

York
Papers in
Linguistics
Series 2

Issue 1

February 2004
ISSN 0307-3238

Editors:
Jonny Butler
Davita Morgan
Leendert Plug
Gareth Walker

DEPARTMENT OF LANGUAGE AND LINGUISTIC SCIENCE
UNIVERSITY OF YORK, HESLINGTON, YORK, YO10 5DD

On Having Arguments and Agreeing: Semantic EPP*

Jonny Butler

*Department of Language and Linguistic Science
University of York*

Abstract

This paper builds on the system of building predication dependencies set out in Adger and Ramchand (2003), providing a semantic characterisation of EPP features (Chomsky 2000) wherein they are treated as syntactic [Λ] features on argument-related heads that map directly to the semantics as a predicate abstraction operator λ . [Λ] features enter into binding dependencies via Agree with [ID] features on V, which are interpreted as variables; they may also be seen here as a featural reinterpretation of θ -roles. The interaction of [Λ] and [ID] is responsible both for (re-)introducing arguments, and for interpreting the structures so created. An extension of the system to cover other kinds of dependencies, such as quantification and control, is sketched.

1 The evolution of EPP

EPP has a long and chequered history; its universality and indeed existence have been defended and denied with equal vehemence. Even for those to whom it exists, its nature has undergone radical shifts over the past twenty years. Below is presented a brief run-down on the development of EPP, exclusively as seen through the eyes of Chomsky: this is of course grossly unfair to anyone else who has worked on EPP but it will do here.

*The usual big thanks to David Adger, George Tsoulas, and Nim Chimpsky for comments, discussion, and the like. I have not heeded all of their advice (scope ambiguity intended). Versions of this paper have been presented at the May 2003 meeting of the North East Syntax Society, York, the September 2003 meeting of the LAGB, Oxford, and the October 2003 CASTL Kick-off conference, Tromsø, as well as various seminars in York; thanks to audiences at same, particularly Annabel Cormack for forcing me to make things clearer that weren't. Any remaining errors serve to enhance falsifiability. This work was undertaken with the support of AHRB award number 02/61154. Correspondence: jrcb100@york.ac.uk.

- Chomsky (1981: 26): ‘[sentences] require subjects for some structural reason; call it the principle *P*’.
- Chomsky (1986: 4 ... 92 fn. 5) ‘the specifier of IP is required by the Extended Projection Principle ... Perhaps derived from the theory of predication’.
- Chomsky (1995: 55): ‘The *Extended Projection Principle* (EPP) states that [Spec, IP] is obligatory, perhaps as a morphological property of I or by virtue of the predicational character of VP (Williams 1980; Rothstein 1983)’.
- Chomsky (1995: 199): ‘The Extended Projection Principle, which requires that [Spec, IP] be realized (perhaps by an empty category), reduces to a morphological property of T: strong or weak NP-features’.
- Chomsky (1995: 232): ‘the Extended Projection Principle (EPP) plausibly reduces to a strong D-feature of I’.
- Chomsky (2000: 102): ‘Each CFC [Core Functional Category: C, T, *v*] ... allows an extra Spec beyond its s-selection ... For T, the property of allowing an extra Spec is the Extended Projection Principle (EPP). By analogy, we can call the corresponding properties of C and *v* *EPP-features*, determining positions not forced by the Projection Principle ... EPP-features are uninterpretable (nonsemantic ...) though the configuration they establish has effects for interpretation’.
- Chomsky (2001: 10): ‘The extra edge position in α required by internal Merge [Move] is optional, and has no theta-role. Assuming options to be determined in LEX [the lexicon], the head H of α must have a feature that makes this position available: an EPP-feature in standard terminology; from another point of view, the feature OCC that means “I must be an occurrence of some β .” Optimally, OCC should be available only when necessary: that is, when it contributes to an outcome at SEM [semantics] that is not otherwise expressible, the basic Fox–Reinhart intuition about optionality. Hence H has OCC only if that yields new scopal or discourse-related properties ... If H has OCC, then the new interpretive options are established if OCC is checked by internal Move ... Informally, we can think of OCC as having the “function” of providing new interpretations’.

Reading carefully through the above, we see an evolution of EPP such that at its current stage it is a very different thing from its 1981 incarnation. That is, we really have two EPPs here.

The first EPP is the standard, original one (Chomsky’s 1981 ‘principle *P*’), which required only that sentences have subjects; by 1986 this is formalised as the requirement

that the specifier of IP (in current terms, [Spec, TP]) be projected and filled. I shall call this ‘the EPP’.

The second EPP is the more recent version (the ‘OCC’ of Chomsky 2001). I will call this ‘the EPP feature’. The EPP feature is much more general than the EPP, not being tied in specifically to TP, but simply being a feature that appears on a head, with the effect that that head then obtains a specifier.

There are thus two related, but distinct, questions. The first, probably most commonly addressed, is why the EPP on T should obtain at all. That is, however we choose to formalise the EPP, we would want to know *why* TP should need a filled Spec. The second is more general, and relates to how we should formalise EPP features in general: what is the structure when we introduce, or reintroduce, an argument, and how do we interpret that structure?

In this paper I will be looking not so much at the first question (the EPP), but rather at the second (EPP features): exploring the notion of a feature that is globally responsible for introducing arguments, and looking at how the resulting structures are interpreted.

I formalise this using an extension and adaptation of Adger and Ramchand’s (2003) featural syntax for interpreting relative clause constructions, wherein two features, [Λ]¹ and [ID], conspire to create predicate abstract constructions that are directly read by the semantics: [Λ] as a predicate abstraction operator λ , [ID] as a variable.

2 The semantics of EPP

Chomsky (2000) above claims EPP features are non-semantic, even though the structures they give us have semantic effects. But given that they clearly do have these effects, we are somewhat un-subtly sidestepping the issue if we don’t have a semantic way of dealing with them. One commonly claimed semantic property of EPP is predication (e.g. Williams 1980; Rothstein 1983; Heycock 1991; Áfarli and Eide 2001). The idea I will pursue here is that EPP features — wherever they appear — can in fact be given a well-defined semantics as generalised λ -abstraction operators, instantiated by the [Λ] features of Adger and Ramchand (2003). EPP effects can be captured by these [Λ] features binding [ID] features — interpreted as variables — introduced on root lexical categories like V. This creates λ -abstracts that require arguments for saturation, which is achieved by Merge of (typically) a DP as specifier. EPP features and their interpretive effects are thus dealt with by one and the same set of features interacting.

Where [Λ] appears on a thematic heads like *v*, arguments are introduced by First Merge, bearing a direct thematic relation to the properties encoded by those heads. Where they appear on non-thematic heads, such as T, arguments are reintroduced by

¹ Λ = capital λ , lambda

Move, or what I will call Rmerge (cf. Starke 2001) as I assume a non-movement analysis of multiple occurrences. I argue basically that the non-thematic properties of heads like T are what bars the introduction of a new argument in these cases: there is no role for such a new argument to play in the thematic structure of the clause, so there would be no way to interpret it.

3 First Merge

In this section I go through the system as it applies for First Merge of arguments, inside *vP*.² In Section 3.1 I set out the basic assumptions I make about the structure of the verbal complex, and in Section 3.2 I run through the details of a derivation to show how the system pans out.

3.1 Assumptions

I assume a decomposition of the verbal complex into V and *v* layers. I take V to be a lexical Root (Pesetsky 1995; Marantz 1997), which is pretty underspecified: some kind of encyclopaedic entry giving us the big ‘meaning’ semantics — it tells us what property is to be predicated. It doesn’t do much else though: specifically, it doesn’t do predication, situation structure, and argument structure. I take these to be dealt with by *v*-level heads (Harley 1995; Pykkänen 2002; Ramchand 2003; etc.). An intransitive verb will have a single *v* head; a transitive will have two *v* heads. Each *v* head introduces a (sub-)situation³ into the representation: stative, causative, inchoative, or whatever. These will build up compositionally to give the verbal complex the denotation of a particular kind of macro-situation. So if the V Root gives us the property of, say, laughter, and *v* denotes a process, we get a verbal complex denoting an event whereby a process of laughter occurs, realised in English as the intransitive verb *laugh*. A more detailed presentation of this will be given in Section 4.1.2. All *v* heads can (but don’t necessarily) introduce an argument as specifier: the highest *v* introduces the external argument, lower heads introduce internal arguments. Here on in I shall discuss just the external argument, since everything I say will go equally for the others.⁴

These arguments are introduced by EPP features. How? Although I treat V as an underspecified Root category, I take it that it does give us a certain amount of information that will be useful in building the *v* layer. In particular, I take it that it tells us enough that we know whether we are talking about a property or a relation; i.e. an intransitive

²I take *vP* as a typical case; the same thing will go for other argument taking predicates — nominal, adjectival, etc.

³I use *situation* as a cover term for events, states etc., similar to Bach’s (1986) term *eventuality*.

⁴For ditransitive verbs obviously a more complex structure is required, but the mechanics for introducing arguments will still be the same so I shan’t go into this.

or a transitive concept. The syntax will thus ‘know’ enough to structure the situation in the v layer. I formalise this syntactically with Adger and Ramchand’s [ID] features: a property-denoting V Root will have a single [ID] feature; a relation-denoting V Root will have two [ID] features. [ID] features are interpreted as variables. In this case, they are variables over the holders of the property or relation. In this sense they are like the argument slots in a selectional grid; we might see them as a featural reinterpretation of θ -roles (cf. Hornstein 1998; Manzini and Roussou 2000).

As variables, the [ID] features need to be bound and given a value. The means of binding that I will mostly consider here is binding by a [Λ] feature. [Λ] features, I take to be introduced in the verbal complex by v heads. Each [Λ] that is introduced will bind one [ID] — so if we have a V Root with two [ID] features, and we Merge the internal v with a [Λ] feature, that [Λ] will bind one [ID], triggering λ -abstraction over the variable, and we will get a predicate.⁵ This will be satisfied by Merge of an argument as Spec. We then merge the external v with another [Λ]; that [Λ] binds the second [ID], we have another predicate so we Merge another argument. We thus have a cycle within v P of building ‘small’ propositions from each sub-situation and then abstracting over them.⁶

I take [ID] features to be obligatory on the Root V, since they serve to structure the verbal complex, and we know that even when we don’t see overt arguments, non-overt or implicit arguments are well motivated. However, I take [Λ] features to be optional. Where v has no [Λ], we don’t get an overt argument.

I will go in detail through a derivation based on these assumptions in the next section; the optionality of [Λ] will be discussed in Section 5.

3.2 Details

The review above provided a basic explanation of how the system of argument introduction works. There are a few specific details left to be filled in, however, relating to how the binding of [ID] by [Λ] occurs.

I follow Adger and Ramchand (2003) in assuming a mechanism for agreement based largely on Chomsky (1999) and subsequent related works. This mechanism makes use of two notions not discussed in the overview above. One of these is UNINTERPRETABLE features; the other is UNVALUED features. Both uninterpretable and unvalued features serve to make the element on which they occur ACTIVE for syntactic purposes: active simply means able to take part in syntactic processes. Conversely, if an element is not active it is not able to take part in syntactic processes. Uninterpretable features need to

⁵An important point later (Section 4.1.1) will be that this is the *only* way predicates are created in this system. If we have [ID] without [Λ], we don’t have a predicate.

⁶This entails that Root V is of type t — it has its arguments, they are just severely underspecified, i.e. variables. But to get a well-defined truth value in the ‘real world’, the arguments *need* to be specified, i.e. the variables need to be given a value.

be checked and deleted, unvalued features need to be valued. Once these conditions are met — i.e. once an element has checked its uninterpretable features and/or valued its unvalued features — it is rendered inactive. If the conditions are not met, we get a crash since the structure will not be semantically coherent.

Both checking and valuation take place via feature matching and the operation AGREE. An active element will search or PROBE its complement to find another active element or GOAL with matching features. Matching features are features of the same type, but not necessarily with the same value — specifically, a valued probe can match an unvalued goal. If it finds one they will Agree, which means their (relevant) features will be checked. This will not necessarily render them inactive since they may have other features that need checking still.

What this means for the current proposal is that both V and v need to be active in order for the Agree relation between $[\Lambda]$ and $[ID]$ to obtain. I follow Adger and Ramchand’s story in assuming that v has, in addition to an interpretable $[\Lambda]$ feature, an uninterpretable $[ID]$ feature, notated following Pesetsky and Torrego (2001) as $[uID]$. v is thus active. The $[ID]$ features on V I take to be interpretable, but unvalued, notated as $[ID:]$. Adger and Ramchand propose that $[ID]$ features can have either of two values⁷: they can be $[ID:\phi]$ or $[ID:\Lambda]$. The first of these is to be read as $[ID]$ with the value ϕ ; that is, a variable with ϕ -features. This is essentially a pronoun and I won’t really be concerned with it here.⁸ The second is to be read as $[ID]$ with the value Λ . This is a variable that is bound by a $[\Lambda]$ feature. $[ID]$ can potentially either enter the derivation with this value, in which case it requires binding by $[\Lambda]$ (so we get something like selective binding; see Section 4.1.1), or it can receive the value Λ by Agreeing with a higher $[\Lambda]$ feature (this is more like unselective binding). I take it that the latter option is what we get in the case under discussion, perhaps in general. The way this goes is as follows.

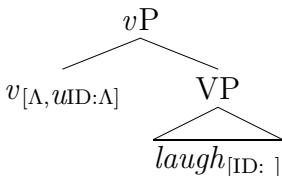
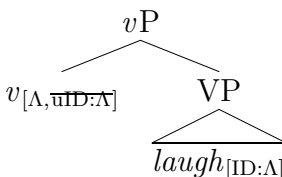
We start off with a Root V ; concretely, as above, expressing the notion LAUGH. This being a one-place property, it has on it one unvalued $[ID:]$ feature.⁹ V is thus active. It can’t really probe at this point (or can do so only vacuously) so we go on.

We Merge a process-denoting v head. This v has on it an interpretable $[\Lambda]$ feature, and an uninterpretable $[uID:\Lambda]$ feature (1a). It too, then, is active. It probes its domain for a feature matching its $[uID:\Lambda]$, and it finds the unvalued $[ID:]$ on V . The $[uID:\Lambda]$ on v is checked (notated by striking it out), and the $[ID:]$ on V is valued $[ID:\Lambda]$. We end up with (1b).

⁷I extend this proposal in Section 4.1.1 to allow additional binding possibilities.

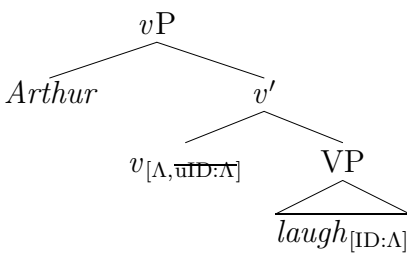
⁸In fact it won’t end up being compatible with the reinterpretation of Adger and Ramchand’s ideas I set out, at least as it stands. A pronoun would probably need to be treated just like a full DP in my system.

⁹This ignores the possibility of an ‘event argument’ (Davidson 1967; Kratzer 1995) — see Section 4.1.2 for this.

- (1) a. 
- b. 
- c. λ . laugh (δ)

(1b) maps to the semantics as (1c): a λ -abstract predicating the property of laughter over some value for V's [ID] variable. I use δ here as a kind of meta-variable, or variable across variable types — so δ can end up ranging across individuals (\mathbf{x}), situations (\mathbf{s}), or anything else we might want to have variables for; I only restrict it here to variables over atomic entities. The reason for generalising λ -abstraction away from the usual $\lambda \dots \mathbf{x}$ is that we can of course get arguments that seemingly don't denote individuals — such as gerunds, CP subjects, or locative PPs.

As a predicate, (1b) needs to be saturated, which is achieved by merging an argument of the appropriate type as a specifier for v P. v 's situational properties in this case give it the denotation of a process, so the argument is interpreted as the undergoer or experiencer of that process. We get (2a), which is read by the semantics as (2b).

- (2) a. 
- b. λ . [laugh(δ)] (Arthur)
= Arthur laugh

This seems to be just the result we want for building v P; it leaves open a question of how exactly this occurrence of the subject is interpreted relative to the later occurrence in [Spec, TP], which will be dealt with in Section 4.1.1.¹⁰

¹⁰A question that comes up here is how we get hierarchical argument structure, if a [Λ] can bind any of V's [ID]s — how do we make sure it binds the 'right' one? Actually, I am assuming there isn't a right one — all the [ID] features do is encode arity on the most basic level, i.e. number of arguments. Argument hierarchy is then encoded by v P structure, which I take to be given by UG, and it doesn't matter which [ID] associates with which v -head, since it's the v -heads that are ordered. Annabel Cormack (p.c.) suggests that this isn't sufficient, and that V should encode argument order; one way of saving the analysis here she suggests is using 'named' variables for each argument. Natural language would then have a small inventory of named variables that would associate with 'named' λ -abstractors on v -

4 Rmerge

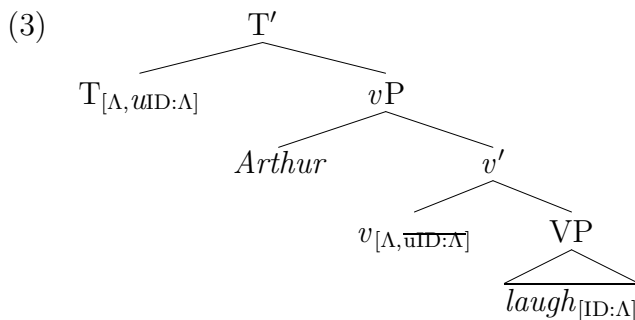
That Move or Rmerge of XPs in general involves λ -abstraction is argued by Heim and Kratzer (1998); Nissenbaum (1998); Sauerland (1998). In each of these analyses, though, the extra syntax required to set up the relevant structure — i.e. insert the λ — is either unclear (Sauerland) or clumsy (basically counter-/non-cyclic: Heim and Kratzer; Nissenbaum). Here I pursue the theory given above for First Merge to cover these cases also.

I take Rmerge to be triggered by exactly the same mechanisms we have seen above for First Merge — the interactions of $[\Lambda]$ and $[\text{ID}]$. I go through this for A-movement¹¹ in Section 4.1, and I discuss A'-movement in Section 4.2.

4.1 A-movement

The original EPP just required that TP have a (filled) Spec. Whatever the ultimate reason for this principle, it is clear that if in this system $[\Lambda]$ features are responsible generally for creating specifiers, then we have to deal with it using $[\Lambda]$ features.

After (2a), we merge T. If T has to have a Spec, then in line with the discussion above it has to have a $[\Lambda]$. In order for this $[\Lambda]$ to be able to do anything, T needs to be active. I assume that, exactly like v , T is active because it hosts a $[u\text{ID}:\Lambda]$ feature. This gives (3).



I assume with Pesetsky and Torrego (2001) that checked features remain available for matching with higher features.¹² The $[u\text{ID}:\Lambda]$ on T, then, probes to find matching features. It finds the $[u\text{ID}:\Lambda]$ on v and they Agree. This creates a semantic dependency heads. This would bring $[\text{ID}]$ features much closer to traditional θ -roles. I mention this as an interesting idea, but I don't go along with it, at least in part because I don't think we can say for sure that Root categories should encode order, given that the whole idea behind them is that they only really have an interpretation in a particular syntactic environment.

¹¹As noted in Section 2, I don't really assume a movement analysis of multiple occurrences, but A-Rmerge just doesn't sound right.

¹²In fact I follow Pesetsky and Torrego's more specific claim that checked features *at a phase edge* (Chomsky 1999) (the phase edge here being v and $v\text{P}$) remain available for matching with features in the immediately higher phase; this doesn't really matter here though.

between the two instances of the $[\Lambda]$ feature — essentially, they end up abstracting over the same $[\text{ID}]$ variable on V (see Adger and Ramchand 2003 on long distance relativisation for relevant discussion). It is not therefore possible to Merge a new argument at this point since we would in effect be trying to provide two values for the same variable: we can only Rmerge the argument we already Merged as the argument for the lower λ -abstract (or something related to it; see Section 4.1.2) or we will get semantic incoherence.¹³

It remains mysterious why T should have a $[\Lambda]$ feature at all on this view — i.e. it provides no explanation for Chomsky’s (1981) principle P (though as stated in Section 1 that isn’t the intention of this paper). I take it to be a reflex of something else — case, specificity, or one of the myriad other reasons that have been proposed for the classical EPP.

Two questions come up here:

1. **For multiple occurrences (copies):** How do we deal with them?
2. **For expletives:** Why are expletives an option? More specifically: what are expletives in this system, and how do we deal with them?

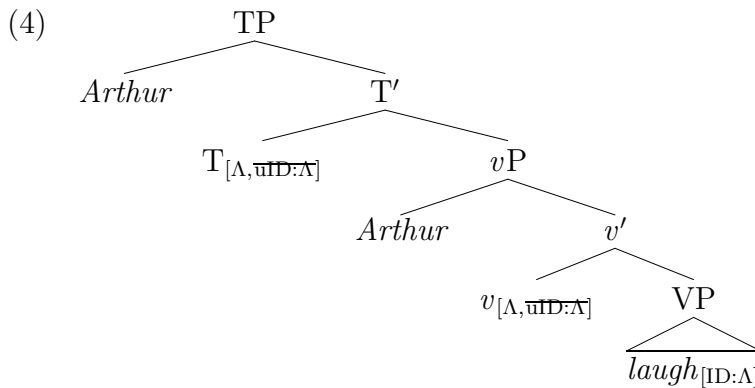
4.1.1 Multiple occurrences

The question regarding multiple occurrences is more precisely, what exactly do we interpret? It is sometimes suggested that we interpret one copy in a chain, and ignore the rest. There are known to be problems with this view with respect to partial reconstruction (for *wh*-constructions, for example), and in the system I have set out so far there is actually no way it could work: because of the way the structures are built and interpreted, simply ignoring occurrences would result in malformed semantics.

I propose instead that, along the lines of Sportiche (2002), we interpret everything which is interpretable; i.e. strong(est) version of Full Interpretation. So we continue (3)

¹³It is reasonable to be suspicious of requiring uninterpretable features for ‘active’ status — cf. Brody (1997); Manzini and Roussou (2000); Roberts and Roussou (2002); etc. It would be quite possible to formulate the dependencies discussed in the text differently. For example, we could have just interpretable $[\Lambda]$ features on v and T , with no $[\text{uID}]$. Since $[\Lambda]$ is interpreted as a λ -operator it needs to probe in order to try and find a variable to operate over, otherwise we would have something equating to vacuous quantification, which is generally considered bad. Assume this happens in vP , and $[\Lambda]$ finds an $[\text{ID}:]$ feature. It binds it. The $[\Lambda]$ feature on T probes down, but before it gets to a variable it comes to another $[\Lambda]$. We can suppose it can’t ‘look past’ this lower $[\Lambda]$ for semantic reasons — it would be analogous to having mismatched brackets in a semantic formula. The only thing it can do is to form a semantic dependency with the lower $[\Lambda]$, meaning they end up abstracting over the same variable, just as in the text. This is somewhat similar to the treatment of A-movement in terms of feature ‘attraction’ in Manzini and Roussou (2000), though they don’t have a vP -internal subject mediating the dependency; it would be interesting to investigate whether their story for control also carries over to this system. I would still take uninterpretable features, as well as unvalued features, to exist under this view, but not as a necessary condition for creating Agree dependencies. See also fn. 17.

by remerging *Arthur* in [Spec, TP], and get (4).



$$\begin{aligned}
 &= \lambda. [\text{Arthur laugh } (\delta)] (\text{Arthur}) \\
 &= \text{Arthur laugh } (\text{Arthur}) \\
 &= \text{'Arthur is such that Arthur laughed'}^{14}
 \end{aligned}$$

Why would we want to interpret everything? In (4), with a proper name, it isn't immediately obvious that it makes a difference. But when you have a full DP/QP it does.

Sportiche (2002) claims that you don't saturate predicates with DP but with NP — i.e. (for him) another predicate. This NP then associates with some definiteness/quantificational operator higher up in the structure. There are various ways to implement this; *contra* Sportiche I take predicates to be saturated syntactically by DPs/QPs (since syntactic constituency tests straightforwardly tell us this), but I assume that these have uninterpretable [DEF]/[QUANT] features which need to agree with interpretable [DEF]/[QUANT] features higher up — specifically, at the top of *vP*, and just above TP in CP (cf. Beghelli and Stowell 1997; Hallman 2000; Brody and Szabolcsi 2003). So $[\Lambda]$ will be satisfied, since it's just a syntactic feature and doesn't need to know anything about the actual eventual semantic interpretation of its argument: it just wants something which could be interpreted the right way in principle (i.e. as atomic). The semantics, though, will pan out based on interpretability, as you would expect.¹⁵

To exemplify, what this means is that if we had, say, *every boy* rather than *Arthur* in (4), where *every* is given its semantic value by a \forall operator in CP, the interpretation would be something like 'every boy was a boy that laughed' (I will make this explicit in (5)); and ditto for any other quantifier/determiner. As Sportiche (2002) notes, the result of this is that we either derive or represent, depending how you want to look at it, the conservativity of natural language determiners (Keenan and Faltz 1985) overtly in the syntax — on this view, there's no way they *couldn't* be conservative.

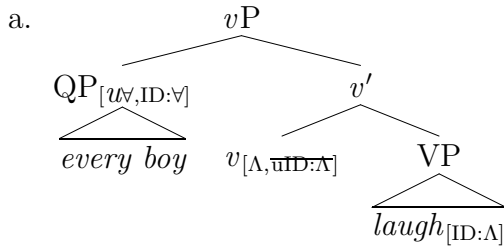
¹⁴I am assuming past tense for concreteness.

¹⁵This means the derivation in (1–4) is actually a little more complicated than presented since it should take into account [DEF] to give *Arthur* the right denotation. This would work just along the lines seen in (5).

Formally, we can obtain this result by taking \forall to be another possible value for [ID]: a feature [ID: \forall] will result in the selective binding of the variable it associates with by a higher \forall operator. A derivation for this kind of structure is shown in (5).

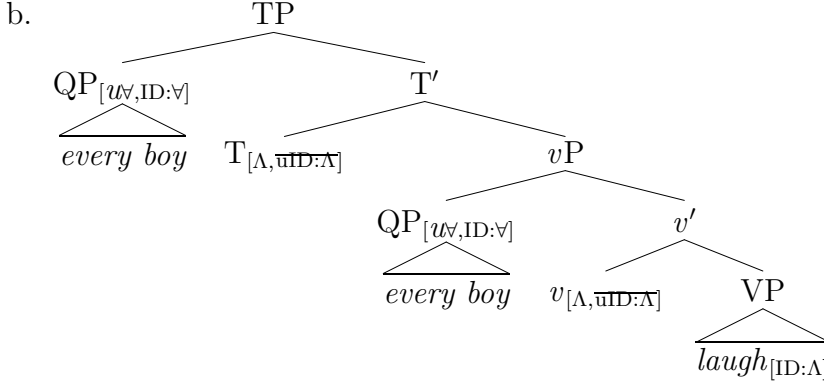
In (5a), the (syntactic) predicate v' is satisfied by the QP *every boy*. *Boy* introduces another [ID] feature into the representation, and *every* introduces a [$u\forall$] feature.¹⁶ In (5b) the QP is remerged in [Spec, TP] in essentially the same way, like *Arthur* was above; in (5c) the interpretable [\forall] is introduced and binds the relevant [ID]s by way of its [$u\text{ID}:\forall$] feature.¹⁷

(5) Every boy laughed



$$= \lambda. [\text{laugh}(\delta)] (\text{boy}(\delta_1))$$

$$= \text{laugh} (\text{boy}(\delta_1))$$



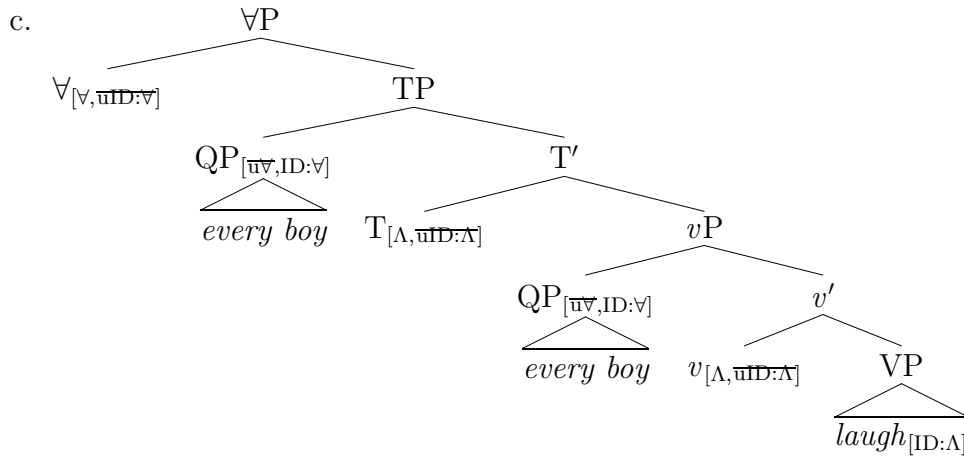
$$= \lambda. [[\text{laugh} (\text{boy}(\delta_1))] (\delta)] (\text{boy}(\delta_1))$$

$$= [\text{laugh} (\text{boy}(\delta_1))] (\text{boy}(\delta_1))$$

¹⁶Explicitly, I presume that *boy* introduces an unvalued [ID:], and that *every* introduces [$u\forall, u\text{ID}:\forall$]: matching occurs, and [ID:] is valued as [ID: \forall], just as for the discussion of matching of [ID:] and [Λ] above. I leave the notation of this off the trees, for fear they become confusing.

¹⁷As in fn. 13, we could again probably get away without making things active via uninterpretable features: [\forall] needs to Agree with some [ID] feature(s) to be interpreted, so it probes for them. It will only be able to Agree with [ID: \forall] or [ID:], though, given the way matching works, so it will bind past any differently valued [ID]s it comes across, basically buying us relativised minimality — see Section 5.

I leave aside the question of whether we want to do some kind of covert movement of the QP to [Spec, \forall P], since I think this is probably independent of anything specific to the story proposed here.



= $\forall. [\text{laugh}(\text{boy}(\delta_1))] (\text{boy}(\delta_1))$

= ‘for every δ_1 such that δ_1 is a value for boy, δ_1 is a value for a boy that laughs’

The syntax–semantics mapping looks idiosyncratic here: for example, in (5a) a λ -abstract wanting an atomic argument is being satisfied by $\text{boy}(\delta_1)$, which looks like it should be a predicate. However, on the view espoused here it isn’t actually a predicate *per se*, because the variable isn’t bound by a $[\Lambda]$, and this is what I claim makes predicates. Rather, $\text{boy}(\delta_1)$ is something more like an unspecified value for the property ‘boy’ — i.e. some (unspecified) member of the set of boys.¹⁸

This looks odd under standard assumptions if we have a universal, but actually it is treating all quantified NPs exactly like Heim (1982)-style indefinites. That is, any quantified NP here just provides a variable and a restriction for that variable. Rather than being an indefinite subject to abstract \exists -closure, though, when we have an overt quantifier like *every* we get an ‘indefinite’ that is (necessarily) subject to binding by a higher \forall operator because its [ID] feature specifies this (because of the way the QP was built; see fn. 16) — i.e. it is selectively bound.

4.1.2 Expletives

As to the question regarding expletives, it really depends how we view them. If they really are expletive, i.e. meaningless place-holders, it isn’t obvious at all how this system could deal with them — we could say the ‘real’ subject LF-replaces/adjoins or whatever, but this gives definiteness/specificity effect problems. However, this is not necessarily a sensible view of expletives anyway, as (like Chomsky’s view of EPP features) it assumes that expletives are essentially non-semantic, despite having clear and well-studied semantic effects.

¹⁸The relevant distinction is that between sets and characteristic functions of sets. $\text{Boy}(\delta)$ is basically picking out an unspecified member of a set; a predicate like $\lambda. \text{laugh}(\delta)$ on the other hand is the characteristic function of a set — a function from members of a set to truth values.

Various people have convincingly propounded an alternative view of expletives that does take into account their semantic effects, where they relate not to the subject of vP , but rather its situation interpretation. Higginbotham (1987) treats expletives as overt existential quantifiers over the event variable of Davidson (1967); Stowell (1991) treats them as overt existential situation argument QPs (more like Kratzer’s 1995 event argument), merged as the outermost argument of vP and thus selected (essentially because of locality) for raising by EPP on T; Ramchand (1996) treats them similarly, as the (raised) overt realisation of Kratzer’s (1995) event argument. These can all be seen as different instantiations of basically the same idea, that expletives function as event anchors.

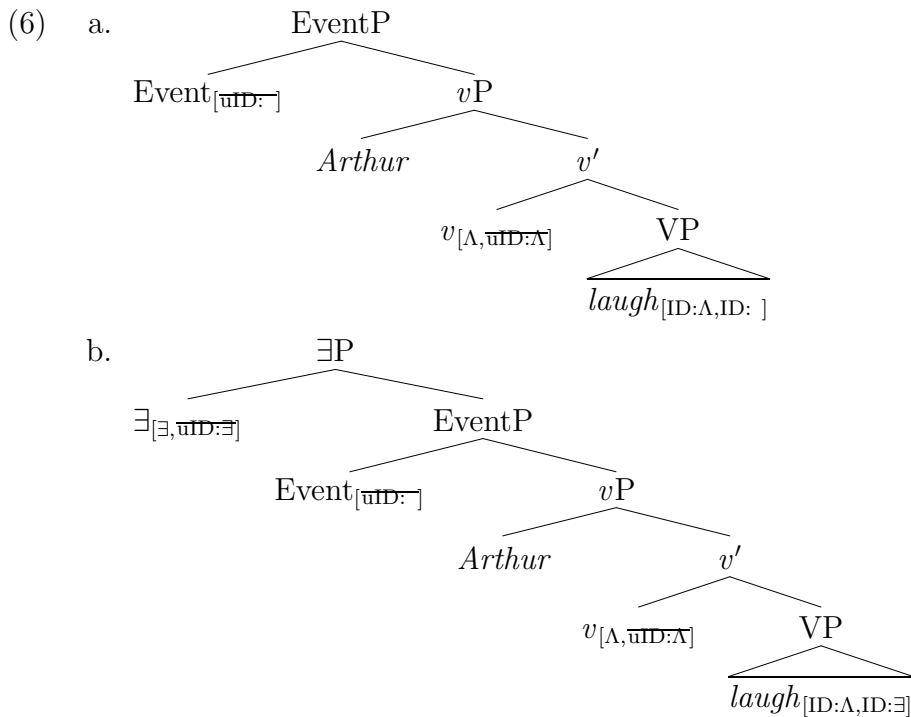
We can get this in the current system in the following way, which corresponds most closely to Stowell’s approach. Stowell (1991) follows Kratzer (1995) (who herself follows on from Davidson’s classic 1967 paper) in assuming that (some) verbs take in addition to their ‘usual’ arguments (subject, object) an event argument, which I shall refer to here as a situation argument¹⁹ (see fn. 3). This argument is the highest argument in the verbal complex, and is usually not overtly realised. Its function in my terms is to formally assign a value to the macro-situation denoted by the composed sub-situations in vP , with the purpose of making it explicitly available for operations such as binding (in the form of \exists -closure).

If this is genuinely an argument, then it should be introduced in the same way as other arguments: via an [ID:] feature on the Root V. (This will obviously make the derivations presented so far a little more complex, but not substantively different.) This in turn entails an additional v -level head, which I will notate as Event^0 (cf. Travis 1994) simply for clarity, without intending to claim that it is formally distinct from the other v heads. We can say that Event^0 has the denotation of a spatiotemporal location, and that any variable associated with it therefore has the value of the ‘holder’ of that spatiotemporal location (in the same way that an argument of a state-denoting v head is interpreted as the holder of that state; an argument of a process-denoting v head is interpreted as the undergoer of that process; etc.). Given the compositional nature of the v -layer, this will be interpreted as the location for the whole complex below it, in the same way that a cause-denoting v on top of a state-denoting v will be interpreted not just as a cause generally but a cause of that state. Event^0 on top of this would give us the location of the cause of the state. In order for Event^0 to be associated with the [ID:] on V, we want it to be active. Just as for v in the examples so far, then, we assume it has a [u ID:] feature. Given that the situation argument is generally covert, we don’t take Event^0 generally to have a [Λ] feature. So what happens?

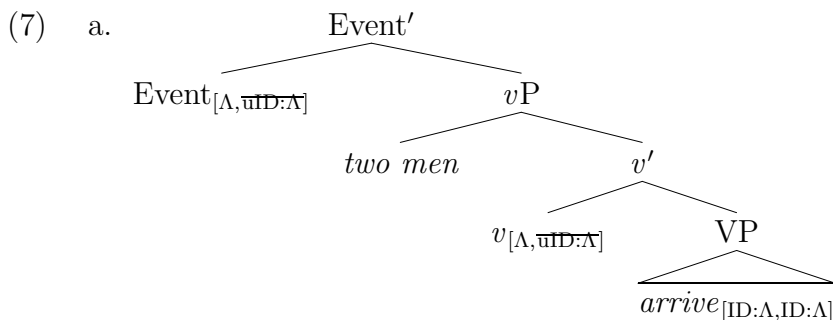
We have a structure like (6a). V has introduced two [ID:] features. One of these has

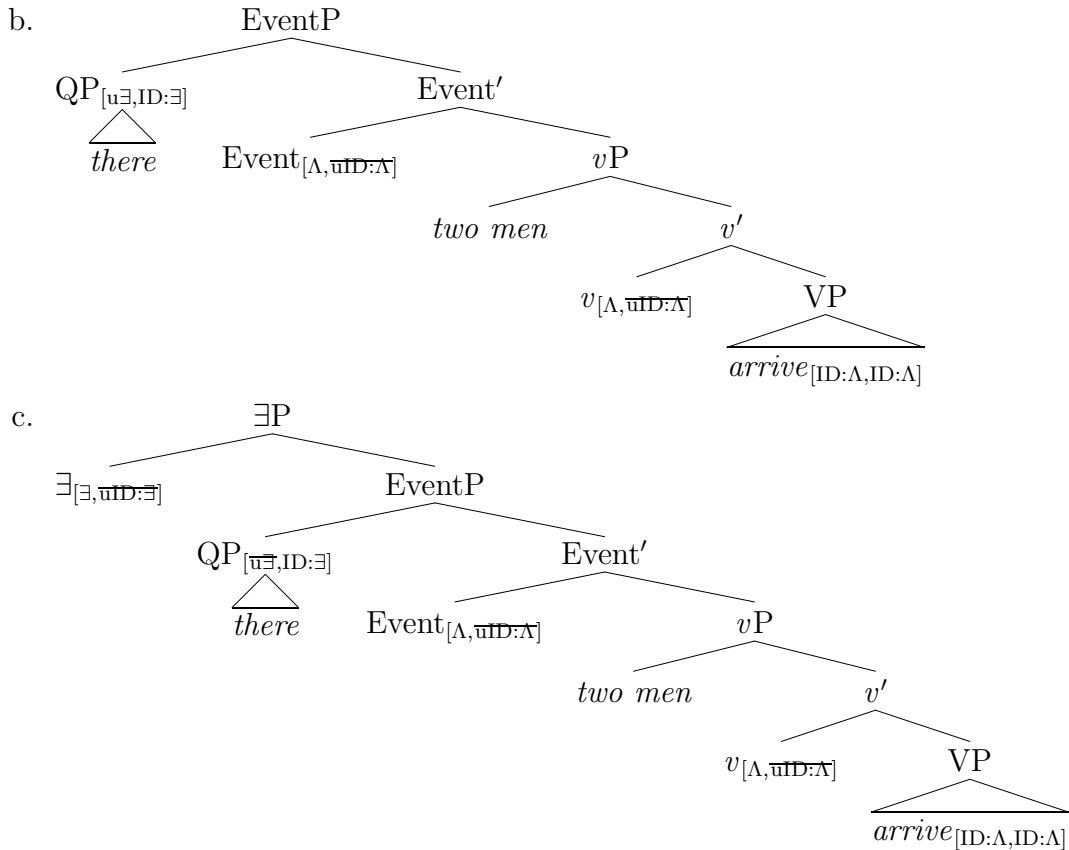
¹⁹I disagree here and assume that all verbs take a situation argument; the distinction between eventivity and non-eventivity is then encoded by distinguishing different types of situation that this argument can denote — event, state, etc. — as in Kratzer (1996).

been bound by $[\Lambda]$ on v , creating a predicate that has been satisfied by Merge of *Arthur* as Spec. The other has matched with the $[uID:]$ on Event^0 , but has not been valued (bound). It is thus a variable over holders of the spatiotemporal location of the process of Arthur laughing. As a variable it is available for binding, and I make the pretty standard assumption that it is bound by \exists -closure. I will represent the operation of \exists -closure simply by merging in an existential operator \exists at this point, without further discussion. \exists introduces an interpretable $[\exists]$ feature and an uninterpretable $[uID:\exists]$ feature, in line with what we saw for \forall in (5) above. $[uID:\exists]$ probes for matching features. It finds the $[uID:]$ on Event^0 , matches it, and values it $[uID:\exists]$; given the Agree relation that already obtains between Event^0 's $[uID:]$ and V's $[ID:]$, V's $[ID:]$ is also valued $[ID:\exists]$. We have (6b).



This is how I take \exists -closure to work in the cases seen so far. For expletive constructions, I follow Stowell in taking expletive *there* to be (in my terms) an existential QP over situations, realised as an overt argument of Event^0 . The derivation is then as in (6b), following just the same lines as the derivation featuring QP *every boy* in (5).





(7c) means something like ‘there exists at some spatiotemporal location an event of arrival by two men’. We Merge T, and the derivation continues basically as we have already seen: T probes and finds matching $[uID:\Lambda]$ on $Event^0$. They Agree and a semantic dependency is obtained between $[\Lambda]$ on T and the $[ID:\Lambda]$ on V that is bound by $Event^0$'s $[\Lambda]$. We thus have to Rmerge *there* as [Spec, TP] for the same reasons we had to Rmerge *Arthur* in (4) and *every boy* in (5). As to deriving the right word order, I adopt the standard position that V *arrive* moves up to the highest *v*-level head, i.e. $Event^0$.²⁰

4.2 A'-movement

What about A'-movement? Again, we can't say anything but that it has to work in the same way.

A question then arises, though, as to why it is different from A-movement (if we accept the arguments that it is). I suggest this is down to the landing site of the movement: A-movement is movement that lands in an A-position²¹; A'-movement is movement that

²⁰It has been pointed out to me that I predict the possibility of transitive expletive constructions (TECs) universally on this view, and therefore have a problem when it comes to languages, like English, that don't have them. In fact, I don't really have any more of a problem than anyone else since any story for expletives needs to explain this distinction. It seems to me, also, that predicting TECs universally, and then having to explain the exceptions, is the right way to go, rather than treating TECs as the exceptions, since it gives us a more general theory.

²¹Of course, questions obtain in a minimalist framework as to what an A- or an A'-position actually is.

lands in an A' -position. Is there a difference between A - and A' -positions that would lead to the empirical differences? I claim yes: A -positions are in the DOMAIN of a phase, whereas A' -positions are in the EDGE of a phase. Why does this matter?

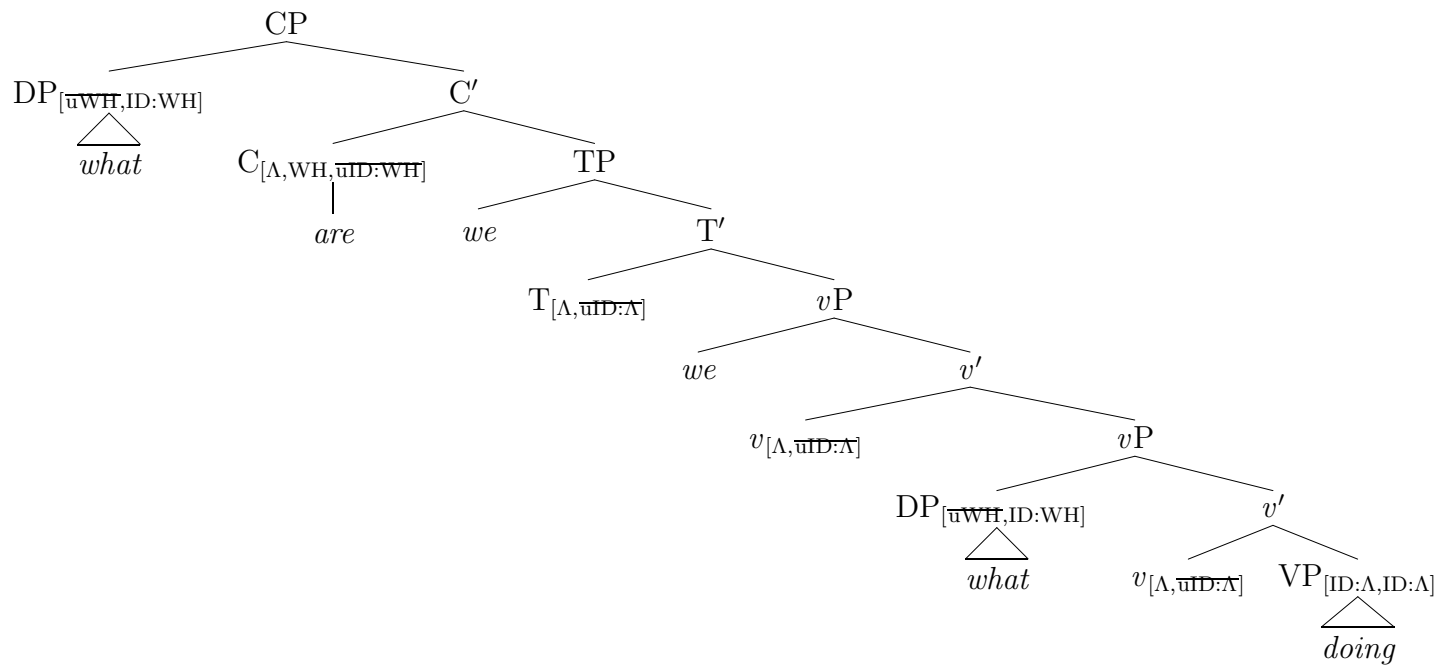
By assumption, only the edge of a phase is visible to higher operations (Chomsky 1999; etc.). A -movement, being not in the edge but in the domain of a phase (i.e. below the edge) therefore has to stop as soon as its phase is evaluated, whereas A' -movement can keep going, edge to edge to edge (successive cyclic movement). With edge to edge dependencies, each position is derivationally linked to the one below, creating a series of semantic dependencies right down the tree to the base position of the movement (the tail of the chain) (see discussion in Section 4.1.1 above of the dependency between the $[\Lambda]$ on T and the $[\Lambda]$ on v , and Adger and Ramchand 2003 on long distance relativisation in Gaelic, for details). These dependencies will be active right until the edge of the phase containing the uppermost element (the head of the chain) is evaluated. There is therefore much more freedom with regard to interpretation in these cases (reconstruction etc.), as compared to the A -movement cases where the position of the moved element is typically ‘frozen’ much sooner.

Apart from this, something extra needs to be said to get relativised minimality effects. In certain cases, we need to ‘see past’ arguments that are close to get to ones further away: for example in an English object *wh*-question we need to Rmerge the *wh*-object ‘past’ the subject, although both will involve a $[\Lambda]$ at some level on this story. I take it that this boils down to what exactly it is that’s probing: for A -movement I have assumed that all that is probing is $[uID:\Lambda]$.²² $[uID:\Lambda]$ should be able to see either an $[ID]$ feature or a $[\Lambda]$ feature. We thus get local Rmerge — so I assumed in Section 4.1.2 that T raises whatever is the highest argument in vP , whether this is the canonical subject or an expletive, basically because of locality between the $[uID:\Lambda]$ on T and the highest $[\Lambda]$ in vP . In a *wh*-construction, though, I take it the probe would be $[uID:WH]$ which would have no reason to be sensitive to $[\Lambda]$ -related features on T , and thus the subject. (Cf. the treatment of \forall in (5) above, where $[uID:\forall]$ probed for $[\forall]$ or $[ID:\forall]$ features — it could thus have seen past an indefinite subject to a universal object, had that been the structure.) We still need a $[\Lambda]$ on C to trigger/interpret the Rmerge of the *wh*-phrase, but this is in a sense secondary to the Agree between the probing $[uID:WH]$ and the *wh*-phrase — something like Pesetsky and Torrego’s (2001) claim that the EPP feature should be seen as a subfeature of a feature.

I am not really concerned about this here, but for those who are, for concreteness I will take A -positions to be case positions, A' -positions to be non-case positions, in addition to the distinctions discussed in the text.

²² If the discussion below of A' -movement is right, then maybe we want to say that actually case, or specificity, or whatever you choose to believe is ultimately responsible for the subject having to Rmerge, is the feature that is relevant for probing, and $[\Lambda]$ is somehow parasitic on this, as I claim for A' -movement in the text.

- (8) a. What are we doing?
 b.



So in (8b), C hosts $[\Lambda, WH, uID:WH]$. Being active, this probes for matching features, and finds (some of) them in $[uWH, ID:WH]$ on the *wh*-DP. Agree takes place. This still leaves $[\Lambda]$ on C. To Remerge *what*, $[\Lambda]$ has to Agree with the $[ID:\Lambda]$ on the internal *v* head that introduced *what* in the first place, in order to be interpreted as a λ -abstract over the same variable. Since $[\Lambda]$ on C is not in itself active, this can only happen parasitically.

For this to be able to happen, it looks like we have to allow that a probing feature gives its feature bundle access not just to the feature bundle that it Agrees with, but also, if it is Agreeing with something in Spec, to the feature bundle in the head that it is specifier of, presumably as a reflex of general Spec-head agreement.²³ As mentioned in fn. 22, this may hold not just in A' cases, but also in the general case if $[\Lambda]$ Agree is parasitic in A-movement too. This may seem undesirable; alternatively it may point to the correctness of a Starke (2001) view wherein the distinction between heads and Specs is neutralised, and a category can project either as an X (traditional head) or as an XP (traditional Spec). Working out the alternative architecture this would require would take me too far afield here.

5 No $[\Lambda]$

So how does this differ from previous claims that EPP effects/EPP features are tied in to predication? These come in two main varieties:

1. Predication is read off syntactic configurations (Williams 1980; Rothstein 1983; Heycock 1991). All these boil down fundamentally to the idea that the syntax reads any maximal projection as a predicate, which needs to be saturated.
2. Predication is instantiated by a specialised head Pred (Áfarli and Eide 2001; cf. Bowers 1993; Svenonius 1994). On this view, VP would denote not a predicate but a property, which is converted to a predicate by a predicative head Pred selecting it.

The first proposal, which generalizes to the principle that the syntax reads anything that is the right configuration (which for me would be any X' rather than any XP), doesn't fit in with various claims made so far. For example, I assumed in Section 4.1.1 that NPs didn't necessarily denote predicates, but could just denote sets (see fn. 18). Of course this may not be the case, but certainly we see NPs that, if they are predicates, don't take arguments. Whether this is a problem probably depends on what other assumptions you make. More specifically problematic, I treated $[\Lambda]$ features as optional in Section 4.1.2, so that Event⁰ could either Merge an expletive if it had $[\Lambda]$, or just be subject to \exists -closure if it didn't. The analysis of expletives I presented was radically different

²³Perhaps also vice versa — i.e. a probe that finds a head goal may have access to that head's Spec.

from that proposed by any of the authors cited here, but below I will suggest that $[\Lambda]$ is optional in the general case too. If this is on the right track, there is no way we could deal with EPP effects just in configurational terms.

The second proposal allows for such optionality, inasmuch as the head Pred can be assumed either to appear or not to appear. However, it isn't really clear that this is getting us anything that treating predication featurally doesn't, other than more structure. In addition, Åfarli and Eide only assume a Pred head over VP. They also take higher layers of predication to obtain, at the IP and CP levels, but they take these to be instantiated just by the heads I and C themselves. There is a major inconsistency with this, and there are two ways of fixing it: one is to say that in fact I and C also may be selected by a Pred head; the other, akin to my claims here and seemingly more minimal, is to say that I and C can instantiate predication just when they have the relevant features, and to extend this to V too. One thing that has been claimed to give empirical support to the PredP view is that we sometimes see morphemes that can be analysed as the Pred head, distinct from the verb. However, when this is so, it is just as possible that these morphemes are spelling out just the $[\Lambda]$ feature on a v head (or other $[\Lambda]$ -hosting 'light' head, if we are dealing with a non-verbal small clause). An important point is that we don't see such morphemes over I or C, even when we clearly have predication going on at these levels. Again, then, the general case for positing a specialised (optional) Pred head is weakened.²⁴

EPP features are quite generally taken to be optional for at least some cases. If we equate these with $[\Lambda]$ features, then we buy the optionality of predication straightforwardly. We already saw this with expletives: I examine the consequences further here, sketching a proposal where the lack of a $[\Lambda]$ binder for $[\text{ID}:]$ gives us implicit arguments like PRO.

5.1 Alternative binders

If there is no $[\Lambda]$ binding $[\text{ID}:]$ it needs to be bound by some alternative binder, such as external quantificational elements like adverbs or closure operators, or anaphora — since otherwise we would have free variables hanging about and giving us bad semantics. The quantificational options are standardly assumed for the situation argument $[\text{ID}:]$ that associates with Event^0 , anaphora less so; I argue here that both options are available for the $[\text{ID}:]$ s that associate with the v heads.

²⁴Another difference between the approach here and both the other approaches — or at least the instantiations of them cited — is that in those approaches, expletives don't really semantically saturate the predicate: they just fill up a 'syntactic' argument slot, and leave the semantics to sort it out later. If there is really such a strong correlation between EPP effects and predication as they claim, then this is something of an anomaly. In my approach, expletives are brought fully into line with other arguments, filling argument slots in exactly the same way, both syntactically and semantically.

What status would these [ID:]s end up with on this view? It would depend on what was doing the binding:

1. If [ID:] was closed off, it would have the interpretation of some kind of ‘abstract argument’ — so on a cause-denoting v we would interpret it as some kind of abstract causer, we wouldn’t know what. Interpretation should vary predictably depending whether it was existentially or generically closed.
2. If it was otherwise quantified over, say adverbially, it would be similarly construed as a quantified-over abstract argument, its interpretation depending on what was doing the quantification.
3. If it was bound by anaphora, it would refer back to its antecedent.

I propose that generic closure of [ID:] gives us PRO_{ARB} ; anaphoric binding gives us control PRO; and existential closure gives us the subject of passives. I won’t discuss adverbial quantification; see Kratzer (1995) for discussion and references.

5.2 PRO_{ARB}

PRO_{ARB} is an abstract argument with arbitrary reference — that is, it says that a predicate is true of more or less any value you choose for that argument. This is usually formalised by giving it (semi-)universal or generic force (though the formalisation varies widely; see e.g. Chomsky 1981; Manzini 1983; Epstein 1984; Lebeaux 1984). The classic example, from Epstein (1984), is (9).

- (9) It is fun [PRO to play baseball]

The interpretation here is that if you pick a value for PRO, it will generally be fun for the referent of that value to play baseball. Genericity is often taken to be introduced into the representation by a generic closure operator GEN, and I will assume this here. PRO_{ARB} , then, is basically interpreted as a variable subject to closure by GEN. The derivation of this interpretation should clear enough on the theory here: v introduces a [$u\text{ID:}$] but no [Λ]. It thus matches with [ID:] on V, but doesn’t value (bind) it.

If we take infinitives to lack TP and CP (cf. Wurmbrand 2001 for references and a recent version of the claim that this is true for at least some cases), that’s it; if we take them to have TP and CP then we introduce T, which has on it [$u\text{ID:}$] but no [Λ], just like v ; T probes and matches its [$u\text{ID:}$] with that of v as we have already seen. Either way, GEN is introduced with features [GEN, $u\text{ID:GEN}$]. It probes, finds the lower unvalued [$u\text{ID:}$](s) and they match. A dependency is created, with the result that GEN ends up

valuing and binding the [ID:] on *V* as [ID:GEN]. We get the interpretation usually notated as PRO_{ARB} .²⁵

5.3 Control PRO

Control PRO is an abstract argument whose interpretation is not fixed by GEN (or whatever we may take to impart arbitrary reference to PRO_{ARB}) but rather by anaphoric reference to some higher element: the subject or object of the immediately higher clause, as in (10a–10b).

- (10) a. Susan_i promised Arthur [PRO_i to wash the dishes]
 b. Susan believed Arthur_i [PRO_i to like peanut butter]

The derivation of this reading in the current system should again be fairly clear. Briefly: the derivation would be the same as for PRO_{ARB} up to immediately before the introduction of GEN. In the control case, instead of GEN, a control operator is introduced in the matrix *vP* (presumably, to be consistent with the other cases, as something like an interpretable feature [CONTROL]). A dependency with [ID:] on *V* is created via Agree, and [ID:] gets valued [ID:CONTROL], receiving an interpretation anaphoric to one of the matrix arguments; which one depending on whether it is a subject or an object control verb.²⁶

This general treatment of PRO recalls in some ways the old arguments that PRO is not actually a non-(PF)overt pronoun, but rather syntactically unexpressed — cf. Partee and Bach (1980); Chierchia (1984); Klein and Sag (1985); and particularly Williams' (1985: 314) claim that implicit arguments are 'unlinked . . . argument slots in the argument structure' (see the suggestion in Section 3.1 that the [ID] features on *V* correspond to a featural reinterpretation of θ -roles).

²⁵In a very nice paper, to which the current section owes a lot, Bhatt and Izvorski (1998) argue that in fact PRO_{ARB} is just like control PRO, but that it is controlled by a higher arbitrary (often) implicit argument — so (9) would really look something like (1).

- (1) It is fun (for implicit argument_i) [PRO_i to play baseball]

They argue that it is actually this implicit argument that is generically construed, so PRO gets a generic construal too by virtue of being controlled. There is then a question of whether the higher implicit argument is also a PRO or what, though; in any case, under this analysis I would analyse the higher implicit argument as being a GEN closed [ID:] in the matrix *vP*, and the lower [ID:] (PRO) as being controlled anaphorically by the higher.

²⁶A more detailed exposition of this might reduce the subject/object control property of the verb to whether the control operator is introduced by the subject-introducing *v* head or the object-introducing *v* head.

5.4 Passive

It is beyond the scope of this paper to propose a complete theory of the structure of passives. However, independently of any detailed proposal we know that the logical subject of a passive, though not overt, needs to be represented at some level: it can license a ‘by’ phrase that makes its identity explicit (11b); it can control a PRO subject in an adjunct rationale clause (11c); and it can be associated with agentive adverbs such as *deliberately* (11d).

- (11) a. The biscuit was eaten
 b. The biscuit was eaten (by Susan)
 c. The biscuit was eaten (PRO to annoy Arthur)
 d. The biscuit was eaten (deliberately)

If we suppose that all these properties require a syntactic reflex, not just a semantic representation, then we have also to suppose that the logical subject of a passive has some syntactic reality. Since the system I am proposing provides a strict one-to-one mapping between the syntax and the semantics, then this assumption is forced.

I propose that the logical subject of passives is similar to the analysis of PRO_{ARB} that I gave above, where it is a closed-off $[\text{ID}:]$ variable on V , with the difference that it is closed off by \exists rather than GEN . If we look at the examples in (11), the implicit subject is clearly interpreted existentially: along the lines of ‘there exists some δ such that δ ate the biscuit’. The derivation would run something like this: V *eat* introduces two $[\text{ID}:]$ features. The object-introducing v head is Merged, and it introduces the object *the biscuit* in the same way that we have seen (for subjects) thus far. The subject-introducing v is Merged and introduces a $[u\text{ID}:]$, but no $[\Lambda]$. $[u\text{ID}:]$ probes and matches the second $[\text{ID}:]$ on V , but doesn’t value (bind) it. As a free variable, this $[\text{ID}:]$ is subject to \exists -closure, as described in Section 4.1.2, and we get the interpretation given above: ‘there exists some δ such that δ ate the biscuit’. When the $[u\text{ID}:\Lambda]$ on T is merged, it probes to find a matching feature. It can’t match with the higher, subject v , because that has been valued $[u\text{ID}:\exists]$ by existential closure. It can match with the object v , because that is valued $[u\text{ID}:\Lambda]$, like itself. There is therefore a semantic dependency created between T and the object v . The DP that was Merged as Spec of this v (*the biscuit*) then must Rmerge as Spec of T , exactly as we find.

The implicit logical subject of a passive, then, is represented syntactically on this view as an \exists -closed $[\text{ID}:]$ variable on V . What about the cases in (11b–11d)? Briefly: in (11b), I suppose that $[\text{ID}:]$ is valued by something like control out of the ‘by’ phrase, instead of by \exists (or perhaps the ‘by’ phrase forms a restriction for \exists). In (11c), I take it that \exists binds into the adjunct, thus giving the PRO $[\text{ID}:]$ the same value as the passive subject $[\text{ID}:]$. In (11d), I take it that the agentive adverb just does whatever it normally does,

not really caring whether the subject is an overt argument or not, as long as agentivity is adequately represented, which it is.

Note that rationale adjuncts also seem to be able to be licensed by the situation argument [ID:] associated with Event⁰, as in (12), from Williams (1985).

(12) Grass is green [PRO to promote photosynthesis]

Clearly *grass* isn't the controller of PRO in the rationale clause; and it doesn't seem sensible to claim that *green* has an implicit argument like a passive. What promotes photosynthesis is the situation of grass having the property of greenness.²⁷ That is, it looks like the \exists -closed situation argument is able to control PRO in the infinitive, exactly parallel to the way I am claiming the \exists -closed implicit logical argument of a passive does in (11c).

This kind of treatment of the logical subject of passives is reminiscent in certain ways of that presented by Baker (1988), and also of the 'suppressed arguments' of Grimshaw (1990).

6 Conclusion

I have presented a system of argument introduction and interpretation in which the EPP features of Chomsky (2000) are represented as the syntactic [A] features of Adger and Ramchand (2003), which map transparently to the semantics as λ . These bind [ID] features, interpreted as variables, introduced on Root V, cyclically creating predicates over the property V denotes. As predicates, these need satisfying, which is achieved by Merge of a DP argument in Spec of the [A] introducing head. The same feature is thus responsible both for the introduction (and reintroduction, i.e. Rmerge/Move) of arguments, and for their interpretation *as* arguments.

This has numerous corollaries for argument structure/interpretation generally. First, it is possible to see the [ID] features on V as a featural reinterpretation of θ -roles. This brings them into line with the aims of the minimalist program generally, where features are taken to be responsible for as much as we can get them to be, hopefully everything.

Second, because of the way the system is set up, it provides an explanation for the locality of Rmerge (=Move) in the cases of the standard EPP on T: the (relevant) featural content of T 'sees' as far as the next [A] feature down, which creates a dependency between the [A] on T and the [A] on that head. They are therefore required to have the same argument, otherwise they would basically be trying to provide different values for the same variable.

Third, it provides a generalised system of binding by operators other than λ (quantifiers, etc.), wherein both selective and non-selective binding can be represented, but

²⁷This is biologically a naïve statement of the facts, but no matter.

where both are derived by exactly the same set of processes. This additionally gets us relativised minimality effects.

Fourth, it provides a simple means of representing \exists /GEN-closure syntactically, which helps provide analyses both of expletives and of non-overt arguments generally, including the situation argument, arbitrary and control PRO, and the logical subject of passives.

References

- Adger, D. and G. Ramchand (2003). Merge and Move: Wh-dependencies revisited. Ms. QMUL/University of Oxford.
- Bach, E. (1986). The algebra of events. *Linguistics & Philosophy* 9, 5–16.
- Baker, M. (1988). *Incorporation: A Theory of Grammatical Function Changing*. Chicago: University of Chicago Press.
- Beghelli, F. and T. Stowell (1997). Distributivity and negation: the syntax of *each* and *every*. In A. Szabolcsi (Ed.), *Ways of Scope Taking*, pp. 71–107. Dordrecht: Kluwer.
- Bhatt, R. and R. Izvorski (1998). Genericity, implicit arguments and control. *Proceedings of the Student Conference in Linguistics (SCIL) 7*. Distributed by MITWPL.
- Bowers, J. (1993). The syntax of predication. *Linguistic Inquiry* 24, 591–656.
- Brody, M. (1997). Perfect chains. In L. Haegeman (Ed.), *Elements of Grammar: Handbook in Generative Syntax*, pp. 139–167. Dordrecht: Kluwer.
- Brody, M. and A. Szabolcsi (2003). Overt scope in Hungarian. *Syntax* 6, 19–51.
- Chierchia, G. (1984). *Topics in the Syntax and Semantics of Infinitives and Gerunds*. Ph. D. thesis, University of Massachusetts at Amherst.
- Chomsky, N. (1981). *Lectures on Government and Binding*. Dordrecht: Foris.
- Chomsky, N. (1986). *Barriers*. Cambridge, MA: MIT Press.
- Chomsky, N. (1995). *The Minimalist Program*. Cambridge, MA: MIT Press.
- Chomsky, N. (1999). Derivation by Phase. *MITOPL* 18.
- Chomsky, N. (2000). Minimalist Inquiries: the framework. In R. Martin, D. Michaels, and J. Uriagereka (Eds.), *Step by Step: Essays on Minimalist Syntax in Honor of Howard Lasnik*, pp. 89–155. Cambridge, MA: MIT Press.
- Chomsky, N. (2001). Beyond explanatory adequacy. Ms. MIT.

- Davidson, D. (1967). The logical form of action sentences. In N. Rescher (Ed.), *The Logic of Decision and Action*. Pittsburgh: University of Pittsburgh Press.
- Epstein, S. (1984). Quantifier-*pro* and the LF representation of PROarb. *Linguistic Inquiry* 15, 499–505.
- Grimshaw, J. (1990). *Argument Structure*. Cambridge, MA: MIT Press.
- Hallman, P. (2000). *The Structure of Predicates: Interactions of Derivation, Case and Quantification*. Ph. D. thesis, UCLA.
- Harley, H. (1995). *Subjects, Events, and Licensing*. Ph. D. thesis, MIT.
- Heim, I. and A. Kratzer (1998). *Semantics in Generative Grammar*. Oxford: Blackwell.
- Heycock, C. (1991). *Layers of Predication: The Non-lexical Syntax of Clauses*. Ph. D. thesis, UPenn.
- Higginbotham, J. (1987). Indefiniteness and predication. In E. Reuland and A. ter Meulen (Eds.), *The Representation of (In)definiteness*, pp. 43–70. Cambridge, MA: MIT Press.
- Hornstein, N. (1998). Movement and chains. *Syntax* 1, 99–127.
- Keenan, E. L. and L. M. Faltz (1985). *Boolean Semantics for Natural Languages*. Dordrecht: D. Reidel.
- Klein, E. and I. A. Sag (1985). Type driven translation. *Linguistics and Philosophy* 8, 163–202.
- Kratzer, A. (1995). Stage-level and individual-level predicates. In G. N. Carlson and F. J. Pelletier (Eds.), *The Generic Book*, pp. 125–175. Chicago, IL: University of Chicago Press.
- Kratzer, A. (1996). Severing the external argument from its verb. In J. Rooryck and L. Zaring (Eds.), *Phrase Structure and the Lexicon*, pp. 109–137. Dordrecht: Kluwer.
- Lebeaux, D. (1984). Anaphoric binding and the definition of PRO. *Proceedings of NELS 14*, 253–274.
- Manzini, R. (1983). *Restructuring and Reanalysis*. Ph. D. thesis, MIT.
- Manzini, R. and A. Roussou (2000). A minimalist theory of A-movement and control. *Lingua* 110, 409–447.
- Marantz, A. (1997). No escape from syntax: Don't try morphological analysis in the privacy of your own lexicon. *University of Pennsylvania Working Papers in Linguistics* 4.2, 201–225.

- Nissenbaum, J. (1998). Movement and derived predicates: Evidence from parasitic gaps. In U. Sauerland and O. Percus (Eds.), *The Interpretive Tract*, Volume 25 of *MIT Working Papers in Linguistics*, pp. 247–295. Cambridge, MA: MITWPL.
- Partee, B. and E. Bach (1980). Anaphora and semantic structure. In J. Kreiman and A. Ojeda (Eds.), *Papers from the Parasession on Pronouns and Anaphora*. Chicago: Chicago Linguistics Society.
- Pesetsky, D. (1995). *Zero Syntax: Experiencers and Cascades*. Cambridge, MA: MIT Press.
- Pesetsky, D. and E. Torrego (2001). T to C movement: causes and consequences. In M. Kenstowicz (Ed.), *Ken Hale: A Life in Language*, pp. 355–426. Cambridge, MA: MIT Press.
- Pylkkänen, L. (2002). Verbal domains: Causative formation at the root, category and phase levels. Paper presented at the Linguistic Society of America Annual Meeting 2002.
- Ramchand, G. (1996). Two types of predication in Scottish Gaelic. *Natural Language Semantics* 4, 165–191.
- Ramchand, G. (2003). First phase syntax. Paper presented at the workshop on the linguistics/philosophy Interface, USC, February 2003.
- Roberts, I. and A. Roussou (2002). The extended projection principle as a condition on the Tense Dependency. In P. Svenonius (Ed.), *Subjects, Expletives, and the EPP*, pp. 125–156. Oxford: Oxford University Press.
- Rothstein, S. (1983). *The Syntactic Forms of Predication*. Ph. D. thesis, MIT.
- Sauerland, U. (1998). *The Meaning of Chains*. Ph. D. thesis, MIT.
- Sportiche, D. (2002). Movement types and triggers. Paper presented at the TiLT workshop, GLOW 2002, Utrecht.
- Starke, M. (2001). *Move Dissolves into Merge: A Theory of Locality*. Ph. D. thesis, University of Geneva.
- Stowell, T. (1991). Alignment of arguments in Adjective Phrases. In S. Rothstein (Ed.), *Perspectives on Phrase Structure: Heads and Licensing*, Volume 24 of *Syntax and Semantics*, pp. 105–135. San Diego: Academic Press.
- Svenonius, P. (1994). *Dependent Nexus: Subordinate Predication Structures in English and the Scandinavian Languages*. Ph. D. thesis, UCSC.

- Travis, L. (1994). Event phrase and a theory of functional categories. In *Proceedings of the 1994 Annual Conference of the Canadian Linguistics Society*, pp. 559–570.
- Williams, E. (1980). Predication. *Linguistic Inquiry* 11, 203–237.
- Williams, E. (1985). PRO and the subject of NP. *Natural Language and Linguistic Theory* 3, 297–315.
- Wurmbrand, S. (2001). *Infinitives: Restructuring and Clause Structure*. Berlin: Mouton de Gruyter.
- Áfarli, T. A. and K. M. Eide (2001). Predication at the Interface. *ZASPiL* 26. http://www.zas.gwz-berlin.de/papers/zaspil/articles/zaspil26_afa.pdf.