Labeling by denotational types

Valentina Bianchi

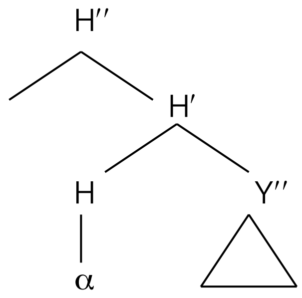
The labeling algorithm proposed by Chomsky (2013), when justified by a principle of legibility at the interface with semantics (Rizzi 2016), requires a many-to-one correspondence between syntactic labels and denotational types. In this paper I explore the view that syntactic nodes are directly labeled by denotational types, via a limited set of rules which only make reference to the labels of the daughter nodes, complying with compositionality/minimal search. I show that such a labeling system accounts for successive cyclic wh-movement as a derivational step driven by detectable type mismatch within narrow syntax; this also extends to Quantifier Raising, which, from this perspective, is triggered in the same way. The labeling system also allows for arguments saturating the valence of a predicate and for free adjunction of two nodes bearing conjoinable types, accounting for Chomsky's "unstructured coordination" and for depictive secondary predicates.

Keywords: Labeling, Quantifier Raising, wh-movement, adjunction, secondary predicates.

1. *Introduction*

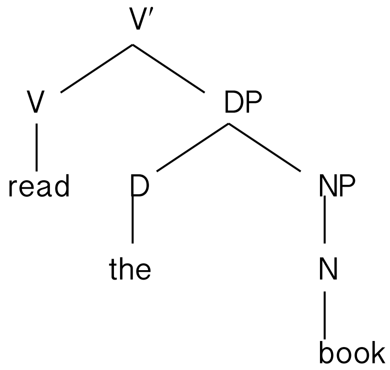
The minimalist approach to phrase structure initiated by Chomsky (1995, 241-249) has rejected the traditional X-bar projection schema of the Principles and Parameters framework, according to which every node immediately dominating a terminal symbol α is labeled by a syntactic category H, and the nodes dominating it inherit that syntactic category, augmented with bars to indicate the projection levels, as in (1):

(1)



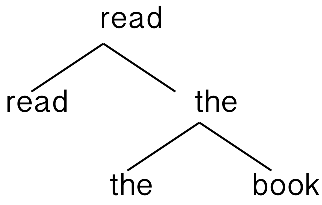
Chomsky argues that the bar diacritics violate the Inclusiveness Condition,[1] since they are not part of the information associated with the lexical item; Inclusiveness therefore requires that the projections of a head inherit the label of that head in identical form.
In addition, the category labels (V, N, A, P, T, C etc.) constitute a departure from minimal design in that they require a partial projection of the information contained in the lexical items that they dominate. Therefore, minimal design implies that non-terminal nodes are directly labeled by the lexical items themselves.
Combining these two ideas, the phrase structure of a complex expression like *read the book* would be as in (3), instead of the more familiar (2):

(2)



[1] "A "perfect language" should meet the condition of inclusiveness: any structure formed by the computation... is constituted of elements already present in the lexical items selected for [the numeration] N; no new objects are added in the course of computation apart from arrangements of lexical properties" (Chomsky 1995, 228).

(3)



Even the phrase structure in (3) is non-minimal in that it contains information about the linear order of the merged syntactic objects: but linear ordering should be though of as a requirement imposed at the interface with the Sensori-Motor systems, and not as a requirement constraining the syntactic computation (Chomsky 1995, 334-335, *contra* Kayne 1994). Dropping linear order, Merge can be reduced to set formation, and (3) is further reduced to (4):[2]

(4) {read, {read, {the, {the, book}}}}

With Merge reduced to set formation, two more issues arise. First, whenever two labeled syntactic objects are merged together, it is necessary to univocally determine which one of the two will trasmit its label to the mother node: in other terms, it is necessary to devise a *labeling algorithm* (LA). In principle, LA should require no further information apart from that which is locally available on the labels of the two merged nodes.

Second, the deeper question arises of why labels are needed at all, in view of the strong minimalist thesis whereby grammar is an optimal solution to the interface requirements posed by the external systems. The answer to this second question has an impact on the first issue as well. One possible view is that a node must be labeled in order to be visible for the subsequent syntactic computation: that is, labels are required by the working of narrow syntax itself.[3] This implies that all non-terminal nodes must be labeled as soon as they are formed by an application of External or Internal Merge. An alternative view is that labels are required at the interfaces, in order for syntactic objects to be legible to the external systems. This has different implications for the labeling algorithm: labeling can be delayed until the phase level, when the syntactic object is transferred to the external systems.

In this paper I discuss this second view of labeling, focussing on Chomsky's (2013) proposal, further developed and modified in Rizzi (2016) (section 2), and I reconsider the issue of what type of labeling exactly would be required by the external interfaces (section 3). Building on Rizzi (2016), I discuss the hypothesis that category labels are required at the interface with semantics. I argue that in order for syntactic labels to be of any use to the semantic component, each label must correspond to exactly one

---

[2] We will return to this kind of phrase structure representation in section 2. However, following common practice, we will often retain the more familiar tree notation for convenience.

[3] See Cecchetto & Donati (2015, 156-157) for discussion.

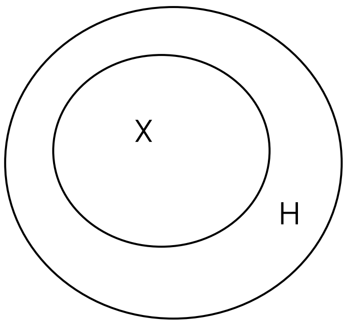
semantic type, but it is actually the type that makes the strucure legible to compositional interpretation rules. Therefore, in section 4 I put forth the hypothesis that syntactic objects are directly labeled by the types of their denotations. In sections 5 to 7 I discuss some consequences of this hypothesis concerning successive-cyclic wh-movement, the problematic status of unstructured coordinations, and Quantifier Raising. In section 8 I apply the proposed labeling algorithm to the empirical domain of depictive predicates, and I show that it allows for a maximally simple analysis. Finally, in section 9 I summarize the main points and briefly discusses one remaining issue.

2. *Chomsky (2013) and Rizzi (2016) on labeling*

Chomsky (2013, 42-48) argues that the simplest conception of Merge is that it takes two syntactic objects α and β and creates the set $\{\alpha, \beta\}$, with no projection – i.e. no label for the set provided by the Merge operation itself. Since Merge leaves the newly formed set unlabeled, a Labeling Algorithm (LA) must apply. From a minimalist perspective, LA simply consists in minimal search: that is, the set inherits the label of the closest element.

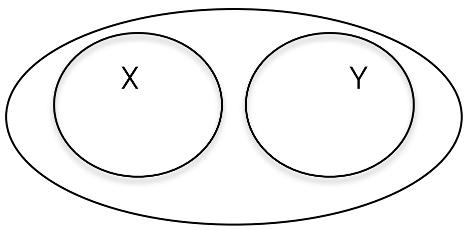
Consider Merge of a head H with a non-head (dubbed XP for convenience), yielding the set {H, XP}. Here XP stands for a set properly containing another head X, hence, more precisely, we have {H, {X, … }}. As the Venn diagram (5) shows, H is a member of the outermost set, whereas X is not: thus, H is clearly closer than X to the outermost set and labels it. This corresponds to the idea that when a head selects a phrase, it is the head that projects (cf. (3) above).

(5)

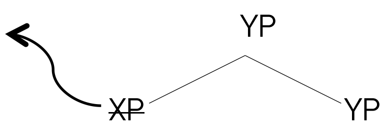


When Merge combines two non-heads XP and YP, instead, a labeling problem arises: the head of neither constituent is closer to the outermost set than the other, as shown in the Venn diagram (6). Thus, LA cannot determine a unique label for the set {XP, YP}.

(6)



It follows that any configuration {XP,YP} in which two non-heads are merged is problematic for the Labeling Algorithm. Chomsky discusses two possible solutions.
The first possibility is that in {XP, YP}, one of the two elements moves away, say XP. (Note that in Chomsky's view labeling is only required at the phase level.) Move/Internal Merge creates another occurrence of XP which is not contained in {XP, YP}. Assuming that a syntactic object α is a term of β iff *all* occurrences of α are contained in β, it follows that after movement, XP is no longer a term of {XP, YP}, and its single occurrence becomes invisible to LA. Thus, the unmoved sister YP univocally transmits its label to the mother set:

(7)


Chomsky suggests that this solution of the labeling paradox via movement of one of two phrasal sisters accounts for the so called Extended Projection Principle, i.e. the fact that when an External Argument is merged with the phrase headed by v, it must move to a higher position. Consider the configuration in (8):

(8) [T [$_{\beta}$ (EA) [$_{\delta}$ v* [V IA]]] (adapted from Chomsky 2013, (17))

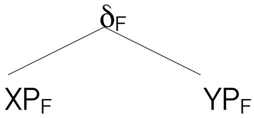
Here the External Argument (EA), a non-head, is merged with the projection δ of v*, which is also a non-head: therefore, the mother set β cannot be labeled. The solution is to move away the External Argument and attach it to the projection headed by T: In this way, β can inherit the label of δ (that is, v*). (Attaching the EA to the projection of T creates another problematic configuration, to be discussed in (13) below.)
A second configuration where solution (7) applies is in case of Internal Merge of a wh-phrase with the constituent labeled by a declarative Complementizer:

(9) * They thought [$_{\alpha}$ [in which Texas city] [$_{\beta}$ C [JFK was assassinated]]]?
(Chomsky 2013, (21))

Here, again, the wh-phrase merged with β creates a set α which cannot be univocally labeled. For this reason, Chomsky argues, the wh-phrase is forced to move away, so that the mother α can be labeled by C. In this way, the requirement of unique and deterministic labeling forces the successive-cyclic movement step, which has always been problematic from the perspective of feature-driven movement.
In (8)-(9), the moved phrase must find a final landing site. Hence, it is necessary to devise an alternative solution to the labeling problem that can license a phrase attached to another phrase – the "halting problem" for successive cyclic movement,

in the terms of Rizzi (2010). Building on previous work by Rizzi, Chomsky proposes that an {XP,YP} configuration can also be licensed when the two phrasal sisters share a prominent feature: the latter can be transmitted as a label to the mother node, as schematically represented in (10).

(10)



Consider for instance the configuration created by Internal Merge of a moved wh-phrase with a root phrase headed by the interrogative Complementizer, C_Q:

(11) [$_{\delta\text{-Q}}$ [in which Texas city]$_Q$ [$_{\gamma\text{-Q}}$ C_Q-did [you think [$_\beta$ ~~in which Texas city~~ [$_\alpha$ C [JFK was assassinated ~~in which Texas city~~]]]]]

While, as discussed above around (9), the problem of labeling β is solved by moving away the wh-phrase, the same problem resurfaces at the next landing site of the wh-phrase, as sister to phrase γ. However, γ is labeled by the interrogative Complementizer, carrying the feature Q, and the very same feature Q is shared by the wh-phrase: thus, Q can be transmitted from both daughters to the mother node δ as a label. The same solution also applies in case of local movement of a wh-phrase attaching to an embedded clause headed by the interrogative C_Q:

(12) They wondered [$_{\alpha\text{-Q}}$ [in which Texas city]$_Q$ [$_{\beta\text{-Q}}$ C_Q [JFK was assassinated]]]
(adapted from Chomsky 2013, (22))

Another configuration in which (10) applies is the landing site of the subject as sister to the phrase headed by T:

(13) C [$_\alpha$ NP TP]] (Chomsky 2013, (16))

Chomsky proposes that TP and the subject NP share prominent features, namely the agreeing ϕ-features, which can be transmitted to the mother node α as a label.
Finally, in addition to (7) and (10), it is necessary to devise a third way to license two phrasal sisters in an "unstructured coordination" like (14), where a number of predicative APs are attached to one another while remaining semantically independent (i.e., each one is individually predicated of the subject):

(14) John is tall, bronde, brown-eyed,....

Clearly, the attachment of multiple APs cannot be solved by movement as in (7), nor is it plausible to assume any kind of feature sharing between them. The solution devised by Chomsky is to analyse the attachment in terms of Pair-Merge – a special instance of Merge creating an ordered pair instead of a set. We will return to structure (14) in section 7.

Options (7) and (10) are also adopted by Rizzi (2016), who redefines in more explicit terms the notion of "closeness" embodied in Chomsky's Labeling Algorithm:

(15) Labeling by the closest head: α created by merge receives the label of head H1 such that:

- I. α contains H1, and
- II. there is no other head H2 such that
  - i. α contains H2, and
  - ii. H2 c-commands H1. (Rizzi 2016, (3))

In addition to movement of a wh-phrase away from a non-criterial landing site, as in (9) above, Rizzi also derives from the labeling algorithm the so called "freezing" effect, namely the fact that a phrase sharing a feature with its sister projection is frozen in place and cannot move away, as shown in (16b):

(16) a. John wonders [α=Q [which$_Q$ book]$_Q$ [Q [Bill read ~~which$_Q$ book~~]]]

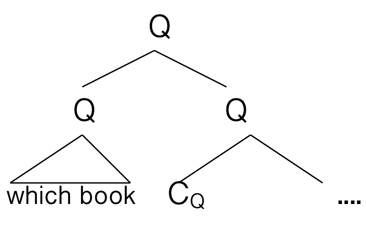
b. *[β [which$_Q$ book] [Q [John wonders [α=Q ~~which$_Q$ book~~ [Q [Bill read ~~which$_Q$ book~~]]]]]] (Rizzi 2016, (23b-c))

The freezing effect is derived from the following Maximality principle:

(17) Maximality: Only maximal objects with a given label can be moved.
(Rizzi 2016, (24))

The idea is the following. In (16b), the intermediate clausal node α inherits the label Q from both the wh-phrase and the phrase headed by C_Q, as schematically represented in (18) below; hence, the wh-phrase no longer qualifies locally as the maximal object with the label Q (since it is dominated by a node with the same label), and by principle (17), it cannot be moved.

(18)



In sum, the Labeling Algorithm can explain the special status of configurations involving two phrasal sisters, distinguishing those in which one phrase is forced to move (e.g. (9)) from those in which both phrases can remain *in situ* via feature sharing (e.g. (11)-(12)), and in fact *must* do so (cf. (16b)).

3. *Why labeling?*

Both Chomsky and Rizzi assume that every node in a syntactic tree must be labeled at the interfaces. This requirement is explicitly stated by Rizzi in the following way:

(19) Uniform labeling: At the interfaces, a tree must be completely labeled.
(Rizzi 2016, (1))

The next question is what motivates such a requirement. Chomsky (2013, 43) proposes that labeling is forced by requirements imposed by the external systems at the interfaces with syntax:[4]

> For a syntactic object SO to be interpreted, some information is necessary about it: what kind of object is it? Labeling is the process of providing that information. Under PSG and its offshoots, labeling is part of the process of forming a syntactic object SO. But that is no longer true when the stipulations of these systems are eliminated in the simpler Merge-based conception of UG. We assume, then, that there is a fixed labeling algorithm LA that licenses SOs so that they can be interpreted at the interfaces, operating at the phase level along with other operations.

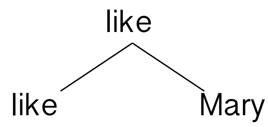
But exactly which interface(s) require labeled nodes?
Consider first the interface with morphology (assuming the Distributed Morphology framework). Chomsky maintains that lexical (open class) items correspond to a-categorial stems which are selected by semi-functional heads (little n, little v, etc.) that assign them a specific categorial status. For instance, the lexical stem LIE can be selected by n, yielding the noun *lie,* or by v, yielding the verb *(to) lie*. The head selecting a lexical stem can thus be seen as determining a specific syntactic category.
If we consider languages that are morphologically more complex than English, such as Italian or Latin, we realize that the semi-functional heads cannot carry just a categorial label, for in such languages a given syntactic category has more than one declension class: therefore, at the interface with morphology the head must provide more specific information than just a syntactic category, and this information must be visible to inflectional heads, such as T. Apart from these heads, it is far from obvious that this interface would require uniform labeling of all nodes, and in particular, of those which dominate two phrasal sisters.
As for the interface with phonology, it is fair to say that at the current stage of theorizing, it is not entirely clear how much syntactic information is required for the syntactic structure to be legible to the phonological component(s). If we assume cyclic spellout, as in e.g. Kratzer & Selkirk (2007), the interface with phonology must be able to identify the complement of each phase head in order for Transfer to apply; apart from the identification of phase heads, it is unclear what use phonological rules would

[4] In general, any principle can be motivated either by the working of the syntactic computation or by external interface requirements (Chomsky 2004).

make of syntactic labels.[5,6] If stress assignment by the Nuclear Stress Rule is sensitive to the depth of embedding – as proposed in Cinque (1993) and Zubizarreta (1998) – the only necessary information is the hierarchical structure, now expressed in terms of set membership: phrase labels do not seem necessary to encode hierarchical relations between nodes. As for the linearization of terminals, if we assumed that linear order is determined by asymmetric c-command between non-terminal nodes (cf. Kayne 1994, Moro 2000), a symmetric configuration like (10) above could not be linearized, while a head-complement configuration like (5) could be linearized even without labeling the mother node. In sum, uniform labeling seems not to be sufficient, and maybe not even necessary, for the structure to be legible at the interface with phonology.
Rizzi (2016, 105) jusifies principle (19) by invoking the interface between syntax and semantics:

> More generally, uniform labeling could be a consequence of interpretive principles, which may need labels to properly interpret structure. Intuitively, this makes sense: a DP, a VP and a CP are interpreted differently, and interpretive principles may be sensitive to the "canonical structural realizations" of semantic types.

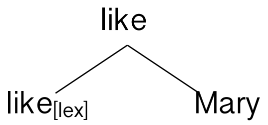
Under this view, category labels are a way of making the syntactic structure legible to type-driven compositional rules. Note that if we want to pursue this idea in a consistent way, we are forced to assume that each syntactic label corresponds to exactly one semantic type. This has non-trivial consequences: for instance, a declarative complement CP and a relative CP must have different labels, since their denotations are of different semantic types (roughly, a proposition vs. a one-place predicate). This difference can only be implemented in a featurally rich system (as in Rizzi 1997, 2010), where a relative CP would be endowed with a distinguished [rel] feature.
Another consequence of this view emerges as soon as we consider the layers below the maximal projection: it turns out that it is impossible for a head to select more than one argument. Consider for instance the predicative phrase *like Mary*:

(20)


Assume that the verb *like* denotes a two-place predicate: the mother node will denote a one-place predicate, since the internal argument position is saturated by the noun phrase *Mary*. If the head and the mother node had exactly the same label, as in (20), the mapping from syntactic labels to denotational types would be one-to-many, *contra* Rizzi's hypothesis. As a matter of fact, Rizzi (2016) argues for independent reasons

[5] According to Bošković (2016, 59-60) this can be obtained if labeling of head-complement merger, as in (5), takes place immediately as part of the syntactic computation, rather than being delayed.
[6] But see Cheng & Dawning (2016) for a different view.

that elements drawn from the lexicon are marked by a feature [lex] which is not inherited by the phrasal mother node. (It is fair to say that the [lex] feature constitutes a departure from inclusiveness.) This will be sufficient to distinguish the head from the mother node, as shown in (21):

(21)


But since presence vs. absence of the [lex] feature is the only available distinction, it is impossible for any head to select more than one argument while projecting phrasal layers: in fact, the layers dominating any arguments beyond the first one would lack [lex], hence they would have the same label but distinct semantic types. Thus, Rizzi's hypothesis requires that any arguments beyond the first one be introduced by separate heads (little a, little n, analogous to little v), forcing the kind of decompositional approach to argument structure that has become common in the minimalist field.[7]

Even though the interface with semantics can enforce uniform labeling (assuming that every node is interpreted), Rizzi's proposal raises one conceptual issue. Under the assumption that every syntactic label corresponds to exactly one denotational type, a syntactic tree can be directly mapped into an LF "decorated" with the denotational type of each node. This kind of enriched LF is indeed assumed in von Stechow (2011: section 4.1), as it allows for straightforward type-driven interpretation. But then, the question arises of whether we really need the nodes to carry both a syntactic label and a denotational type. Recall that by hypothesis, labels are not required by narrow syntax, and as far as the interface is concerned, denotational types seem to be sufficient. Recall also that we have already departed from inclusiveness by distinguishing heads from non-heads, in order to allow for heads to combine with at least one argument (cf. the discussion around (21)). At this point, we may entertain the possibility of doing away with syntactic labels entirely, that is, replace them with the corresponding denotational types. This is the path that I will explore in the remainder of this paper.

4. *Labeling by denotational types*

First of all, let us review the standard inventory of denotational types. This is defined by two basic types (22a,b), a recursive rule yielding complex extensional types (22c), and another rule yielding intensional types (22d):

(22) Recursive definition of denotational types

a. e is a type (*entities*)

b. t is a type (*truth-values*)

c. If σ and τ are types, the ordered pair $\langle\sigma,\tau\rangle$ is a type.

d. if σ is a type, the ordered pair $\langle s, \sigma\rangle$ is an (intensional) type.

e. Nothing else is a type.

[7] I return to this decomposition in section 8.

Each type corresponds to a specific denotational domain:

(23) Recursive definition of denotational domains

a. D_e is a domain of entities (members of the 'universe of discourse')

b. D_t is the set of truth values {0,1} (= true, false)

c. $D_{\langle\sigma,\tau\rangle}$ is the domain of the functions from D_σ to D_τ

d. $D_{\langle s,\sigma\rangle}$ is the domain of the functions from possible worlds/indices to D_σ.

To illustrate, $\langle e,t\rangle$ is the type of functions that take in input entities and return truth values. An $\langle e,t\rangle$ function characterizes a set of entities, i.e. the set of entities for which it returns the value 1. Similarly, t – the type of truth values – is the type of the extension of a simple declarative sentence at a given possible world; the corresponding intensional type $\langle s,t\rangle$ is a function from possible worlds to truth-values, and it can be seen as characterizing the set of possible worlds in which the sentence is true – technically, the *proposition* expressed by the sentence. 2-place predicates are dealt with via function nesting: For instance, a predicate like *love* denotes a function of type $\langle e,\langle e,t\rangle\rangle$, which takes in input an entity *a* and returns the characteristic function (type $\langle e,t\rangle$) of the set of entities that stand in the loving relation with *a*.
Let us now explore a labeling algorithm that directly uses denotational types:

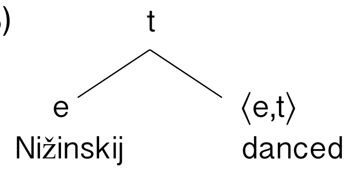
(24) Labeling by denotational types

R1. For terminal nodes (i.e. lexical items), the denotational type is drawn from the lexicon.

R2. For non-terminal nodes: given two daughter nodes $\{\alpha, \beta\}$, if the type of α is an ordered pair $\langle\sigma, \tau\rangle$, and the type of β is the first member of that ordered pair (σ), the mother node has the type corresponding to the second member of that pair (τ).[8]

To illustrate, consider (25) (I use extensional types for the sake of simplicity, and I retain the usual tree format):

(25)

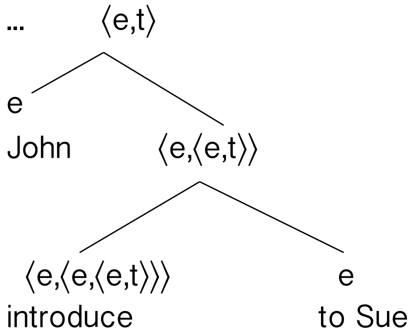


The denotational type of the proper name *Nižinskij* is determined in the lexicon (e – entity); similarly for the type of the one-place predicate *danced* ($\langle e,t\rangle$). When these two syntactic objects are merged together, (R2) applies: the type of the first daughter node

[8] This formulation will do for the combination of extensional types. On intensional types, see note 11.

(e) corresponds to the first element of the pair introduced by the second daughter node (⟨e,t⟩); hence, the mother node will have as denotational type the second element of the pair (t).
The type reduction rule (R2) corresponds straightforwardly to the basic operation of function application in the semantic component: the function of type ⟨e,t⟩ denoted by the predicate, characterizing the set of people who danced, takes in input the entity denoted by the subject and returns a truth-value as the denotation of the whole clause.
Note that, contrary to Rizzi's formulation, labeling by denotational types does not rule out predicates with valence more than 1: whenever a node denotes an n-place predicate and its sister denotes an appropriate input, the mother node will have a denotational type corresponding to an n-1-place predicate. For instance, consider (26):

(26) (Mary) [introduced John to Sue]



Recall that a two-place predicate like *love* is an ⟨e,⟨e,t⟩⟩ function taking in input an entity (e) and returning a one-place predicate (⟨e,t⟩). For a three-place predicate like *introduce*, function nesting is iterated, so its denotation is a function from entities to two-place predicates, i.e. ⟨e, ⟨e, ⟨e,t⟩⟩⟩.
Looking at tree (26), we can see that there is nothing wrong with its labeling: each node carries a well-formed denotational type. In particular, the sister node and the mother node of *John* have different labels, due to the reduction of valency that obtains by (R2) at the level of the mother node: this being the case, nothing forces one of the two phrasal sisters to move away.
With syntactic nodes labeled by types, principle (19) can be restated as follows:

(27) At the interface, each node in a tree must have a well-formed denotational type.

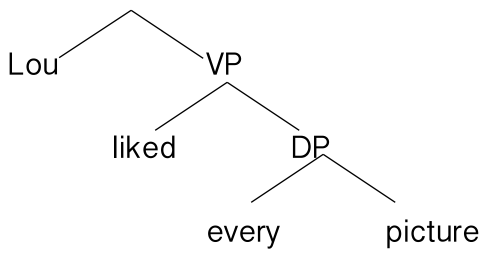
(27) makes the syntactic structure legible to type-driven interpretation rules, namely rules that compose the meaning of two sister nodes simply by looking at their types.
In the following sections, I will explore some empirical consequences of this approach.

5. *Quantifier Raising*

One interesting consequence of labeling by denotational types concerns the covert movement rule of Quantifier Raising.

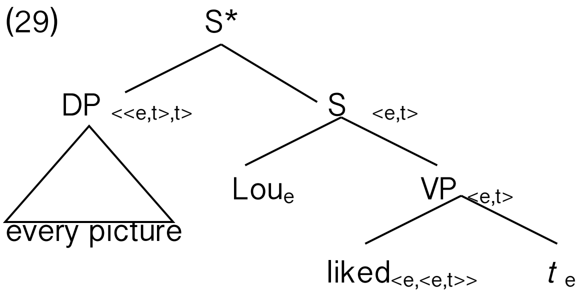
Starting from May's (1977) seminal work, it is commonly assumed that a quantifier is assigned scope by Quantifier Raising. Consider a sentence with a quantificational DP sitting in internal argument position:

(28)



The transitive verb *liked* is a two-place predicate (type ⟨e,⟨e,t⟩⟩) and requires in input an entity. However, the sister DP denotes a quantifier, (the characteristic function of) a family of sets (type ⟨⟨e,t⟩,t⟩) which in turn requires in input a one-place predicate (type ⟨e,t⟩) to return a truth value. This being the case, functional application is impossible in either direction, and the structure is uninterpretable.
May proposed that the structure can be fixed by raising the quantificational DP and adjoining it to the clause:

(29)



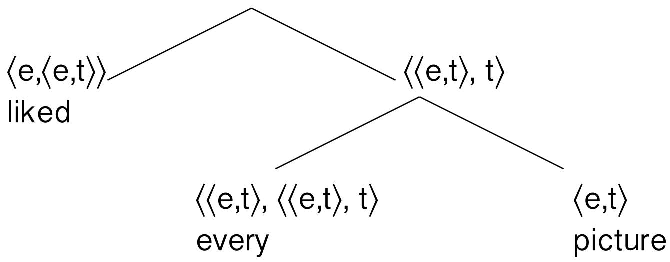
The effects of Quantifier Raising are two-fold. First, the trace left by the DP is interpreted by the compositional rules as a variable ranging over entities of type *e*: this can be the input to the two-place predicate (type ⟨⟨e,t⟩,t⟩), solving the type mismatch problem. Second, the sister node of the attached DP (S) is interpreted via functional abstraction over the variable, thus defining a one-place predicate (type ⟨e,t⟩) which be the input to the quantifier denoted by the DP (cf. Heim & Kratzer 1998, 186-187 for an accessible discussion).
Notice that Quantifier Raising is a syntactic rule, yet it is generally assumed to be triggered by denotational type mismatch, rather than by an active feature involved in a probe-goal relation.[9] Now, type mismatch can only be detected by making reference to the types of two sister nodes: consequently, Quantifier Raising must be seen as an *interface-driven movement*, i.e. a movement that is triggered by an interface

[9] In fact, it is for this reason that Quantifier Raising was considered problematic in the early minimalist framework: see Hornstein (1995), Kennedy (1997).

requirement (cf. Kennedy 1997, 684-685; Reinhart 1995/2006 and Fox 2000 have defended this view and provided empirical arguments in support of it.)[10] The upshot is that Quantifier Raising is exceptional in two respects: (i) it is invisible to the interface with phonology; (ii) it is triggered from outside the narrow syntactic component. This makes its assimilation to the general rule Move/Internal Merge somewhat less compelling.

If, however, syntactic nodes are labeled by denotational types, the type mismatch can be directly detected within the syntactic component. The VP of tree (29) will correspond to (30):

(30)



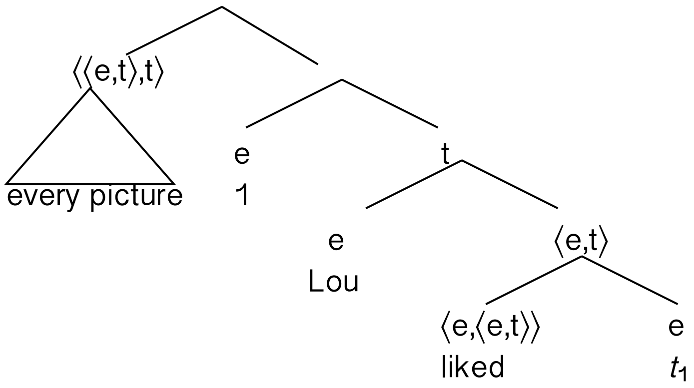
Here, the node immediately dominating the transitive verb and the ⟨⟨e,t⟩,t⟩ quantifier cannot be labeled by the reduction rule (R2), due to type mismatch. Exactly as in (7) above, the problem is solved by movement of one of the two phrasal sisters: the ⟨⟨e,t⟩,t⟩ node moves away and leaves in the argument position a trace with label e, whose mother node can be properly labeled by an application of (R2). The relabeling of the DP trace with type e can be seen as a direct consequence of the independently required Trace Conversion rule of Fox (2000), which converts the trace into a definite description.[11]

The Quantifier Raising structure requires one more labeling rule: this is because, as discussed above, the sister node of the quantifier must be lifted to type ⟨e,t⟩ in order to be able to compose with the quantifier. On standard assumptions, this type adjustment is triggered by the presence of an index of type e attached to the sister node of the quantifier:

[10] See also Zubizarreta (1998) for the proposal of syntactic movement driven by the interface with prosodic phonology.

[11] This rule converts the copy left behind by the quantificational DP into a definite description, retaining the quantifier's restriction conjoined with an indexed trace:
(a) Variable Insertion: (Det) Pred -> (Det) [Pred $\lambda y(y=him_n)$ (where n is the index of the moved quantifier);
(b) Determiner Replacement:(Det) [Pred $\lambda y(y= him_n)$] -> the [Pred $\lambda y(y=him_n)$]

(31)



We need a third labeling rule (R3) to cope with this configuration:

(32) R3. Given two daughter nodes $\{\alpha,\beta\}$, if α contains an index of type e and the type of β is t, the mother node has type ⟨e,t⟩.

An application of (R3) will label the sister node of the quantifier as ⟨e,t⟩; finally, the root node can be labeled by another application of (R2).
One more consequence is worth considering. It is generally assumed that Quantifier Raising must target a node with type t, so that functional abstraction on its trace will yield a one-place predicate (see Heim & Kratzer 1998, 214 ff. for extensive discussion). Once again, if denotational types are not available to the syntactic computation, this constraint on Quantifier Raising must be determined at the interface with semantics; on the other hand, the constraint can be purely syntactic if the tree nodes are labeled by denotational types.

6. *Wh-chains*

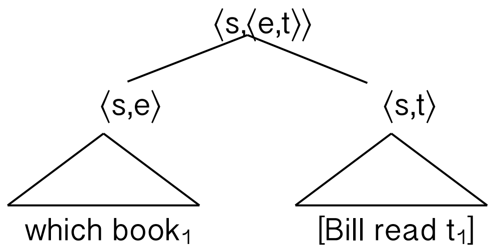
One important result of the Chomsky-Rizzi approach to labeling (section 2) is the principled explanation of successive cyclic movement of a constituent through intermediate chain links and, conversely, freezing in the "criterial" position where the moved phrase shares a prominent feature with its sister node. Let us consider the predictions of labeling by denotational types for interrogative wh-chains.
Since we want the type labels to correspond to what is needed by the semantic compositional rules, the proper labeling for wh-chains depends on the choice of a specific semantic analysis for interrogative clauses. For the sake of concreteness, I assume here the functional approach to the semantics of interrogatives, which will allow us to maximize the use of the labeling rules already introduced.
The basic tenets of the functional approach are the following: in a simple wh-question,
(i) the wh-phrase introduces functional abstraction over its trace (i.e. the trace is interpreted as a variable of type *e* bound by a lambda operator);

(ii) the lexical restriction of the wh-phrase is interpreted as a presupposition that restricts the domain of that function;[12]

(iii) the denotation of the question is then (the intension of) a propositional function which takes in input a denotation of the type of the bound trace/variable, and returns a truth value.

In a wellformed interrogative clause, as in (33), the wh-phrase is internally merged with the constituent headed by the interrogative complementizer Q and lifts the denotation of the mother node α to the appropriate functional type. This result can be obtained by an application of rule (R3),[13] assuming that the indexed daughter node *which book*$_1$ bears the label $\langle s,e \rangle$ (the intensional counterpart of e):

(33) John wonders [α [which$_Q$ book]$_1$ [Q [Bill read ~~which book~~$_1$]]]



Consider now what happens when the wh-trace moves to the Specifier of a declarative Complementizer, as in (34) (this example is parallel to (9) above):

(34) * John [$_\beta$ thinks [α [which$_Q$ book]$_1$ [C$_{decl}$ [Bill read ~~which book~~$_1$]]]]

Here, the matrix propositional attitude verb *think* selects for a proposition (type $\langle s,t \rangle$): its (extensional) type is $\langle\langle s,t \rangle, \langle e,t \rangle\rangle$ (as determined in its lexical entry, cf. (R1)). On the other hand, the indexed wh-phrase attached to α triggers the application of rule (R3), so that α will be of type $\langle s,\langle e,t \rangle\rangle$, i.e. the functional question type (as in (33) above). Consequently, the mother node β cannot be labeled: given the two type labels $\langle\langle s,t \rangle, \langle e,t \rangle\rangle$ and $\langle s,\langle e,t \rangle\rangle$, there is no way to apply type reduction. This accounts for the ill-formedness of (34).[14]

---

[12] See e.g. Krifka (2011, 1753, (55b)).

[13] The intension we want to obtain is of type $\langle s,\langle e,t \rangle\rangle$ (s being the type of possible worlds/indices): a function from possible worlds to one-place predicates. The intensional version of (R3) can be formulated as follows: R3′. Given two daughter nodes $\{\alpha, \beta\}$, if α contains an index of type $\langle s,e \rangle$, and the type of β is $\langle s,t \rangle$, the mother node has type $\langle s, \langle e,t \rangle\rangle$.

[14] A type mismatch would also arise if we assumed instead the proposition set approach to interrogative clauses. In this approach, the wh-phrase would lift the type of the embedded declarative clause α to a set of propositions, while the selecting verb requires its sister to bear the type of a single proposition in order for type reduction to be possible.

Recall that a structure like (34) can be rescued if the wh-phrase moves further and attaches to the projection of a matrix interrogative complementizer, as in (35) (which is parallel to (10) above):

(35) [δ [which$_Q$ book] [Q [John [~~[which$_Q$ book]~~ [γ thinks [β ~~[which$_Q$ book]~~ [α C$_{decl}$ [Bill read ~~[which$_Q$ book]~~]]]]

The root node δ, immediately dominating the highest occurrence of the wh-phrase, will receive the label ⟨s,⟨e,t⟩⟩) by (R3). On the other hand, the embedded clause β must bear the label ⟨s,t⟩ (proposition) selected by the matrix verb *think*, in order for type reduction (R2) to be possible, so as to label the verb phrase γ. To this effect, it is necessary to stipulate that intermediate traces of wh-chains are invisible to labeling: in other terms, the intermediate trace will *not* trigger an application of (R3) at the level of the embedded clause β. β will thus inherit the ⟨s,t⟩ type of the other daughter node α.
Despite the resemblance to (7) above, note that the stipulated invisibility of intermediate traces cannot extend to the lowest trace of the movement chain: the latter must be interpreted at the interface as a bound variable. One crucial difference is that the lowest trace is contained in the domain of a phase head, while the higher traces are in phase edges. Assume that the labeling of a phase edge is delayed up to the next cyclic domain. In (35), the labeling of the embedded clause β (whose edge contains an occurrence of the wh-phrase) is delayed until the verb phrase γ is completed (assuming, for concreteness, that γ is the next phase / cyclic domain): at this point, the labeling mismatch that would be caused by applying rule (R3) at the level of β can be detected; therefore, the wh-phrase is forced to move further to the edge of γ, and the intermediate trace attached to β is ignored by the labeling algorithm. The same mechanism will also account for further movement from the edge of the verb phrase γ to the edge of the root node δ; at this point, the application of (R3) will not cause any type mismatch, and δ will be labeled by the question type ⟨s,⟨e,t⟩⟩.
Finally, let us reconsider the ill-formedness of (16b) above, repeated here as (36):

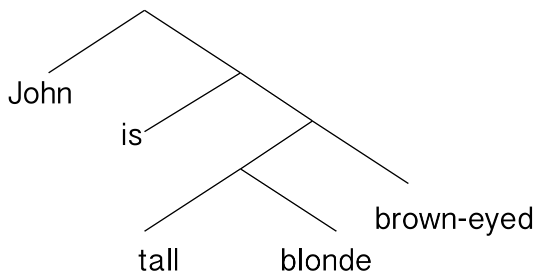
(36) *[γ [which$_Q$ book] [Q [John [β wonders [α ~~[which$_Q$ book]~~ Q [Bill read ~~[which$_Q$ book]~~]]]

Here the matrix verb *wonder* selects for a questions type; therefore, at the level of the cyclic node β, the embedded clause can be labeled by an application of (R3). This being the case, labeling does not trigger any further movement of the wh-phrase; this accounts for the underivability of (36).
In sum, labeling by denotational types can express the basic insight of the Chomsky-Rizzi labeling approach to wh-chains: a wh-phrase cannot stop in an intermediate position where an application of rule (R3) would yield a detectable type mismatch, whereas it can – and in fact, must – stop in a position where the functional type yielded by an application of (R3) is legal, that is, either at the root or at the edge of an interrogative clause appropriately selected by the matrix verb.

7. *Unstructured coordination*

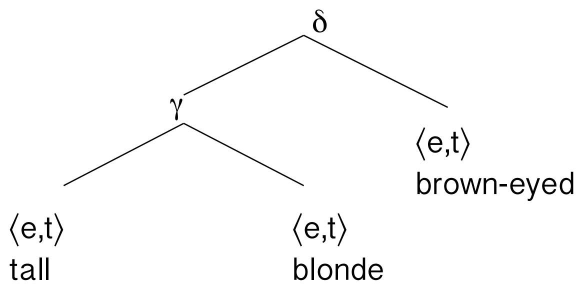
Recall from (14) above that Chomsky takes unstructured coordination to be problematic for the labeling algorithm, since it involves the attachment of two phrases neither of which moves or shares a prominent feature with the other, as exemplified in (14), repeated here as (37):

(37) John is tall, blonde, brown-eyed,...



Chomsky proposes that unstructured coordination is an adjunction configuration involving Pair-Merge. Let us reconsider this configuration in the light of labeling by semantic types:

(38)



Each adjective is a one-place predicate of (extensional) type ⟨e,t⟩. We also know what the type of the coordination node δ should be: an ⟨e,t⟩ type, which can be saturated by *John*. To this effect, let us assume the following labeling rule (R4):

(39) R4 (to be revised). Given two daughter nodes {α,β}, if the type of both α and β is ⟨e,t⟩, the type of their mother node is ⟨e,t⟩ as well.

(R4) bears some resemblance to the Chomsky-Rizzi labeling principle whereby, when two daughter nodes share a prominent feature, the mother node is labeled by the shared feature (cf. (10) above); here, two daughters with an identical type label transmit this label to the mother node. In (38), node γ is labeled ⟨e,t⟩ by an application of (R4), and node δ is also labeled ⟨e,t⟩ by (R4). The iterability of application accounts for the fact that unstructured coordination allows for an unlimited number of stacked predicates.

At the interface with semantics, the configuration so labeled can be interpreted by a compositional rule like Heim & Kratzer's (1998, 65) Predicate Modification, which creates out of two one-place predicates a single, complex one-place predicate.

Summing up, the proposed labeling algorithm consists of four rules (R1-R4), plus the assumed invisibility of intermediate movement trace. Note that, even though these rules are not reducible to minimal search, they only make reference to the labels of the two daughter nodes: hence, they do not require a more powerful search algorithm than Chomsky's minimal search. We have seen that this proposal offers a conceptual advantage over labeling by syntactic categories, in that it allows Quantifier Raising to be triggered within narrow syntax. In the following section, I will provide an empirical argument in support of it, showing that it allows for a maximally simple analysis of depictive predicates.

8. *Depictive predicates*

Depictive secondary predicates constitute an interesting case study for the labeling algorithm, since they involve attachment of a phrase (the secondary predicate) to another phrase (the main predicate).

A preliminary question to be addressed concerns the attachment site of depictives. Two empirical arguments suggest a low attachment site. First, in Italian a depictive predicate can be predicated of a post-verbal subject in a broad focus sentence, as in (40):[15]

(40)	E' arrivato/ comparso	Nižinskij	vestito	con un costume esotico.
	is arrived/ appeared	Nižinskij	dressed	with a costume exotic
	'There arrived/ appeared Nijinsky wearing an exotic costume.'			

It can be shown that such post-verbal subjects (in declarative clauses) are syntactically low (see Cecchetto 1999 for one detailed argument based on reconstruction effects). Given the standard hypothesis that predication requires c-command, we are led to conclude that in (40) the secondary predicate must be attached to a low projection.

Second, we observe obligatorily disjoint reference between a cliticized argument of the main predicate and an R-expression contained in the depictive predicate:

(41) a. *	Gli_i	è apparso	Nižinskij	[travestito	da $Djagilev_i$].
	him.DAT	is appeared	Nižinskij	disguised.M.SG.	as Djagilev

[15] The licensing of post-verbal subjects under broad focus is constrained by the requrement that the sentence must introduce a temporally and spatially delimited "topic situation" (cf. Klein 2008 for discussion). This condition is satisfied in our examples.

b. * Lo$_i$ inseguiva Nižinskij [travestito da Djagilev$_i$].
him.ACC followed Nižinskij disguised M.SG as Djagilev

Disjoint reference can be analysed as a Principle C effect if the clitic (by hypothesis licensed in an inflectional layer of the clause) c-commands the depictive.
Depictive secondary predicates have different interpretations depending on the nature of the main predicate. With change of state predicates, the state described by the depictive predicate holds in concomitance with the result state of the matrix clause event (cf. Rothstein 2000, 253-255):

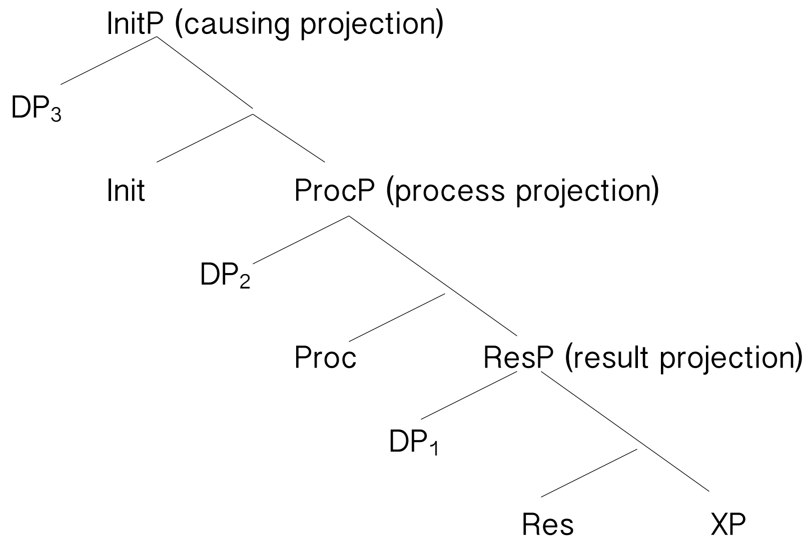
(42) Nižinskij è arrivato/ uscito/comparso vestito con un costume esotico.
Nižinskij i s arrived/gone-out/appeared dressed with a costume exotic
'Nijinksy arrived/went out/appeared wearing an exotic costume.'

With activity verbs, instead, the state described by the depictive predicate holds throughout the process phase of the event:

(43) Nižinskij danza/ dorme/ vaga vestito con un costume esotico.
Nižinskij dances/ sleeps / wanders dressed with a costume exotic
'Nijinsky is dancing/sleeping/wandering wearing an exotic costume.'

The decompositional approach to event structure proposed by Ramchand (2008) allows us to straighforwardly characterize this difference:

(44)



(Ramchand 2008, 39, (1))

In this approach, the topmost projection encodes the causing eventuality (a state) and the argument related to it (DP$_3$) is the INITIATOR of the eventuality. The central projection encodes the dynamic part of the event (the process) and licenses the

argument undergoing the process (DP_2, dubbed UNDERGOER). Finally, the lowest projection encodes the result state (if any) and licenses the argument that comes to hold the result state (DP_1, dubbed RESULTEE).
The basic rule of event composition[16] expresses the 'leads to' relation connecting the subevents within a complex eventuality:

(45) Event Composition Rule (Ramchand 2008, 44, (5))
e = e1→ e2: e consists of two subevents e1, e2, such that e1 causally implicates e2.

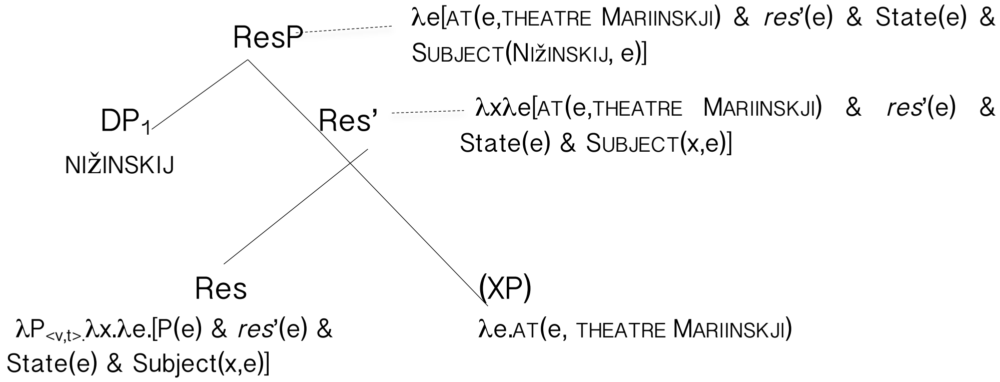
(46) Res = λPλxλe[P(e) & *res*'(e) & State(e) & Subject(x,e)]

(47) Proc = $\lambda P \lambda x \lambda e \exists e_1, e_2$[P($e_2$) & *proc*'($e_1$) & Process($e_1$) & e=($e_1 \rightarrow e_2$) & Subject (x,$e_1$)]

(48) Init = $\lambda P \lambda x \lambda e \exists e_1, e_2$[P($e_2$) & *init*'($e_1$) & State($e_1$) & e=($e_1 \rightarrow e_2$) & Subject (x,$e_1$)]

For a change of state verb like *arrive* in (40)/(42), we have the following structure (I retain the usual X-bar notation for convenience):

(49)



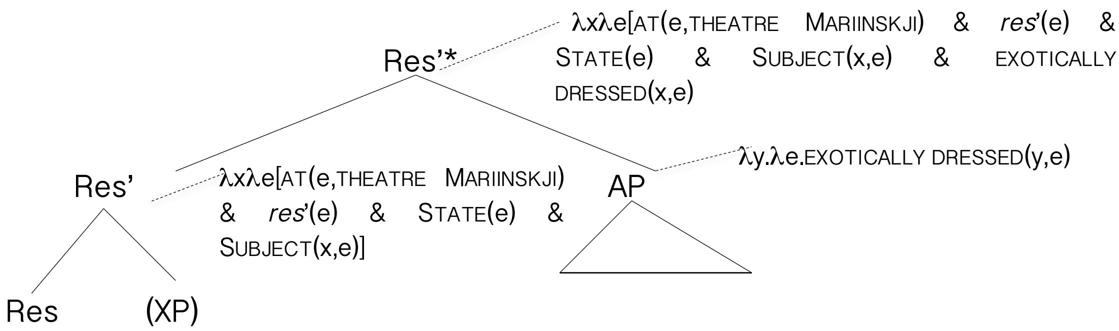
Let us assume for concreteness that the implicit Locative argument introduces the set of states of being at Theatre Marinskji. The Res head denotes a function that takes in input this set of states (type ⟨v,t⟩, where v is the type of event(ualitie)s) and introduces the valence for a Subject; this valence is then saturated by merging the resultee DP in Spec,ResP.
Let us now consider how a depictive predicate can be merged in such a structure. The simplest analysis for a depictive like "exotically dressed" in (42) is that it denotes a relation between an individual and a stative eventuality (cf. Rothstein 2000), of type ⟨e,⟨v,t⟩⟩:

(50) λy_e. λe_v.EXOTICALLY-DRESSED(y,e)

[16] Ramchand (2008) uses 'event' rather than 'eventuality' as a general cover term. I retain her terminology in reporting her definitions.

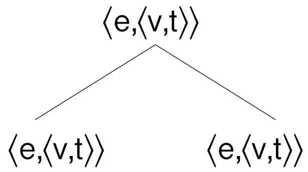
Note now that the intermediate projection Res' of (49), with the subject role yet unsaturated, has exactly the same semantic type as the depictive predicate of (50), namely a relation between an individual and an eventuality. We can therefore assume that the depictive predicate adjoins to this node *before* the subject argument is merged, as shown in (51). The two functions, being of a conjoinable type (⟨e,⟨v,t⟩⟩), receive a conjunctive interpretation, whereby they are incorporated in the value description of a single function (again of type ⟨e,⟨v,t⟩⟩) at the level of the mother node (Res'*).

(51)



From the viewpoint of labeling by denotational types, the adjunction configuration (51) can be characterized as follows:

(52)



This configuration is parallel to that of unstructured coordination in (38) above: two daughter nodes bear the same conjoinable type ("ending in t"),[17] and the mother node receives exactly the same label; the configuration is interpreted at the interface via generalized conjunction (in the sense of Partee & Rooth 1984). In order to cover both (38) and (52), we can generalize rule (R4) as follows:

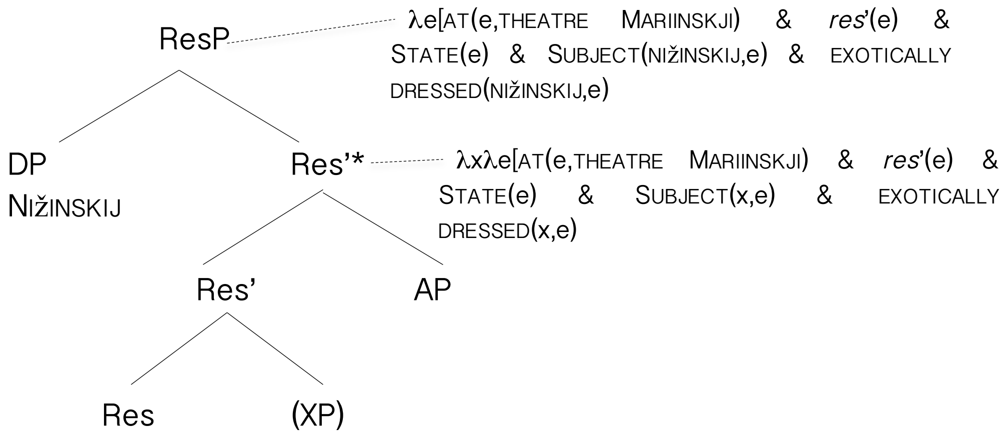
(53) R4 (final). Given two daughter nodes {α,β}, if α and β bear the same conjoinable type, the mother node receives the same type as well.

---

[17] Conjoinable types (definition):
i. t is a conjoinable type
ii. if b is a conjoinable type, then for all a, ⟨a,b⟩ is a conjoinable type. (Partee & Rooth 1984, (4)

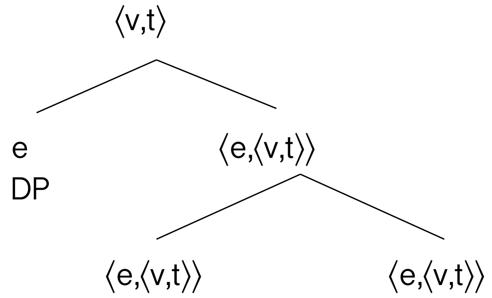
When the Resultee is merged with (51), it simultaneously becomes the subject of the result state and of the depictive predicate (by converting the x variable):

(53)



Note that (53) is wellformed from the labeling viewpoint: the DP merged with the adjunction structure is of type e, so that the type reduction rule (R2) can apply, yielding a wellformed denotational type ⟨v,t⟩ for the mother node:

(54)

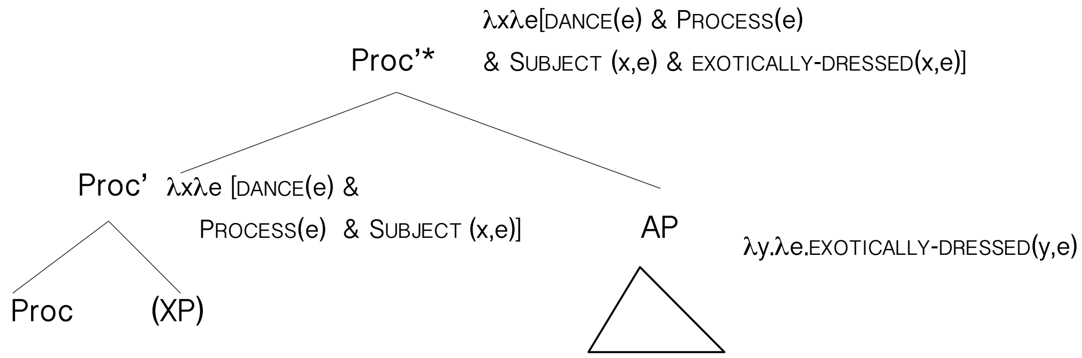


Thus, labeling by denotational types does not require an external argument to move away, contrary to Chomsky's LA (cp. the discussion around (8) above).[18] More generally, whenever an argument saturates the valence of a predicate, (R2) can apply unproblematically.

The conjunctive analysis proposed above for change of state verbs can also be adopted for depictive secondary predicates combining with an activity verb, as in (43). Let us assume that the relevant activity verbs, like *dance* or *perform*, do not include a Result Phrase in their event structure, and that the complement of Process is a silent cognate object; the depictive predicate adjoins to the intermediate projection of the Process head (cf. Ramchand 2008, 104):

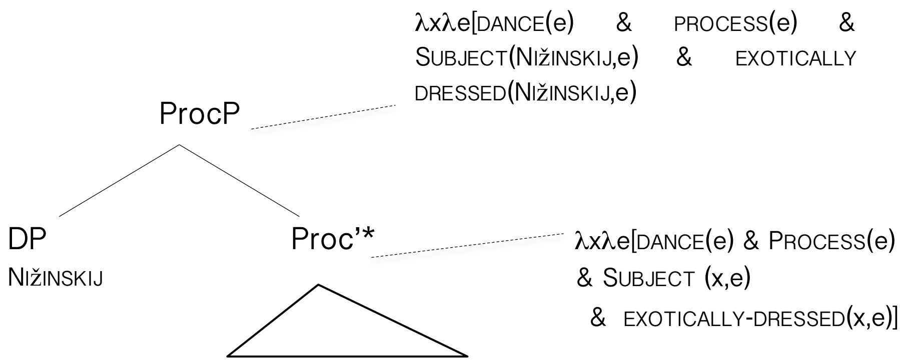
[18] From this perspective, movement to Spec,TP is triggered by a requirement independent of labeling (cf. Rizzi& Shlonsky's 2007 Subject Criterion).

(55)



Again, the main and the secondary predicate are interpreted as conjoined, so that the secondary predicate is simultaneous to the process phase of the event; when the Undergoer is Merged in Spec,ProcP, it saturates the complex predicate:

(56)



Concerning the internal structure of the secondary predicate, Legendre (1997) argues that it contains a PRO, on the grounds that it behaves as a minimal domain for binding purposes: in (57a), it can contain an anaphor bound by the matrix object, but not a pronoun coreferent with it:

(57) a. Pierre a vu Marie$_i$ [PRO$_i$ satisfaite d'elle-meme$_i$/*d'elle$_i$]
Peter has seen Mary satisfied.F.SG of herself / *of her
b. Pierre$_j$ a vu Marie$_i$ [PRO$_i$ satisfaite de lui$_j$/* lui-meme$_j$]
Peter has seen Mary satisfied.F.SG of him/*himself
(French; Legendre 1997, 54, (17))

PRO acts as a local antecedent for the anaphor and qualifies the predicate's projection as a minimal binding domain. PRO also acts as a local goal for the Agree relation involving the gender and number features of the depictive predicate.

From the present perspective, nothing precludes this syntactic analysis, provided that PRO is interpreted as an operator implementing lambda-abstraction over the predicate's external argument position. On the other hand, it is also possible to do away with PRO if we modify our assumptions about binding and agreement. Concerning binding, we might adopt Kratzer's (2009, 193-194) proposal that the binder for the anaphor is the closest c-commanding head which introduces an external argument: in (57), the two-place predicative Adjective will carry an index binding the anaphor.[19] Kratzer also argues that the anaphor binder must be endowed with phi-features to be transmitted to the anaphor; this condition is satisfied here, since the secondary predicate carries agreement features.

Concerning the agreement mechanism itself, we might assume that the secondary predicate Agrees with a DP that is in a minimal configuration with it, according to the following definition:

(59) X is in a *minimal configuration* with Y with respect to local relation R only if there is no Z such that
i. Z c-commands Y and Z does not c-command X, and
ii. Z is of the same type as X with respect to R.
(Rizzi 2016, 106, (4))

In (53), both the Res head and the head of the adjoined AP are in a minimal configuration with the subject DP; therefore, the AP Agrees with the DP. The same assumption will also account for agreement in (56).

In sum, the hypothesis that syntactic adjunction of depictive predicates is interpreted via generalized conjunction can account for the relation of the depictive with different subevents in the case of change of state vs. activity verbs. This maximally simple analysis is compatible with labeling by denotational types, whereas it is problematic for the Chomsky-Rizzi labeling algorithm, which would need to have recourse to Pair-Merge (cf. the discussion following (14) above).

9. Concluding remarks

The labeling algorithm proposed by Chomsky (2013), when justified by a principle of legibility at the interface with semantics (Rizzi 2016), virtually imposes a many-to-one correspondence between syntactic labels and denotational types. In this paper I have explored the view that denotational types may directly replace syntactic labels; the label of a mother node results from the interplay of the labels of the two daughters, via a limited set of labeling rules which have well-defined correlates in the compositional interpretation process.

This view still allows for an account of movement out of intermediate (non-criterial) chain links, and of the freezing effect affecting criterial positions. Contrary to the Chomsky-Rizzi algorithm, however, it does not force movement of an argument merged with a predicative phrase, nor does it impose any limit to the number of arguments that can be licensed within a predicate's maximal projection.

---

[19] Kratzer (2009: 194 ff.) actually claims that only a verbal (little v) head can carry a binding index, but an example like (57) forces us to extend this possibility to adjectival heads.

Two novel consequences are worth stressing. First, under labeling by denotational types, the type mismatch caused by a quantifier merged in an internal argument position can be directly detected within the narrow syntactic component, triggering Quantifier Raising as a repair strategy, without any additional stipulations. Second, two phrasal sisters with an identical conjoinable type can transmit that type as the label to the mother node, with a concomitant conjunctive interpretation: this accounts for Chomsky's "unstructured coordination" of one-place predicates and also provides a maximally simple analysis for depictive secondary predicates.

One limitation of this labeling system is that it has nothing to say about the interface between syntax and morphology. As noted above, morphological processes cannot exploit such general labels as syntactic categories or, for that matter, denotational types; I will tentatively assume that they make reference to the actual content of terminal nodes plus containment relations.

An issue that I cannot address here is how the proposed adjunction configurations can determine a unique linearization of the terminal nodes. As for merger of the external argument, as in (53)-(54) and (56), one possible solution would be to encode the direction of functional application in the types themselves: however, this move would give up the idea that hierarchical relations determine the linear order of terminal nodes (Kayne 1994, Moro 2000), and moreover, it would not extend to the fully symmetric adjunction configurations in (51)-(52) and (55). It seems to me, however, that labeling by denotational types is in no worse position than the Chomsky-Rizzi labeling system with respect to linearization. Another aspect of the problem is whether to spell out the highest or the lowest link of a movement chain (see Bobalijk &Wurmbrandt 2012). These issues remain open for future research.

References

Belletti, A. 2004. Aspects of the low IP area. In *The Structure of CP and IP,* ed. L. Rizzi, 16-51. New York/Oxford: Oxford University Press.

Bobalijk, J. & S. Wurmbrandt. 2012. Word Order and Scope: Transparent Interfaces and the ¾ Signature. *Linguistic Inquiry* 43, 371-421.

Bošković, Z. 2016. On the timing of labeling: Deducing Comp-trace effects, the Subject Condition, the Adjunct Condition, and tucking in from labeling. *The Linguistic Review* 33, 17–66.

Cecchetto, C. 2000. Doubling Structures and Reconstruction. *Probus* 12, 1-34.

Cecchetto, C. & C. Donati. 2015. *(Re)labeling*. Cambridge, Mass., MIT Press.

Cheng, L. & L. Downing. 2016. Phasal Syntax = Cyclic Phonology? *Syntax* 19, 156-191.

Chomsky, N. 1995. *The Minimalist program*. Cambridge, Mass., MIT Press.

Chomsky, N. 2004. Beyond explanatory adequacy. In *Structures and beyond: the cartography of syntactic structures*, vol. 3, ed. A. Belletti, 104-131.

Chomsky, N. 2013. Problems of projection. In *Lingua*, 130, Special Issue "Core ideas and results in syntax", ed. by Luigi Rizzi, 33-49.

Cinque, G. 1993. A null theory of phrasal and compound stress. *Linguistic Inquiry* 24,

239–297.
Fox, D. 2000. *Economy and semantic interpretation*. Cambridge, Mass.: MIT Press.
Heim, I. & A. Kratzer. 1998. *Semantics in generative grammar*. Oxford: Blackwell.
Hornstein, N. 1995. *Logical Form: From GB to Minimalism*. Oxford: Blackwell.
Kayne, R. 1994. *The Antisymmetry of Syntax*. Cambridge, Mass: MIT Press.
Kennedy, C. 1997. Antecedent-Contained Deletion and the syntax of quantification. *Linguistic Inquiry* 28, 662-688.
Klein, W. 2008. The topic situation. In *Empirische Forschung und Theoriebildung: Beiträge aus Soziolinguistik, Gesprochene-Sprache- und Zweitspracherwerbs-forschung: Festschrift für Norbert Dittmar*, eds. B. Ahrenholz, U. Bredel, W. Klein, M. Rost-Roth & R. Skiba, 287-305. Frankfurt am Main: Lang.
Krifka, M. 2011. Questions. *Semantics: an international handbook of natural language meaning, Vol I,* eds. C. Maienborn, K. Von Heusinger, P. Portner, 1742-1785.
Kratzer, A. 2009. Making a pronoun: fake indexicals as windows into the properties of pronouns. *Linguistic Inquiry* 40, 187–237.
Kratzer, A. & E. Selkirk. 2007. Phase theory and prosodic spellout: the case of verbs. *The Linguistic Review* 24, 93-135.
Legendre, G. 1997. Secondary predication and functional projections in French. *Natural Language and Linguistic Theory* 15, 43-87.
May, R. 1977. *The grammar of quantification*. Doctoral dissertation, MIT.
Moro, A. 2000. *Dynamic antisymmetry.* Cambridge, Mass.: MIT Press.
Ramchand, G. 2008. *Verb meaning and the lexicon: A first phase syntax*. Cambridge, Mass.: MIT Press.
Reinhart, T. 1995/2006. *Interface strategies.* Cambridge, Mass.: MIT Press.
Rizzi, L. 1997. The fine structure of the left periphery. In *Elements of grammar. Handbook in generative syntax*, ed. L. Haegeman, 281–337. Dordrecht: Kluwer.
Rizzi, L. 2010. On some properties of criterial freezing. In *The Complementizer Phrase*, ed. E. P. Panagiotidis, 17-32. New York/Oxford: Oxford University Press.
Rizzi, L. 2016. Labeling, maximality, and the head-phrase distinction. *The Linguistic Review* 33, 103–127.
Rizzi, L. & U. Shlonsky. 2007. Strategies of subject extraction. In *Interfaces + recursion = language? Chomsky's minimalism and the view from syntax-semantics,* eds. H. M. Gärtner & U. Sauerland, 115-160. Berlin: Mouton de Gruyter.
Rothstein, S. 2000. Secondary predication and aspectual structure. *ZAS Papers in Linguistics* 17, 241-264.
von Stechow, A. 2011. Syntax and semantics: an overview. In *Semantics: an international handbook of natural language meaning, Vol 3*, eds. C. Maienborn, K. von Heusinger & P. Portner, entry 82.