The term *code* was coined by Firth (1968), and a neat illustration of his terminology is quoted in Palmer (1988) pp.20:

Do you think he will?
I don't know. He might.
I suppose he ought to, but perhaps he feels he can't
Well, his brothers have. They perhaps think he needn't.
Perhaps eventually he may. I think he should and I very much hope he will.

The 'key to the code' is *join the army*.

(20) Did you see the race last night? Yes, I did.

(21) Can I stay up and watch the game? No, you can't.

2.1.4 Emphatic affirmation

The final NICE property is Emphatic affirmation, where the stress or accent is on the auxiliary verb, as shown in examples 22 - 25. As Palmer (1988) states, this is a less clear-cut property, as lexical verbs can also take the accent, however;

> What is essential about the use of the auxiliaries is that they are used for emphatic affirmation of a doubtful statement, or the denial of the negative... (pp.21)

(22) I CAN do it.

(23) He HASN'T finished his homework.

(24) I DID do it. *(And you are wrong to think I didn't.)*

(25) I CAN'T cook. *(Who told you that I can?)*

2.1.5 Additional Characteristics

Another important characteristic exhibited by auxiliary verbs, and not shared by full verbs is contraction, illustrated in examples 26 - 29, below. Although it is only forms of *be* (*'m, 's, 're*), *have* (*'ve, 's, 'd*) and *will/would*, or occasionally *shall/should* (*'ll, 'd*) which contract, in a similar way, all the auxiliary verbs have phonologically weakened forms, which the cliticised versions illustrated below could be seen to be more extreme forms of.

(26) I'll do it. *(I* WILL *do it.)*

(27) He's finished his homework. *(He* HAS *finished his homework.)*

(28) I'd've done it myself. *(I* WOULD HAVE *done it myself.)*

(29) I'm waiting for him. *(I* AM *waiting for him.)*

2.2 Exceptions

Although the picture presented above seems to show that auxiliaries are a well delimited and highly grammaticalised type of function word (at least in modern English), this by no means indicates that there is any agreement on how they should be treated in the literature. Both the existence of exceptions and the diachronic evidence that this is a fairly recent development of English (occurring at some point in or after the 1600's), as well as cross-linguistic evidence (from, for example, Spanish, where auxiliaries share semantic features with English auxiliaries, but do not have the characteristic NICE properties and are thus harder to define) mean that any attempted explanation of the patterns exhibited by auxiliary verbs in modern English has many exceptions and counter-examples to take into account.

2.2.1 Dare/Need

For a start, the ‘core’ modals and forms of *do*, *have* and *be* are not the only lexical items which exhibit auxiliary-like behaviour. The two oft-quoted examples of *need* and *dare* sometimes behave like auxiliaries, and sometimes do not, leading most to conclude that they represent two completely separate lexical words; a full (lexical) verb, and an auxiliary (e.g. Palmer (1988)). In cases of negation and inversion, they can behave in exactly the same way as the modal auxiliaries, with the following verb being a bare infinitive. In declarative contexts, however, if the complement is verbal, then it usually takes the form of a to-infinitive, analogously to lexical verbs such as want. This can be seen in example 30, taken from Hudson (1997).

(30) a. *She dare/need jump.
b. Dare/need she jump?
c. She daren’t/needn’t jump.
d. She dares/needs *to* jump.

Evidence in support of the idea that these are two separate lexical items includes the observation that, as in declarative sentences where the existence of the full lexical verb has in some sense replaced or triumphed over its auxiliary counterpart, so the same is possible in negative and inverted contexts (as in example 31, below). As Hudson (1997) states, “In short, these auxiliaries are severely restricted and becoming more so, and their eventual replacements are already in use”

(31) a. She doesn’t dare/need to jump.
b. Does she dare/need to jump?

However, even this is a simplification, as there are sentences which seem to combine features of the auxiliary and non-auxiliary forms, especially in the case of dare. In both examples 32a and b, the verbal complement is a bare infinitive form as with the auxiliary form of dare, however, do-support is also evident in 32a, and a past tense inflection in 32b, which are both associated with the full verb form.

(32) a. She doesn’t dare jump.
b. She dared not jump.

2.2.2 Ought/Used (to)

Another item whose classification divides commentators, is *ought (to)*. Denison (1993) regards it as a 'marginal modal' due to morphological and syntactic differences between it and the 'core' modals, but acknowledges that semantically it lies on a continuum with *must* and *should*. Contrarily, Coates (1983) claims that "apart from the to-infinitive, it presents no problem: it has all the formal characteristics listed above [the NICE properties and lack of inflectional forms CH], besides clearly belonging to the same semantic set." In contrast, another marginal auxiliary, *used to* is usually considered to fall outside the class of auxiliaries, and seems to be becoming a full verb for many speakers, as illustrated in example 33.

(33) a. ?Used she to go to Brownies?
b. Did she used to go to Brownies?

2.2.3 Possessive Have / Existential Be

The interference between lexical and auxiliary verbs which have identical forms (as seen with *dare* in 2.2.1, above) is also observed in the non-auxiliary forms of *have* and *be*, which exhibit the NICE properties, even in their lexical forms. The main difference between these uses and the auxiliary uses is in the fact that they have a non-verbal complement. Interestingly, some of the examples in 35 sound archaic, more normally replaced by *have got*, as in 36. According to Hudson:

> The alternatives make the future of possessive auxiliary *HAVE* uncertain. The loss of this transitive auxiliary can be seen as the last battle in the war against transitive auxiliaries which has been taking place since the 17th century ..., when *WILL* stopped allowing an object (e.g. *I will an apple*).

(34) a. It's a dog.
b. It isn't a dog. *(negation)*
c. Is it a dog? *(inversion)*
d. It's a dog, isn't it? *(code / ellipsis)*
e. It IS a dog. *(emphatic affirmation)*

(35) a. I've a dog.
b. I haven't a dog. *(negation)*
c. Have you a dog? *(inversion)*
d. I have a dog, haven't I? *(code / ellipsis)*
e. I HAVE a dog. *(emphatic affirmation)*

(36) a. I've *got* a dog.
b. I haven't *got* a dog. *(negation)*
c. Have you *got* a dog? *(inversion)*
d. I have *got* a dog, haven't I? *(code / ellipsis)*
e. I HAVE *got* a dog. *(emphatic affirmation)*

2.2.4 Do

Auxiliary *do* is almost universally recognised as an exceptional case. It does not seem to contribute any semantic information at all, merely being used in exactly (and only) those situations when an auxiliary is required but not already present - as in cases of *negation* (section 2.1.1), *inversion* (section 2.1.2), *code* (section 2.1.3) and *emphasis* (section 2.1.4). This means that strings that already contain an auxiliary *cannot* contain auxiliary *do*, and explains why, despite sharing many similarities with the modal auxiliaries, auxiliary *do* can never co-occur with other auxiliaries as the modals can (see example 37d, below). In addition, although like the modals non-finite forms of auxiliary *do* are not possible (as in 37f and g), unlike the modals, tensed forms are, as noted in section 2.1.1. Just to further confuse the picture, unlike possessive *have* and existential *be* (discussed above in section 2.2.3), the lexical form of *do* does not exhibit the NICE properties of negation and inversion (see example 38)

(37) a. I can jump.
b. I do jump.
c. I can have jumped.
d. *I do have jumped.
e. I have been jumping.
f. *I have done jump.[8]
g. *I am doing jump.

(38) a. I do the dishes.
b. *Do I the dishes?
c. *I don't the dishes.

2.2.5 Quasi-auxiliaries

Of course, these are not the only exceptions, and whole chapters have also been devoted to the *quasi-auxiliaries*, which are a group of complex phrases which exhibit semantically auxiliary-like properties. Examples are *be going to, have to* and *be able to*. However, since this paper is focussing on the syntactic properties of the auxiliary system, I will not be going into detail about them here.

[8]The superficially similar "I have done *a* jump" is clearly using the lexical form of *do*, not the auxiliary.

3 Existing Accounts of the Auxiliary System

Previous attempts to account for the auxiliary system are varied and complex. Some lay their focus on the semantic characteristics of auxiliaries, whilst others focus exclusively on syntactic properties, which, as Denison (1993) points out, can be a moot point for “those sceptical about the autonomy of syntax” (pp.325). Indeed as Palmer (1988) notes;

> It is almost certainly the case that any semantic distinction can be matched somewhere in the language by a formal one and that any formal regularity can be assigned some kind of meaning. It is not, then, a matter of form versus meaning, but of the weighting to be given to obvious formal features and to fairly obvious semantic ones.

Additionally, some explanations use diachronic or cross-linguistic evidence in their support, whilst others take a ‘snapshot’ of existing language use, usually acknowledging, but not exploring, dialectic differences.

3.1 Transformational approaches

The classic transformational approach to auxiliary verbs, dating back to Chomsky (1957, 1965), posits new categories and rules akin to those shown in 39, below. Although these are able to handle the linear sequencing of auxiliaries (with one optional modal, and perfective forms preceding progressive) it poses no additional restrictions on the form of the verb phrase following the modal, nor does it adequately capture semantic facts (e.g. of scope, where multiple auxiliaries are present; see Langacker (1978) for an early discussion)

(39) a. S → NP AUX VP
b. AUX → (M) (*have* + *en*) (*be* + *ing*)
c. M → *may, will, can, shall, must, . . . ,*

Although transformational grammars have moved on since those early days, their explanatory power is diminished by the additional machinery which they posit in order to account for language universals. Without going into too much detail, the basic assumption of a modern movement based account is that, although in earlier forms of English, as well as, for example, modern Romance languages such as French and Spanish, *all* verbs could be raised to a higher structural position, in modern English, only auxiliary verbs are able to undergo the same process. These differences are usually couched in terms of head-movement, which, it is alleged “accounts for lots of nifty variation among languages” (Harley (2002)). Although analyses seemed to show promising results, e.g. in the postulation of a single parameter of V-to-T movement to account for aspects of language differentiation, the Principles and Parameters approach leads to what Kim & Sag (2002) refer to as "the supposition of unwelcome ancillary devices or nontrivial complications in other grammatical components", meaning that this approach does not account for the behaviour of English auxiliary verbs nearly as neatly as it claims to. Further problems for the principles and parameters approach, with reference to the historical development of the English auxiliary system, are discussed in section 3.3.1, below.

3.2 Lexical Approaches

In contrast to this type of analysis, there are several lexicalised approaches, which look to differences in the way in which lexical items are stored and accessed in order to account for both general patterns (including those relating to auxiliaries as outlined above) and exceptions (e.g dare/need in section 2.2.1, above). The way in which a few of these have attempted to account for auxiliary verbs are outlined below.

3.2.1 HPSG

Head-driven Phrase Structure Grammar deals with the class of auxiliary verbs by treating them as a sub-group of subject raising verbs. Like subject-raising verbs (such as *tends*), auxiliaries take verbal complements, and 'share' their subject semantically with their verbal complement (see Sag et al. (2003), Chapter 13 for further details). However, although this account is able to capture the data adequately, it does so by the postulation of numerous additional features, such as [AUX ±], and [INV ±] (for *aux*iliary and *inv*erted, respectively). Many of their so-called lexical rules, which are assumed to apply to all verbs, for example, have each of the auxiliaries (especially the modal ones) listed as exceptions. As well as introducing unnecessary redundancy into the system, this criticism is, perhaps, one that is avoided by approaches such as Falk's (outlined below, 3.2.2) which does not treat all auxiliaries as a homogenous group. In addition, despite HPSG supposedly being a system where semantic and syntactic information is built up in tandem, they have no way of explaining the ambiguities inherent in the modal auxiliaries, which, as noted as early as Ross (1969), means that in some cases auxiliary verbs behave like subject control verbs, not subject raising verbs (see example 40, below). By restricting the auxiliary lexeme class to being a subset of subject raising verb lexemes, they do not capture these semantic facts, and it is not easy to see how they would do so without either a radical reshuffling of the verbal types or duplicating lexical entries.

(40) a. I can put the kettle on. *(There is the possibility of my putting the kettle on - subject raising interpretation)*
b. I can put the kettle on. *(I have the ability to put the kettle on - subject control interpretation)*

It is interesting to note that Meza & Pineda (2002a,b) encounter exactly this problem in their analysis of auxiliary verbs in Spanish, which they resolve by postulating different lexical entries for the subject raising and subject control readings of *poder* (be able to / can).

3.2.2 LFG

Like Sag et al., Falk (2003), recognises that there are two basic different approaches to auxiliary verbs in the literature. These he describes in Lexical Function Grammar (LFG) terminology as *aux-predicate* and *aux-feature* analyses. These two approaches represent different syntactic relations between the auxiliary and lexical verb elements of a sentence as illustrated by 41, below (his example 1) whereby 41b represents the *aux-predicate* analysis and 41c the *aux-feature* analysis.

(41) a. The children will take syntax.
b. *will* is the head of the sentence, and *[take syntax]* is a complement of *will*
c. *take* is the head of the sentence, and *will* is a “modifier” or morphological marker expressing/realising future tense.

In essence, the earlier approaches which considered *AUX* to be a separate category (like the early transformational approach discussed in section 3.1) can be categorised in Falk’s terms as an *aux-feature* approach, whilst HPSG, for example (section 3.2.1, above), which considers auxiliaries to be verbs which take verbal complements, is an *aux-predicate* approach.

Falk’s analysis, however, concludes that neither approach is sufficient to account for the behaviour of all the auxiliary verbs. He states that there “is no single analysis that covers all auxiliaries”. In consequence, he categorises progressive *be* and the modal verbs (with the notable exception of *will* and *would*) as of the *aux-predicate* type, whilst supportive *do* and perfective *have*, as well as *will* and *would*, are mere feature carriers (this analysis treats *will* as a carrier of the future tense feature). However, although Falk offers some compelling arguments for his categorisation of auxiliaries, even if we accept his arguments that not all auxiliaries can be treated alike, his solution is not the only possible one.

Schütze (2003), for example, proposes a different division, and also offers persuasive arguments in support of it. In Schütze’s account, all the modal verbs, including *will* and *would*, form one set, of which supportive *do* is also a member (which Schütze terms M, for mood, in his movement account), in contrast to *be* and *have*, which are members of V (verb). This distinction, based purely on morphosyntactic facts, is illustrated by many different types of example, of which Mad Magazine sentences (from Akmajian (1984)) as in example 42, below, are just one, relating to the fact that auxiliary *do*, like the modals, has no non-finite forms.

(42) a. What? Her be out all night? Never!
b. What? Him be drinking at 9 in the morning? Never!
c. ?What? John not have finished his homework by 9pm? Never!
d. *What? Him do/does not pick up the kids on time? Never!
e. *What? Him should/must/could leave the firm? Never!

In both accounts briefly outlined above, however, there are questions regarding how to account for the fact that wherever the division is placed, if there are two (or more) distinct groups posited for what have traditionally been termed auxiliary verbs, then the task becomes one of explaining why these distinct classes share so many idiosyncratic syntactic features (the NICE properties, discussed in section 2.1, above). The question remains whether to treat auxiliaries as a distinct class of linguistic objects with some distinct properties, of which certain elements are exceptional in one or more ways, or as more than one distinct set of objects which are exceptional in (some of) the same ways. This type of view seems to me to add additional, and unnecessary levels of complexity. Like the HPSG approach (section 3.2.1), I believe that the apparent differences between members of the set of auxiliaries are best explained at the level of the lexicon, so that the similarities between them becomes the focus.

3.3 Historical Approaches

One of the reasons that auxiliary verbs are such a well studied group of words, apart from the fact that they are a strictly limited class with important and specific properties, is due to their historical development. In English, it is well attested that certain linguistic facts, which are now limited to auxiliary verbs (e.g. *negation* and *inversion* as discussed in sections 2.1.1 and 2.1.2, above) were in fact available for all verbs, even as recently as Shakespearean times, as shown in the examples taken from Othello (43), below, although as can be seen from 43d, this was just one alternative, with the modern construction involving periphrastic *do* also available.

(43) a. But if you know not this... *negation* (I.1.130)
b. Are they married, think you? *inversion, intransitive case* (I.1.168)
c. How got she out? *inversion, transitive case* (I.1.170)
d. Where didst thou see her? (I.1.164)

Looking back further, (as, for example, in Getty (2000)), the evidence shows that the words from which the auxiliary verbs evolved had corresponding full forms, including full semantic meanings and various inflections. With minor disputes, these facts are agreed on (see Denison (1993) for a comprehensive overview). What is less clear is the question of why or how such changes took place.

3.3.1 A Principles and Parameters Explanation

A principles and parameters approach to this question postulates that a single movement parameter is responsible. In earlier forms of English (roughly up to the 1600s), movement from V-to-I (or V-to-T, depending on the specific account) was allowed for all verbs, allowing them to raise to a structurally higher position. The modern English setting of the parameter restricts it so that it is only applicable to the auxiliary verbs.

A major problem of this approach is that is requires us to treat languages with differently set parameters as distinct grammatical systems. Although this may not be a problem when we consider the differences between, say, English and Spanish, it is a less satisfactory conclusion when we are concerned with diachronic changes in a single language. In Kroch (1994), for example, he says

> Because the variants of the syntactic changes we have studied are not susceptible of integration into a single grammatical analysis, the variation does not stabilize and join the ranks of a language's syntactic alterations. Instead, the languages always evolve further in such a way that one or the other variant becomes extinct.

Presented with evidence that the rise of do-support in English did not represent the step-change that such a view predicts, with two statistically distinct periods of change apparent, Han & Kroch (2000) add an additional *mood* node to the already complex tensed node (T). They can then claim that the rise of do-support corresponded to two separate parameter changes, one of which was the loss of M-to-T. This increased complexity in order to deal with one small data problem in order to keep the rise of do-support fully explained by the Principles and Parameters program appears to be an arbitrary one. Furthermore, it fails to explain why, if two grammars (with one parameter set differently in each) are conflicting, certain lexical items are affected (or switch to the 'new' parameter setting) at different times, both within and between speakers. In Dynamic Syntax, as it is a lexicalised grammar, we can predict that certain lexical items might be affected by any change earlier than others, and that some might not be affected at all. In the case of auxiliary verbs it is hypothesised that the change that occurred was in the 'trigger' conditions, or constraints, for each verb (see section 6.1). This possibility fits in with Engblom's observation, back in 1938, that;

> It is scarcely true to say that it was unusual for the particle not or the subject of an interrogative sentence to follow any verb but an auxiliary, as such phrases as *I went not*, or *Heard you?* were very frequently used for centuries and, in many cases, are still in use. (pp.31)

In short, the evidence does not seem to support the claim of Traugott (1972) (which seems to be a basic assumption for those following the Principles and Parameters program) that "...change is instantaneous. There is nothing gradual about acquiring a pattern; the moment it becomes part of one's competence, ..., one's competence is instantaneously changed." (pp.13).

3.3.2 Cognitive Explanations

An alternative approach to the data, exemplified by Warner (1993), is in terms of cognitive classes. The grammaticalisation of the auxiliaries into a readily identifiable distinct class of words is taken to show the gradual redistribution of characteristics relating to full verbs and auxiliaries. Table 3, below (taken from Hudson (1997)), shows the different characteristics originally applicable to both sets of words, and roughly when the detailed changes occurred. However, unlike the parameter setting approach, this change is assumed to have been a gradual one, with competing pressures eventually resulting in preferred choices driving out the alternatives. According to Warner, this long-term development demonstrates Rosch's principle of cognitive economy, which has "to do with the function of category systems and asserts that the task of category systems is to provide maximum information with the least cognitive effort" Rosch (1978). To this end, the cognitive system is presumed to favour distinct categories with highly correlated features. In this way, Rosch's general principle of cognitive effort and Relevance Theory's cognitive principle of relevance, as applied to language processing, despite being unrelated, are both concerned with the minimisation of cognitive effort, and therefore rely on mutual assumptions.

Distinctive characteristic of auxiliary verb	Exceptions	Example	Date	Changed
V allows VP ellipsis and pseudo-gapping		...it would — me —	OE	—
V < adverb		*ran never	15c	full
V < subject		*ran you?	15c	full
V < not		*ran not	15c	full
V reducible to clitic	OUGHT, USED, DARE, NEED	's going	16c	aux
V in tag question		..., is he?	16c	aux
V needs VP complement	HAVE_{poss}, BE	*will coffee	17c	aux
V not ~ V-n't		isn't	17c	aux

Table 3: Characteristics of auxiliary verbs in Modern English (from Hudson (1997), pp.57)

The dating facts, as shown in Table 3 lead Hudson (1997) to conclude that

> Full verbs changed before auxiliaries did...Consequently, rather than seeing the change as the rise of the auxiliary class, it would be better to see it as simply the separation of two classes, neither of which has any particular priority.

This view contrasts directly with the principles and parameters approach outlined above in section 3.3.1 and advocated by Kroch (1994), for example.

Most commentators agree, however, that without *do* being available as a dummy auxiliary, the language changes (that all agree occurred) would not have taken place. Denison (1993) outlines several explanations that have been offered as the reasons for the origins of periphrastic *do*. These include emphatic usage, ambiguity resolution between nouns and verbs (e.g. *sin*) when the English inflectional system was reduced (meaning that, in its role as a generic action verb, substitute-*do* was added to avoid confusion), and language contact effects (which explanations range from a Germanic V2 like effect to Celtic, to French). Engblom (1938) (and later Ellegård (1953)) argues that, as periphrastic *do* was first recorded in verse, not prose, it arose as a metrical device, whose usage spread due to other, functional, pressures. Although it is not the purpose of this paper to speculate as to which, if any, of these theories on the rise of periphrastic *do* is correct, the question is an important and unresolved one. Without the existence of a "meaningless and optional auxiliary verb" (pp.62), the functional and cognitive changes which Hudson (1997) claims led to the delineation of the class of auxiliary verb cannot be motivated.

Given this presupposition, Hudson hypothesises on the particular functional pressures which might have occurred, which include a range of constructions whose tendency to include an auxiliary might have influenced those situations where either an auxiliary or a full verb construction was possible. These include ambiguity resolution, as illustrated in example 44. In the case of transitive verbs, especially those with a potentially interchangeable subject and object (because for example, they both refer to people), an empty auxiliary could be used to disambiguate between possible left dislocation and inverted subject question environments (which in speech, but not in print, could be identified by intonation). Two possible interpretations of 44a are shown in 44b and c. Another contributing factor could have been the small but statistically significant preference for negation to co-occur with an auxiliary verb, related, possibly, to those contexts in which negation is likely to occur, for example in conducive questioning, where a negative yes/no question is considered to be a conducive way of finding things out and not just a direct negation of its positive counterpart (see e.g. 45, from Hudson, pp.62).

(44) a. Saw Mary John?
b. Did Mary see John? *inverted subject reading*
c. Did John see Mary? *left-dislocated topic reading*

(45) a. Can't you swim?
b. Can you swim?

Hudson identifies three rough stages in the development of the English auxiliary system. they are as shown in 46 - 48, (Hudson's examples 11 - 13, pp.63) below.

(46) Stage A (as in early Middle English)

a. You may invert any verb, whether auxiliary or full.

b. You may negate any verb, whether auxiliary or full.

(47) Stage B

a. You may invert any verb, whether auxiliary or full; but;

a.′ Do not invert a full verb if it is negated or has an object.

b. You may negate any verb, whether auxiliary or full

(48) Stage C (as in Modern English)

a. You may invert any auxiliary verb.

b. You may negate any auxiliary verb.

In his terminology, the transition from Stage A to Stage B is driven by functional pressures (as in examples 44 and 45, above) whilst that from Stage B to Stage C is driven by cognitive ones (relating to Rosch's Principle of Cognitive Effort, because the grammar in Stage B is a "... cognitively [more] complex grammar ... " (pp.63)).

Many commentators, however, doubt that the types of construction in 44 and 45 could be a locus of change, because such processing preferences seem neither sufficient or necessary to account for such categorical change. A further, frequently asked question relates to the fact that there are two concurrent ways of expressing simple present or past tense, and questions why we still have any tensed forms, given the possibility of replacing "I went" with "I did go". These questions are orthogonal to the current discussion, however the question of whether the two sentences are, in fact, exactly synonymous has a bearing on how we analyse auxiliary do (see section 2.2.4). It is true that the above examples seem small and insignificant when faced with the task of bringing about wholesale change to the verbal system, however, using the principles of Relevance Theory, I believe we can begin to come up with a plausible story of that change.

3.3.2.1 A Relevance Theory Explanation of the Diachronic Data

According to the principles of Relevance Theory (outlined above in section 1.2), in communications, all efforts are made to minimise cognitive costs. In examples like (44), this desire for a clearly disambiguated question would be naturally preferred, if there were no other factors (e.g. previous discourse) which led to its disambiguation. With 44b being possible, this pragmatic consideration would, over time and in affected circumstances, lead to the routinisation of the actions required - in this case, the inclusion of an auxiliary verb in inverted situations where an object is present. This, in turn, would lead to the situation where different strategies were available (as illustrated by Shakespeare in 43, above), which, again, would require additional cognitive effort to process. As auxiliary usage became the standard in negation and transitive verb cases, we should expect that this strategy, in principle also available for other instances should come to be used more widely. In other words, it is cognitively more efficient to take the already encoded (situation specific) routinisation and apply it more generally. This brief hypothetical account, from a Relevance Theory perspective, mirrors Hudson's conclusions exactly.

Moreover, Relevance Theory can also offer insights into semantic change. In grammaticalisation studies so-called *semantic bleaching* is often considered to be a key factor, whereby a full verb in some sense loses its meaning. Kuteva (2001), however, argues against such a simplification, as what is sometimes seen as bleaching can lead to a word actually gaining new meanings. In essence, I agree with her arguments, which assert that although meanings usually do change as part of the process of grammaticalisation, this is by no means always in the direction of simplification, and that to talk of *semantic bleaching* is at best misleading and at worst completely mistaken.

That said. in simple cases, such as *will* and *would*, we can see how Relevance Theory can be invoked to account for the change in meaning from the Old English *willan*, meaning 'to want' or 'to desire', into its current (main) meaning of futurity. The inference associated with wishing for something, or desiring it, is that you will try to achieve it, or make it happen. It is a short step from this being merely an implicature, to an intrinsic part of the word's meaning, to its main or sole meaning. In Chaucerian English, examples using *wil/wol* or *wolde* are often ambiguous between a desire reading and a futurity meaning, as in e.g. 49a, below, and both uses of *will* were still available at least up to Shakespearean times (as can be seen in 49b). In addition, in certain calcified uses, we can see that even today, it can have associated meanings, as in 49d and e. Similar observations are possible for *can* and *could*, (see 49c) where the original meaning translates as "know how to" (compare this to Scots *ken*).

(49) a. *For sothe, I wol nat kepe me chaast in al*
Forsooth, I will not keep me chaste in all
'Forsooth, I don't want to/won't be chaste at all'
From Chaucer's The Wife of Bath, (line 46).

b. Where will you that I go? *From Shakespeare's Othello, (I.2.85).*

c. *For half so boldely kan ther no man / Swere and lyen, as a woman kan*
For half so boldly can there no man / swear and lie as a woman can
'For no man knows how to/can swear / half as well as a woman'
From Chaucer's The Wife of Bath, (line 226).

d. What would you have me do?

e. Would that I could.

It is interesting to note that *want* with an infinitival complement, also has the implication of futurity in its meaning. These and related observations led Bolinger (1980), to conclude that "the moment a verb is given an infinitive complement, that verb starts down the road of auxiliaryness." For further discussions and evidence regarding the explanatory power of Relevance Theory in cases of auxiliation, see Kuteva (2001).

4 Dynamic Syntax Framework

In order to give an account of the English auxiliary verb system in Dynamic Syntax, which tries to accommodate the data discussed in section 2, and has as much or more explanatory power as accounts based in different frameworks, as discussed in section 3, it is first necessary to provide an overview of the formal tools used in Dynamic Syntax.

4.1 Formal Tools

In Dynamic Syntax, trees are built up incrementally. Strings of words (in a sentence or utterance) are processed in a strictly left-to-right manner, in the order in which they are encountered. The trees that are thus produced are not traditional syntactic trees, which preserve the word order of the string in the final tree, but interpretations, analogous to Montagovian semantic trees. For this reason, despite different word orderings, different strings can lead to the same final tree, e.g. *John loves Mary* and its semantic equivalent with a left dislocated topic *Mary, John loves* (or even *Mary, John loves her*). The steps used to reach the output tree are therefore of vital importance, as it is through these partial trees that the notion of growth and the syntactic impact of any particular word in the parse can be seen. Any string encountered is grammatical just in case there is a sequence of steps which leads to a completed tree when all the words have been parsed.

The nodes in the output tree and in all partial trees during the parse are annotated with various decorations. These give information about the current tree node, telling us, for example, what type of node it is. In Dynamic Syntax, there are three basic types; propositions (full sentences) are of type t, for truth conditional, which is depicted as $Ty(t)$. Noun phrases are $Ty(e)$, for entity, and common nouns are $Ty(cn)$. Complex types are built up from the basic types, for example, verb phrases (one-place predicates) are functions from entities to truth values, i.e. $Ty(e \rightarrow t)$. Two-place predicates, such as transitive verbs, are $Ty(e \rightarrow (e \rightarrow t))$, and so on.

Other decorations include a formula value (analogous to lambda terms in Montagovian grammars) of the form $Fo(John')$, which, by the rules of the grammar can be combined to form complex expressions such as $Fo(Loves'(Mary')(John'))$. A further, vital, decoration is the tree node address. This is stated in relative terms, based on the root node of the tree under construction being $Tn(0)$, and each daughter node being assigned an additional one for a functor daughter or zero for an argument daughter. For example, $Tn(010)$ would be the argument daughter of the functor daughter of the root node. Importantly, the tree node address can also be described, using the logic of finite trees (LOFT, Blackburn &

Meyer-Viol (1994)), by its relation to any other tree node. This uses two modal operators to signal daughters or mothers of a node ($\langle\downarrow\rangle$, $\langle\uparrow\rangle$). Additional subscripts 0 or 1 indicate whether this refers to an argument or functor daughter/mother ($\langle\downarrow_0\rangle$, $\langle\downarrow_1\rangle$, $\langle\uparrow_0\rangle$ and $\langle\uparrow_1\rangle$ respectively), and can be underspecified using the Kleene star as a subscript. $\langle\uparrow_*\rangle Tn(0)$, for example, means that the root node is somewhere above (or at) the current node.

All these labels can either be complete, stating something about what has already been successfully parsed, or requirements, indicating what else is required to complete the current partial tree. Unlike complete descriptive decorations, requirements are preceded by a question mark; $?Ty(t)$ is a requirement for a proposition, $?\exists\mathbf{x}.Tn(\mathbf{x})$ is a requirement for a fixed tree node address ($\mathbf{x}$ is a variable indicating that any tree node address will fulfil the requirement). Only when all requirements are fulfilled can the string be said to be grammatical. In addition, there is a pointer in the tree, depicted by the $\diamondsuit$ symbol, which indicates the node currently under construction. This is an important feature of Dynamic Syntax, as it immediately explains the ungrammaticality of certain strings. This is because it sets an immediate restriction on what rules or lexical actions can occur at any given point in the parse. For example, the string *John hits sing* would be ungrammatical precisely because when the parser attempts to parse *sing*, the pointer will be at a node with a requirement for a $Ty(e)$, which will not match the information held in the lexical entry for *sing*.

4.2 Lexical Actions

Like Head-driven Phrase Structure Grammar (HPSG, see Sag et al. (2003)) and Combinatory Categorial Grammar (CCG, see e.g. Steedman (2000)), Dynamic Syntax is a lexicalised grammar, acknowledging the fact that complexity in language relies to a large extent on information that is stored in the lexicon. Evidence for lexicalised grammars is extensive and varied. That the lexicon must be rich enough to allow for a multitude of idiosyncrasies is highlighted by the fact that many words do not behave in the way expected by other elements in the same syntactic category. For example, in English, the word "beware" only appears in its base form ('Beware of the monster!' is grammatical, for example, whilst '*she bewared of the monster' and '*she is bewaring of the monster' are not). Many cross-linguistic differences can also be explained by appeals to the lexicon, and the way it is organised. In HPSG, for example, the organisation of lexical items which take agreement features in Spanish includes determiners, whilst in English it does not (el, la, los, las = the). In the case of Dynamic Syntax, what is stored in the lexicon is a set of procedures, known as lexical actions. This contrasts with HPSG and CCG which store collections of feature structures and category specifications respectively.

Some simple Dynamic Syntax lexical items are shown below. The first, shown in 50, will be accessed when the word *John* is encountered in a string, and can be read as saying "if there is a requirement for an entity at the current node (determined by the position of the pointer $\diamondsuit$), then put the decorations for a type entity, and formula $Fo(John')$ at the current node.” The last decoration in this lexical entry $[\downarrow]\bot$ (‘below me, the falsum holds’), is the bottom restriction, which simply means that the node cannot be developed further. Finally, the lexical entry says that if there is not a requirement for an entity at the current node, then abort the parse process, i.e. fail to parse the string, as in the example *John hits sing*, discussed above.

(50) *John*

IF	$?Ty(e)$
THEN	$\texttt{put}(Ty(e), Fo(John'), [\downarrow]\bot)$
ELSE	$\texttt{Abort}$

The lexical entry for the verb *died* (51) is similar, but its trigger requirement is $?Ty(e \rightarrow t)$ and it uses modal operators to indicate that tense information should be put at the functors mother node. It can be read as stating “if there is a requirement for a $Ty(e \rightarrow t)$ at the current node, then go up one functor node, and put the decoration $Tns(PAST)$ at that node. Then go back down the functor node (to the original trigger node) and add the decorations for $Ty(e \rightarrow t)$, $Fo(Die')$ and the bottom restriction.”

(51) *died*

IF	$?Ty(e \rightarrow t)$
THEN	$\texttt{go}(\langle\uparrow_1\rangle); \texttt{put}(Tns(PAST));$
	$\texttt{go}(\langle\downarrow_1\rangle); \texttt{put}(Ty(e \rightarrow t), Fo(Die'), [\downarrow]\bot)$
ELSE	$\texttt{Abort}$

In the third case, (52), the transitive verb *loved*, the lexical entry creates new nodes for the verb and its object, using the operator $\texttt{make}()$ to create an argument and functor daughter for the $Ty(e \rightarrow t)$ node, and leaving the pointer at the argument daughter (object) node where it has placed a requirement for an entity ($?Ty(e)$).

(52) *loved*

IF	$?Ty(e \rightarrow t)$
THEN	$\texttt{go}(\langle\uparrow_1\rangle); \texttt{put}(Tns(PAST));$
	$\texttt{go}(\langle\downarrow_1\rangle); \texttt{make}(\langle\downarrow_1\rangle); \texttt{go}(\langle\downarrow_1\rangle);$
	$\texttt{put}(Ty(e \rightarrow (e \rightarrow t)), Fo(Love'), [\downarrow]\bot);$
	$\texttt{go}(\langle\uparrow_1\rangle); \texttt{make}(\langle\downarrow_0\rangle); \texttt{go}(\langle\downarrow_0\rangle);$
	$\texttt{put}(?Ty(e))$
ELSE	$\texttt{Abort}$

4.3 Rules

Basic rules, which can apply at any point in the parse (Dynamic Syntax does not give us a strategy for choosing a rule at any given point, although of course there are restrictions on when they can apply) include the AXIOM (the default starting position for any parse), INTRODUCTION, and PREDICTION, shown below.

(53) AXIOM

$$\{\ldots ?Ty(t) \ldots \Diamond\}$$

(54) INTRODUCTION

$$\frac{\{\ldots\{\ldots ?Ty(Y)\ldots\Diamond\}\ldots\}}{\{\ldots\{\ldots ?Ty(Y), ?\langle\downarrow_0\rangle Ty(X), ?\langle\downarrow_1\rangle Ty(X\rightarrow Y),\ldots\Diamond\}\ldots\}}$$

(55) INTRODUCTION - SUBJECT AND PREDICATE

$$\frac{\{\ldots\{Tn(n), ?Ty(t), \Diamond\}\}}{\{\ldots\{Tn(n), ?Ty(t), ?\langle\downarrow_0\rangle Ty(e), ?\langle\downarrow_1\rangle Ty(e\rightarrow t), \Diamond\}\}}$$

(56) PREDICTION

$$\frac{\{\ldots\{Tn(n),\ldots, ?\langle\downarrow_0\rangle\phi, ?\langle\downarrow_1\rangle\psi, \Diamond\}\ldots\}}{\{\ldots\{Tn(n),\ldots ?\langle\downarrow_0\rangle\phi, ?\langle\downarrow_1\rangle\psi\}, \{\langle\uparrow_0\rangle Tn(n), ?\phi, \Diamond\}, \{\langle\uparrow_1\rangle Tn(n), ?\psi\}\ldots\}}$$

(57) PREDICTION - SUBJECT AND PREDICATE

$$\frac{\{\ldots\{Tn(0), ?\langle\downarrow_0\rangle Ty(e), ?\langle\downarrow_1\rangle Ty(e\rightarrow t), \Diamond\}\}}{\begin{array}{c}\{\ldots\{Tn(0), ?\langle\downarrow_0\rangle Ty(e), ?\langle\downarrow_1\rangle Ty(e\rightarrow t)\}, \{\langle\uparrow_0\rangle Tn(0), ?Ty(e), \Diamond\}, \\ \{\langle\uparrow_1\rangle Tn(0), ?Ty(e\rightarrow t)\}\}\end{array}}$$

The AXIOM merely states that we begin a parse with a single node consisting of a requirement for a proposition, and with the pointer at that node. INTRODUCTION (55) states that if we have a requirement for a proposition $Ty(t)$, then we can break the requirement down into requirements for a subject and a predicate; equivalent to a noun phrase ($Ty(e)$) and verb phrase ($Ty(e \rightarrow t)$), or, more generally (54), that given a requirement at the current node, we can break it down into requirements for a functor daughter and an argument daughter. PREDICTION takes these requirements and builds the nodes specified, decorating them with the appropriate requirements, and leaving the pointer at the argument daughter, again, the general rule is shown (56) as well as its specific form for the subject and predicate case (57). These three stages, using the subject and predicate forms of the general rules, effectively create a blank tree waiting for a subject and predicate, as shown in figures 1 and 2.

$$?Ty(t), ?\langle\downarrow_0\rangle Ty(e), ?\langle\downarrow_1\rangle Ty(e\rightarrow t)$$

Figure 1: Single node tree following INTRODUCTION

Given a simple example *John died* using the lexical entries shown in (50) and (51), the next step in the parse would be to parse the word *John*, as per the lexical actions discussed above, leaving the resulting tree, as in figure 3.

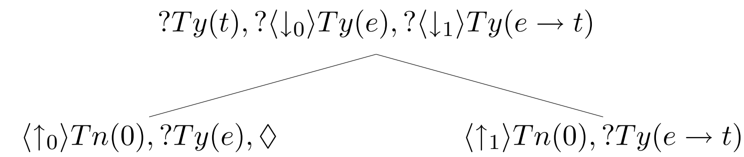

Figure 2: Tree following PREDICTION

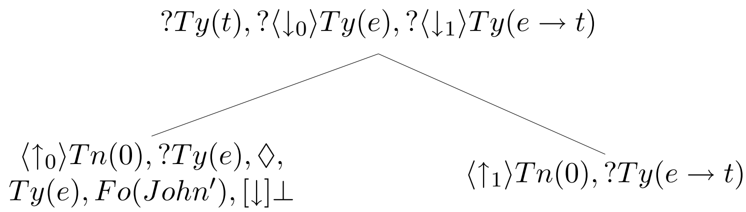

Figure 3: Parsing *John*

Further rules allow us to update the tree such that the pointer is at the node with a requirement for a type $Ty(e \rightarrow t)$, where we can parse the word *died*. They are THINNING (58), which allows us to remove a requirement decoration from a node if the completed form is also present (e.g. in the above tree, the decorations $?Ty(e)$ and $Ty(e)$ are both present on the *John* node), COMPLETION (59), which allows us to move the pointer to a mother node if no requirements are outstanding at the current node, and ANTICIPATION (60), which allows us to move the pointer to any daughter node with outstanding requirements.

(58) THINNING

$$\frac{\{\ldots\{\ldots,\phi,\ldots,?\phi,\ldots,\Diamond\}\ldots\}}{\{\ldots\{\ldots,\phi,\ldots,\Diamond\}\ldots\}}$$

(59) COMPLETION

$$\frac{\{\ldots\{Tn(n),\ldots\},\{\langle\mu^{-1}\rangle Tn(n),\ldots,Ty(X),\ldots\Diamond\}\ldots\}}{\{\ldots\{Tn(n),\ldots,\langle\mu\rangle Ty(X),\ldots,\Diamond\},\{\langle\mu^{-1}\rangle Tn(n),\ldots,Ty(X),\ldots\}\ldots\}}$$
$$\mu^{-1} \in \{\uparrow_0,\uparrow_1,\uparrow_*,L^{-1}\}, \mu \in \{\downarrow_0,\downarrow_1,\downarrow_*,L\}.$$

(60) ANTICIPATION

$$\frac{\{\ldots\{Tn(n),\ldots,\Diamond\},\{\langle\uparrow\rangle Tn(n),\ldots,?\phi,\ldots\}\ldots\}}{\{\ldots\{Tn(n),\ldots\},\{\langle\uparrow\rangle Tn(n),\ldots,?\phi,\ldots,\Diamond\}\ldots\}}$$

(61) ELIMINATION

$$\frac{\{\ldots\{Tn(n),\ldots,?Ty(X),\langle\downarrow_0\rangle(Fo(\alpha),Ty(Y)),\langle\downarrow_1\rangle(Fo(\beta),Ty(Y\rightarrow X)),\ldots,\Diamond\},\ldots\}}{\begin{array}{c}\{\ldots\{Tn(n),\ldots,?Ty(X),Fo(\beta(\alpha)),Ty(X),\\ \langle\downarrow_0\rangle(Fo(\alpha),Ty(Y)),\langle\downarrow_1\rangle(Fo(\beta),Ty(Y\rightarrow X)),\ldots,\Diamond\},\ldots\}\end{array}}$$
$$\text{Condition : } \langle\downarrow_i\rangle?\phi, i\in\{1,0\}, \text{ does not hold.}$$

Subsequent to supplying the lexical actions induced by *died*, use of the rules of THINNING and COMPLETION then allow us to complete the tree. All that remains to be done is the collation of the formula values of the daughter nodes to give a complete, complex formula at the root node. This formula combination is completed using the rule of ELIMINATION (61), leaving no outstanding requirements and therefore constituting a successful parse, as shown in the completed parse tree in figure 4.

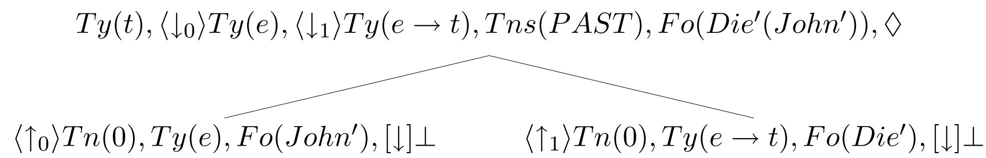

Figure 4: Parsing *John died*

In the transitive verb case, for example in parsing the sentence *John loved Mary*, the parse would proceed in an identical fashion up until *loved* was encountered. The application of the lexical actions in 52 would result in the tree shown in figure 5.

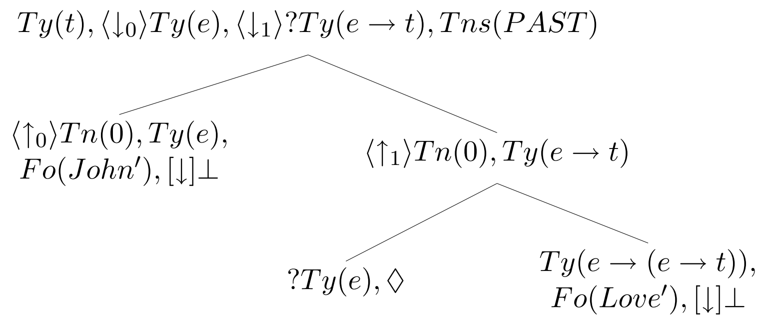

Figure 5: Parsing *John loved ...*

As can be seen, this leaves the pointer at the 'object' node, where the trigger condition for *Mary* ($?Ty(e)$) would be met, and the tree can then be completed using THINNING, COMPLETION and ELIMINATION, as before, resulting in the tree shown in figure 6.

With the addition of a few simple rules (see Kempson et al. (2001) for formal definitions and justifications), this framework allows us to account for a wide range of phenomena that other grammars have to postulate additional machinery to deal with. Examples include, relative clauses (see section 4.6, below), Hanging Topic Left Dislocation (e.g. 'As for Mary, Bill loves her') and Wh-questions. In addition, the combination of rule applications, lexical actions and the strict incrementality of the parse means that certain puzzling

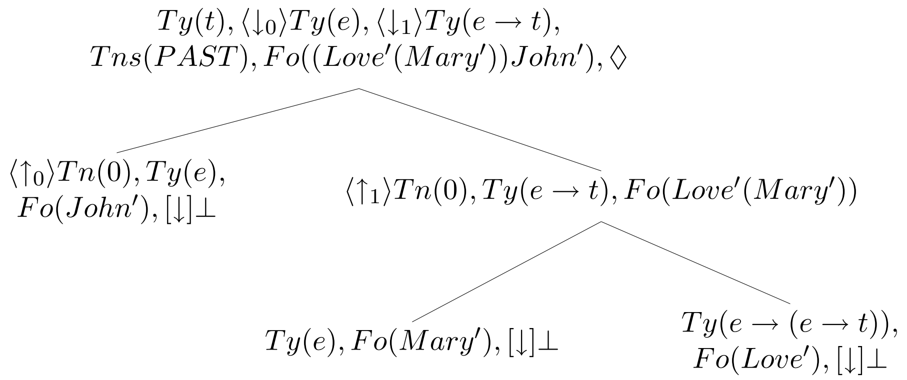

Figure 6: Parsing *John loved Mary*

phenomena of language 'fall-out' from the syntax naturally, such as the Right Roof Constraint, and the asymmetries between effects at the left and right peripheries (important when we consider verb final language such as Japanese which have historically caused problems for traditional grammars).

4.4 Unfixed Nodes

Central to a Dynamic Syntax analysis of relative clauses, left topic dislocation and Wh-questions is the notion of unfixed nodes. Unfixed nodes are nodes whose position in a tree is initially underspecified, with a requirement to be fixed at a later point in the parse. What this means in practise is that a parse tree can unfold with certain elements not yet in the positions they will occupy in the final semantic tree, without having to resort to the notion of movement. A canonical example of this is left topic dislocation, as in *Mary, John loved*. In this case, although it is the first item encountered in the string, *Mary* is not the subject, but the object of *loved*. In a transformational account, it is assumed that it is moved from its usual object position, as a focus effect, but Dynamic Syntax does not need to make such assumptions.

If we begin the parse as before, using INTRODUCTION and PREDICTION, it is clear that we will immediately run into problems, because, following the parse of *Mary*, the pointer will only be allowed to move through the tree to the $?Ty(t)$ or $?Ty(e \rightarrow t)$ node, but the next word in the string is *John*, which has a $?Ty(e)$ trigger. In fact, what we need is a weaker relation rule, *ADJUNCTION (62).

(62) *ADJUNCTION:

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

The rule of *ADJUNCTION simply says "if I am at a $?Ty(t)$ (propositional) node then I can make a $?Ty(e)$ (entity) node somewhere below me - exact location as yet unknown" (as indicated by the Kleene star modal operator $\langle\uparrow_*\rangle$, which means at this node or any node above it the following holds). Following *ADJUNCTION, the results of which are shown in figure 7, we are at a $?Ty(e)$ node, from where *Mary* can be parsed, as in figure 8.

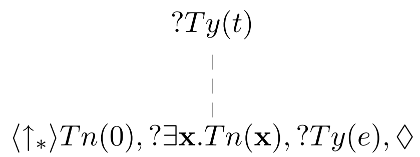

Figure 7: Using *ADJUNCTION

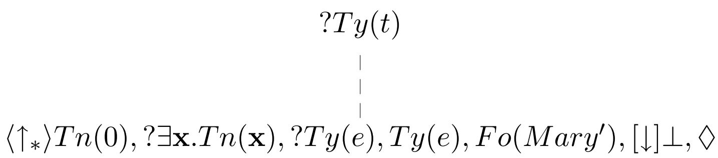

Figure 8: Parsing *Mary, . . .*, after *ADJUNCTION

THINNING and COMPLETION leave the pointer back at the $?Ty(t)$ node, from where INTRODUCTION and PREDICTION can be used to proceed as before, parsing *John*, and *loved*. The results of this are shown in figure 9.

As can be seen from figure 9, the parse cannot yet be completed, because the pointer is at a node with a requirement ($?Ty(e)$). Additionally, there is a requirement for a fixed position in the tree on the unfixed node. Both these requirements can be satisfactorily met by the merging of the two nodes (formally defined in 63, below, where $\mathbf{DU}$ stands for a nodes decorations). The result of this MERGE is the exact same tree as that shown in figure 6, above, with the only difference being in the steps used to get there.

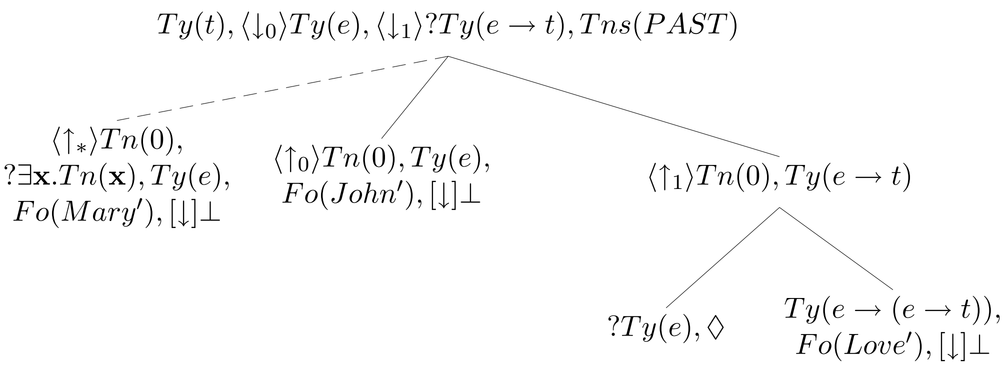

Figure 9: Parsing *Mary, John loved . . .*

(63) MERGE:

$$\frac{\{\ldots\{\ldots \mathbf{DU}, \mathbf{DU}'\ldots\}\ldots\}}{\{\ldots\{\ldots \mathbf{DU} \sqcup \mathbf{DU}'\ldots\}\ldots\}}$$
$$\Diamond \in \mathbf{DU}'$$

4.5 Metavariables

The location of a node (at a given point in a parse) is not the only thing that can be underspecified in Dynamic Syntax. As already mentioned, one of the main points of divergence between Dynamic Syntax and other grammar formalisms, is in the notion of context. In effect, the aim of Dynamic Syntax is to provide "...a formal characterisation of an expression's lexical meaning in terms of some specification which is only the input to an interpretation process that is sensitive to the context in which the expression is uttered" (Cann et al. (2005), pp.11). Dynamic Syntax does not just try to explain how syntactic forms alone relate to the grammaticality or otherwise of a sentence. In this way, pronouns, anaphora and ellipsis, for example, demonstrate how we use context to interpret strings.

In a Dynamic Syntax framework, a sentence such as *He loves cakes* would be ungrammatical if there were no contextual indication of who to interpret *he* as. This underspecification is important in that it assumes that, as processors, we constantly update our interpretations of utterances based on what we know about the world, previous discourse, or other perceptual indicators (e.g. pointing). Parsing a string with a pronoun in it involves a pragmatic process of SUBSTITUTION; for example, if the string *He loves cakes* follows *John ate all the meringues*, we would be able to substitute the formula value $Fo(John')$ for *he* in the second string, resulting in the parse leading to the complete formula $Fo(Love'(Cake')(John'))$.

This idea is formalised in the lexical entry for the pronoun. The lexical entry for *he* (see 64) contains a formula value that is underspecified; this is in the form of a *metavariable*, $Fo(\mathbf{U}_{Male'})$ whose identity can be assigned by the pragmatic process of SUBSTITUTION. There is also a requirement for a fixed formula value ($?\exists\mathbf{x}.Fo(\mathbf{x})$), which must be filled for the parse to be complete. This is why strings with pronouns in a null context are viewed by Dynamic Syntax as incomplete and therefore ungrammatical, as, as already discussed, no requirements must be outstanding for a parse to be successful. The metavariable ($Fo(\mathbf{U}_{Male'})$) in the lexical entry imposes additional constraints on what the substituted formula can be, and in this case, further constraints are imposed by the case condition $?\langle\uparrow_0\rangle(Ty(t) \wedge \exists\mathbf{x}.Tns(\mathbf{x}))$, preventing strings such as *Joan likes he*, because when 'he' is encountered in the string, the mother node will be of $Ty(e \rightarrow t)$, thus conflicting with the stated requirement.

(64) *he*

$$\left|\begin{array}{ll}\text{IF} & ?Ty(e)\\ \text{THEN} & \texttt{put}(Ty(e), Fo(\mathbf{U}_{Male'})),\\ & ?\langle\uparrow_0\rangle(Ty(t)\wedge\exists\mathbf{x}.Tns(\mathbf{x}))\\ & ?\exists\mathbf{x}.Fo(\mathbf{x}),[\downarrow]\bot)\\ \text{ELSE} & \texttt{Abort}\end{array}\right.$$

With this underspecification and use of metavariables, pronouns can thus be seen as placeholders for some other information to be assigned from context. As interpretations in context are a key feature of Dynamic Syntax, it can be seen as an attempt to formalise some of the ideas which have traditionally been left to pragmatics, such as those articulated by adherents of Relevance Theory, that all utterances are context dependence.

4.5.1 Subject Pro-drop Languages

In Spanish, as with other languages which are subject pro-drop, the subject need not be explicitly stated. Dynamic Syntax handles this in the lexical entry for verbs, which also puts a metavariable in the tree at the subject node, which can be determined pragmatically from context (which in this case includes the verb form), as outlined above. The example in 65 is legitimate if the lexical actions for *creces* are triggered by a requirement for a $?Ty(t)$, and a subject metavariable (dependent on the form of the verb) is placed at the subject node (as would be actioned by the lexical entry shown in 66).

(65) *Creces*
Grow$_{2ndSG}$
'You grow.'

(66) *creces*

$$\left|\begin{array}{ll}\text{IF} & ?Ty(t)\\ \text{THEN} & \texttt{put}(Tns(PRES));\\ & \texttt{make}(\langle\downarrow_1\rangle);\texttt{go}(\langle\downarrow_1\rangle);\\ & \texttt{put}(Ty(e\rightarrow t), Fo(Crec'),[\downarrow]\bot);\\ & \texttt{go}(\langle\uparrow_1\rangle);\texttt{make}(\langle\downarrow_0\rangle);\texttt{go}(\langle\downarrow_0\rangle);\\ & \texttt{put}(?Ty(e), Fo(\mathbf{U}_{Hearer'}), ?\exists\mathbf{x}.Fo(\mathbf{x}))\\ \text{ELSE} & \texttt{Abort}\end{array}\right.$$

Notice that the trigger for the lexical action, unlike in English, is $?Ty(t)$. This means that INTRODUCTION and PREDICTION do not have to have been used before the verb is processed, in keeping with the fact that due to the pro-drop nature of Spanish, the verb may very well be the first word in the string to be processed. The lexical entry for the verb itself contains actions for creating a subject node. Of course, it could be the case that the subject is specifically stated, as in *Juan crece*, in which case INTRODUCTION and PREDICTION could have been used with the lexical action in effect harmlessly 'creating' a node that already exists (and thus collapsing together). Alternatively, an unfixed node could be created when the explicit subject is processed (by *ADJUNCTION, as in the left topic dislocation example, in section 4.4, above) which would later be merged with the subject node. Evidence that this is in fact what happens in Spanish comes from Belletti (1999), who argues that a lexically specified subject is 'clause external'.

4.5.2 Expletive *it*

We have already seen how, in English, because it is not subject pro-drop, there needs to be a syntactic subject. This is the case even in examples where the semantic subject of a sentence has been *extraposed* to the end of the sentence, as for example in 67a, below, which can be paraphrased by 67b. Because of this, we can see that *it* in this example is not referential in the same way that the pronoun *he* is, taking its value from context (as seen above in section 4.5), rather it takes its value from what follows the verb. This is known as *expletive it*, and leads some theories to accept that there is an unavoidable disjunct between syntax and semantics (as in HPSG, see Sag et al. (2003)). In Dynamic Syntax, the problem is tackled in a different way.

(67) a. It bothers Louise that John loves Mary.
b. That John loves Mary bothers Louise.

Basically, *it*, in its expletive use, is a pronoun which has lost its bottom restriction, and its function is to move the pointer from the subject node, in order to allow the parse to continue. The important point to note about the fact that it does not have a bottom restriction is that, unlike pronouns, for example, there is, in principle, no limit to the complexity of the structure with which it can be updated. The lexical entry for expletive *it* (taken from Cann et al. (2005), pp.195) is shown in 68, below.

$$
\text{(68)} \quad it_{expl} \left| \begin{array}{lll} \text{IF} & ?Ty(t) & \\ \text{THEN} & \text{IF} & [\uparrow]\bot \\ & \text{THEN} & \texttt{Abort} \\ & \text{ELSE} & \texttt{put}(Ty(t), Fo(\mathbf{U}), ?\exists\mathbf{x}.Fo(\mathbf{x})); \\ & & \texttt{go}(\langle\uparrow_0\rangle\langle\downarrow_1\rangle) \\ \text{ELSE} & \texttt{Abort} & \end{array} \right.
$$

There is an assumption here, that a subject node can be of $Ty(t)$ (as in 67b, above), and that INTRODUCTION and PREDICTION can apply to create a blank tree with a $Ty(t)$ and a $Ty(t \rightarrow t)$ node. Once these rules have been applied, the lexical actions in 68, can be processed (with the added restriction $[\uparrow]\bot$ preventing the parse from continuing if we are at the top node).

Once *bothers Louise* has been parsed, the tree is in the state shown in figure 10[9]. Although all type requirements have been fulfilled, there is still a requirement for a formula value

[9]From now on, where decorations do not add explanatory power they will be omitted.

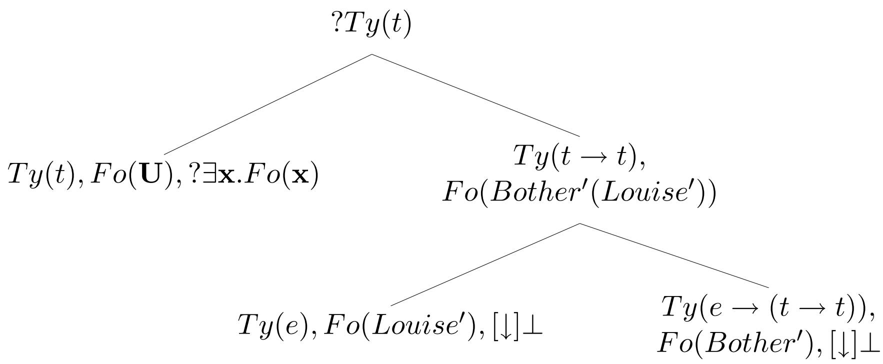

Figure 10: Parsing *It bothers Louise* . . .

at the subject node, so we cannot yet evaluate the overall formula value and complete the parse. Instead, we can move the pointer through the tree to the node with the outstanding requirement. Once at this $Ty(t)$ node, because there is a metavariable at the node, we can either complete the parse from context (as would be the case if there were a dialogue along the lines of that shown in 69), or from following lexical material. To do this, we need to introduce a rule of LATE *ADJUNCTION. Like *ADJUNCTION, discussed earlier, this introduces an unfixed node. Unlike *ADJUNCTION, however, the unfixed node is of the same type as the one from which it is projected.

(69) a. Did you know that John loves Mary?
b. Yes. It bothers Louise.

(70) LATE *ADJUNCTION:

$$\frac{\{Tn(n), \ldots \{\uparrow_* Tn(n), Tn(a), \ldots, Ty(X), \Diamond\}, \ldots\}}{\{Tn(n), \ldots \{\uparrow_* Tn(n), Tn(a), \ldots, Ty(X)\}, \{\langle\uparrow_*\rangle Tn(a), ?Ty(X), \exists\mathbf{x}.Tn(\mathbf{x}), \Diamond\}, \ldots\}}$$

Following the application of LATE *ADJUNCTION, then, we can parse the proposition *that John loves Mary*. Subsequently, we can MERGE the two completed $Ty(t)$ nodes, as shown in figure 11, after which the usual processes of THINNING, COMPLETION and ELIMINATION will allow us to complete the parse.

The resulting formula value will be $Fo((Bother'(Louise'))((Love'(Mary'))John'))$, which will be the same for both 67a and b, as it should be.

4.6 Relative Clauses

An inevitable consequence of the fact that a formula value from one tree can be pragmatically substituted into another, is that trees are not constructed in isolation. This has

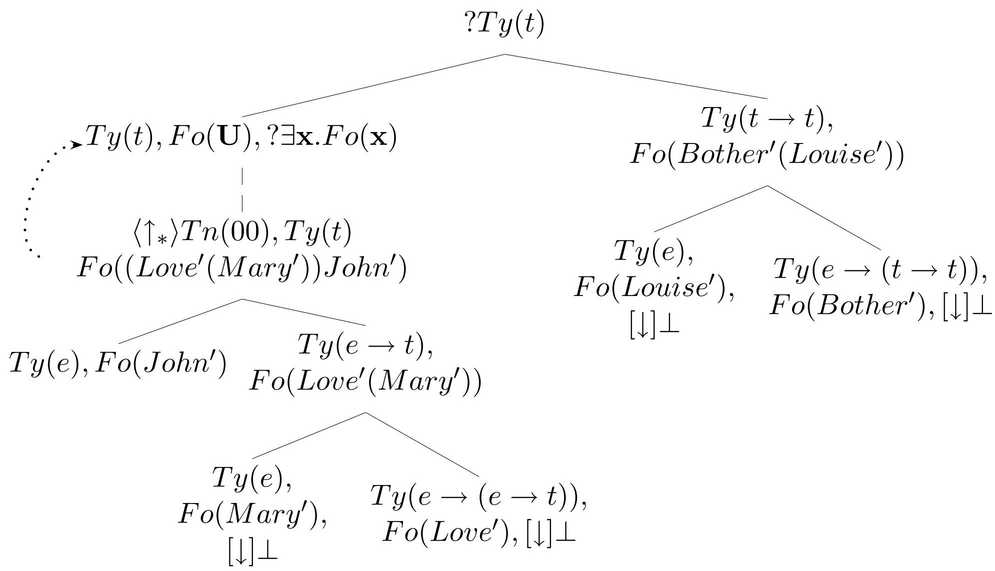

Figure 11: Parsing *It bothers Louise that John loves Mary*

implications for many different types of construction, including coordination and relative clauses. The way these are dealt with in the Dynamic Syntax framework is in the building of separate, but linked informational trees, in tandem. The rule of LINK ADJUNCTION (71) allows us to construct a new tree, in some sense linked to the tree currently under construction. The rule forces us to carry a copy of the formula value from the node at which LINK ADJUNCTION is applied, and we can see how this works with a relative clause example.

(71) LINK ADJUNCTION:

$$\frac{\{\ldots\{\{Tn(a), Fo(\alpha), Ty(e), \Diamond\}\}\ldots\}}{\{\ldots\{\{Tn(a), Fo(\alpha), Ty(e)\}, \{\langle L^{-1}\rangle Tn(a), ?Ty(t), \langle\downarrow_*\rangle Fo(\alpha), \Diamond\}\}\ldots\}}$$

In combination, the rules of LINK ADJUNCTION and *ADJUNCTION allow us to project a linked tree which must contain a copy of the node it is linked to, so that sentences such as *John, who Mary said smoked, died* can be parsed as shown in figure 12. Later in the parse, shown in figure 13, the unfixed node is merged to a fixed node position, carrying the copy of the node the linked tree is from. This can then be evaluated using LINK EVALUATION (72), and gives the correct interpretation that it is John who both smoked and died.

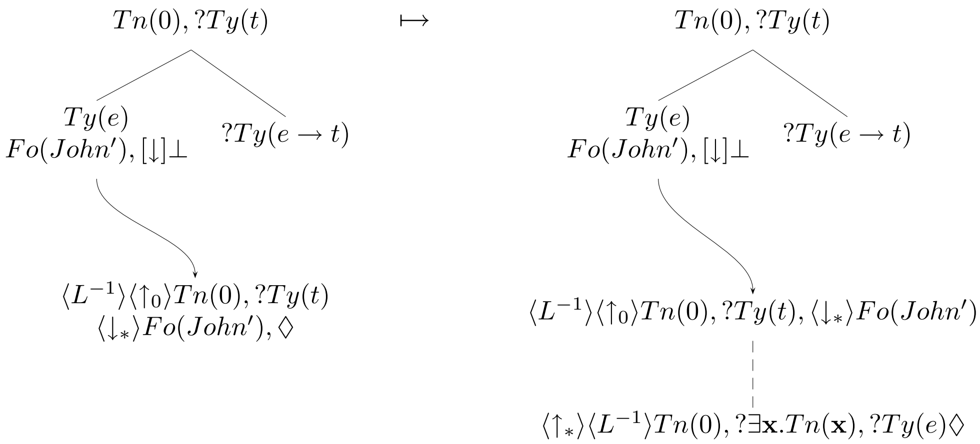

Figure 12: Parsing *John, who* . . . , using LINK ADJUNCTION and *ADJUNCTION

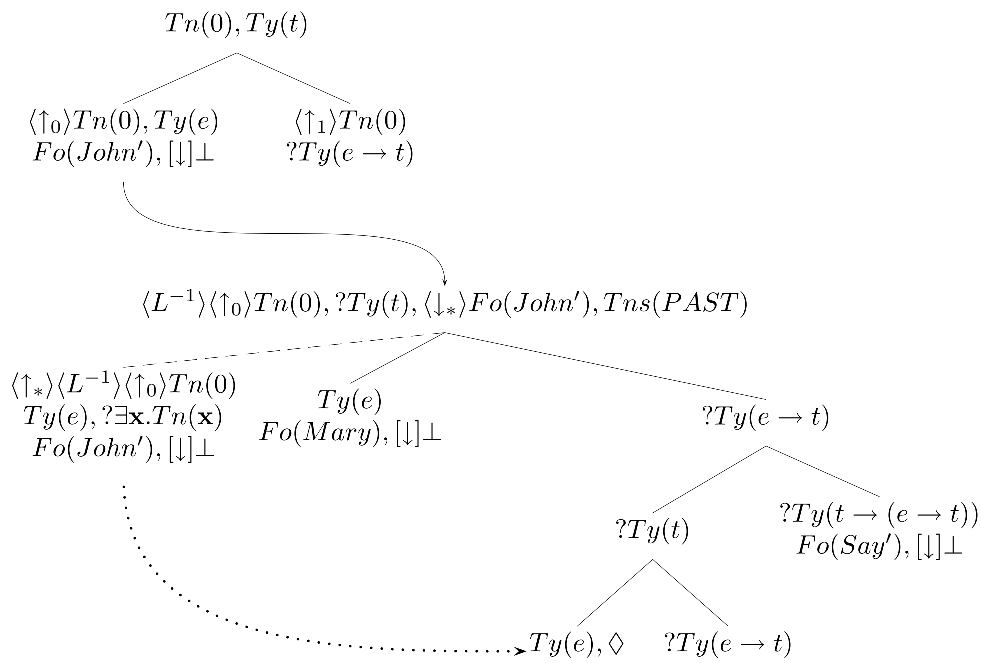

Figure 13: Parsing *John, who Mary said* . . . Merging the unfixed node

(72) LINK EVALUATION (Non-restrictive construal[10]):

$$\frac{\{\ldots\{Tn(a),\ldots,Fo(\phi),Ty(t),\Diamond\}\},\{\langle L^{-1}\rangle MOD(Tn(a)),\ldots,Fo(\psi),Ty(t)\}}{\{\ldots\{Tn(a),\ldots,Fo(\phi\wedge\psi),Ty(t),\Diamond\}\},\{\langle L^{-1}\rangle MOD(Tn(a)),\ldots,Fo(\psi),Ty(t)\}}$$
$$MOD\in\{\langle\uparrow_0\rangle,\langle\uparrow_1\rangle\}^*$$

5 Potential Dynamic Syntax Approaches to the Problem of Auxiliary Verbs

Due to the flexibility inherent in the Dynamic Syntax system, there are several potential ways in which to analyse the English auxiliary system. I will examine some of these in turn, and try to identify which method most adequately captures the known syntactic facts without neglecting the semantic contribution of the auxiliaries.

As a first assumption, I will take it as self-evident that, subject to a few item-specific exceptions (to be investigated more fully later), auxiliary verbs do contribute semantic information to a sentence or utterance. The semantics of the modal auxiliaries are much discussed in the literature (e.g. Coates (1983); Leech (1987)), so, in the below first approximations of a Dynamic Syntax analysis of auxiliaries, this will be assumed, on the basic premise that it is better to have a more inclusive generalisation, including a majority of the items under consideration, than the other way round. As semantic contributions in Dynamic Syntax are represented by decorations (e.g. Formula values) at the level of the node, one way in which this assumption can be implemented is by ensuring that each auxiliary verb should occupy its own node in a parse tree. This is not, however, the only possibility, as certain words in Dynamic Syntax do not occupy their own node, rather, they annotate existing nodes (possibly with a metavariable, for example, which is later replaced with other, fixed information), trigger specific rules or merely contribute to pointer movement. However, while it may require revision, for the sake of clarity I will start from this assumption that, like lexical verbs, auxiliaries occupy their own node.

5.1 Auxiliaries as Verbs which take Verbal Complements

Taking an approach similar to that of HPSG (see section 3.2.1), one possibility is that auxiliary verbs are simply verbs which require verbal complements. This would be analogous to other, non-auxiliary verbs, which can also take verbal complements, e.g. *want* (ignoring the contribution of the infinitival marker *to*, for now, but see section 6.6). Equivalent lexical entries might look like those shown in 73 and 74 below.

[10] See Cann et al. (2005), Chapter 3, for explanation and details of restrictive relative clauses.

(73) *wanted*

$$\left|\begin{array}{ll} \text{IF} & ?Ty(e \rightarrow t) \\ \text{THEN} & \texttt{put}(Ty(e \rightarrow t)) \\ & \texttt{go}(\langle\uparrow_1\rangle); \texttt{put}(Tns(PAST)); \\ & \texttt{go}(\langle\downarrow_1\rangle); \texttt{make}(\langle\downarrow_1\rangle); \texttt{go}(\langle\downarrow_1\rangle); \\ & \texttt{put}(?Ty((e \rightarrow t) \rightarrow (e \rightarrow t)), Fo(Want'), [\downarrow]\bot); \\ & \texttt{go}(\langle\uparrow_1\rangle); \texttt{make}(\langle\downarrow_0\rangle); \texttt{go}(\langle\downarrow_0\rangle); \\ & \texttt{put}(?Ty(e \rightarrow t)) \\ \text{ELSE} & \texttt{Abort} \end{array}\right.$$

(74) *could*

$$\left|\begin{array}{ll} \text{IF} & ?Ty(e \rightarrow t) \\ \text{THEN} & \texttt{put}(Ty(e \rightarrow t)) \\ & \texttt{go}(\langle\uparrow_1\rangle); \texttt{put}(Tns(PAST)); \\ & \texttt{go}(\langle\downarrow_1\rangle); \texttt{make}(\langle\downarrow_1\rangle); \texttt{go}(\langle\downarrow_1\rangle); \\ & \texttt{put}(?Ty((e \rightarrow t) \rightarrow (e \rightarrow t)), Fo(Could'), [\downarrow]\bot); \\ & \texttt{go}(\langle\uparrow_1\rangle); \texttt{make}(\langle\downarrow_0\rangle); \texttt{go}(\langle\downarrow_0\rangle); \\ & \texttt{put}(?Ty(e \rightarrow t)) \\ \text{ELSE} & \texttt{Abort} \end{array}\right.$$

However, whilst at first glance this might appear to adequately capture the basic facts of auxiliary verbs, it instantly runs into difficulties, thus highlighting certain problems. To see why this is so, consider the tree produced when multiple verbs which take verbal complements are chained, as in figure 14. An equivalent tree for a similar length chain of auxiliaries is shown in figure 15. Quite apart from the contribution of the infinitival *to*, in figure 14, the only verb which contributes any tense information is the first one, in this case *wanted*. This is not the case in figure 15, although, of course, there are strict co-restrictions on exactly which form (and therefore what tense information) each element can contribute.

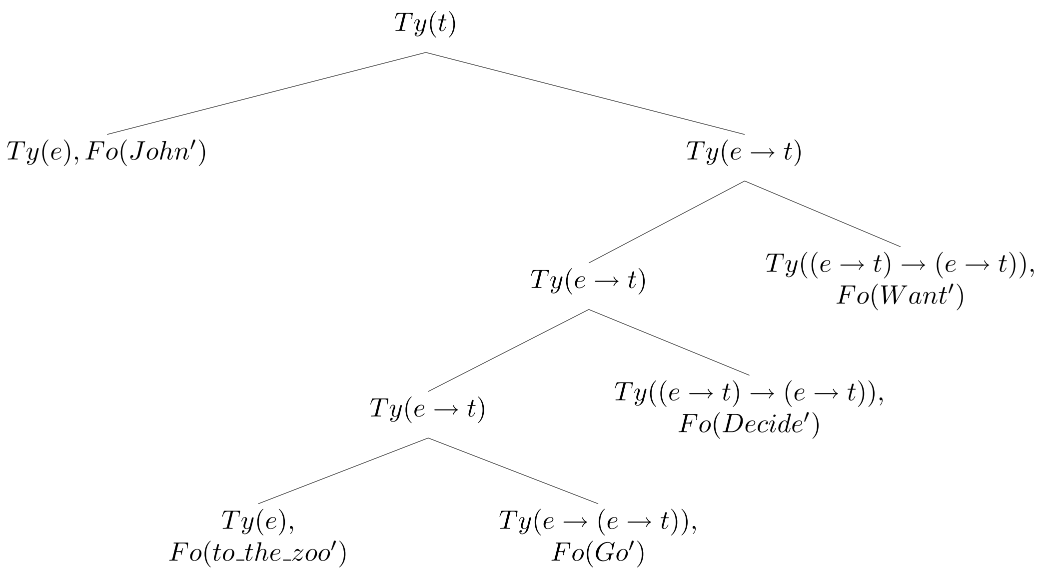

Figure 14: Parsing *John wanted to decide to go to the zoo*

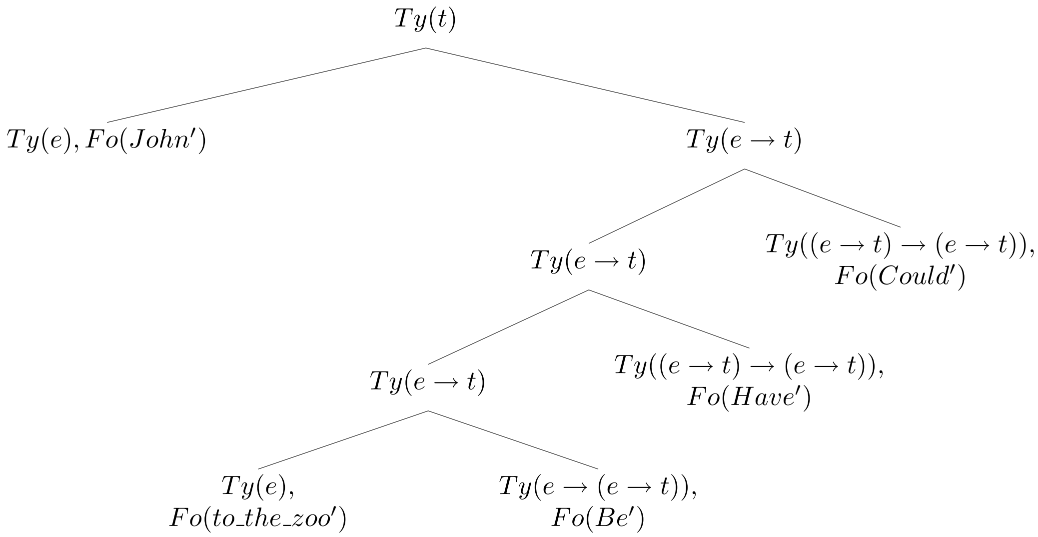

Figure 15: Parsing *John could have been to the zoo* (treating auxiliaries as verbs which take verbal complements)

(75) a. John went to the zoo.
b. John is going to the zoo.
c. John has been to the zoo.
d. John has been going to the zoo.
e. John could go to the zoo
f. John could be going to the zoo.
g. John could have been to the zoo.
h. John could have been going to the zoo.

In each of the examples in 75, each of the consecutive auxiliaries contributes some tense information, and it is not clear, in the existing Dynamic Syntax framework, how this would occur in a principled fashion. One possible solution, would be for the lexical entries of each item, depending on the ending, to contribute additional tense information to the top node (the only place in Dynamic Syntax where tense information is collected), however, not only is there no way of knowing how far back up the tree the root node would be when a particular item were encountered (e.g. the progressive form *going* would occur further down the tree in 75h than in 75b), but the information contributed would lead, in the more complex cases to a proliferation of decorations at the root node (e.g. using mnemonic terms for progressive, perfective, etc) which would offer no clues as to how these were to be interpreted. The complex issue of how tense is to be dealt with in Dynamic Syntax is beyond the scope of this paper, but note that without one, a completely inclusive analysis of the auxiliary verb system in English is impossible.

Although this itself highlights an existing area of Dynamic Syntax which is underdeveloped, there are additional issues with this type of approach based purely on syntactic properties. The main reason that this approach does not adequately capture the facts of auxiliary use is in the very point that it is exactly analogous to all verbs which take verbal complements. If this were taken to its logical conclusion, then additional, ad hoc constraints would have to be proposed to account for the syntactic NICE properties (see section 2.1) and why they are, in modern English, restricted to auxiliaries, and do not occur with all verbs which take verbal complements. Although it might be thought that this explanation can exploit the fact that most non-auxiliaries take *to*-infinitives, not bare infinitives, the existence of both auxiliaries which exhibit the nice properties despite taking a complement with a *to*-infinitive (e.g. *ought*) and non-auxiliaries which take a bare infinitive complement (e.g. *help*) mean that this is insufficient to account for the data.

5.2 Auxiliaries as an Independent Category

An alternative approach, in line with early transformational accounts as discussed in section 3.1 is to treat auxiliary verbs as an independent category. Although there are many ways in which this could be realised, the simplest, illustrated in figure 16, is to assume that the auxiliary category creates and annotates its own node (given $Ty(aux)$ here). Problems associated with this type of analysis are immediately apparent when we consider the type decorations in the tree in figure 16. Although the problems of the production of arbitrary seeming types such as $Ty(e \rightarrow (aux \rightarrow (aux \rightarrow (e \rightarrow t))))$ are not insurmountable, and, in fact, might even be necessary in considerations of adjuncts and verbs which take optional arguments, as argued in Marten (2002)[11], this approach introduces additional complexity into the system, and does not resolve the issues raised in 5.1, above.

5.3 Multiple Update of a Single Node Using LATE *-ADJUNCTION

An earlier account of the English auxiliary system in Dynamic Syntax is outlined in Garcia-Miller (2005). This account treats all auxiliary verbs as placeholding metavariables which, in a similar way to expletive *it* (as discussed in section 4.5.2) are updated using the rule of LATE *ADJUNCTION. This essentially means treating auxiliaries as elements which do not decorate their own node. As discussed in 5, this, to my mind, means that semantic information carried by the auxiliaries may be lost. Indeed, Garcia-Miller acknowledges this in the case of the modal verbs and annotates the metavariable with an arbitrary subscript, e.g. $Fo(\mathbf{V}_{inf})_{MAY}$. Another complication of this approach is reflected in the apparatus required for cases where multiple auxiliaries are present in a string. In order to accommodate multiple verbal entries at the *same* node, Garcia-Miller

[11]This example of types $Ty(aux^* \rightarrow (e \rightarrow t))$ mirrors his $Ty(e^* \rightarrow t)$ notation, which might even be desirable if we accept his arguments for type underspecification of nodes, however, Marten has nothing to say about how his analysis could be extended to verbs which take verbal complements.

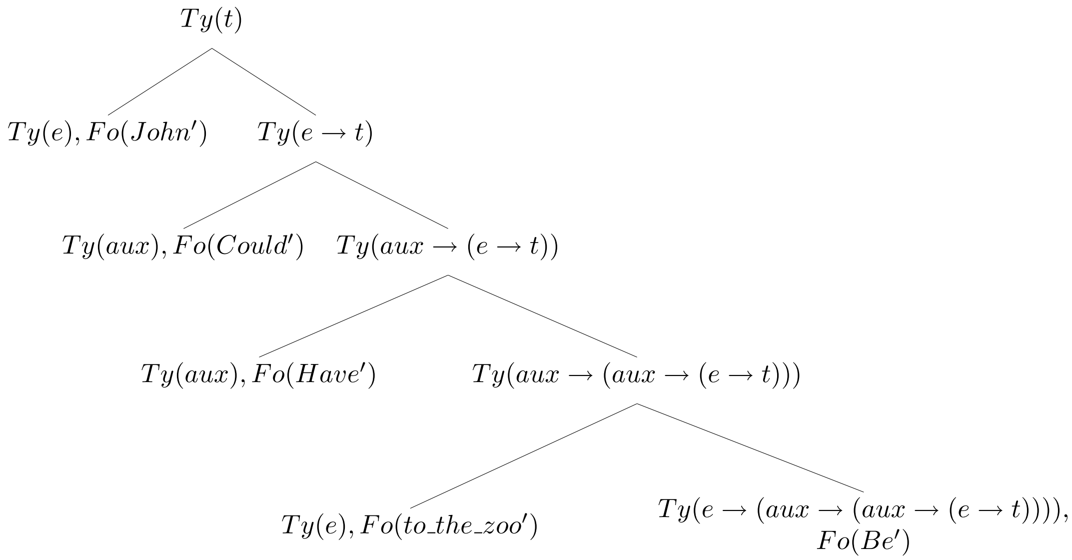

Figure 16: Parsing *John could have been to the zoo* (treating auxiliaries as a separate type category)

introduces the notion of a *verbal clause*, which she defines as in (76), below. This allows verbs to annotate the same node, without altering the requirement value, which is only ‘closed off’ to additional update when a lexical verb is encountered.

(76) VERBAL CLAUSE

$$\frac{\{\ldots\{\langle\uparrow_1\rangle Tn(0), ?Ty(e \rightarrow t), \Diamond\} \vee \{\langle\uparrow_1\rangle Tn(0), ?Ty(e \rightarrow t)_{?1\ldots n\perp}, \Diamond\}\ldots\}}{\{\ldots\{\langle\uparrow_1\rangle Tn(0), ?Ty(e \rightarrow t)_{?1\ldots n\perp}, (Fo(\mathbf{V}_{aux})_1, \ldots Fo(\mathbf{V}_{aux})_n, \ldots Fo(\mathbf{V}_{main})\perp)_n, \Diamond\}\ldots\}}$$

Although in theory this means that she can introduce additional $Ty(e \rightarrow t)$ elements (at different values of *n*), in practise, it means that she is forced to adopt hugely disjunctive lexical entries for words, depending on where in the verbal clause they can appear. Additionally, the application of this apparatus leads to either a proliferation of arbitrary decorations (either at the node under development or at the root node) to indicate the contribution of each successive auxiliary (e.g. in her example *John had been reading*, the decorations include **Perf.**, **Part.**, and **Prog.**) or the loss of information as to how the structure was built up. Both of these possibilities are untenable. In the former, the intuitively semantically different sentences *John had been reading* and *John reads* result in the same formula value (with the only difference being in what arbitrary decorations are present and raising questions as to whether tense information should be bound up with the formula value), and, more problematically, in the latter situation it would be impossible to unpick the sentence into its constituent parts. To see why this matters, we only have to consider the ellipsis possibilities sentences with multiple auxiliaries present, as shown in example 77, below. Without a structural cue as to how the complement of each of the respective ellipses is possible, we have no way of accounting for them.

(77) John might have been being kissed, and . . .

  a. . . . Julia might, too.
  b. . . . Julia might have, too.
  c. . . . Julia might have been, too.
  d. . . . Julia might have been being, too.

Ultimately, I believe the major failing of this approach is in its inability to account adequately for the strict co-occurrence restrictions of auxiliaries, and in its treatment of their semantic contribution (although, as discussed above, this highlights a failure of Dynamic Syntax as it stands to explain or account for features of tense, aspect and mood in a principled way). In choosing what Falk (section 3.2.2) refers to as an *aux-feature* account, the treatment misses important factors, and, to my mind, any adequate account must, like the auxiliary verbs themselves, be in some sense interpretable as in the *aux-feature* analysis, and in some sense, at the same time, be interpretable in an *aux-predicate* way.

I also think that Garcia-Miller is incorrect in her highlighting of what she considers to be a problem - namely that auxiliary *be* can have either a present or a past participle as its complement, which she specifies in the lexical entry. In my view, either both possibilities would be explored in tandem, or the onus would be in the lexical entry for the complement itself, which would be constrained for what it could follow. This would enable the constraint for *-ing* forms to specify that before it must be an event verb (of which *be* is one possibility, with others being, for example, *start*, *keep*, etc), whilst the constraint for *-en* forms would restrict what they could follow to forms of *have* or *be*.

6 Analysing Auxiliaries with LINKed Structures and *Type* Underspecification

If my criticisms of the above potential methodologies for accounting for the syntactic properties of auxiliary verbs in Dynamic Syntax are sound, then where does that leave us?

The methodology I propose makes use of the Dynamic Syntax notions of underspecification and LINKed structures to explain the NICE syntactic properties of auxiliary verbs. The analysis is also applied to the exceptional items discussed in 2.2, as well as there being a brief discussion of the implications of such an analysis on the interpretation of all verbs which take verbal complements and the contribution of *infinitival-to*.

6.1 A Dynamic Syntax Account of Inversion

Firstly, I will outline the way in which I propose to deal with Inversion in Dynamic Syntax, which, due to the flexibility inherent in the system, there are several potential candidates for. One option would be to have a rule of inversion (as postulated by Garcia-Miller (2005)). However, due to the nature of rules in Dynamic Syntax, which can occur at any point when the criteria for their application is met (the part of the rule above the line which describes the existing tree formation required for the rule to apply), this would mean that lexical entries for non-auxiliary verbs would require additional constraints in order to prevent the rule of inversion occurring with full verbs. The same is true if we assume that auxiliaries can create and decorate unfixed nodes immediately in the parse. Of course, such additional constraints may prove to be desirable, and even necessary, however, as a first approximation, it seems to me that the most appropriate way of accounting for the facts of inversion which can only occur with auxiliary verbs and their subjects, notably leading to cases where auxiliary verbs are the first word encountered in a string is at the level of lexical actions.

The proposal adopted here is that auxiliary verbs have a trigger requirement of $?Ty(t)$. This is in contrast to full, lexical verbs in Modern English, which are depicted in Dynamic Syntax as being triggered by $?Ty(e \rightarrow t)$ (see e.g. 51 and 52 in section 4.2, above). A working hypothesis regarding the diachronic data would then be that part of the recategorisation of full verbs (as discussed in section 3.3.2) would have been the change in trigger requirement for full verbs from $?Ty(t)$, a change which did not affect auxiliary verbs. Further motivations for this come from cross-linguistic data, e.g. Spanish, where, due to the subject pro-drop nature of the language, all verbs have a $?Ty(t)$ trigger. What this suggests is that verbs in Early Modern English were more similar to those in Modern Spanish than they are today.

A first approximation for the lexical entry of the modal verb *can* is shown in 78, below.

(78) Lexical entry for *can* (first approximation).

	IF	$?Ty(t)$	
	THEN	IF	$\langle\downarrow_1\rangle Ty(e \rightarrow t)$
		THEN	Abort
		ELSE	$\texttt{put}(Tns(PRES))$;
can			$\texttt{make}(\langle\downarrow_1\rangle); \texttt{go}(\langle\downarrow_1\rangle)$;
			$\texttt{put}(Ty(e \rightarrow t), Fo(Can'))$;
			$\texttt{go}(\langle\uparrow_1\rangle); \texttt{make}(\langle\downarrow_0\rangle); \texttt{go}(\langle\downarrow_0\rangle)$;
			$\texttt{put}(?Ty(e))$
	ELSE	Abort	

A couple of points should be noted about the lexical entry shown in 78. Firstly, like the intransitive verb lexical entry for *died* in 51, and unlike the example in 74, it does not project any additional structure. The reasons for this, and its implications will be examined in more detail, later. Secondly, as well as having a $?Ty(t)$, trigger, there is a second constraint present, on the second line of the lexical entry, $\langle\downarrow_1\rangle Ty(e \to t)$. This ensures that if there is already a fulfilled $Ty(e \to t)$ node then the auxiliary cannot be parsed, thus preventing the licensing of ungrammatical strings such as **John died can*. It is important to note, however, that this restriction is not against a *requirement* for a $?Ty(e \to t)$ node. This is because the existence of such a node would not in fact interfere with the lexical actions in 78. This is exactly what is required in a situation where inversion has not occurred and the sentence initial subject has occupied the subject node after INTRODUCTION and PREDICTION, for example in the sentence *John can jump*. In essence, what would then occur (after COMPLETION had moved the pointer back to the root node) is that the lexical actions would force the creation of a node which already existed. While this would cause the parse to abort if there were any conflicting decorations, in this example there would not be, and the node created by PREDICTION and that created by the lexical actions of *can* would be indistinguishable. In these circumstances, the Logic of Finite Trees (LOFT, Blackburn & Meyer-Viol (1994)) tells us that they are not separate nodes at all. Notice that this means, in a similar fashion, that in both the inverted and non-inverted case, the pointer would end up at the subject node, either awaiting the input of a lexical subject (in the inverted case), or on the already completed node in the standard case.

One consequence of this is that the final parse tree for the inverted or non-inverted case would be identical. It is clear, though, that "Can I?" and "I can" have very different semantic meanings and this needs to be reflected in the tree. In previous treatments, it has been assumed that the inversion operation invokes the clause typing of the sentence being processed as a question (represented on the root node by a $Cat(Q)$ decoration. However, it is not by any means clear to me that this is a satisfactory representation of the data, as it is, of course, possible to ask questions without any change in word order, merely by intonation (which is not yet well accounted for in Dynamic Syntax), as in example 79, below. In addition, questions are not the only contexts in which inversion occurs, as seen in 12, repeated here as 80.

(79) a. Did you see her?
b. You saw her?

(80) Had I known you were late, I would have waited. *(Conditional)*

It seems to me, therefore, that inversion, like intonation in languages where scrambling is possible (e.g. Korean; Kiaer (2005)) would be better represented as a pragmatic marker to the hearer that something non-canonical is going on (which might be usually interpreted as a question, but importantly not always). For the moment I will leave these considerations to one side, but assume that the inverted case does leave some trace on the parse tree; a decoration which must be pragmatically combined with the other information available. For the purposes of clarity, and subject to further investigation regarding the implications of adding extra decorations which are not well-defined, I shall simply call this $Cat(INV)$ for inverted, the possibility that such a decoration has come to almost exclusively represent a question in English (and therefore perhaps does have a clause-typing effect) notwithstanding. This means that the lexical entry for the modal verb needs amending, as shown in 81, below. The additional actions shown are in an embedded IF clause, and it is important to note that although the actions before this embedded clause will always occur when *can* is encountered in a string (provided the initial trigger conditions are met), the inclusion of the embedded clause means that the effects of parsing *can* are slightly different depending on the state of the existing parse tree when it is encountered. Details of all lexical transitions are in Kempson et al. (2001), section 3.2.2, where they note that "...item internal disjunctions are for cases in which an individual word projects a number of discrete actions, which share a core subset of properties." Essentially the only difference in what occurs in this lexical entry, compared to 78, is that, after making the argument daughter (subject node), but *before* moving the pointer to this newly created node, it checks to see whether there is already a $Ty(e)$ node in place. If there is, this means that this is *not* an inverted case, and therefore we can move the pointer down the argument daughter node, as before. In the contrary case (as actioned by the ELSE statement), there is no existing $Ty(e)$ node in the subject position, so we are dealing with an inverted case. Consequently, the decoration $Cat(INV)$ is placed at the current ($Ty(t)$) node before moving down to the subject node, and putting in a requirement for $?Ty(e)$.

(81) Lexical entry for *can* (second approximation).

$$
\textit{can}\;\left|\;\begin{array}{llllll}
\text{IF} & ?Ty(t) & & & & \\
\text{THEN} & \text{IF} & \langle\downarrow_1\rangle Ty(e \to t) & & & \\
 & \text{THEN} & \texttt{Abort} & & & \\
 & \text{ELSE} & \multicolumn{4}{l}{\texttt{put}(Tns(PRES));} \\
 & & \multicolumn{4}{l}{\texttt{make}(\langle\downarrow_1\rangle); \texttt{go}(\langle\downarrow_1\rangle);} \\
 & & \multicolumn{4}{l}{\texttt{put}(Ty(e \to t), Fo(Can'));} \\
 & & \multicolumn{4}{l}{\texttt{go}(\langle\uparrow_1\rangle); \texttt{make}(\langle\downarrow_0\rangle);} \\
 & & \text{IF} & \langle\downarrow_0\rangle Ty(e) & & \\
 & & \text{THEN} & \texttt{go}(\langle\downarrow_0\rangle) & & \\
 & & \text{ELSE} & \texttt{put}(Cat(INV)); & & \\
 & & & \texttt{go}(\langle\downarrow_0\rangle), \texttt{put}(?Ty(e)) & & \\
\text{ELSE} & \texttt{Abort} & & & &
\end{array}\right.
$$

The results of using this lexical entry to parse both *John can* ... and *Can John* ... are shown in figure 17.

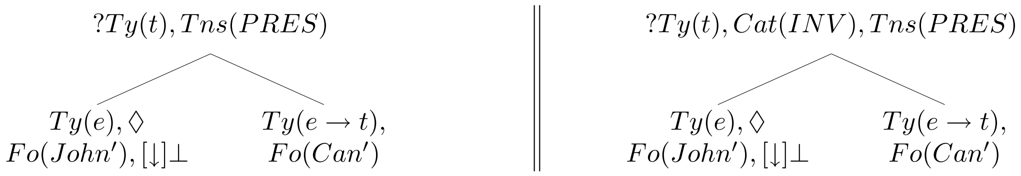

Figure 17: Parsing *John can* ... and *Can John* ...

6.2 A Dynamic Syntax Account of Code (Ellipsis)

It seems apparent, however, that this is insufficient. This is because, as it stands in figure 17, the rule of ELIMINATION could be applied, because there are no outstanding requirements, meaning that "John can" or "Can John?" would be valid sentences in their own right. Although in some sense these utterances can stand alone (as in cases of ellipsis as discussed in section 2.1.3), they beg the question "Can what?". As I have already rejected the possibility that auxiliary verbs are merely verbs taking verbal complements, it is clear that any extension to the lexical entry shown in 81 will not merely be an instruction to create new nodes. Then what?

It is here that the intuition that "John can" can stand alone only if uttered in context, and the Dynamic Syntax notion of underspecification needs to be invoked. Just as personal pronouns were underspecified for their referent (as discussed in section 4.5), and ungrammatical if none could be specified, so it is with the auxiliary verbs. Essentially, what the formula value of the auxiliary needs to project is a requirement for exactly the information asked for by the earlier intuitive question "Can what?". As with pronouns, this is done by the introduction of a metavariable. However, unlike in the pronoun case, we do not want to completely lose the semantic contribution of the auxiliary (as discussed above) and I therefore propose that the formula value introduced by the lexical entry will be of the form $Fo(Can'(\mathbf{AUX}))$ (see 82, below). Just as with the pronoun case, however, this metavariable (mnemonically called $\mathbf{AUX}$ here, which represents whatever follows it[12]) can be updated from context, previous discourse, or lexical material still to be processed.

(82) Lexical entry for *can* (third approximation).

$$
\textit{can} \left| \begin{array}{llll}
\text{IF} & ?Ty(t) & & \\
\text{THEN} & \text{IF} & \langle\downarrow_1\rangle Ty(e\rightarrow t) & \\
 & \text{THEN} & \texttt{Abort} & \\
 & \text{ELSE} & \texttt{put}(Tns(PRES)); & \\
 & & \texttt{make}(\langle\downarrow_1\rangle);\texttt{go}(\langle\downarrow_1\rangle); & \\
 & & \texttt{put}(Ty(e\rightarrow t), Fo(Can'(\mathbf{AUX})), ?\exists\mathbf{x}.Fo(\mathbf{x})); & \\
 & & \texttt{go}(\langle\uparrow_1\rangle);\texttt{make}(\langle\downarrow_0\rangle); & \\
 & & \text{IF} \quad \langle\downarrow_0\rangle Ty(e) & \\
 & & \text{THEN} \quad \texttt{go}(\langle\downarrow_0\rangle) & \\
 & & \text{ELSE} \quad \texttt{put}(Cat(INV)); & \\
 & & \qquad\qquad \texttt{go}(\langle\downarrow_0\rangle);\texttt{put}(?Ty(e)); & \\
\text{ELSE} & \texttt{Abort} & &
\end{array}\right.
$$

The inclusion of $?\exists\mathbf{x}.Fo(\mathbf{x})$, which is a requirement for a fixed formula value, ensures that a string in which the metavariable has *not* been replaced by some fixed formula value will not be grammatical in Dynamic Syntax terms, because this requirement will remain outstanding. The notion that only auxiliary verbs introduce this metavariable and accompanying requirement for a fixed formula value which may be fulfilled from context

[12]Or, in the case of ellipsis, whatever the 'missing' material would be

correlates precisely with the idea of VP ellipsis, which is available with auxiliary verbs, but not with lexical verbs (discounting for a moment the contribution of the *to* infinitival marker in "John wants to go, but Julia doesn't want to", which will be explored further in section 6.6).

This leads to the question of whether this complex formula can annotate a node directly, as in the lexical entry shown in 82, or leads to the postulation of internal structure of $Ty(e \rightarrow t)$ nodes with auxiliary verbs (in an analogous way to how $Ty(e)$ nodes are structured when they contain common nouns (see chapter 3, Cann et al. (2005) for further details)) In principle, a simplistic picture such as the one shown in figure 18 could be posited, where the node of $Ty(\mathbf{A})$ is, at present, type underspecified in a way to be made clear later.

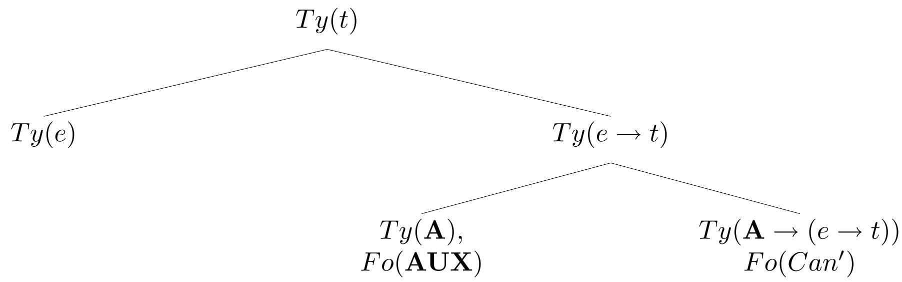

Figure 18: Potential internal structure of $Ty(e \rightarrow t)$ node

6.2.1 Implications for *All* Sentences with Auxiliary Verbs

The major question, then, concerns how the parse continues if the metavariable is updated by following lexical material. According to Garcia-Miller, this is done using LATE *ADJUNCTION, though we have already seen how this runs into problems when we consider examples with multiple auxiliary verbs. In my account, the process of continued tree growth is accomplished by the creation of a LINKed tree.

If we recall, from the relative clause example, the introduction of a LINKed tree allows trees to be built up in tandem and provides a copy of the node from which the LINK operation came. In the relative clause example, "John, who Mary said smoked, died", the subordinate relative clause (demarcated in this case by the relative pronoun *who*) in some sense gives us more information regarding John and led to a final evaluation that translated as "John died, and Mary said John smoked." In the auxiliary case under consideration, the additional information also seems to be regarding the subject. For example, in "John can jump", what is being asserted is that *there is the possibility that John jumps* (which could potentially be realised by John having the ability to jump). In other words, *John jumps* is in some sense a component of the sentence "John can jump", just as it is of the sentence "John could have been jumping"; just the implication we get using the notion of LINKed structures.

After parsing *can* in the sentence “John can jump”, due to the lexical entry for can (see 82, as already discussed), the pointer will be at the subject node (already annotated with the decoration $Fo(John')$). The possibility therefore arises of using the existing LINK ADJUNCTION rule (as stated in section 4.6). There are advantages to this approach. Not only does it not require us to introduce additional machinery to Dynamic Syntax, but, as already alluded to, it seems right that the subject be copied over to the LINKed structure, as is the case with the previously discussed rule of LINK ADJUNCTION. Figures 19, 20 and 21 show (ignoring internal structure of $Ty(e \rightarrow t)$ nodes for the moment) how, using exactly the processes of LINK and *ADJUNCTION, this parse would proceed.

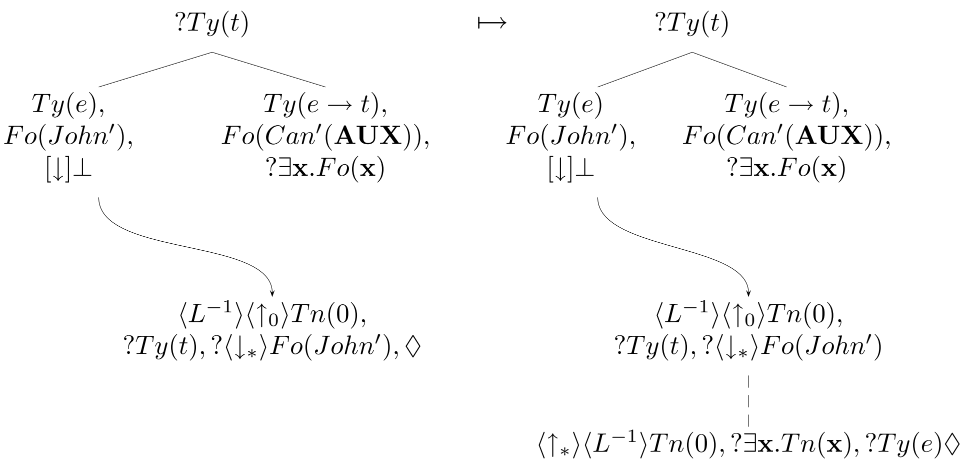

Figure 19: Parsing *John can ...*, using LINK ADJUNCTION and *ADJUNCTION

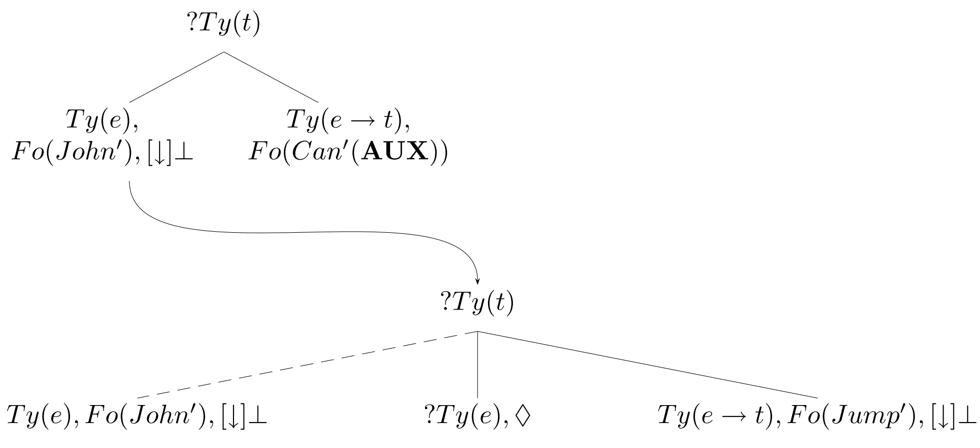

Figure 20: Parsing *John can jump*

A problem arises, however, when we consider the LINK EVALUATION function used in the relative clause example. If applied to the output shown in figure 21, then the result would be as in 83, below.

(83) $Fo((Can'(\mathbf{AUX}))John') \wedge ((Jump')John')$

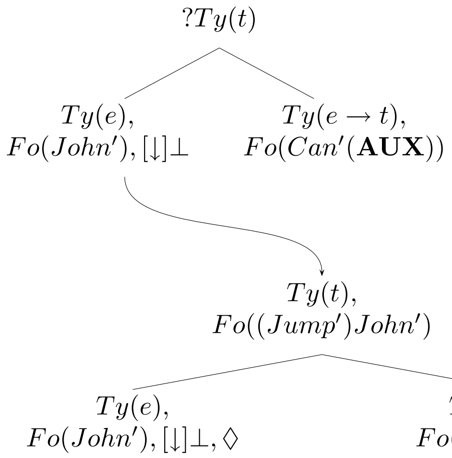

Figure 21: Parsing *John can jump*, after MERGE

Even if we assume that the formula results of the LINKed structure could be substituted for the metavariable $\mathbf{AUX}$, we would still be left with the result shown in 84, when arguably what we are actually looking for is the formula value shown in 85[13].

(84) $Fo((Can'((Jump')John'))John') \wedge ((Jump')John')$

(85) $Fo((Can'((Jump')John'))John')$

A further problem concerns the necessary restrictions on the lexical entry for the base infinitive form of the lexical verb, in this case *jump*. If the LINK relation were created from the subject node, the constraint condition would be extremely complex, with modal operators akin to $\langle\uparrow_1\rangle\langle L^{-1}\rangle\langle\uparrow_0\rangle\langle\downarrow_1\rangle Fo((\mathbf{AUX})x)$. This hardly seems parsimonious, and, together with the apparent necessity to come up with an entirely new LINK EVALUATION function, leads me to the conclusion that the subject node is not the locus for the LINKed structure. Intuitively, assuming the hypothesised internal structure shown in figure 18 (at least for auxiliaries), then, after parsing *can* in the string "John can jump", the pointer, currently at the subject node, can move through the tree using COMPLETION and ANTICIPATION, as far as the only node with unfulfilled requirements, namely the $Ty(\mathbf{A})$ node.

In this case, we need to formulate a new form of the LINK ADJUNCTION rule, to induce a LINKed structure from a $Ty(\mathbf{A})$ node.

(86) A-LINK ADJUNCTION:

$$\frac{\{\ldots\{\{Tn(n), \langle\uparrow_0\rangle\langle\uparrow_1\rangle\langle\downarrow_0\rangle(Ty(e) \wedge Fo(\alpha)), ?Ty(\mathbf{A}), \Diamond\}\}\ldots\}}{\{\ldots\{\{Tn(n),\ldots\}, \{\langle L^{-1}\rangle Tn(n), ?Ty(\mathbf{A}), \langle\downarrow_0\rangle Fo(\alpha), \Diamond\}\}\ldots\}}$$

[13] I will assume that this is so for now, my criticisms of the HPSG formulation in Sag et al. (2003) for precisely this type of assumption notwithstanding, with this line of enquiry to be further developed later.

This rule can be more transparently described using the syntax of lexical actions (as, indeed, all rules in Dynamic Syntax can be viewed as collections of the same tree operations as are carried out by the lexical items, analogously to computational macros), as shown in 87, below.

(87) *A-LINK Adjunction*

IF	$?Ty(\mathbf{A}) \wedge \langle\uparrow_0\rangle\langle\uparrow_1\rangle\langle\downarrow_0\rangle(Ty(e) \wedge Fo(\alpha))$
THEN	$\mathtt{make}(\langle L\rangle); \mathtt{go}(\langle L\rangle);$
	$\mathtt{put}(?Ty(\mathbf{A}), \langle\downarrow_0\rangle Fo(\alpha))$
ELSE	`Abort`

As can be seen in 86, above, the only differences in the trigger condition for this type of LINK ADJUNCTION and that encountered in the relative clause example earlier (the rule is shown in 71), are in the type value of the node from which the link is created, and in the copying of the information at the 'subject' node $Fo(\alpha)$ requiring modal operators. This may prove to be too restrictive, and be replaced by $\langle\uparrow_*\rangle\langle\downarrow_0\rangle Fo(\alpha)$, however, it suffices to illustrate the operations involved for now. The only difference in the resulting LINKed tree is that the position of the subject is fixed. Again, this may turn out to be an unnecessarily strict condition.

Obviously, in order for this rule to be triggered, a $?Ty(\mathbf{A})$ node will have to have been created. As mentioned earlier, this will be in the lexical entry for the auxiliary verbs. The additional lexical actions, shown in 88, below, are instructions to make the $Ty(e \rightarrow t)$ node into an argument and functor daughter, analogously to the transitive verb case (loved; 52), discussed earlier.

(88) Lexical entry for *can* (fourth approximation).

can

```
IF      ?Ty(t)
THEN    IF      ⟨↓1⟩Ty(e → t)
        THEN    Abort
        ELSE    put(Tns(PRES));
                make(⟨↓1⟩); go(⟨↓1⟩); put(Ty(e → t));
                make(⟨↓1⟩); go(⟨↓1⟩);
                put(Ty(A → (e → t)), Fo(Can′), [↓]⊥)
                go(⟨↑1⟩); make(⟨↓0⟩); go(⟨↓0⟩);
                put(?Ty(A), Fo(AUX), ?∃x.Fo(x));
                go(⟨↑0⟩); go(⟨↑1⟩); make(⟨↓0⟩);
                IF      ⟨↓0⟩Ty(e)
                THEN    go(⟨↓0⟩)
                ELSE    put(Cat(INV));
                        go(⟨↓0⟩); put(?Ty(e));
ELSE    Abort
```

The sentence *John can jump* would then proceed as shown in figure 22, before the A-LINK ADJUNCTION rule was applied (figure 23). At this point, $\mathbf{A}$ is left underspecified,

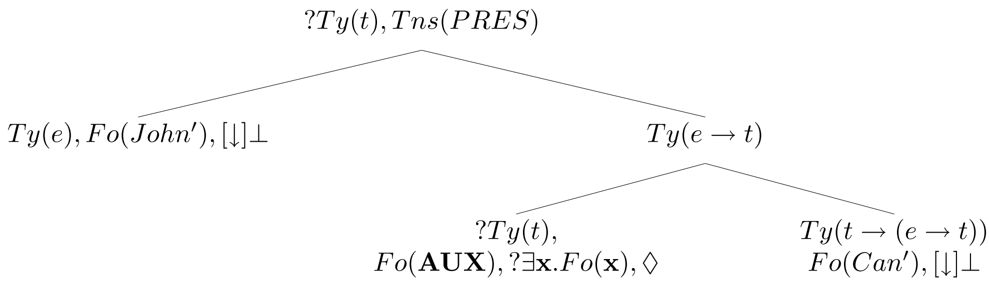

Figure 22: Parsing *John can . . .*

however, it is clear that this needs to be specified in order for a parse to proceed. For the purposes of clarity, I will assume for the time being that in this case it is of $Ty(t)$, and that this is an appropriate type for the rule of A-LINK ADJUNCTION, with possible alternatives to be discussed further in section 6.2.2. INTRODUCTION and PREDICTION would leave the pointer at the $Ty(e \rightarrow t)$ node (or this could be specified in the A-LINK ADJUNCTION rule), whereby *jump* could be parsed.

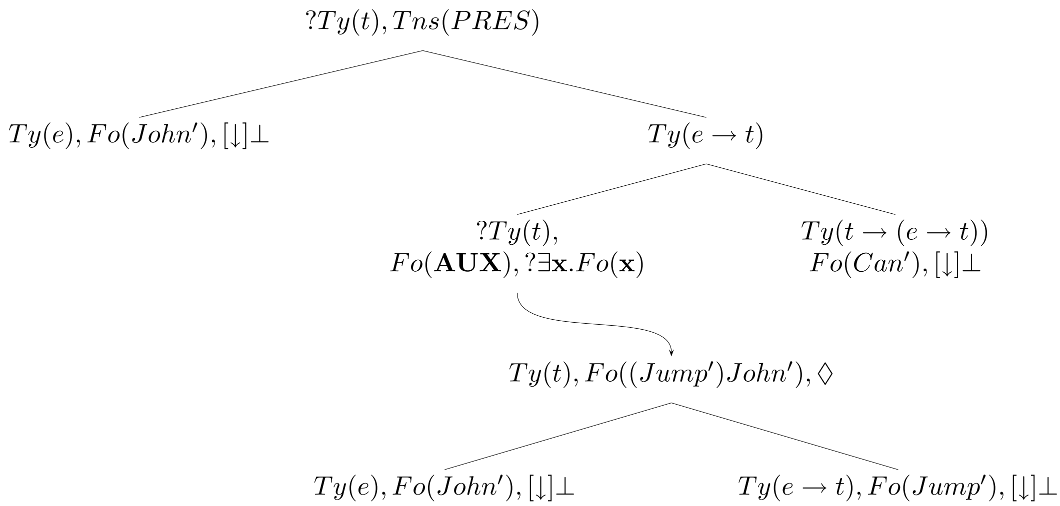

Figure 23: Tree state after parsing *John can jump* using A-LINK ADJUNCTION

All that remains is for the evaluation function to be modified, and, following Cann et al. (2005), I would split the evaluation rule into a two-stage process; "...one that first writes the formula on the LINKed structure onto its host node and then tries to resolve the apparent contradictory result". As they conclude

> It may be, ...that this is the only evaluation rule needed (with types suitably left unrestricted). The information copied onto the host node is then passed up the tree and resolved as locally as possible according to the type of the copied formula and that of the host node.

Like them, I leave this refinement for another time, but the LINK EVALUATION rule that I propose to use here (shown in 89, below) is identical to their formulation except in the type value of the node, which I leave underspecified. The possible type values for $\mathbf{A}$ will be discussed later, in section 6.2.2, but I assume for now that $Ty(t)$ is an appropriate type.

(89) A-LINK EVALUATION:

$$\frac{\{\ldots\{\{Tn(n),\ldots\ Fo(\alpha), ?Ty(\mathbf{A}),\ldots\}\},\{\{\langle L^{-1}\rangle Tn(n),\ldots,Fo(\beta),Ty(\mathbf{A}),\ldots\Diamond\}\}\ldots\}}{\{\ldots\{\{Tn(n),\ldots\ Fo(\alpha),Fo(\beta),Ty(\mathbf{A}),\Diamond\}\},\{\{\langle L^{-1}\rangle Tn(n),\ldots,Fo(\beta),Ty(\mathbf{A}),\ldots\}\}\ldots\}}$$

The application of the A-LINK EVALUATION function shown here would result in the pair of LINKed trees shown in figure 24. The apparent conflict between the multiple formula values at the $Ty(t)$ node, of $Fo(\mathbf{AUX})$ and $Fo((Jump')John')$ is illusory, since a metavariable can subsume any value at all, thus being exactly equivalent to $Fo((Jump')John')$, and leading, through the usual operations of THINNING, COMPLETION and ELIMINATION, to the final output formula $Fo((Can'((Jump')John'))John')$, exactly that shown in 85, as required. In this particular instance, we can see that the $Ty(\mathbf{A})$ node is instantiated as a $Ty(t)$.

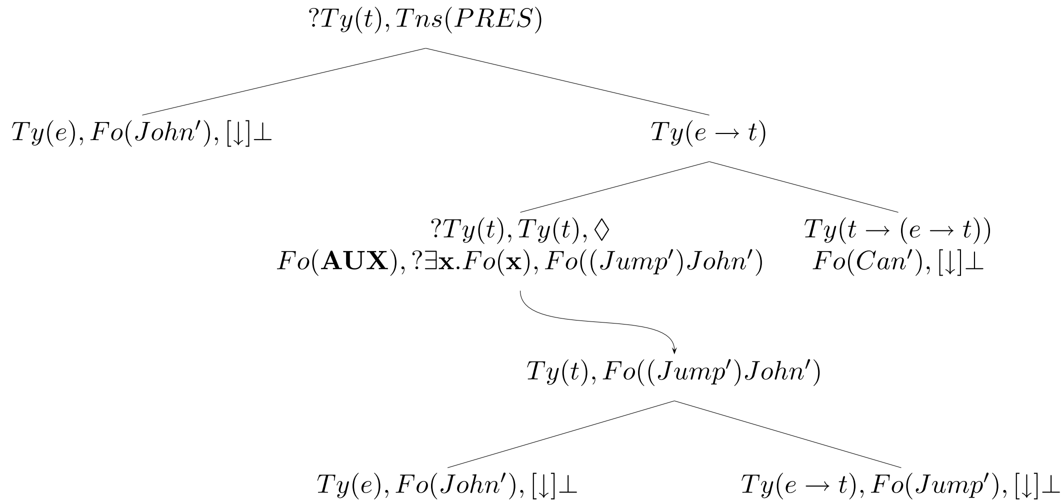

Figure 24: *John can jump* - Tree state after applying A-LINK EVALUATION Rule

More complex examples with multiple auxiliaries proceed in the same fashion, with the assumption that the lexical entry for the verb form that follows restricts what it can occur after. For example *-en* forms would be restricted to $\langle\uparrow_1\rangle\langle L^{-1}\rangle Fo(\mathbf{HAVE})$ or $\langle\uparrow_1\rangle\langle L^{-1}\rangle Fo(\mathbf{BE})$. An example is shown in 25. Repeated applications of the A-LINK EVALUATION RULE, as outlined above, would result in a final formula value as shown in 90, below, however, please note that as the necessary restrictions on the *-ing*, *-en* and base infinitive forms of verbs are not explicitly explored in this paper, (which is a necessary extension of this work to account for the noted ordering restrictions) **John been might have jumping* is currently equally permitted.

(90) $Fo((Might'((Have'((Be'((Jump')John'))John'))John'))John')$

The formula value in 90, above, might seem unnecessarily complex, but does capture the semantic relationships of each ‘segment’ being a component part of the whole. In this view, each of the single trees linked together to form a string with multiple auxiliaries is in some sense a ‘basic’ unit. This helps to account for the ellipsis data mentioned in example 77, earlier, whereby the ellipsis can grammatically occur at many points in the string, in each case needing the final element to be an auxiliary verb. Perhaps, too, the complexity of the formula explains why there is a limit to the possible length of the chain of multiple auxiliary verbs.

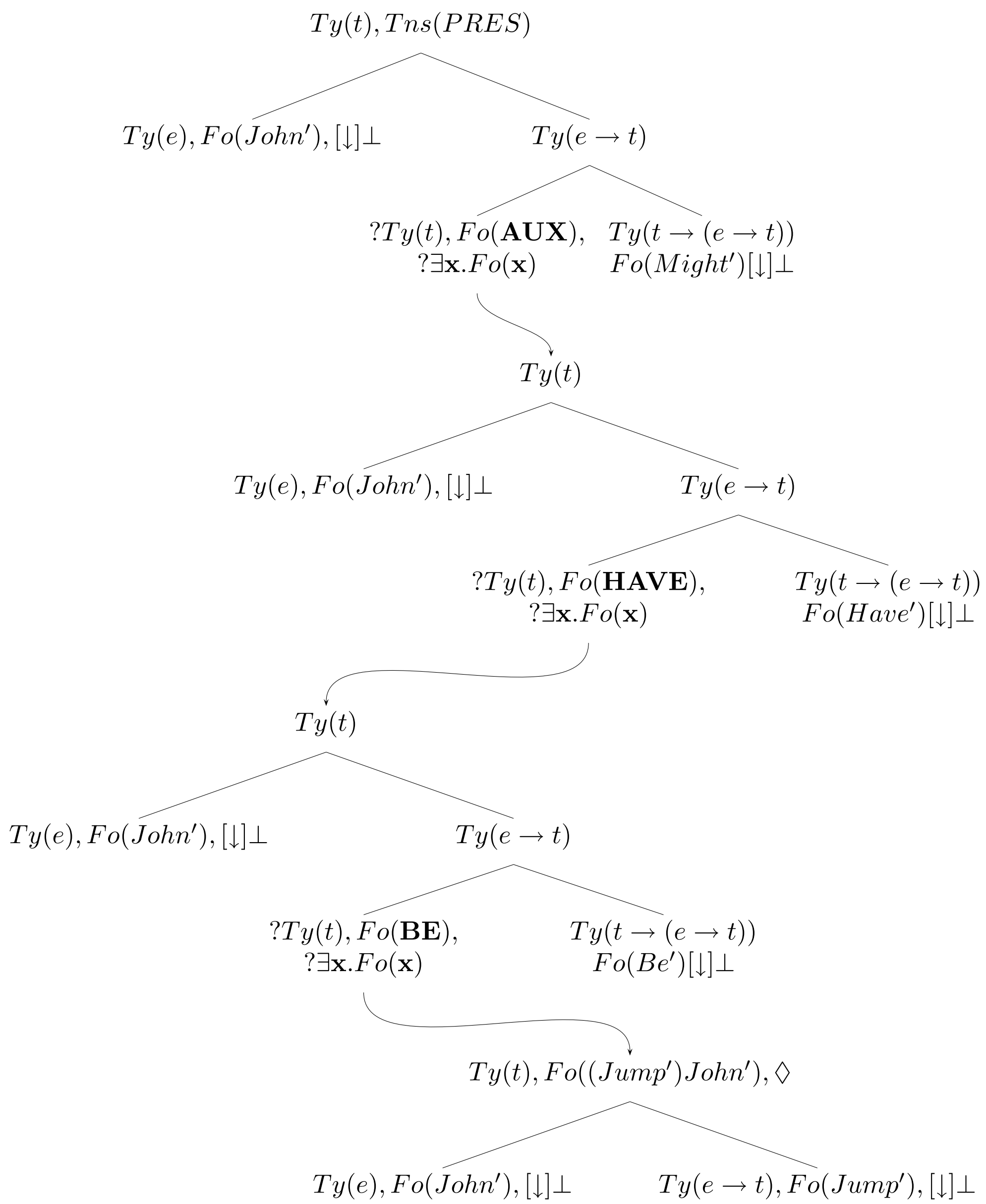

Figure 25: One result of parsing *John might have been jumping*

6.2.2 The Ambiguity of Modals

Does this then mean that I am committed to one basic interpretation for sentences with auxiliaries, encapsulated by the resulting formula values shown in the above? There are two possibilities. Either I am committed to one *basic* interpretation; in which case, like, for example, Papafragou (1998) I would have to suggest that the ambiguity inherent in sentences with modal auxiliaries (for example) be pragmatically resolved *after* an initial syntactic analysis, or, more in keeping with the ethos of Dynamic Syntax, I will have to claim that there is an alternative derivation possible. Choosing the latter option, it is here that I will invoke the notion of *type* underspecification.

There are intriguing questions regarding the ambiguity of the modal auxiliaries, and although it is not my intention to discuss the semantics of modal verbs in depth (which has been done far better and in extensive detail in, for example Coates (1983); Leech (1987); Lewis (1986); Palmer (1988)) I would like to explore briefly some of the implications that the different possible interpretations have for Dynamic Syntax.

To do so, I will briefly outline the differences between so-called *subject raising* and *subject control* verbs, and how this impacts on the modal auxiliaries. These differences are illustrated by the two sentences in 91, which, while superficially similar in structure, embody very different semantic relationships.

(91) a. Jane tends to avoid people.
b. Jane tries to avoid people.

In 91a, what *tends* (to happen) is Jane's avoidance of people. Contrarily, in 91b, what *tries* is Jane, and what is tried is Jane's avoidance of people. From this rough description, it should be clear that semantically, *tends* has only one argument, whilst *tries* has two. From this observation, linguists going back to Chomsky (1957) have categorised these as two distinct classes of verbs; *subject raising* verbs, e.g. *tends*, and *subject control* verbs, e.g. *tries*. In Transformational Grammars, where these names originated, it was proposed that quite different transformations were applicable in each case, expressing the difference between the one and two place predicates. These differences are shown schematically in 92, below.

(92) a. Jane tends [~~Jane~~] to avoid people.
b. Jane$_i$ tries PRO$_i$ to avoid people.

In 92a, the semantic subject of *to avoid people*, *Jane* is raised to the subject position of *tends*. This fact, that *tends* is in some way transparent is handled by traditional Transformational Grammar by a movement operation which *raises* the deep structure subject of the lower verb phrase to a higher position, where it is not assigned any semantic role. In 92b, on the other hand, a silent subject (*PRO*) is posited, meaning that for semantic purposes there are two subjects. This unvoiced subject is co-indexed with the subject of *tries*, which therefore *controls* it, hence the traditional names. The different semantic properties of these two different constructions are also reflected in distinct syntactic behaviours, as shown in 93, below.

(93) a. *Semantically empty subjects*
(i) There tends to be salt in the food.
(ii) *There tries to be salt in the food.
b. *Semantically empty subject - Expletive it*
(i) It tends to be cold in the bedroom.
(ii) *It tries to be cold in the bedroom[14].
c. *Paraphrasing with passives*
(i) Julia tends to visit John = John tends to be visited by Julia
(ii) Julia tries to visit John ≠ John tries to be visited by Julia

Because the auxiliary verbs pass the above tests, and are able, for example, to have semantically empty subjects, an HPSG account (see Sag et al. (2003) for details) treats them as a subset of subject raising verbs, but despite this, it seems clear to me that at times they act as subject raising verbs, and at others as subject control verbs. This is an observation which dates back at least to Ross (1969), and it is clear that which interpretation we choose depends on the semantic structure. For example, even the simple sentence *John could jump*, could mean two different things depending on whether a subject raising or subject control interpretation is assumed[15]. In a subject control interpretation, it could mean that *John once had the ability (for John) to jump* (perhaps his legs are currently broken so he can't right now) or it could equally be interpreted in a subject raising way to mean that *there is the possibility that John will jump* (maybe he is thinking about whether or not to dive off a high board). This state of affairs pertains (to a greater or lesser extent) to all modal auxiliaries, although some may have a preferred reading, and which reading is intended is usually recoverable from context, however, my intention here is not to explore these factors but merely indicate how a single sentence may be interpreted in both ways in Dynamic Syntax.

(94) $Fo((Can'((Jump')John'))John')$

(95) $Fo((Can'(Jump'))John')$[16]

The two different readings should, ideally, be equally derivable, but result in different formula values, corresponding to the idea of there being one or two (identical) subjects,

[14]This sentence could be grammatical with a *referential* subject, but not an *expletive* one. See section 4.5.2 for details.

[15]Of course, in HPSG where there can be a disjunct between syntax and semantics this does not necessarily pose a problem.

[16]This may prove to be inaccurate, and better expressed as $Fo(Can'((Jump')John'))$, but this possibility will be explored further in section 6.3.2.1

as discussed above. It is here that I intend to invoke the notion of type underspecification, and explore a little more fully what is meant by $Ty(\mathbf{A})$. Essentially, my intuition is that the formula value in a subject control reading (with two subjects) should correspond to that shown in 85 (and repeated here as 94), whilst a subject raising interpretation, with a single semantic subject, should correspond to that shown in 95[17].

What this means in terms of our type underspecified $Ty(\mathbf{A})$ node is that it can be instantiated either as a $Ty(t)$ node (as in the examples in section 6.2, above), or as a $Ty(e \rightarrow t)$. The resulting pair of LINKed trees would, in this case, be as shown in figure 26.

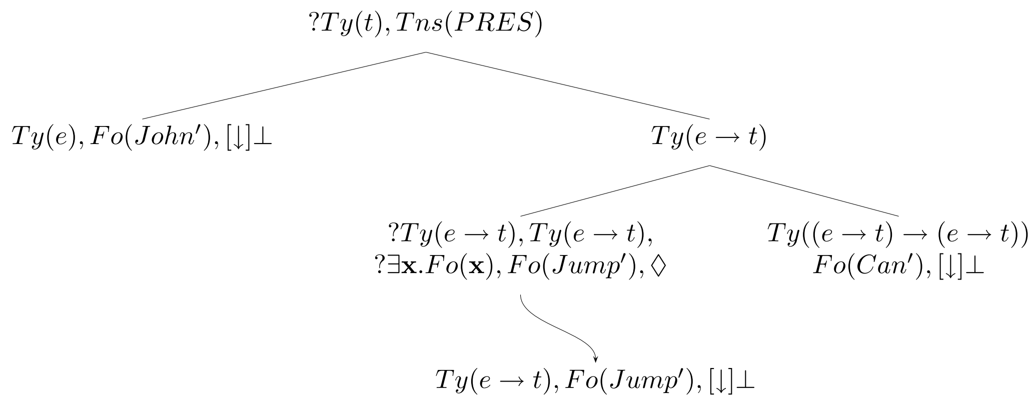

Figure 26: Second possible result of *John can jump*, following A-LINK ADJUNCTION and EVALUATION

Of course, this has implications for the rule of A-LINK ADJUNCTION, because if the $Ty(\mathbf{A})$ is instantiated as a $Ty(e \rightarrow t)$ then the subject does not need to be copied over. What this means in effect is that there are two separate sets of actions that the rule of A-LINK ADJUNCTION can carry out, whose precise effect depends on the trigger condition. In the terminology of lexical actions, as before, this idea can be expressed as a disjunctive lexical entry, as shown in 96. The rule of A-LINK EVALUATION (as shown in 89) merely requires us to specify $\mathbf{A} \in \{t, (e \rightarrow t)\}$, so that acceptable values that can be substituted into the rule of A-LINK EVALUATION for $Ty(\mathbf{A})$ are $Ty(t)$ or $Ty(e \rightarrow t)$. However, it is possible that this over complicates the issue, as when the potential LINKed tree were of a $Ty(e \rightarrow t)$, because as we do not need to copy over any information, there is no reason that we need to project a LINKed structure at all, as we could simply parse the word *jump* at the current node. This would have an impact on the restrictions for non-finite forms of verbs, which will not be explored in detail in the current paper.

[17] I am making the assumption (as per Kempson et al. (2001), pp.33) that the formula values can be interpreted as lambda terms (see, e.g. Carpenter (1997)), meaning that $Fo(Jump)$ should really be read as $\lambda x.Jump(x)$. In this analysis therefore, $Fo(Can')$ is either a one place predicate $\lambda x.Can(x)$, or a two place predicate $\lambda x \lambda y.Can(y)(x)$. The common assumption that the subject control reading corresponds to the two-place predicate interpretation is made implicitly.

(96) *A-LINK Adjunction*

$$\left|\begin{array}{llll}
\text{IF} & \langle\uparrow_0\rangle\langle\uparrow_1\rangle\langle\downarrow_0\rangle(Ty(e)\wedge Fo(\alpha)) & & \\
\text{THEN} & \text{IF} & ?Ty(t) & \\
 & \text{THEN} & \texttt{make}(\langle L\rangle);\texttt{go}(\langle L\rangle); & \\
 & & \texttt{put}(?Ty(t),\langle\downarrow_0\rangle Fo(\alpha)) & \\
 & \text{ELSE} & \text{IF} & ?Ty(e\rightarrow t) \\
 & & \text{THEN} & \texttt{make}(\langle L\rangle);\texttt{go}(\langle L\rangle); \\
 & & & \texttt{put}(?Ty(e\rightarrow t)) \\
 & & \text{ELSE} & \texttt{Abort} \\
 & \text{ELSE} & \texttt{Abort} & \\
\text{ELSE} & \texttt{Abort} & &
\end{array}\right.$$

How I envisage this proceeding is with the notion of potential parses building up in parallel. In other words, for a sentence such as *John can jump*, when *can* was parsed, this would lead to the creation of two separate parallel trees; one in which the underspecified $Ty(\mathbf{A})$ was instantiated as $Ty(t)$ and one in which it was instantiated as a $Ty(e \rightarrow t)$. This parallel processing could explain why both interpretations are in principle available, with the choice of which is ultimately chosen resting on additional factors, for example, the preceding context.

6.3 A Dynamic Syntax Account of Negation

The preceding sections, motivated by the syntactic phenomena of Inversion (discussed in section 2.1.2), and Ellipsis (discussed in section 2.1.3), need revising in the light of certain puzzling facts, for example, regarding the distribution of certain negative polarity adverbs such as *never* and *rarely*, as shown in example 97, below.

(97) a. I [never want to be bothered].
b. I want [never to be bothered].
c. ??I want to be [never bothered][18].
d. I want to [never be bothered].
e. *[Never want I to be bothered].
f. I [never could be bothered].
g. I could [never be bothered].
h. ??I could be [never bothered].
i. [Never could I be bothered].

[18]For some speakers this is completely unacceptable, possibly due to semantic factors. The example below, which adds information about the past participle, seems more acceptable.

(i) ?I want to be [never bothered about money].

This also holds for example (h), and, possibly, the equivalent sentences in 98.

One potential solution to this distributional problem is to have a disjunctive lexical entry for *never*, such that if it is the first item encountered in a string, it triggers an inverted context, which could only be met by an auxiliary verb. However, this would also lead to additional complexity in the lexical entry for auxiliary verbs, and the sentences in 97 lead me to believe that it is actually a more general phenomena, which can be articulated as "I must be followed by a verb or verbal element." If this is the case and *never* is sentence initial, it automatically follows that the only type of verb that is allowed to follow in a successful parse is one that can precede its subject; i.e. an auxiliary. In fact, the distribution has parallels to that displayed by *not* (shown in 98).

(98) a. *I [not want to be bothered (ever)].
b. I want [not to be bothered (ever)].
c. ?I want to be [not bothered (ever)].
d. I want to [not be bothered (ever)].
e. *[Not want I to be bothered (ever)].
f. I [don't want to be bothered (ever)].
g. *I [not could be bothered (ever)].
h. I could [not be bothered (ever)].
i. ?I could be [not bothered (ever)].
j. *[Not could I be bothered (ever)].

Kim (2000) believes that these parallels are because both *never* and *not* act as negative polarity adverbs, which are, in the HPSG terms in which his analysis is based, modifiers of the following verb. Whilst this appears to be able to account for the syntactic placement of these elements (though, importantly, not without raising other issues, especially in regard to VP-ellipsis), it does not seem to capture the semantic facts. For Kim, this is not a major issue, because in HPSG there can be a separation between the syntactic contribution of a word and its semantic contribution (seen most notably in the HPSG treatment of the different forms of *be*, which sometimes is semantically empty, but sometimes not, thereby supposedly accounting for what Sag et al. (2003) consider to be syntactic homonymy but without semantic homonymy), however, this is not the case for Dynamic Syntax where the syntactic factors build up the semantic picture.

What can be seen from the combination of insights that *never* appears to negate the following clause, but also seems to incorporate a $Ty(e)$ element (*ever*) can be accounted for in Dynamic Syntax by examining the diachronic roots of *never*. In essence, *never* is derived from two different elements, corresponding, in Modern English, to *not* and *ever*. I propose to treat it therefore as an almost prototypical example of routinisation, whereby the negative element, which was originally separate (in Old English *ne*) and the lexical actions it invoked became a subpart of the lexical actions for *never*. In other words, we would expect *never* to perform *all* the actions that *ne* did (which are very similar to those that *not* now performs), as well as introducing a $Ty(e)$ element, as *ever* does. This brief description, to be later elaborated in section 6.3.3, means that we essentially get the placement data which Kim treats as the main characterisation of *never*, for free. The difference, then, is in fact in cases of *sentential*[19] *not*, the only case where auxiliaries are not only permitted, but strictly necessary (and the only cases where contraction is possible, or even preferred), as well as in the introduction of a $Ty(e)$ element in the case of *never*. This suggests that, contrary to many existing analyses, the exceptional case is the one which commentators (e.g. Sag et al. (2003)) usually focus on explaining when they consider the linguistic concept of negation, i.e. *sentential negation*. In my view, a failure to account for *constituent negation* in the same analysis represents a failure to appreciate the facts. Whether or not we accept that there are any real differences between the types of negation does not seem to me to be an important distinction, as, if there are, this is a question of scale (or, indeed, scope), not of principle.

In attempting to motivate my analysis of negation, therefore, I will assume a common structure for constituent and sentential negation, with the only difference being the behaviour of the element *not* when it is used to negate the first verbal element in a string (or, more accurately, the highest $Ty(t)$ node in a given tree). If the element is an auxiliary, unlike in the normal case where the negation precedes the verbal clause that it negates, in this case it follows. In addition, if the verb is a main verb, then although the negation does precede the verb which it negates (as in the other cases of constituent negation), it is somehow too weak an element to stand alone, and therefore requires an auxiliary verb for 'support'. This is the only contrast with the case in Early Modern English, where the initial negator could follow any verb, but was not in itself strong enough to precede it (although there have been a few examples found, dating roughly from the time corresponding to Hudson's Stage B (see 47), when English was allegedly in a state of flux, where *not* precedes the main verb. See Mazzon (2004), for discussion of these examples and also Jesperson's NEG-cycle, which is presumed to explain the historical cross-linguistic patterns of negation). A similar effect is seen with certain clitics in Spanish, for example, whereby so-called weak clitics cannot be sentence initial (see e.g. Bouzouita & Kempson (2006)), leading to the Wackernagel effect. In line with this explanation, if there are multiple auxiliaries in a sentence, then it is only the first one which negation follows, as shown in example 99, below. An advantage of this approach is that it can help to show why the initial *not* in a string is the only one which became enclitic, which can be seen from the ungrammaticality of example 99h-j. That said, I will argue, following Zwicky & Pullum (1983) that the evidence suggests that, although initially an enclitic particle, *n't* has since been reanalysed as an inflectional form - a reanalysis that, as we shall see, leads to a simpler parse tree, and also explains why the preferred reading of example 99b is that indicated by the square brackets.

(99) a. *I not could have been being bothered.
b. I could [not have been being bothered.][20]

[19]The terms *sentential* and *constituent* negation refer to the scope of the negative element. In *sentential* negation, the whole clause is negated; in *constituent* negation it is some part of it only.

[20]this example is ambiguous between the two readings whereby it could be the 'have been being bothered'

c. I could have [not been being bothered.]
d. I could have been [not being bothered.]
e. I could have been being [not bothered.]
f. I couldn't [not have been being bothered.]
g. I couldn't [not have [not been [not being [not bothered.]]]]
h. *I could haven't been being bothered
i. *I could have beenn't being bothered
j. *I could have been beingn't bothered

The way in which this problem is analysed in Dynamic Syntax relies upon some fairly crucial assumptions regarding where in the final parse tree the negation resides. There are several possibilities, however, due to the nature of the Dynamic Syntax model, a couple of them can be ruled out immediately. This relates to a point made forcefully in Marten; that in Dynamic Syntax, once something exists in a tree, it cannot be changed. Of course, the system seems fairly flexible when we consider such things as unfixed nodes, but the important thing to realise is that information at the node does not change when it finds a fixed position; every decoration that was at the node still holds, even if some new decorations narrow down the possibilities. The reason that this has implications for negation is that, if we consider the effects of negation to hold at the verbal $Ty(e \rightarrow t)$ node (which would match the usual assumption for *constituent* negation, but require us to assume that linguistic negation is in some sense different from logical negation), then we would have to subdivide the node into two; with the original node in effect moving down the tree to a new argument node, and gaining a new functional sister node of the type $Ty((e \rightarrow t) \rightarrow (e \rightarrow t))$ which contains the negation. An alternative methodology might be to annotate the existing $Ty(e \rightarrow t)$ (or indeed the root node) with a formula instruction to negate the formula. Unfortunately, whilst this might seem to be an efficient alternative, the fact that nothing can be taken away in the Dynamic Syntax parse tree would mean that the resulting node would constitute a Reductio Ad Absurdum, because it would be annotated both with the relevant formula value, and its negation. Alternatives which do not alter the underlying structure of the tree, or induce contradictions are few and far between, however, I do not believe the problem to be an intractable one.

There are two basic options available, and the one that we choose depends to a large degree on the presumed structure of negation. The two, conflicting, options are shown pictorially in figure 27. Notice that in the former potential structure, the formula value is $Fo(Neg')$, whilst in the latter it is $Fo(\neg)$. This is because logical negation ($\neg$), by its very definition negates a propositional phrase (depicted in Dynamic Syntax as $Ty(t)$), resulting in another propositional phrase. This being the case, $Fo(\neg)$ can *only* decorate a $Ty(t \rightarrow t)$ node. Analyses like Kim (2000), however, do not treat negation as a logical operator, as they consider it to be a modifier of a verb phrase. It might seem unlikely to assume that linguistic negation does not act as it does in propositional logic but is not unthinkable, if we assume that logical negation ($Fo(\neg)$) may have been the starting

being negated, or the whole sentence, as can be seen from 99f.

point for *not* but it has since been reanalysed as the less restricted $Fo(Neg')$. Of course, although many syntactic analyses do not show the semantic relationships as explicitly as Dynamic Syntax, we could interpret HPSG, for example, as postulating the former type of sister relationship for *constituent negation*, with the latter representing the *sentential* case (see Kim (2000), chapter 3, for an HPSG analysis where *not* sometimes acts as a *complement* and sometimes as a *modifier*).

The difficulties associated with the first type of analysis have already been briefly examined, and are discussed at length in Marten (2002). The second, sentential ($Ty(t \rightarrow t)$) analysis has problems of its own. Primary among these is the fact that, should we consider the traditional starting point in Dynamic Syntax to be as stated, then the *not* element would, when encountered in the string, be required to create a node above the existing root node. An alternative option, to start with a template with the space for a polarity item as a $Ty(t \rightarrow t)$, as shown in the second tree in figure 27 is also possible, but unattractive.

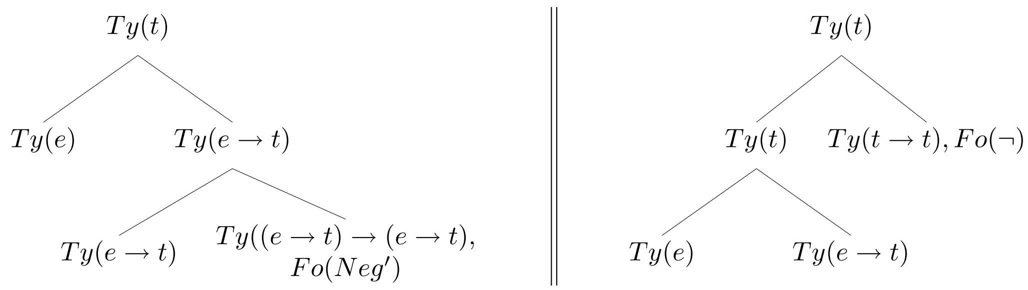

Figure 27: Possible candidates for structure of negatives

6.3.1 Negation in Dynamic Syntax - Option 1

Taking the two options in turn, if we assume that *not* decorates a sister node of the verbal node, be it the required auxiliary if the negated item is the first in the string, or any other verb if it is later in the string, then the obvious place for the negative when it occurs after an auxiliary verb (*sentential negation*) is at the $Ty(\mathbf{A})$ node. Conversely, when considering cases of so-called *constituent negation*, the correct positioning would seem to reflect the structure shown in the first instance of figure 27; at the $Ty(e \rightarrow t)$ node[21]. This would lead to a lexical entry for not like that shown in 100, below.

[21]This leaves aside, for the moment, cases where *not* negates a propositional phrase or a noun phrase as in examples of constituent negation:

a. John loves not Mary, but Julia.

b. I come not from Bedford, but from London.

though there is no principled reason why these could not be incorporated into a wider analysis of *not*.

(100) Lexical entry for *not* - Option 1.

not	IF	$?Ty(\mathbf{A})$	where $\mathbf{A} \in \{t, (e \rightarrow t)\}$
	THEN	IF	$\langle U\rangle Ty(e \rightarrow t) \wedge \langle U\rangle Ty(t)$
		THEN	$\texttt{put}(Ty(\mathbf{A})); \texttt{make}(\langle\downarrow_1\rangle); \texttt{go}(\langle\downarrow_1\rangle);$
			$\texttt{put}(Ty(\mathbf{A} \rightarrow \mathbf{A}), Fo(Neg'), [\downarrow]\bot)$
			$\texttt{go}(\langle\uparrow_1\rangle); \texttt{make}(\langle\downarrow_0\rangle); \texttt{go}(\langle\downarrow_0\rangle);$
			$\texttt{put}(?Ty(\mathbf{A}), Fo(\mathbf{AUX'}), ?\exists\mathbf{x}.Fo(\mathbf{x}));$
		ELSE	Abort
	ELSE	Abort	

To see how this would work with a simple example, consider the sentence *John can not jump*. Having parsed *John can* ..., and moved through the tree to the $?Ty(\mathbf{A})$ node with the outstanding requirement, we will, as before, be in the parse state depicted in the tree shown in figure 22[22]. The next word encountered in the string is *not*. The first restriction $?Ty(\mathbf{A})where(\mathbf{A} \in \{t, (e \rightarrow t)\}$ restricts the node types at which *not* can be parsed to a node with a requirement for a $?Ty(t)$ or $?Ty(e \rightarrow t)$. The second restriction, $\langle U\rangle Ty(e \rightarrow t) \wedge \langle U\rangle Ty(t)$, says that somewhere above or at the current node in the tree, or in a linked tree above the current node[23] are both a satisfied $Ty(e \rightarrow t)$ node and a $Ty(t)$ node[24]. In the current case, both these type restrictions are met, so the lexical actions contained in 100 can be applied. The result, having parsed *not*, is shown in figure 28.

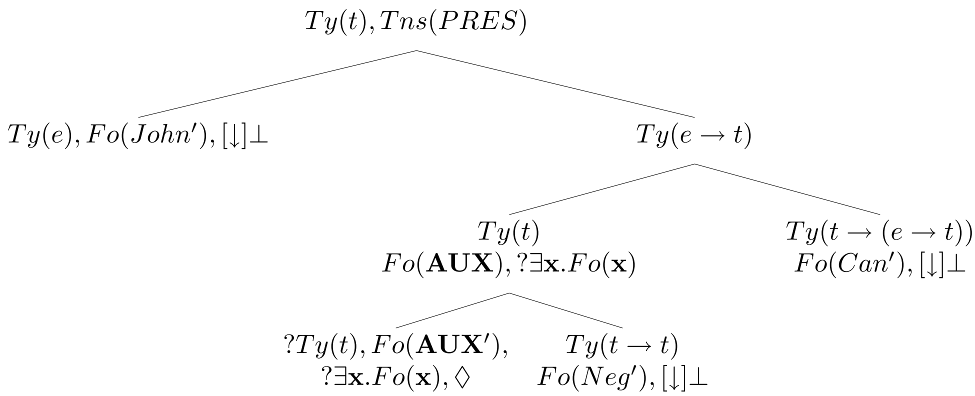

Figure 28: Parsing *John can not* ...

As before, the metavariable on the node where the pointer is ($Fo(\mathbf{AUX'})$) can be supplied from context, or, in this case, the rule of A-LINK ADJUNCTION can be applied so that following lexical material may be parsed. As discussed previously, the criteria for the application of A-LINK ADJUNCTION must be made slightly more general, as shown in 101, below, where the criteria is now $?Ty(\mathbf{A}) \wedge \langle\uparrow_0^*\rangle\langle\uparrow_1\rangle\langle\uparrow_0\rangle(Ty(e) \wedge Fo(\alpha))$. The only

[22]Although I assume that two trees are being built up simultaneously, I will illustrate the current section using the version whereby the $Ty(\mathbf{A})$ is instantiated as $Ty(t)$, for clarity.

[23]$\langle U\rangle$ is a modal operator meaning 'up anywhere', and is a very weak tree relation. Its inverse is $\langle D\rangle$, down.

[24]It is possible that these restrictions will license certain ungrammatical strings when relative clauses are considered, and thus may need replacing with a more restricted constraint.

difference between here and before is that the collection of the subject formula value to be copied onto the LINKed structure can now require us to go up more than one argument node, before going up a functor node and down an argument daughter, as shown by $\langle\uparrow_0^*\rangle$. What this means is that in examples such as that shown in figure 28, the formula value for *John*, which is $\langle\uparrow_0\rangle\langle\uparrow_0\rangle\langle\uparrow_1\rangle\langle\downarrow_0\rangle$ away from the node where the pointer is, can still be collected. Importantly, the more general $\langle\uparrow_*\rangle$ modal operator would be inappropriate because it would allow us to pick up the wrong noun phrase in more complex sentences. For example, in *John thinks Mary can jump*, it would allow us to carry the formula $Fo(John')$ through to the LINKed tree, which is clearly inappropriate[25]. Notice also, that the condition also includes that the node has a requirement for a formula value ($?\exists\mathbf{x}.Fo(\mathbf{x})$). This condition would be met by the node created by the lexical actions of an auxiliary verb, but importantly would not be met by nodes introduced from lexical actions (e.g. in cases where the word creates a node for a $Ty(t)$ or a $Ty(e \rightarrow t)$ argument, as *thinks* or *wants* would do), thus preventing A-LINK ADJUNCTION from being carried out in these cases. This has important implications for the contribution of *infinitival-to* in examples such as *John wants to like Mary*, which will be discussed further in section 6.6.

The parse tree for *John can not jump*, before terms are collected, is shown in 29.

(101) *A-LINK Adjunction*

$$\left|\begin{array}{llll}
\text{IF} & \langle\uparrow_0^*\rangle\langle\uparrow_1\rangle\langle\downarrow_0\rangle(Ty(e) \wedge Fo(\alpha)) & & \\
\text{THEN} & \text{IF} & ?Ty(t) \wedge ?\exists\mathbf{x}.Fo(\mathbf{x}) & \\
 & \text{THEN} & \texttt{make}(\langle L\rangle);\texttt{go}(\langle L\rangle); & \\
 & & \texttt{put}(?Ty(t), \langle\downarrow_*\rangle(Ty(e) \wedge Fo(\alpha))) & \\
 & \text{ELSE} & \text{IF} & ?Ty(e \rightarrow t) \wedge ?\exists\mathbf{x}.Fo(\mathbf{x}) \\
 & & \text{THEN} & \texttt{make}(\langle L\rangle);\texttt{go}(\langle L\rangle); \\
 & & & \texttt{put}(?Ty(e \rightarrow t)) \\
 & & \text{ELSE} & \texttt{Abort} \\
 & \text{ELSE} & \texttt{Abort} & \\
\text{ELSE} & \texttt{Abort} & &
\end{array}\right.$$

In this case, as we can see, the LINKed structure will be of type $Ty(t)$, meaning that *all* occurrences of $\mathbf{A}$ are t. However, the collection of formula terms, using a combination of ELIMINATION and A-LINK EVALUATION leads to the final formula value shown below in 102. It is clear that this is not the correct interpretation, in which case it is either necessary to adjust the ELIMINATION rule where the formula is Neg' such that it would be the left-most item in the formula value and thus have scope over everything else in the formula describing the tree in which it resides, or acknowledge that linguistic negation does in fact mirror logical negation and explore the available options for the second possible structure of negation shown in 27, above. This option will be explored further, in section 6.3.2, below, but in any case, it is important to note that there is an alternative derivation available for the string *John can not jump*, which is as it should be, given the ambiguity between the two readings *John [can not jump]* and *John can*

[25] An alternative restriction using the $\texttt{gofirst}_\uparrow$ operator (defined in Kempson et al. (2001), pp.91) would equally be able to prevent such capturing of inappropriate noun phrases.

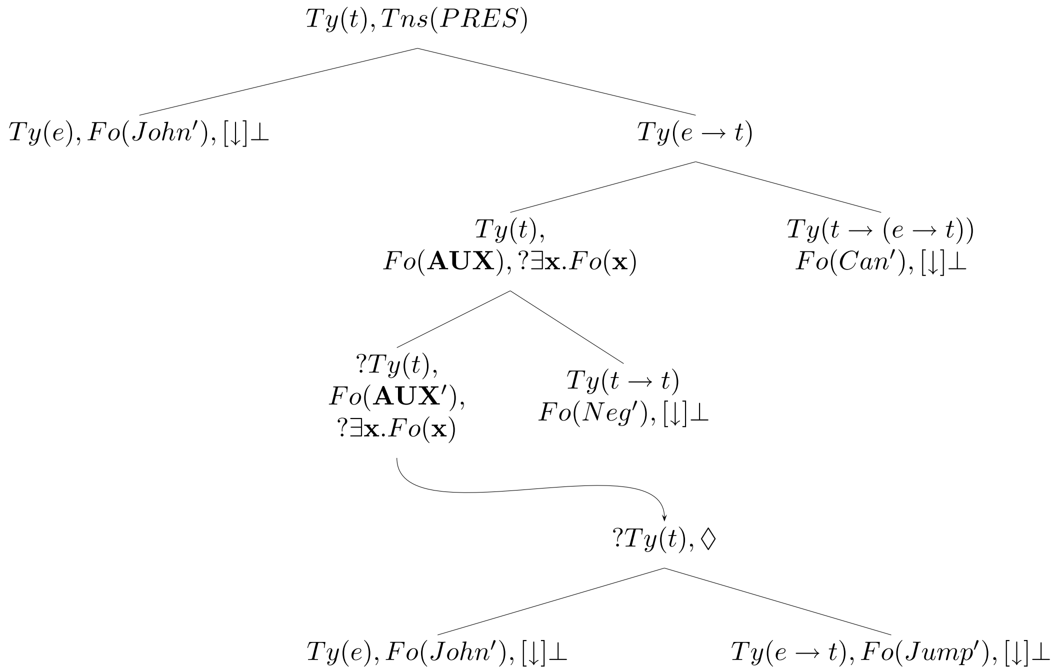

Figure 29: Parse tree after parsing *John can not jump*

[not jump]. The tree for the alternative derivation is shown in figure 30, and the subtly different resulting formula (using A-LINK EVALUATION and the existing ELIMINATION rule) in 103. The fact that there are two different ways of parsing *John can not jump* is an important point, as it allows us to account for sentences with multiple negation, such as *John can not NOT jump*.

(102) $Fo((Can'(Neg'(Jump'(John'))))John')$

(103) $Fo((Can'((Neg'(Jump'))John'))John')$

This account of negation actually allows far more complex examples of multiple negation, such as *John can not not have not jumped* or even *John can not not have not not jumped*, and whilst these sound strange, they do not sound ungrammatical (with appropriate stress), so I would argue that this is a strength of the analysis, and that the fact that such sentences are rarely, if ever, encountered, reflects on their complexity, and not on their ungrammaticality. This is in contrast to strings such as **John not jumps* and **John jumps not*, which are both debarred in this analysis because the first condition in the lexical entry of *not* is not met. Examples such as *Can not I* ... are also ungrammatical in this account, however, in this case it is because, following the parse of *can* the pointer is at a $Ty(e)$ node, which means that the lexical actions for *not* cannot be invoked. As mentioned in footnote 21, this may require changing in the light of contrastive negations as well as in acknowledgement of the fact that sentences beginning, for example, *Can*

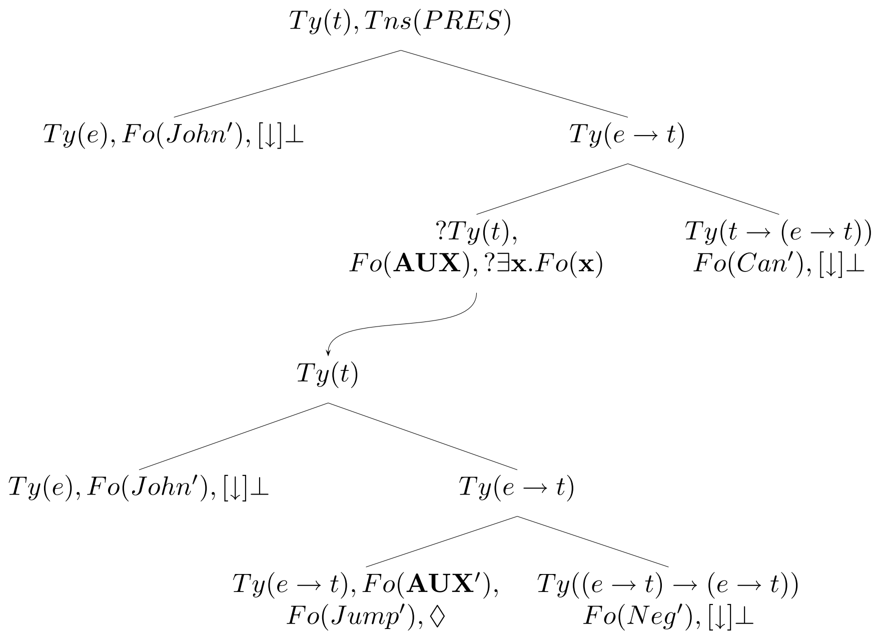

Figure 30: Parse tree after parsing *John can not jump* - alternative derivation

not Mary ... may well be considered archaic, but not necessarily strictly ungrammatical. However, these are subjects for further investigation, and beyond the scope of the current paper.

6.3.2 Negation in Dynamic Syntax - Option 2

Given the formula values in 102 and 103, it seems obvious that these do not adequately characterise negation in English. In fact, the formula value in 102 seems to be an accurate one for the interpretation *John can [not jump]* (repeated here as 104, with the logical operator replacing Neg'), while *John [can not jump]* would be better described by the formula value shown in 105, below. These formula values map onto those which would be the outcome if the second structure shown in 27 represented the final parse tree.

(104) $Fo((Can'(\neg(Jump'(John'))))John')$

(105) $Fo(\neg((Can'(Jump'(John')))John'))$

Treating negation as introducing the logical operator (¬) leads, as discussed above, to the assumption that *not* can only decorate a $Ty(t \rightarrow t)$ node. This means that all the possible placements for *not* are accounted for, with the notable exception of the fact that it cannot be the first element in a sentence, instead requiring that an auxiliary and subject have already been parsed. This leads me to postulate a disjunctive lexical entry, as shown below in 106. In essence, the first restriction ensures that the word is not the first encountered in the string[26], and the first set of lexical actions applies if there has already been an auxiliary verb parsed in the string.

(106) Lexical entry for *not* - Option 2.

$$
\textit{not}\;\left|\;\begin{array}{llll}
\text{IF} & ?Ty(t) & & \\
\text{THEN} & \text{IF} & [\downarrow]\bot \wedge [\uparrow]\bot & \\
 & \text{THEN} & \texttt{Abort} & \\
 & \text{ELSE} & \text{IF} & \langle\downarrow_*\rangle Ty(e) \wedge \langle\downarrow_*\rangle Ty(\mathbf{A} \rightarrow (e \rightarrow t)) \wedge [\uparrow]\bot \\
 & & & \text{where } \mathbf{A} \in \{(e \rightarrow t), t\} \\
 & & \text{THEN} & \texttt{make}(\langle\uparrow_0\rangle); \texttt{go}(\langle\uparrow_0\rangle) \\
 & & & \texttt{put}(?Ty(t)); \texttt{make}(\langle\downarrow_1\rangle); \texttt{go}(\langle\downarrow_1\rangle); \\
 & & & \texttt{put}(Ty(t \rightarrow t), Fo(\neg), [\downarrow]\bot); \\
 & & & \texttt{go}(\langle\uparrow_1\rangle); \texttt{go}(\langle\downarrow_0\rangle) \\
 & & \text{ELSE} & \texttt{put}(Ty(t)); \texttt{make}(\langle\downarrow_1\rangle); \texttt{go}(\langle\downarrow_1\rangle); \\
 & & & \texttt{put}(Ty(t \rightarrow t), Fo(\neg), [\downarrow]\bot) \\
 & & & \texttt{go}(\langle\uparrow_1\rangle); \texttt{make}(\langle\downarrow_0\rangle); \texttt{go}(\langle\downarrow_0\rangle); \\
 & & & \texttt{put}(?Ty(t), Fo(\mathbf{AUX}'), ?\exists\mathbf{x}.Fo(\mathbf{x})) \\
\text{ELSE} & \texttt{Abort} & &
\end{array}\right.
$$

The important point about this lexical entry is that there are potentially two parse sites in a simple tree. One, if the $\langle\downarrow_*\rangle Ty(e) \wedge \langle\downarrow_*\rangle Ty(\mathbf{A} \rightarrow (e \rightarrow t)) \wedge [\uparrow]\bot$ restriction[27] is met, is the root of the tree. The other, is after an auxiliary or verb taking a verbal complement has been parsed, where, although the node from which it will be parsed has a $?Ty(\mathbf{A})$, this can be (as seen in the positive examples discussed earlier), instantiated as a $Ty(t)$, meaning that $?Ty(\mathbf{A})$ could be $?Ty(t)$, in which case the initial trigger is met. This has implications for negative sentences which will be further discussed later after a few simple examples. The process by which this lexical entry licenses strings such as *John can not jump* in two ways, and debars examples such as *John jumps not*, and *John not jumps*, is illustrated below.

Taking the earlier example of *John can not jump*, we can, as before, parse the string up to *John can . . .* , as in figure 22. The two different options available depend on where the pointer is when we parse *not*. If it is at the root node (to which we can move through the tree using the rule of ANTICIPATION), then the first criterion is met, but the second ($[\downarrow]\bot \wedge [\uparrow]\bot$) is not, because there is already information in the tree. This means we can continue to the next criterion, $\langle\downarrow_*\rangle Ty(e) \wedge \langle\downarrow_*\rangle Ty(\mathbf{A} \rightarrow (e \rightarrow t)) \wedge [\uparrow]\bot$, which is, in this case, met (by the nodes decorated with $Fo(John')$ and $Fo(Can')$, respectively), so the actions are applied. The results of these actions are shown in figure 31. Using

[26] This means for the moment that sentences such as *Not only did she . . .* will be debarred, although it seems evident that this highlights even further the connection between *not* and *never*, which I shall return to in section 6.3.3, later.

[27] This restriction says roughly: "If below me there is already a parsed $Ty(e)$ node and a parsed $Ty(\mathbf{A} \rightarrow (e \rightarrow t))$ node and there is nothing above me"

COMPLETION and ANTICIPATION allows us to move through the tree to the only node with outstanding requirements, where, exactly as before, the $Fo(\mathbf{AUX})$ node can be updated from context, or with following lexical material appearing on a LINKed structure. The resulting formula of parsing (*John can not jump*) in this way would be as shown in 105.

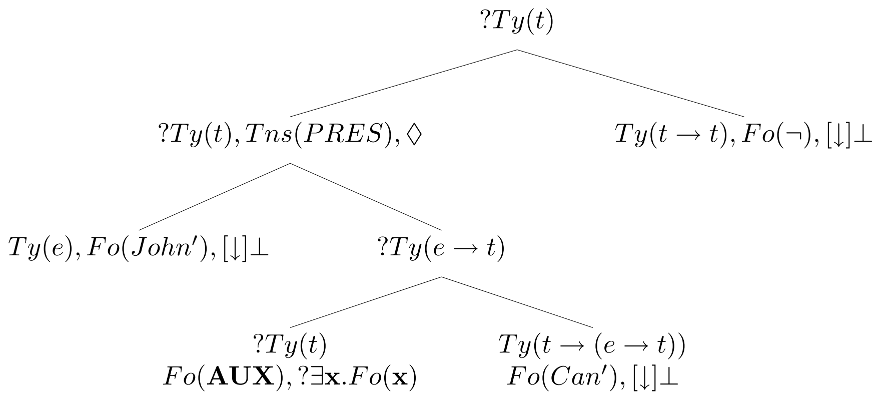

Figure 31: Parsing *John can not ...*

The alternative parse is possible if the pointer is allowed (via COMPLETION and ANTICIPATION) to move through the parse tree to the $?Ty(\mathbf{A})$ node *before* not is processed. In this case, the criterion $\langle\downarrow_*\rangle Ty(e) \wedge \langle\downarrow_*\rangle Ty(\mathbf{A} \rightarrow (e \rightarrow t)) \wedge [\uparrow]\bot$ would *not* be met, so the actions carried out would be those after the second ELSE statement. The results are shown in figure 32, and the resulting formula value following the complete parse would be that shown in 104. The inclusion of the metavariable in the lexical entry for *not* in this case ensures that A-LINK ADJUNCTION (as shown in 101) can apply, and has other implications for ellipsis which will be discussed below, and again in section 6.3.4.

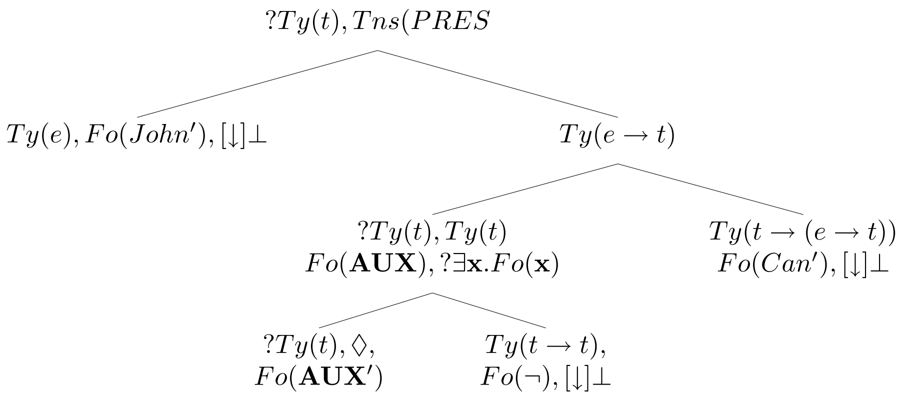

Figure 32: Parsing *John can not ...*

There are some important points to note about these two, parallel parses. The main one is to note what would occur in a tree with multiple occurrences of *not* or multiple auxiliaries. In a sentence such as *John can not not have jumped*, although the first *not* can be sentential (in these terms, introduce a node above the root node, as per the first set of actions in the lexical entry for *not*), the second cannot, and must be attached to the lower $?Ty(\mathbf{A})$ node.

The results of such a parse are shown in figure 33, below. Note that there is nothing in principle that prevents not iterating indefinitely, legitimising strings such as *John can not not not have jumped* but that if it does so it is only at the lower attachment site in each case, and not at the root node. Note also that because *not* also introduces a metavariable, ellipsis is possible in these cases (e.g *?John can not not*). This may be an unwanted conclusion, although, as it is not difficult to construct sentences of this form which seem acceptable it may be that the apparent unacceptability of sentences with additional *not*s is likely to be a processing constraint rather than a grammatical one. It is possible that the lexical action for *not* introducing a metavariable in cases at the lower attachment site is not appropriate, but as such cases of ellipsis seem acceptable to me (although they are completely unacceptable for Kim (2000)) I will leave it in for now, and assume that this could be a source of variability between speakers. I shall return to this point in 6.3.4, below.

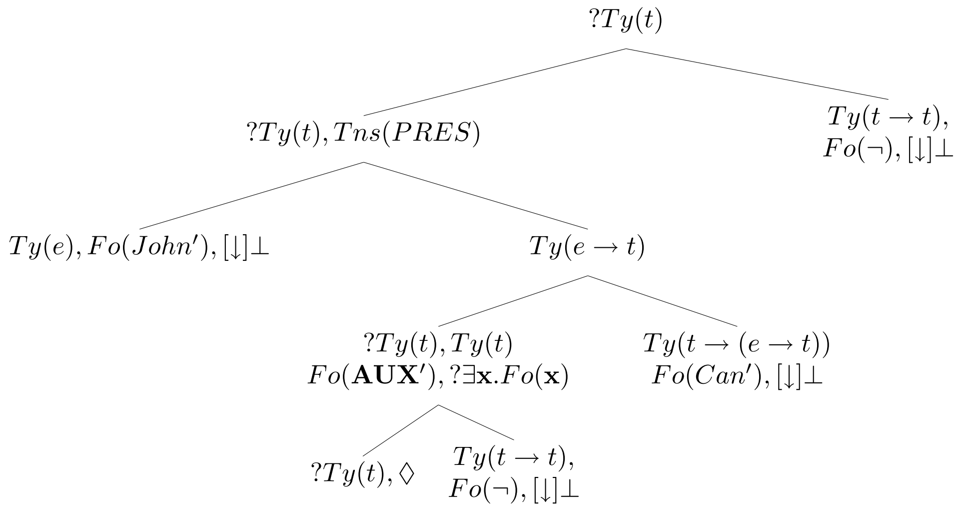

Figure 33: Parsing *John can not not...*

Sentences of the type **John jumps not*, are debarred, because the first set of actions would not be triggered due to there being no $Ty(A \to (e \to t))$ node in the tree, and the second set of actions would result in the functor daughter of the root node being decorated with both $Ty(e \to t)$ and $Ty(t \to t)$, thus causing a contradiction and the parse to fail. Similarly, **John not jumps* would be ungrammatical because, again, there would be no satisfied $Ty(A \to (e \to t)$ in the tree, and again, the second set of actions would lead to a contradiction when *jumps* came to be parsed.

Additionally, despite the apparent type similarities between auxiliaries and verbs with verbal complements, **John not wants to go* would be debarred because there is no satisfied $Ty(A \to (e \to t)$ in the tree, and due to pointer movement (*wants* leaves the pointer at its object node, much as loved did in example 52, earlier), *John wants not to go* can only be interpreted as negating *to go*, and not the whole sentence. It is therefore not equivalent to *John doesn't want to go*. Further evidence that this is appropriate comes from the grammaticality of sentences such as *John doesn't want not to go*

This notwithstanding, an advantage of this approach is that while it debars such sentences, it is easy to describe an earlier stage of English where they would be acceptable, by a simple change in the initial criteria, so that instead of looking for a completed $Ty(A \rightarrow (e \rightarrow t))$ node, a completed $Ty(e \rightarrow t)$ would be sufficient. Thus, what is often considered to be brought about by a wholesale parametric change, can in fact be explained by the introduction of a simple constraint in certain lexical items (possibly driven by processing pressures, as discussed in section 3.3.2, above). In addition, it is necessary to assume that all verbs had a $Ty(t)$ trigger, thus explaining the additional inversion possibilities of Early Modern English (as seen in the Shakespearean examples in section 3.3). This situation has parallels in modern Spanish, where auxiliaries are not a syntactically distinct word class. All the possibilities shown in 107 below are possible in Spanish (though examples with multiple negation may sound strange and constructions using the subjunctive preferred), and this could easily be accommodated in the lexical entry for *no*.

(107) a. *No quieres hacerlo*
neg want$_{2ndSg}$ do$_{inf}$-it
'You don't want to do it.'

b. *No puedes hacerlo*
neg can$_{2ndSg}$ do$_{inf}$-it
'You can't do it.'

c. *Puedes no hacerlo*
Neg can$_{2ndSg}$ do$_{inf}$-it
'You can [not do it].'

d. *No puedes no hacerlo*
neg can$_{2ndSg}$ neg do$_{inf}$-it
'You can't not do it.'

e. *No puedes no querer hacerlo*
neg can$_{2ndSg}$ neg want$_{inf}$ do$_{inf}$-it
'You can't not want to do it.'

6.3.2.1 Potential Problems and Possible Solutions

Unfortunately, while this appears to satisfactorily account for most negative sentences, in cases with relative clauses, the situation is not quite so simple. To see why this is so, consider a simple relative clause akin to the one derived in section 4.6 (108a), and a similar sentence where the embedded sentence is negative (108b).

(108) a. John, who can jump, smokes.

b. John, who can not jump, smokes.

Intuitively, the actions applied by not in the relative clause example should be identical to that in the simple sentence *John can not jump*, and, as discussed above there ought to be (at least) two different ways of producing the parse (corresponding to those shown in figures 31 and 32). However, although the second option (corresponding with the interpretation *John can [not jump]* is successfully parsed, the other is not. This is because the top node of the LINKed tree has a $\langle L^{-1} \rangle$ relationship with the node from which it is projected meaning that if we were to build nodes *above* the top node in the LINKed structure, the new root node would not have such a relationship, rather it would be necessary to go $\langle \downarrow_0 \rangle \langle L^{-1} \rangle$ to get back to the node projecting the LINK, as shown in figure 34.[28]

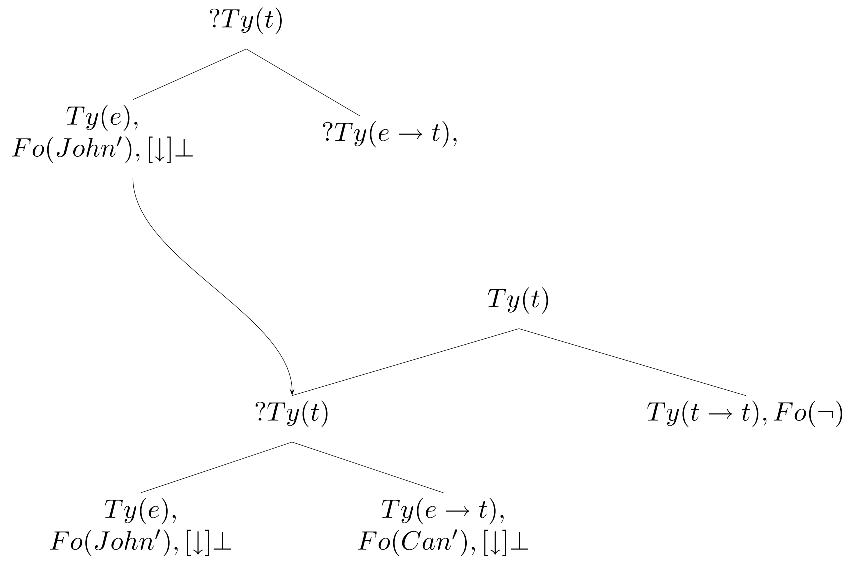

Figure 34: Parsing *John, who can not . . .*

This is clearly inappropriate; the LINKed structure needs to apply to the newly created (by the actions in the lexical entry for *not*) top node. There are two ways this problem can be resolved; the first would be to assume a template for *all* $Ty(t)$ structures, such that there is always an 'empty' space for $Fo(\neg)$. This would be similar to the proposal (in the Principles and Parameters program) of a NegP projection (see e.g. Mazzon (2004)), but it is an unattractive proposal in Dynamic Syntax because it assumes that there is some structure continually present which is not always realised either syntactically or semantically. The alternative solution, which is my preferred option, is, instead to assume that the LINK relation is itself underspecified. The formal properties of such a proposal require more consideration, beyond the scope of this project, but I see no principled reason why, after a LINKed structure were created, a similar operation to LATE *ADJUNCTION should not apply, thus creating a possible 'space' in the parse for the operations of *not*.

Another problem associated with this account is related to the ambiguity of modals discussed in section 6.2.2. Basically, if the *not* element requires a $?Ty(t)$ trigger, then the potential formula values of examples with the negative element *not* in them are more restricted than they should be. Consider the sentence in 109, below.

[28]The $Ty(e \rightarrow t)$ node shown as decorated with $Fo(Can')$ is for simplicity only. In reality it would, of course, be more complex, with a $Ty(\mathbf{A})$ argument daughter and a $Ty(\mathbf{A} \rightarrow (e \rightarrow t))$ functor daughter.

(109) I can not put the kettle on.

Intuitively, this sentence should have four possible interpretations, relating to the paraphrases shown in 110, and the formula values shown in 111.

(110) a. I don't have the ability to put the kettle on.
b. There is no possibility of me putting the kettle on.
c. I have the ability to not put the kettle on.
d. There is the possibility of me not putting the kettle on.

(111) a. $Fo(\neg((Can'((Put\_kettle\_on')Chris'))Chris))$
b. $Fo(\neg((Can'(Put\_kettle\_on'))Chris))$
c. $Fo((Can'(\neg((Put\_kettle\_on')Chris'))Chris))$
d. $Fo((Can'(\neg(Put\_kettle\_on'))Chris))$

However, using the lexical entry for *not* shown in 106, and the lexical entry for *can* shown in 88, we are only able to derive three of them, corresponding to 110 and 111a, b and c. To see why this is so, consider the underivable example in 110 and 111d. Given the current actions of ambiguous examples, the positive equivalent of such a reading would treat the underspecified $Ty(\mathbf{A})$ as being instantiated by $Ty(e \rightarrow t)$, but it is clear that *not* can not be parsed from such a node. Unfortunately for this account, this problem only escalates when we consider examples with multiple negation, and it appears to me that either the first characterisation of *not* must be correct (although see section 6.3.1 for arguments against this), or something else must be going on.

Essentially, the only way I can see of accurately capturing the ambiguous data, if *not* is to retain its type specification of $?Ty(t \rightarrow t)$ is to treat *can* and the other ambiguous auxiliaries as themselves type underspecified, between $Ty(e \rightarrow t)$ and $Ty(t \rightarrow t)$. Thus sentences combining the expletive *it* (as discussed in section 4.5.2) and these ambiguous modals should not share the ambiguity, because the subject forces the typing to be $Ty(t \rightarrow t)$. What this implies is that the ambiguous auxiliaries have two potential parsing strategies, roughly correlating to the *subject raising* and *subject control* readings. Although it is beyond the scope of this paper to go in to this in extensive detail, see section 6.6, for a proposal of how I see this working out whereby the $Ty(e \rightarrow t)$ and $Ty(t \rightarrow t)$ interpretations would parallel the examples of *wants* and *seems*, respectively.

6.3.3 Never revisited

As mentioned earlier, and seen in examples 97, and 98, *never* has a similar distribution to *not*. The differences, as previously discussed, reflect the fact that *never* combines the actions of negation and the introduction of a $Ty(e)$ element. For this reason, we can take the lexical actions shown in 106 as a starting point, and as a first approximation, the resulting lexical entry for *never* is shown below[29].

(112) Lexical entry for *never*.

$$
\textit{never} \left|
\begin{array}{lll}
\text{IF} & ?Ty(t) & \\
\text{THEN} & \text{IF} & \langle\uparrow\rangle\bot \\
 & \text{THEN} & \mathtt{make}(\langle\uparrow_0\rangle);\mathtt{go}(\langle\uparrow_0\rangle) \\
 & \text{END IF} & \\
 & \multicolumn{2}{l}{\mathtt{put}(Ty(t));\mathtt{make}(\langle\downarrow_1\rangle);\mathtt{go}(\langle\downarrow_1\rangle);} \\
 & \multicolumn{2}{l}{\mathtt{put}(Ty(t\rightarrow t),Fo(\neg),[\downarrow]\bot);} \\
 & \multicolumn{2}{l}{\mathtt{go}(\langle\uparrow_1\rangle);\mathtt{make}(\langle\downarrow_0\rangle);\mathtt{go}(\langle\downarrow_0\rangle)} \\
 & \multicolumn{2}{l}{\mathtt{put}(?Ty(t)),\mathtt{make}(\langle\downarrow_*\rangle);\mathtt{go}(\langle\downarrow_*\rangle);} \\
 & \multicolumn{2}{l}{\mathtt{put}(Ty(e)),Fo(Ever'),[\downarrow]\bot} \\
\text{ELSE} & \mathtt{Abort} &
\end{array}
\right.
$$

Although using this lexical entry (which basically, as discussed, carries out the actions of *not* and also introduces a $Ty(e)$ element on an unfixed node) can account for the sentences in example 97, it also raises certain interesting questions.

The main question concerns where in the final parse tree the newly introduced unfixed $Ty(e)$ node (with $Fo(Ever')$) resides. It is clear from the basics of Dynamic Syntax that it must have a fixed position before a parse can be considered successful, but it is by no means obvious where this would be. This question is tackled by Marten, and involves complex questions about the nature of optional arguments and adjuncts, which it is not my intention to address here. See Marten (2002) for a discussion of the $Ty(e^* \rightarrow t)$ type underspecification that he uses to confront this problem, and note merely that this is a question that requires resolution.

Additionally, sentences such as those in 97b and d have interesting implications for the contribution of the *to* infinitival marker. As *never* requires a $?Ty(t)$ trigger, this suggests that infinitival *to* may be a $Ty(t)$ introducing element (fulfilling the same actions as the last part of the auxiliary). This will be discussed further in section 6.6, however, for now it suffices to say that if *to* cannot, at least in principle, be a $Ty(t)$ element, then with the current characterisation of *never*, example 97b would be ungrammatical.

A further implication which needs considerably more investigation is whether subject and predicate INTRODUCTION and PREDICTION can only occur right at the start of a parse. Without adding such a stipulation, it is impossible to debar sentences such as *Never I want to be bothered*, but with it, both this type of sentence and the inversion caused by having

[29]Notice that I have introduced an 'END IF' statement into the lexical actions. This is merely for clarity, where if the condition is met the actions are carried out, but if they are not, the next criterion/action is applied. A strictly disjunctive lexical entry whereby the bulk of the actions were repeated would equally capture the required effects, but in my view is harder to read and disguises the fact that most of the actions are identical.

a sentence initial *never* (and thus necessitating the use of an auxiliary verb) are immediately and simply accounted for. **Never I want* ... would be ungrammatical because if, following the parse of *never*, we could *not* use INTRODUCTION and PREDICTION, hypothetically because either there was already information in the tree, or because we were not now at the root node, then the only way to introduce a new $Ty(e)$ node (after returning to the $?Ty(t)$ node using COMPLETION) would be using *ADJUNCTION to create an unfixed node. We have already seen, however, that if there were two type equivalent unfixed nodes, that these would collapse together, meaning that $Fo(Ever')$ and $Fo(Chris')$ would effectively be trying to occupy the same node, thus causing the tree to abort.

This same concept explains why inversion is not just preferred, but the only grammatical option after a sentence initial *never*. The parsing of *never*, as above, leaves us at a type fulfilled $?Ty(e)$ node (decorated with the formula $Fo(Ever')$. The only movement possible from here (to a node with an unfulfilled type requirement) takes us, using COMPLETION, back to the $?Ty(t)$ node. We cannot parse a subject from here directly, and nor can we parse a full lexical verb (because that requires us to have a $?Ty(e \rightarrow t)$ trigger), and if we are debarred from using INTRODUCTION and PREDICTION, as I have proposed, then the only available option is for the next lexical item to be an auxiliary verb, because, as seen earlier, these proceed from a $?Ty(t)$ trigger and create their own subject node, awaiting a $?Ty(e)$ input.

In addition, certain puzzling phenomena fall out naturally from this characterisation of *never* and *not*. For example, Kim (2000) discusses the differences in cases of VP-ellipsis illustrated by his examples, (pp.90) shown below in 113.

(113) a. Tom has written a novel, but Peter never has.
b. *Tom has written a novel, but Peter has never.
c. Kim told Tom to leave, but Jane told him not to.
d. *Kim told Tom to leave, but Jane told him to not.
e. Tom has written a novel, but Peter has not.

In the Dynamic Syntax account outlined above, it is immediately obvious why *not* can be the last word in cases of VP-ellipsis with auxiliaries, but *never* cannot, by comparing the trees in figures 35 and 36, corresponding to the second clause in 113b and d respectively. As we can see, in the *Peter has not* tree, the node with the pointer on it has a metavariable, which can be filled from context. In the *Peter has never* tree, contrarily, there is an unfulfilled node which requires a $?Ty(t)$ input. Thus, without having to postulate additional constraints or machinery as Kim does, we can easily account for the apparently random differences in the placement of the two negative words.

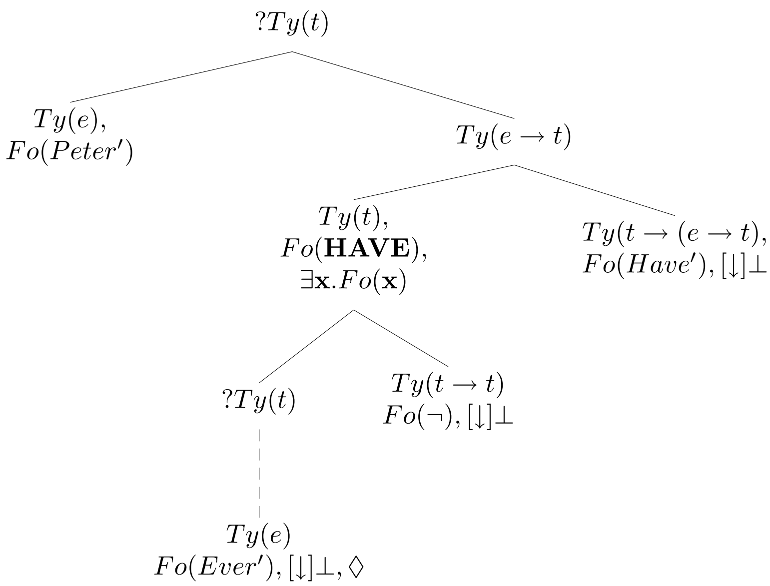

Figure 35: Parsing *Peter has never* . . .

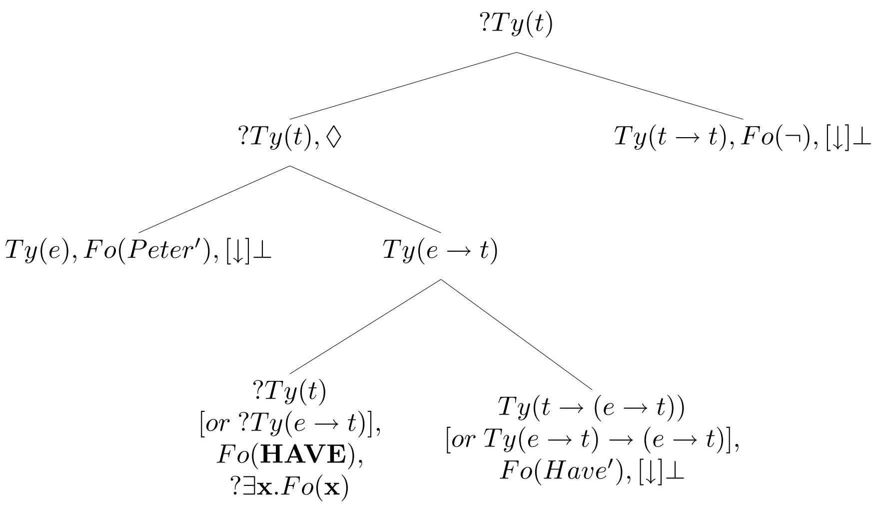

Figure 36: Parsing *Peter has not* . . .

6.3.4 Clitic or Inflectional *n't*

Although many commentators treat *n't* as a cliticised form of the negative particle *not* (enclitic to the first auxiliary in a string), I agree with the extensive arguments in Zwicky & Pullum (1983), that although this was probably true historically, *n't* has since been reanalysed as an inflectional affix. In Dynamic Syntax terms, what this means is that the *n't* forms of the auxiliaries have their own lexical entries, as one of the main differences between clitics and inflectional affixes is that while clitics are syntactic in nature, inflectional affixes are lexical. Other key points made by Zwicky & Pullum are that arbitrary gaps and morphosyntactic and semantic idiosyncracies, of the sort exhibited by the auxiliary + *n't* words, are typical of inflectional affixes, not clitics. The idea that these are separate lexical items also matches the notion that the lexicon is the source of arbitrariness in language. For a full discussion of *n't* forms as inflectional affixes, including numerous persuasive examples, see Zwicky & Pullum (1983), however, for now it suffices to consider their examples 4-7 (pp.506, and repeated here as 114-117).

(114) a. You haven't been here.
b. Haven't you been there?

(115) You have not been there.

(116) *Have not you been there?

(117) a. Have you been there?
b. Have you not been there?

If *n't* was indeed a clitic, then we should expect that (114a) is derived from (115), and (114b) from (116), which is ungrammatical, unlike (117b). Under my analysis, (116) is debarred, as it should be[30], because having processed *have*, the pointer will be at an unfulfilled $?Ty(e)$ node. This is not what we would expect if *n't* was merely a clitic, so, as already mentioned, I propose to treat contracted negative forms of auxiliaries as separate lexical entries. Like the positive modals, the negative forms of auxiliaries can only appear as the first verbal element in a string with multiple auxiliaries, as can be seen from Zwicky & Pullum's examples 13 and 14 (repeated here as 118 and 119).

(118) a. The police have not been informed.
b. The police haven't been informed.

(119) a. Would the police have not been informed?
b. *Would the police haven't been informed?

This is simple to stipulate, as it is only the finite forms which can occur as the first auxiliary and therefore have a contracted form. Some confusion arises because the base form

[30]Though see earlier comments regarding whether examples of this type are strictly ungrammatical and not merely archaic.

of *have* is identical to its present tense (non-3rd singular) form, but for the sake of clarity I will assume that these morphologically equivalent forms have different lexical entries, with relevant constraints. The lexical entry for each negative form of an auxiliary, will therefore be identical to the equivalent positive form, but have the additional instructions inherited from *not* (indented in the lexical entry, for clarity), as shown below in 120. What this means in practise is that the sentences in 114b and 117b will result in identical trees. As with the positive example of *can*, shown in 88, in the lexical entry for *can't* shown below, $\mathbf{A}$ can be instantiated as either t or $(e \rightarrow t)$, and it is assumed that both possibilities will be explored in parallel[31].

(120) Lexical entry for *can't*.

$$
\textit{can't}\;\left|\;\begin{array}{lll}
\text{IF} & ?Ty(t) & \\
\text{THEN} & \text{IF} & \langle\downarrow_1\rangle Ty(e \rightarrow t) \\
 & \text{THEN} & \mathtt{Abort} \\
 & \text{ELSE} & \mathtt{put}(Tns(PRES)); \\
 & & \mathtt{make}(\langle\downarrow_1\rangle); \mathtt{go}(\langle\downarrow_1\rangle); \mathtt{put}(Ty(e \rightarrow t)); \\
 & & \mathtt{make}(\langle\downarrow_1\rangle); \mathtt{go}(\langle\downarrow_1\rangle); \\
 & & \mathtt{put}(Ty(\mathbf{A} \rightarrow (e \rightarrow t)), Fo(Can'), [\downarrow]\bot) \\
 & & \mathtt{go}(\langle\uparrow_1\rangle); \mathtt{make}(\langle\downarrow_0\rangle); \mathtt{go}(\langle\downarrow_0\rangle); \\
 & & \mathtt{put}(?Ty(\mathbf{A}), Fo(\mathbf{AUX}), ?\exists\mathbf{x}.Fo(\mathbf{x})); \\
 & & \mathtt{go}(\langle\uparrow_0\rangle); \mathtt{go}(\langle\uparrow_1\rangle); \mathtt{make}(\langle\downarrow_0\rangle); \\
 & & \quad \mathtt{make}(\langle\uparrow_0\rangle); \mathtt{go}(\langle\uparrow_0\rangle) \\
 & & \quad \mathtt{put}(?Ty(t)); \mathtt{make}(\langle\downarrow_1\rangle); \mathtt{go}(\langle\downarrow_1\rangle); \\
 & & \quad \mathtt{put}(Ty(t \rightarrow t), Fo(\neg), [\downarrow]\bot); \\
 & & \quad \mathtt{go}(\langle\uparrow_1\rangle); \mathtt{go}(\langle\downarrow_0\rangle) \\
 & & \begin{array}{ll}
 \text{IF} & \langle\downarrow_*\rangle Ty(e) \\
 \text{THEN} & \mathtt{go}(\langle\downarrow_0\rangle); \mathtt{put}(?Ty(e)); \\
 \text{ELSE} & \mathtt{put}(Cat(INV)); \\
 & \mathtt{go}(\langle\downarrow_0\rangle); \mathtt{put}(?Ty(e));
 \end{array} \\
\text{ELSE} & \mathtt{Abort} &
\end{array}\right.
$$

A question arises, if *n't* is taken to be an inflection, as to whether strings such as those shown in 121, below, are grammatical. Under the current analysis, they would be, however, as before I would be inclined to argue that they are not, in fact, ungrammatical per se, but perhaps sound odd because they are rarely heard. Additionally, under the current analysis, VP-ellipsis is possible in the case of 122, because when *not* is associated with the lower tree position (in a completely analogous way to the example shown in figure 33), as it must be here, it introduces a metavariable in exactly the same way that the auxiliary verbs do. This seems intuitively correct to me, as I believe that, with appropriate stress, 122 could be a perfectly legitimate response to a question like *Did you go and visit your mother on her birthday?* It is possible that this is an area of current language change;

[31] This means that, in effect, the lexical entry in 120, could be viewed as two separate entries which carry out identical actions with the relevant type value being substituted wherever an $\mathbf{A}$ appears.

a possibility that is, perhaps, supported by the acceptability of (sarcastic) examples like 123, to younger speakers of the language.

(121) I haven't not NOT been thinking about you.

(122) I couldn't not. *(ellipsis)*

(123) John can jump. NOT!

6.4 Inversion Revisited - WH-Questions

A further question with regards to auxiliary use in English, involves the well documented question of sentences involving Wh-words[32,33]. The data indicates that if a question is asked whereby the question begins with the Wh-word, but the questioned element is *not* the subject, then the usual Subject-Verb-Object (SVO) English word order is not respected. Instead, an auxiliary verb and inversion must be used, as shown in example 124.

(124) a. Who loves Mary? *Subject questioned; no auxiliary or inversion*
b. Mary loves who? *Who in situ; no auxiliary or inversion*
c. Who does Mary love? *Direct object questioned; auxiliary and inversion*
d. *Who Mary loves? *Intended: direct object questioned; no auxiliary or inversion*

In addition, the string in between the Wh-word and the questioned article, can be far more complex, as shown in example 125 (the usual position of the questioned material is shown as —). Note that in all these cases, it is also possible to have *who* in situ, as shown.

(125) a. Who does Mary think John loves —?
b. Mary thinks John loves who?
c. Who does Mary think — loves John?
d. Mary thinks who loves John?
e. Who does Mary think John spent the night with —?
f. Mary thinks John spent the night with who?

It is easy to see why a transformational account postulates movement from the position of the noun phrase that the Wh-word stands in for (the element being questioned), to the front of the string, because it seems fairly obvious that in these examples, *who* is in some sense 'misplaced'. What this means for a Dynamic Syntax account, of course, is that

[32] A group that contains *who*, *what*, *when*, *where*, *which* and *how*, for example, though I shall be focussing on *who* for the purposes of this section.

[33] See Kempson et al. (2001), chapter 5, for an extensive discussion of wh-questions, incorporating cross linguistic data.

when encountered in such cases, *who* should occupy an unfixed node, with its position to be specified before the parse can be completed. In addition, although it seems obvious that *who* should provide a metavariable that is to be updated from context, if we use the same methodology as, for example, pronouns, then this will allow us to derive the ungrammatical string shown in example 126, since *John* will be able to occupy the same node as *who*.

(126) *Who John loves Mary?

Although it might be argued that this sentence is, in fact, grammatical (with appropriate, if strange, intonation), I will assume for the moment that it is another feature of the Wh-words that they must occupy their own node in the tree under construction. This means, as noted in Kempson et al. (2001), that the metavariable is not accompanied with the substitution requirement, $\exists \mathbf{x}.Fo(\mathbf{x})$, and may therefore remain as variables in the final structure; unlike pronouns, for example. However, even allowing for the inclusion of these stipulations, the situation is not that simple.

Consider the sentences in example 127, below, where *who* is part of an embedded clause in the sentence (of which relative clauses are just one possibility).

(127) a. Mary, who loves John, smokes.
b. Mary, who John loves, smokes.
c. Mary, who does love John, smokes.
d. I wonder who loves John.
e. I wonder who John loves.
f. I wonder who does John love.
g. *I think who loves John.
h. I can't think who loves John.

What we can see from this is that the ungrammatical example 124d, is perfectly acceptable if *who* is not sentence initial. In embedded and relative clauses, the Wh-element can be left-dislocated, but if it is sentence initial then it cannot. This is a strange quirk of English, especially when we consider that *John, Mary loves.* is a grammatical string, so we might expect, as *who* is arguably just representing a generic (person) noun phrase, that it could appear in all contexts in which a fully specified one can.

One way of dealing with this problem in Dynamic Syntax is by effectively limiting the type of clause that is being processed when a Wh-word is encountered first in the string. What this would then effectively do is put a requirement for a decoration onto the top node (e.g. $Cat(Q)$, or $Cat(INV)$ as discussed in section 6.1, above), which would then be matched by the auxiliary verb if it had not already parsed a subject. The problem is, this would then cause problems with the apparently simpler example of *Who likes John?*, whereby the lexical verb *likes* cannot match the question requirement. A solution, which assumes that the requirement is only present if *who* is present on an unfixed node (i.e. has not been parsed as a subject following INTRODUCTION and PREDICTION) is certainly feasible, however, it leads to a disjunctive lexical entry which does not necessarily account for the reasons why embedded clauses with *who* in them can behave differently. If an element of clause typing is necessary (and it may be so to account for why even items such as *never* prevent INTRODUCTION and PREDICTION occurring after them if they are sentence initial, as discussed in section 6.3.3) then I would propose that it is not clause typing for a question per se, but for an inverted environment that is normally interpreted as a question in English (as discussed in section 6.1). One of the main reasons for this is that if the clause typing were for a question, then we should expect that the tag $Cat(Q)$ is applied to the root node even in cases where the Wh-word appears in situ, as in *John loves who?*. However, while it is easy to stipulate applying the clause-typing effect when *who* is sentence initial, it is less obvious how to do so when it is not. In other words, either we accept that even in cases of relative and embedded clauses, the tree (or subtree) containing the wh-word is in some sense a question, or we restrict the clause typing to cases of sentence initial *who*, and make it the less restrictive $Cat(INV)$. It could be argued that one way to debar 127g would be to have the lexical entry for *think* put a restriction ($Cat(-Q)$) on the complement node, as many syntactic analyses do, however, the availability of 127h should indicate that this is inappropriate. I have no further comments to make on the differences between these two examples, where the latter is grammatical, but the former is not, (or whether, for example, the difference should be considered syntactic or semantic in nature), but merely note that it is a question requiring resolution.

(128) Lexical entry for *who* - First approximation.

who	IF	$?Ty(e)$
	THEN	$\texttt{put}(Ty(e), Fo(WH_{Person'})), [\downarrow]\bot)$
	ELSE	$\texttt{Abort}$

This said, the simplest possibility for a lexical entry for who, and as such a useful starting point, is to treat it as exactly analogous to one of the noun phrases which it is a ‘wildcard’ character for, as shown in 128. If this were the case, of course, *Who, Mary loves?* would be as acceptable as *John, Mary loves*, which, as previously discussed is not the case[34]. In order to prevent it, whilst retaining the possibility of such a word order occurring in embedded and relative clauses, either we need to say something altogether more general about the restrictions of the whole language, or we need to impose restrictions on the lexical entry. While the former might ultimately be more appropriate (with, for example, all verbs requiring a $?Ty(t)$ trigger and INTRODUCTION and PREDICTION being done away with altogether), this raises more questions than it resolves, so for the time being I shall assume that a weak form of clause typing does occur with certain lexical items, e.g.

[34]Note also the acceptability of *Who is it Mary loves?* whilst **John is it Mary loves* is ungrammatical. Similarly, the ordering changes for the simple confirmatory question compared to one which requires a name answer; compare *Who is it?* and *Is it John?*, (and not **Is it who?* or *??John is it?*) both of which ought to map to the same positive structure *It is John*. This data is intriguing, and requires additional research, taking into account the complex interactions of the copula and *it*.

never, and *who*.

With *never* it is easy to identify whether it is the first element processed, because it has a $Ty(t)$ trigger and we can therefore simply query whether anything else exists in the tree at all. If we retain the $Ty(e)$ trigger for *who*, however, this is less easy to stipulate. There are two options available for where *who* can reside in the tree under construction if it the first element encountered. The first is on an unfixed node (after *ADJUNCTION has applied), and the other is on the subject node of the tree (after INTRODUCTION and PREDICTION). However, as *who* behaves normally if it is on a fixed subject node (as in, for example *Who loves Mary?*), it is only the first of these cases that we need to identify.

(129) Lexical entry for *who* - Second approximation.

	IF	$?Ty(e)$				
	THEN	$\mathtt{put}(Ty(e), Fo(WH_{Person'})), [\downarrow]\bot)$				
		IF	$?\exists\mathbf{x}.Tn(\mathbf{x})$			
		THEN	IF	$\langle\uparrow_*\rangle Tn(0)$		
who			THEN	$\mathtt{go}(\langle\uparrow_*\rangle Tn(0)); \mathtt{put}(?Cat(INV))$		
			ELSE	IF	$\langle\uparrow_*\rangle\langle L^{-1}\rangle Fo(\alpha)$	
				THEN	$\mathtt{put}(Fo(\alpha))$	
			END IF			
		END IF				
	ELSE	Abort				

This lexical entry has several disjunctive elements, but the key thing to notice is that the first set of actions will apply regardless of which, if any, of the other actions will. In cases where *who* is the first element and on an unfixed node, the requirement for a $Cat(INV)$ will also be applied, which only an auxiliary verb (as already defined) will be able to fulfill. This therefore has the desired effect of preventing the use of INTRODUCTION and PREDICTION (because if there is a subject already in place when the auxiliary, or any other verb, is encountered, the decoration $Cat(INV)$ is not applied). The third set of actions is that which is needed to copy the formula value of the host node in the relative clause examples (see Cann et al. (2005), Chapter 3 for further details).

While it might seem that this lexical entry can adequately capture the facts regarding Wh-word placement in full and embedded clauses, unfortunately it is not that simple. A further problem arises when we consider the fact that I have characterised auxiliaries as projecting a LINKed structure. This is because, as it currently stands, one of the features of unfixed ($\langle\downarrow_*\rangle$) nodes is that they can only be fixed locally; i.e. in the tree under construction, and not a tree LINKed to the tree under construction. While it is not hard to imagine or formulate a rule which creates an unfixed node with a less restrictive $\langle D\rangle$ relation on which $Fo(\mathbf{WH})$ can reside so that it may MERGE with any node in the tree under construction including any LINKed trees, this has unexpected and undesirable consequences, especially when we consider relative clauses, and other typical LINKed constructions (for example coordination). This problem does not just affect Wh-words, because all sentences which begin with an unfixed node and then have an auxiliary verb will also need to be on a less restricted unfixed node (see example 130a, and its required derivation in figure 37).

(130) a. John, Mary can love.
b. ?John, Mary, who likes Julia, can't love.
c. *Julia, Mary, who likes —, can't love John
d. Who can Mary love —?
e. Who can Mary, who likes Julia, love —?
f. *Who can Mary, who likes —, love John?
g. Mary likes John and Julia.
h. *Who can Mary like John and —?

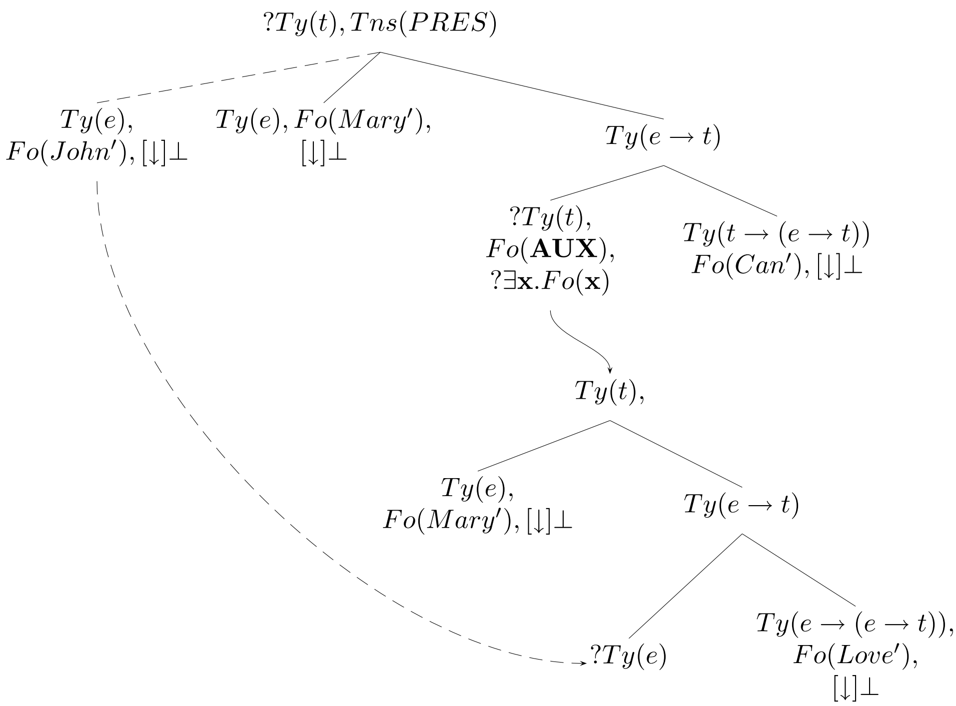

Figure 37: Tree state after parsing *John, Mary can love* before MERGE

The problem, which is a form of the well-documented puzzle of *Complex NP Constraints*, as identified in Ross's 1967 PhD Thesis (published as Ross (1986)), can be described as requiring A-LINK connected structures to allow unfixed nodes to be passed down through them, whilst at the same time preventing other LINKed structures from doing so.

In essence then, either I am completely mistaken in my formulation of the English auxiliary system and there are no LINKed structures involved, or there is something tangibly different about the LINKed structures produced by the A-LINK rule. Taking the first possibility, the embedded, currently LINKed tree could, in principle, be created in the existing tree. This however introduces unwanted complexity, as the onus for copying the subject details into the embedded structure would have to be taken up in the lexical actions for the auxiliaries.

Alternatively, and how I envisage this problem being most parsimoniously resolved, the actions induced by the A-LINK ADJUNCTION rule require revision. Roughly, what the rule needs to do, in addition to the actions it currently carries out, is to check to see what, if any, unfixed nodes are attached to the nearest $Ty(t)$ node above the node it is being implemented from, and carry them through to the new LINKed tree (where they will remain unfixed for the time being, either to be fixed in the new tree or carried through again in another application of A-LINK ADJUNCTION). The precise formal details of such a procedure and considerations about its possible implications require significant working out, beyond the scope of this paper. It may in fact turn out to be the case that all unfixed nodes should be on a node with a $\langle D \rangle$ relation, with the difference being in the normal rule of LINK ADJUNCTION, which somehow blocks the MERGE operation.

6.5 Exceptional Items

As Dynamic Syntax is a lexicalised grammar, it is clear that the correct level for exceptional items to be differentiated at is the level of the lexicon.

6.5.1 Have

Starting with *have*, I assume that, like *not*, the best way to accommodate the data is in the notion of type underspecification. However, whilst with *not* this manifested itself in the trigger condition of the lexical item, with *have*, we can see how this enables us to expect different types of potential complements. Taking the lexical entry for *can* (88) as a starting point, I believe the lexical entry in 131 allows us to capture the facts about *have*, where $\mathbf{A}$ can be either $(e \rightarrow t)$ or e[35].

The main difference between this lexical entry and that for *can* (as seen in 88) is in the formula values $Fo(Have')$ and $Fo(\mathbf{HAVE})$, which are self explanatory[36]. If, when the lexical item is parsed, the option $Ty(e \rightarrow t)$ is chosen, then A-LINK ADJUNCTION can be applied, whilst if it creates a $Ty(e)$ node then a noun phrase may be parsed. Notice that as the perfect forms of have do not share the subject raising versus subject control ambiguity of modals like *can*, that a $?Ty(t)$ complement is not appropriate. As with the application of its rules, Dynamic Syntax does not explicate the reasons for choosing one option over another, and again, it is assumed that both options are explored in parallel, with one eventually ‘winning’ when the next word is encountered. The advantage of this

[35]The assumption is still that both possible trees are constructed until one is ruled out by following lexical input, or substitution from context.

[36]$Fo(\mathbf{HAVE})$ stands for ‘something that can be had, state or object’ such that it allows appropriate noun phrases or -en forms. This is left aside for future precise definition.

type underspecification is that it neatly accounts for the otherwise puzzling fact (discussed in 2.2.3, and illustrated in examples 35 and 36) that possessive *have* acts like an auxiliary syntactically, but not semantically. It also explains the increasing tendency for *have got* + *NP* constructions to replace *have* + *NP*. In frameworks like HPSG, which postulate multiple lexical entries for words like *have*, the differences between the entries are emphasised and the similarities downplayed, or treated as a historical quirk, relating only to the words single lexical ancestor. In this account, contrarily, with the simple choice of type value, we can see how the entry is either exactly analogous to the other auxiliaries, or analogous to lexical verbs which take a noun phrase complement, but, crucially, still allowing the syntactic NICE properties (discussed in section 2.1).

(131) Lexical entry for *have*.

$$
\textit{have}\;\left|\;\begin{array}{llll}
\text{IF} & ?Ty(t) & & \\
\text{THEN} & \text{IF} & \langle\downarrow_1\rangle Ty(e\rightarrow t) & \\
 & \text{THEN} & \texttt{Abort} & \\
 & \text{ELSE} & \texttt{put}(Tns(PRES)); & \\
 & & \texttt{make}(\langle\downarrow_1\rangle);\texttt{go}(\langle\downarrow_1\rangle);\texttt{put}(Ty(e\rightarrow t)); & \\
 & & \texttt{make}(\langle\downarrow_1\rangle);\texttt{go}(\langle\downarrow_1\rangle); & \\
 & & \texttt{put}(Ty(\mathbf{A}\rightarrow(e\rightarrow t)), Fo(Have'), [\downarrow]\bot) & \\
 & & \texttt{go}(\langle\uparrow_1\rangle);\texttt{make}(\langle\downarrow_0\rangle);\texttt{go}(\langle\downarrow_0\rangle); & \\
 & & \texttt{put}(?Ty(\mathbf{A}), Fo(\mathbf{HAVE}), ?\exists\mathbf{x}.Fo(\mathbf{x})); & \\
 & & \texttt{go}(\langle\uparrow_0\rangle);\texttt{go}(\langle\uparrow_1\rangle);\texttt{make}(\langle\downarrow_0\rangle); & \\
 & & \text{IF} \quad \langle\downarrow_*\rangle Ty(e) & \\
 & & \text{THEN} \quad \texttt{go}(\langle\downarrow_0\rangle);\texttt{put}(?Ty(e)); & \\
 & & \text{ELSE} \quad \texttt{put}(Cat(INV)); & \\
 & & \qquad\quad\; \texttt{go}(\langle\downarrow_0\rangle);\texttt{put}(?Ty(e)); & \\
\text{ELSE} & \texttt{Abort} & &
\end{array}\right.
$$

Further evidence for this type of treatment (though obviously, precise details need fully fleshing out) comes from the diachronic data discussed in section 3.3.2.1. This suggests that *will* was once similarly type underspecified, with the frequency of one type value eventually being so much higher than the other that the lesser used option became obsolete. This preference is currently occurring with *have* and for some native speakers, it seems that the $Ty(e)$ option is no longer available. This analysis also has implications for all verbs which can take more than one type of complement (see section 6.6, below).

6.5.2 Be

We could interpret *be* as acting in a way completely analogously to *have*, i.e. as projecting a type underspecified complement node (with the possible types being of a wider variety than those for *have*, as seen in example 132, taken from Cann (2007)). However, Cann et al. (2005) argue that although this could capture the syntactic facts, it would not adequately capture the semantic facts as well.

(132) a. Mary is happy / in the gym / a student.
b. John is the teacher.
c. There's a riot on Princes Street.
d. It's me.
e. It is Mary who is the dancer.
f. What I want is a good review.
g. Neuroses just ARE (they don't need a cause).
h. Kim was running to the shops.
i. The fool was hit by a truck.

Because of the availability of examples with a bare copula (as in example 132g, above), they propose that *be* in fact does not have any complements at all. The apparent contradiction that seems to then occur (that there usually does seem to be a complement) is resolved by its projection of a metavariable, which, as before, may be filled from context. Like the example with expletive *it* that we saw in section 4.5.2, earlier, the fact that this can be arbitrarily complex merely implies that if the following lexical material is parsed using LATE *ADJUNCTION (which is one, but not the only, possibility, as the formula value could also be updated from context using the pragmatically motivated rule of SUBSTITUTION), then the lexical entry for *be* should not introduce the bottom restriction. The lexical entry for *is*, shown below (133), is taken from Cann et al. (2005).

(133) Lexical entry for *is*.

$$
\textit{is}\;\left|\;\begin{array}{lll}
\text{IF} & ?Ty(t) & \\
\text{THEN} & \text{IF} & \langle\downarrow_1\rangle Ty(e \rightarrow t) \\
 & \text{THEN} & \texttt{Abort} \\
 & \text{ELSE} & \texttt{put}(Tns(PRES)); \\
 & & \texttt{make}(\langle\downarrow_1\rangle); \texttt{go}(\langle\downarrow_1\rangle) \\
 & & \texttt{put}(Ty(e \rightarrow t), Fo(\mathbf{BE}), \exists\mathbf{x}.Fo(\mathbf{x})) \\
 & & \texttt{go}(\langle\uparrow_1\rangle); \texttt{make}(\langle\downarrow_0\rangle); \texttt{go}(\langle\downarrow_0\rangle); \\
 & & \texttt{put}(?Ty(e)) \\
\text{ELSE} & \texttt{Abort} &
\end{array}\right.
$$

In effect, what this means is that *be*, in English at least, is in some sense the 'default' verb (stepping in, for example when we want to attribute a noun phrase or adverb to a person, as in example 132a). This shifts the burden of explanation of *be* from an assumption that it has a core meaning (usually thought of as existence or identity) to "an account of inference in context that derives the correct interpretations of sentences." (pp.343, Cann et al. (2005)). This means that *be*, unlike the other auxiliaries thus far discussed, does not add semantic meaning itself, merely facilitating the interpretation and allowing the parse to continue.

This has interesting implications for languages which permit a *null copula*, (i.e. do not require the presence of a copula verb like *be* in certain contexts), like African American Vernacular English (AAVE). In AAVE, unlike standard British English sentences such as those shown in 134 (taken from Sag et al. (2003), pp.457) are perfectly acceptable.

(134) a. Chris at home.
b. We angry with you.
c. You a genius!

In Dynamic Syntax there are two possible ways to tackle this problem. One way is similar to the HPSG approach, which postulates a phonetically unrealised lexical entry, which supplies the necessary features in HPSG, but would supply the necessary metavariable in Dynamic Syntax. The alternative in Dynamic Syntax is to assume that in languages like AAVE, the metavariable introduced by the copula in standard British English was introduced at the same time as the requirement for a $?Ty(e \rightarrow t)$, e.g. in the rule of Prediction. Both options have problems. Phonetically unrealised lexical items go against the basic tenets of a lexicalised grammar (as discussed in Sag et al. (2003), but the alternative would lead to overgeneration and the acceptability of a wide range of ungrammatical strings. The further investigation of these possibilities is left to another day.

The account outlined in Cann et al. (2005), Chapter 8, and explored in more depth in Cann (2006, 2007) meshes neatly with that for auxiliaries given here, so it is not my intention to discuss the copula *be* in further detail, or to comment on the pros and cons of limiting what the metavariable *be* can be replaced with in terms of K-states. See Cann et al. (2005); Cann (2006, 2007) for more details.

6.5.3 Do

While *be* can be loosely described as the default verb for *states*, *do*, at least in its lexical form, is often considered to be the default verb for *actions*. It is clearly the case that in its non-auxiliary use, it in some sense stands in for another verb, (whose precise identity may depend on context) as can be seen in example 135, below.

(135) a. I'll do dinner. = I'll *cook* dinner.
b. Can you do the dishes? = Can you *wash* the dishes?

However, despite this, it is only in its auxiliary use that it is usually considered to be semantically empty. In my view, if it is semantically empty (akin to *be*, discussed in section 6.5.2, above) then it must require a metavariable to be filled form context in both its full and auxiliary use. This is a contentious question, because although *do* is often considered to be semantically empty in cases of so-called *do-support* (as discussed in section 2.2.4), it is not usually considered to be so in cases where the full lexical verb is used (in which case the basic meaning is assumed to be something along the lines of 'generic action'). Contrarily, some commentators have pointed out that, for example, in positive declarative sentences, *do* cannot be used unless it is in an emphatic way. If *do* were truly semantically empty, they argue, it should be possible to have an unstressed auxiliary *do* in a sentence. For the account outlined here, what this boils down to is whether we treat *do* as in some sense the same as *have*, i.e. with a semantic input, or as in some sense the same as *be*, i.e. as supplying a placeholder for some other semantic information to be supplied from context. For the purposes of this paper, I will assume the former, though note that this is an area for further research, and the only thing that depends on this decision is the lexical entry for *do*.

A major question arises, regardless of which of the above options we pursue. Why, if there is only one lexical entry for *do*, and unlike the case for both *be* and *have*, can it only invert or negate in its auxiliary use, as can be seen from the grammaticality judgements of the examples in 136, below.

(136) a. I don't do the dishes.
b. *I don't the dishes.
c. Doesn't John do the dinner?
d. *Doesn't John the dinner?

As this must be reflected in the lexical entry, what it seems to mean in practise is that finite forms of *do* must have a completely disjunctive lexical entry, as shown in 137, below.

(137) Lexical entry for *does*.

$$
\textit{does}\quad\left|\begin{array}{llll}
\text{IF} & ?Ty(t) & & \\
\text{THEN} & \text{IF} & \langle\downarrow_1\rangle Ty(e\rightarrow t) & \\
 & \text{THEN} & \texttt{Abort} & \\
 & \text{ELSE} & \multicolumn{2}{l}{\texttt{put}(Tns(PRES));\texttt{make}(\langle\downarrow_1\rangle);\texttt{go}(\langle\downarrow_1\rangle);} \\
 & & \multicolumn{2}{l}{\texttt{put}(Ty(e\rightarrow t));\texttt{make}(\langle\downarrow_1\rangle);\texttt{go}(\langle\downarrow_1\rangle);} \\
 & & \multicolumn{2}{l}{\texttt{put}(Ty((e\rightarrow t)\rightarrow(e\rightarrow t)),Fo(Do'),[\downarrow]\bot);} \\
 & & \multicolumn{2}{l}{\texttt{go}(\langle\uparrow_1\rangle);\texttt{make}(\langle\downarrow_0\rangle);} \\
 & & \multicolumn{2}{l}{\texttt{put}(Ty(e\rightarrow t),Fo(\mathbf{DO}),?\exists\mathbf{x}.Fo(\mathbf{x}));} \\
 & & \multicolumn{2}{l}{\texttt{go}(\langle\uparrow_0\rangle\langle\uparrow_1\rangle);\texttt{make}(\langle\downarrow_0\rangle);} \\
 & & \text{IF} & \langle\downarrow_*\rangle Ty(e) \\
 & & \text{THEN} & \texttt{go}(\langle\downarrow_0\rangle);\texttt{put}(?Ty(e)); \\
 & & \text{ELSE} & \texttt{put}(Cat(INV)); \\
 & & & \texttt{go}(\langle\downarrow_0\rangle);\texttt{put}(?Ty(e)) \\
\text{ELSE} & \text{IF} & ?Ty(e\rightarrow t) & \\
 & & \multicolumn{2}{l}{\texttt{put}(Ty(e\rightarrow t));\texttt{make}(\langle\downarrow_1\rangle);\texttt{go}(\langle\downarrow_1\rangle);} \\
 & & \multicolumn{2}{l}{\texttt{put}(Ty(e\rightarrow(e\rightarrow t)),Fo(\mathbf{DO}),?\exists\mathbf{x}.Fo(\mathbf{x}));} \\
 & & \multicolumn{2}{l}{\texttt{go}(\langle\uparrow_1\rangle);\texttt{make}(\langle\downarrow_0\rangle);\texttt{put}(?Ty(e));} \\
 & \text{ELSE} & \texttt{Abort} &
\end{array}\right.
$$

Although like *have*, do can take a $Ty(e)$ or a $Ty(e \rightarrow t)$ complement (but not a $Ty(t)$, as the ambiguity inherent in modals does not apply (*John can jump = It is the case that John jumps*)), if it is to take a $Ty(e)$ complement, this means it is standing in for a full verb, and thus inherits the restrictions on inversion and negation (hence the $?Ty(e \rightarrow t)$ trigger for the second half of the lexical entry). Contrarily, in its auxiliary use, it in some sense stands in for an auxiliary, and thus inherits the properties associated with them. An advantage of this approach is that it is easy to stipulate that the non-finite forms of *do*lexnot2 only have the second set of actions associated with them. This contrasts with both *have* and *be*, for which there is no major difference in the lexical entries of the finite and non-finite forms.

In practise, what the lexical entry in 137 means is that in a sentence that begins with, for example *John does* . . . , there are two possible parse trees up to this point. When the next word is encountered, it will only be able to progress one of these potential parse trees, depending on its type.

6.5.4 Dare and Need

The way I see it, although *do* is considered to be a completely idiosyncratic lexical item by many commentators, there are many similarities between the disjunctive lexical entry described above and one which would account for the distribution data of *dare* or *need*. An example lexical entry for *dares* is shown in 138, below.

(138) Lexical entry for *dares*.

$$
\textit{dares}\;\left|\;\begin{array}{llll}
\text{IF} & ?Ty(t) & & \\
\text{THEN} & \text{IF} & \langle\downarrow_1\rangle Ty(e \rightarrow t) & \\
 & \text{THEN} & \texttt{Abort} & \\
 & \text{ELSE} & \texttt{put}(Tns(PRES)); \texttt{make}(\langle\downarrow_1\rangle); \texttt{go}(\langle\downarrow_1\rangle); & \\
 & & \texttt{put}(Ty(e \rightarrow t)); \texttt{make}(\langle\downarrow_1\rangle); \texttt{go}(\langle\downarrow_1\rangle); & \\
 & & \texttt{put}(Ty(t \rightarrow (e \rightarrow t)), Fo(Dare'), [\downarrow]\bot); & \\
 & & \texttt{go}(\langle\uparrow_1\rangle); \texttt{make}(\langle\downarrow_0\rangle); & \\
 & & \texttt{put}(Ty(t), Fo(\mathbf{DARE}), ?\exists\mathbf{x}.Fo(\mathbf{x})); & \\
 & & \texttt{go}(\langle\uparrow_0\rangle\langle\uparrow_1\rangle); \texttt{make}(\langle\downarrow_0\rangle); & \\
 & & \text{IF} & \langle\downarrow_*\rangle Ty(e) \\
 & & \text{THEN} & \texttt{go}(\langle\downarrow_0\rangle); \texttt{put}(?Ty(e)); \\
 & & \text{ELSE} & \texttt{put}(Cat(INV)); \\
 & & & \texttt{go}(\langle\downarrow_0\rangle); \texttt{put}(?Ty(e)) \\
\text{ELSE} & \text{IF} & ?Ty(e \rightarrow t) & \\
 & & \texttt{put}(Ty(e \rightarrow t)); \texttt{make}(\langle\downarrow_1\rangle); \texttt{go}(\langle\downarrow_1\rangle); & \\
 & & \texttt{put}(Ty(t \rightarrow (e \rightarrow t)), Fo(Dare'), [\downarrow]\bot); & \\
 & & \texttt{go}(\langle\uparrow_1\rangle); \texttt{make}(\langle\downarrow_0\rangle); & \\
 & & \texttt{put}(?Ty(t)); & \\
 & \text{ELSE} & \texttt{Abort} &
\end{array}\right.
$$

While this lexical entry will license both *She dares do it* and *She dares to do it*, it is possible that an increasing preference for the latter can be explained by appealing to the fact that, in strings where a subject is parsed first, the usual continuation of a parse would involve progressing through the tree using COMPLETION and ANTICIPATION, until the pointer is at the $?Ty(e \rightarrow t)$ node. As this usual state of affairs allows us to continue to parse *She dares* . . . , this is the 'chosen' parse in a majority of instances. Contrarily, if inversion or the contracted negative form (*daren't*) are used, then this parse is not available and the auxiliary alternative is the only successful option. This lexical entry, however, also licenses *??Dares she jump?*, which sounds at least odd, and suggests that it may increasingly be the case that the finite forms of *dare* only allow the second half of the lexical entry, with only the base infinitive form allowing the inverted case.

A similar analysis can be extended to, for example, *ought*, but note that its lexical entry should not itself introduce a metavariable. The reasons for this are the same as those discussed in section 6.6, below.

6.6 Infinitival-*to*

As discussed above, this characterisation of lexical items being able to have type underspecified complements has implications for all verbs which can take more than one type of complement, for example, *want*, which can have as its object either a *to*-infinitival verb phrase, or a noun phrase. This, of course, has a knock-on implication for the characterisations of infinitival-*to*. In Sag et al. (2003), infinitival-*to* is characterised as an auxiliary verb. This is motivated by appeals to the similarities between *to* and the auxiliaries, including the facts that both are followed by base infinitives, and both can support VP-ellipsis, as shown in example 139a and b, below.

However, infinitival-*to* also exhibits characteristics that are unlike auxiliary verbs. For a start, if we are committed to treating it as an auxiliary verb, then the other uses of to, which are not verb-like, must be completely separate. It is clear to me that *to* is, in an important sense, *not* a verb, however much it behaves like one in one of its many uses. It also does not display the other NICE properties of auxiliary verbs, and, unlike auxiliaries, there is potentially no limit to the number of infinitival-*to*s in a sentence (provided appropriate host verbs can be thought up), as in example 139c.

(139) a. I want to go to the zoo, but John doesn't want to.
b. Do you want to?
c. I want to need to decide to dare to wait to see my mum.

Additionally, in the superficially similar sentences in example 140a and b, the *to*-infinitival clause is playing a different role; in 140a, it is an adjunct, whilst in 140b, it is a required complement. This is apparent when considering that a paraphrase of 140a, is "I eat in order to make myself fat", but 140b cannot be paraphrased "I want in order to make myself fat".

(140) a. I eat to make myself fat.
b. I want to make myself fat.

Taking these considerations, it seems that although treating *to* as an auxiliary verb would be inappropriate, we can nevertheless capture the similarities between it and the auxiliaries by treating it as decorating an underspecified node. The difference then, between, for example *want* and *have* when both have verbal complements is that when *want* is parsed, it creates a node with a requirement for a type underspecified node $?Ty(\mathbf{A})$ (which in this case can represent either $?Ty(t)$ or $?Ty(e)$), but does not decorate it with a formula value or a requirement for a fixed formula value. In contrast, *have* creates a node which is a requirement for a type underspecified node, but also decorates it with a formula metavariable, and a requirement that a fixed formula value be found before a parse is successful ($?\exists\mathbf{x}.Fo(\mathbf{x})$). As the A-LINK ADJUNCTION rule in 101 can only apply on a node which has a requirement for a formula value (which, in this case would not be met until *to* is parsed, but usually would be supplied by the auxiliary metavariable) this means that *to* merely supplies the necessary criterion, without which A-LINK ADJUNCTION could not apply (and effectively licensing ellipsis as a side-effect).

(141) Lexical entry for *wants*.

wants	IF	$?Ty(e \rightarrow t)$
	THEN	$\texttt{make}(\langle\downarrow_1\rangle); \texttt{go}(\langle\downarrow_1\rangle);$
		$\texttt{put}(Ty(\mathbf{A} \rightarrow (e \rightarrow t)), Fo(Want'), [\downarrow]\bot);$
		$\texttt{go}(\langle\uparrow_1\rangle); \texttt{make}(\langle\downarrow_0\rangle); \texttt{go}(\langle\downarrow_0\rangle);$
		$\texttt{put}(?Ty(\mathbf{A}));$
	ELSE	Abort

(142) Lexical entry for *to*.

to$_{inf}$	IF	$?Ty(\mathbf{A})$ where $\mathbf{A} \in \{t, (e \rightarrow t)\}$
	THEN	$\texttt{put}(Fo(\mathbf{AUX}), ?\exists\mathbf{x}.Fo(\mathbf{x}), [\downarrow]\bot)$
	ELSE	Abort

Although this needs refining to allow different type values and therefore account for additional uses of *to* (for example in prepositional phrases), and prevent overgeneration, we can see how it could be a useful starting point. Conceptually, this has consequences for the output trees and final formula value of a sentence with any verbal complements in it, which will then be of an analogous form to that of the auxiliary (with repeated subject).

While this is fine, and even desirable for some verbs which take verbal complements, eg *wants* (the group of verbs called *subject control verbs* as discussed in section 6.2.2) it is not accurate for others, like *seems* (the *subject raising verbs*). In some sense, what is required for these verbs is that when the subject is passed down the tree, via the A-LINK ADJUNCTION rule, the original subject value is more like expletive-*it*. These so-called *subject-raising verbs*, then, carry a presumption that their syntactic subject is expletive. We do not, in fact, have to add additional machinery to achieve this, as long as we assume, as per the expletive *it* example in section 4.5.2, that subject raising verbs are always of $Ty(t \rightarrow t)$. What this means is that the only way that a sentence such as *John seems to like Julia* can proceed is by the use of *ADJUNCTION, so that *John* will occupy an unfixed node. The parse will then proceed in exactly the same fashion as *It seems that John likes Julia*, with the only question still requiring resolution being how the pointer would move to the $?Ty(t \rightarrow t)$ required before *seems* can be parsed. In my opinion, one way that this could be addressed is by assuming that subject raising verbs, like the auxiliaries, would have a $?Ty(t)$ trigger[37] but with an additional constraint to ensure that a syntactic subject has already been parsed, and prevent inversion. A possible lexical entry is shown in 143, below, with further research required to answer the question of how this analysis can be extended to examples like *It seems normal* or *John seems nice*, potentially using the notions of type specification outlined in the current paper. Questions as to how this might be relatable to the ambiguity of auxiliaries (as mentioned in section 6.3.2.1), is also left for further research.

(143) Lexical entry for *seems*.

seems	IF	$?Ty(t) \wedge \langle\downarrow_*\rangle Ty(e)$
	THEN	$\mathtt{make}(\langle\downarrow_1\rangle); \mathtt{go}(\langle\downarrow_1\rangle); \mathtt{put}(?Ty(t \rightarrow t))$
	END IF	
	IF	$?Ty(t \rightarrow t)$
	THEN	$\mathtt{put}(Ty(t \rightarrow t), Fo(Seem'), [\downarrow]\bot);$
		$\mathtt{go}(\langle\uparrow_1\rangle); \mathtt{make}(\langle\downarrow_0\rangle); \mathtt{go}(\langle\downarrow_0\rangle);$
		$\mathtt{put}(?Ty(t));$
	ELSE	Abort

Going back to the characterisation of *to*, we can now see that a further question which puzzled Kim (2000) is answered immediately, namely why *I want not to* is a valid instance

[37] It is possible that it would be more parsimonious for *all* verbs in English to have a $Ty(t)$ trigger, with the difference that then was introduced being an added restriction on full verbs which did not extend to the group of auxiliary verbs. This would mean that INTRODUCTION and PREDICTION may not be necessary, however, although the advantage of this approach would be that it would mean greater unification of lexical entries for verbs (including cross-linguistically), and it does seem to be a more efficient way of accounting for the data, it also introduces additional problems, for example in cases of left-hanging topic dislocation (e.g. Mary, John loves). While these problems may be resolvable, such a wholesale change in the basics of Dynamic Syntax is left for another time. Nothing in the present paper relies on the distinction, and it is also important to remember that just because something appears more efficient from a static point in time does not mean that it was the most efficient diachronically. It is the nature of calcified routines that when they interact they may not process *all* data in the most parsimonious fashion, though it cannot be denied that the very existence of a routinised set of actions suggests that it was the most efficient mechanism for *some* process or other.

of ellipsis, whilst *I want to not*, is not. Because of the bottom restriction in the lexical entry for *to*, the only possible position of the *not* when it occurs after the *to*, is on the top of the LINKed tree created by the A-LINK ADJUNCTION rule. This means that the copy of the subject will be unfixed, and therefore there are outstanding requirements, meaning the parse tree is not complete.

7 Conclusions

As we saw above, using only the Dynamic Syntax notions of LINKed structures and underspecification, it is possible to come up with a coherent and well-motivated account for various linguistic phenomena surrounding the usage of auxiliary verbs in English, which continue to cause disputes amongst linguists. The analysis takes into account cross-linguistic and diachronic evidence, and seems to provide a good fit with the data. However, although this account solves many of the problems associated with the renowned idiosyncratic English auxiliary system, it also leaves many questions unanswered. The analysis shows that when focussing on a relatively small area of linguistic interest, such as auxiliary verbs, it is possible to come up with what seems to be an inclusive account, but that when the scope is widened to include other linguistic phenomena, it can have unpredictable and undesirable consequences, most notably in this case, in regards to the interactions of Wh-questions and relative clauses (section 6.4).

Additionally, although the type underspecified $Ty(\mathbf{A})$ introduced in this paper seemed at first to offer a useful way to account for and describe the ambiguities of some of the modal auxiliaries, it comes unstuck when we consider cases with multiple negation, and also cases with expletive subjects (e.g. *it*), suggesting that this may not in fact be the correct way to analyse auxiliary verbs in modern English. The question was raised as to whether it would be better to treat these ambiguous modal auxiliaries as themselves type underspecified (corresponding roughly to the differences proposed in the handling of subject raising and subject control verbs), rather than projecting a type underspecified node. Nevertheless, I believe that some form of type underspecification is necessary, and that the idea is more generally applicable (as was seen with verbs which take verbal complements). I feel that this addition to the ways in which a nodes decorations may be underspecified is a strength of this analysis, and its introduction to a basic Dynamic Syntax framework seems pertinent. The details for type underspecification in Dynamic Syntax, however, still require precise formalisation.

These are not the only areas which require significant additional research before the tricky problems of auxiliaries can be adequately accounted for in Dynamic Syntax. As already mentioned, tense is an area which also needs formalising in Dynamic Syntax (the $Tns(PAST)$ decorations are not to be thought of as a serious account of tense), and although a general problem, this has specific relevance for the issues presented by auxiliaries. The semantic content of auxiliaries (which express notions of, for example, time, necessity, obligation and permission) has not even been touched on in this paper, and is an important area for future investigation. Not only does it have a bearing on English, but it also has cross-linguistic implications. In Spanish, for example, the auxiliary verbs do not share the syntactic properties of the English ones (the NICE properties), but they do share semantic characteristics, which they are often defined by.

Dynamic Syntax's underdeveloped notion of tense highlights a more general issue. Unlike the Principles and Parameters program, which has evolved from the Chomskyan linguistics of the 50s, or HPSG, which has its roots in GPSG, and therefore has a 30 year tradition, Dynamic Syntax is relatively young. What this means in practise is that many topics which have been extensively researched and analysed under the alternative research programs, are as yet underexplored in Dynamic Syntax. This means that, although it can account for certain linguistic phenomena, there are many areas in which it cannot offer an explanation, and one can see why other formalisms might therefore be attractive. Additionally, the Dynamic Syntax emphasis on the ability of rules to generalise, whilst allowing us to capture previously unappreciated cross-linguistic generalisations, mean that in language specific circumstances it is sometimes necessary to stipulate convoluted constraints in order to prevent ungrammatical strings.

Furthermore, the complexities of the precise restrictions on the ordering of multiple auxiliaries has not been explored in this paper, which any complete account of auxiliaries must address. Whether placing specific constraints in the lexical entries of certain verb forms (*-ing*, *-en* and infinitive forms) would be sufficient to capture the data requires careful consideration. And all of this is not to mention the complexities introduced by negation and negative words, an area on which whole books have been written on in other frameworks and is understandably, therefore, only sketched in this paper.

Of course, this is by no means an exhaustive list of areas in need of additional research. Whether my briefly outlined hypotheses regarding the diachronic changes to the group of auxiliary verbs actually bears scrutiny is another matter. However, I have included these speculations because it is my opinion that they help shed light on how what look like clumsy disjunctions from a synchronic perspective could have arisen as previously more generally available rules became lexicalised. The auxiliary verbs can be seen, under this view, as a small sub-category of words which evolved to perform a specialised form of update, and that they therefore require a lot of disjunctive specifications should not be surprising, given the constantly shifting system of a natural language. That such language change leads to additional complexity (at the lexical level) perhaps goes against the orthodox view that grammaticalisation necessarily leads to simplification, though see Kuteva (2001) for a comprehensive repudiation of this view.

All of this may suggest that this is not, in fact the only way of analysing these phenomena using the framework of Dynamic Syntax, though I hold out hope that some of the insights can be salvaged. One other possibility, for example, as yet under explored in Dynamic Syntax is the notion of having multiple decorations at a single node, with this only being allowed if they do not contradict each other and the later decoration adds additional information to the existing formula value (in a sense, narrowing down the interpretation search space and thus removing the need for the *verbal clause* postulated by Garcia-Miller). This would have implications for auxiliary verbs, but also for optional arguments of verbs, and adjuncts, as well as for the organisation of the lexicon, and, of course, is yet another area which requires further research.

Ultimately, I firmly believe that the best way to test competing hypotheses, such as those outlined above, would be in a computational implementation. This would also enable efficient comparisons of diachronic and cross-linguistic data, and although it could not in itself offer us a hard and fast 'solution', it could at least allow us to dismiss possibilities which turn out to be too unwieldy. This would be the direction that I would be most interested in pursuing.

Overall, I feel that this Dynamic Syntax analysis of auxiliary verbs, if not completely correct, at least offers some interesting insights into some fairly eclectic phenomena, and most certainly provides intriguing avenues for further research.

References

Akmajian, A., 1984. "Sentence types and the form-function fit." *Natural Language & Linguistic Theory*, *2*(1):1–23.

Bar-Hillel, Y., 1971. "Out of the pragmatic wastebasket." *Linguistic Inquiry*, *2*:401–407.

Belletti, A., 1999. "Inversion as focalisation and related questions." *Catalan Working Papers in Linguistics*, *7*:9–45.

Blackburn, P. & Meyer-Viol, W., 1994. "Linguistics, logic and finite trees." *Bulletin of Interest Group of Pure and Applied Logics*, *2*:2–39.

Bolinger, D., 1980. "Wanna and the gradience of auxiliaries." In: G. Brettschneider & C. Lehmann (eds.), *Wege zur Universalienforschung: Sprachwissenschaftliche Beitrage zum 60. Geburtstag von Hansjakob Seiler*, pp. 292–299. Tübingen: Gunter Narr.

Bouzouita, M. & Kempson, R., 2006. "Clitic placement in old and modern spanish: a dynamic account." In: O. Nedergaard Thomsen (ed.), *Competing Models of Linguistic Change*. John Benjamin.

Cann, R., 2006. "Semantic Underspecification and the interpretation of copular clauses in English." In: K. von Heusinger & K. Turner (eds.), *Where Semantics Meets Pragmatics*. Oxford: Elsevier.

———, 2007. "Towards a dynamic account of be in English." In: I. Comorowski & K. von Heusinger (eds.), *Existence: Syntax and Semantics*. Kluwer.

Cann, R., Kempson, R. & Marten, L., 2005. *The Dynamics of Language*. Oxford: Elsevier.

Carpenter, B., 1997. *Type-Logical Semantics*. Cambridge, MA: MIT Press.

Chomsky, N., 1957. *Syntactic Structures*. The Hague: Mouton.

———, 1965. *Aspects of the Theory of Syntax*. Cambridge, MA: MIT.

Coates, J., 1983. *The Semantics of the Modal Auxiliaries*. Beckenham: Croom Helm.

Denison, D., 1993. *English Historical Syntax*. London: Longman.

Ellegård, A., 1953. *The Auxiliary Do: The Establishment and Regulation of Its Use in English*. Stockholm: Almqvist & Wiksell.

Engblom, V., 1938. *On the Origin and Early Development of the Auxiliary Do*. Gleerup.

Falk, Yehuda N., 2003. "The English auxiliary system revisited." In: *Proceedings of the LFG03 Conference*.

Firth, J.R., 1968. "Linguistic analysis as a study of meaning." *Selected Papers of J.R. Firth*, pp. 12–26.

Fodor, J.A., 1998. *Concepts: Where Cognitive Science Went Wrong*. Oxford: Oxford University Press.

Garcia-Miller, L., 2005. "The Treatment of the English Auxiliary System in Dynamic Syntax." Ms., Kings College London.

Getty, M., 2000. "Differences in the metrical behavior of Old English finite verbs: evidence for grammaticalization." *English Language and Linguistics*, *4*(1):37–67.

Grice, H.P., 1975. "Logic and Conversation." *Syntax and Semantics*, *3*(S 41):58.

Han, Chung-hye & Kroch, A., 2000. "The rise of do-support in English: implications for clause structure." In: *Proceedings of NELS 30*.

Harley, H., 2002. "Why one head is better than two: Head-movement and compounding as consequences of merge in bare phrase structure." In: *Proceedings of the University of Arizona Linguistics Colloquium*.

Hawkins, J. A., 2004. *Efficiency and Complexity in Grammars*. Oxford: OUP.

Huddleston, R.D., 1976. *An Introduction to English Transformational Syntax*. Harlow: Longman.

Hudson, R., 1997. "The rise of auxiliary DO: Verb-non-raising or category-strengthening." *Transactions of the Philological Society*, *95*(1):41–72.

Kamide, Y. & Mitchell, D., 1999. "Incremental pre-head attachment in Japanese parsing." *Language and Cognitive Processes*, *14*:631–662.

Kempson, R., Meyer-Viol, W. & Gabbay, D., 2001. *Dynamic Syntax*. Oxford: Blackwell.

Kiaer, J., 2005. “Incremental parsing in Korean: At the syntax-phonology interface.” Ms., Kings College London.

Kim, Jong-Bok, 2000. *The Grammar of Negation: A Constraint Based Approach*. Stanford: CSLI Publications.

Kim, Jong-Bok & Sag, Ivan A., 2002. “French and English negation without head-movement.” *Natural Language and Linguistic Theory*.

Kroch, A., 1994. “Morphosyntactic variation.” *30th Regional Meeting of the Chicago Linguistics Society*, pp. 180–201.

Kuteva, T., 2001. *Auxiliation: An Enquiry into the Nature of Grammaticalization*. Oxford: OUP.

Langacker, Ronald W., 1978. “The form and meaning of the English auxiliary.” *Language*, *54*(4):853–882.

Leech, G., 1987. *Meaning and the English Verb*. Harlow: Longman, 2 edn.

Lewis, M., 1986. *The English Verb: An Exploration of Structure and Meaning*. London: Commercial Colour Press.

Marten, L., 2002. *At the Syntax-Pragmatics Interface: Verbal Underspecification and Concept Formation in Dynamic Syntax*. Oxford: Oxford University Press.

Mazzon, G., 2004. *A History of English Negation*. Harlow: Longman.

Meza, I. & Pineda, L., 2002a. “The Spanish auxiliary verb system in HPSG.” In: Alexander Gelbuck (ed.), *Computational Linguistics and Intelligent Text Processing, Third International Conference CICLing 2002, Lecture Notes in Computer Science*, vol. 2276, pp. 200–209. Springer-Verlag.

———, 2002b. “Un modelo para la integración de verbos auxiliares y pronombres clíticos del español en HSPG.” In: Julio Gonzalo, Alselmo Pe nas & Antonio Ferrández (eds.), *Memorias de Iberamia 2002: Workshop on Multimodal Information Access and Natural Language Processing*. Sevilla, España.

Palmer, F. R., 1965. *A Linguistic Study of the English Verb*. Harlow: Longman.

———, 1988. *The English Verb*. Harlow: Longman, 2 edn.

Papafragou, A., 1998. “Inference and word meaning: The case of modal auxiliaries.” *Lingua*, *105*(1):1–47.

Pickering, M. & Garrod, S., 2004. “Toward a mechanistic psychology of dialogue.” *Behavioral and Brain Sciences*, *27*:169–226.

Pritchett, B., 1991. “Head positions and parsing ambiguity.” *Journal of Psycholinguistic Research*, *20*:251–270.

Richards, J., 1990. *Language Development and Individual Differences: A Study of Auxiliary Verb Learning*. Cambridge: Cambridge University Press.

Rosch, E., 1978. “Principles of categorization.” In: A. Collins & E. E. Smith (eds.), *Readings in Cognitive Science, a Perspective for Psychology and Artificial Intelligence (1988)*. San Mateo: Smith, Morgan Kaufmann.

Ross, J.R., 1969. “Auxiliaries as main verbs.” *Studies in Philosophical Linguistics*, *1*:77–102.

———, 1986. *Infinite Syntax*. Norwood, NJ: ABLEX.

Sag, Ivan A., Wasow, Thomas & Bender, Emily M., 2003. *Syntactic Theory: A Formal Introduction*. Stanford: CSLI Publications.

Schütze, Carson T., 2003. “When is a verb not a verb?” In: *Proceedings of the 19th Scandinavian Conference of Linguistics*.

Sperber, D. & Wilson, D., 1995. *Relevance: Communication and Cognition*. Oxford: Blackwell.

———, 2002. “Pragmatics, Modularity and Mind-reading.” *Mind and Language*, *17*(1 & 2):3–23.

Steedman, M., 2000. *The Syntactic Process*. Cambridge, MA: MIT Press.

Steedman, M. & Baldridge, J., 2003. “Combinatory categorial grammar.” Unpublished tutorial (Chapter 5).

Traugott, E. C., 1972. *The History of English Syntax*. London: Holt, Rinehart and Winston.

Warner, A. R., 1993. *English Auxiliaries: Structure and History*. Cambridge: Cambridge University Press.

Wilson, D. & Sperber, D., 2002. “Truthfulness and Relevance.” *Mind*, *111*(443):583.

Zwicky, A. M. & Pullum, G. K., 1983. “Cliticization vs inflection: English n’t.” *Language*, *59*(3):502–513.