Right Node Raising, Co-ordination and The Dynamics of Language Processing

Ronnie Cann, Ruth Kempson, Lutz Marten, Masayuki Otsuka
Edinburgh, King's College London, SOAS

1 Right Node Raising

Although reportedly a phenomenon that is rare in spontaneous spoken language, Right Node Raising is surprisingly consistent in its properties across those languages that exhibit the construction. We illustrate the phenomena in (**??**) with data from English. The basic construction is exemplified in (**??**a,b) showing a rightward dependency into the VPs of two conjoined constituents. (**??**c) shows that there can be more than one right dislocated expression, giving rise to apparent non-constituent co-ordination, while (**??**d) shows that the dependency can be into a strong island. The examples in (**??**e,f) reveal the strong relation between the properties of the second conjunct and the right dislocated expression which include the licensing of negative polarity items and the satisfaction of selectional properties.

(1) a. Syntax students dislike, or at least barely tolerate, four hour exams.
  b. John wants to visit, but has forgotten how to contact, his aunt.
  c. John passed on, and Harry distributed, Ruth's lecture notes to anyone that asked for them.
  d. John wants to buy, and Sam knows the name of someone who is willing to sell, a 1950's Jaguar.
  e. John has read, but he hasn't understood, any of my books.
  f. Fiona wanted, but Bill wouldn't let her, (*to) eat chocolate.

Apart from the requirement for a dislocated expression to have a dependency into all conjuncts (**??**a), left dislocation from conjoined expressions shows discrepant properties from their right dislocated counterparts. Thus, multiple left dislocated expressions are not acceptable (in English) (**??**b);

a left dislocated expression cannot have a dependency into a strong island (??c); and negative polarity items are not licensed on the left (??d).

(2) a. His aunt, John wants to visit, but has forgotten how to contact.

b. *Ruth's lecture notes to anyone that asked for them, John passed on, and Harry distributed.

c. *A 1950's Jaguar, John wants to buy, and Sam knows the name of someone who is willing to sell.

d. *Any of my books, John has read, but he hasn't understood.

In addition, right dislocation may permit a dependency of a sort not licensed by a left dislocated expression. Such a situation is reported in McCloskey (1986) with respect to Modern Irish, where left dislocation does not allow preposition stranding, but Right Node Raising does.

These data are notoriously recalcitrant to straightforward analysis and it is notable that discussion of them is rather thin on the ground. All the analyses that we are aware of[1] find some aspect of the construction difficult to incorporate into the theory they propose. Although it is not possible to give a full account of Right Node Raising in a short paper, we show below how modelling the process of assigning an interpretation to a string in context as a left-right process of tree growth, provides an explanatory account of Right Node Raising that captures directly the asymmetry between it and its left dislocated counterparts noted above.

The central intuition behind our analysis is that the characteristic intonation associated with Right Node Raising licenses the postulation of a

---

[1] See particularly Hartmann (1998), Postal (1998), Levine (2001), McCawley (1988). The one exception is Steedman (1996) but his account involves a concept of syntactic category exactly as rich as is required to reflect the surface nonconstituent coordination of Right Node Raising so the descriptive success of the analysis is not surprising. Moreover his account faces the problem of expecting symmetry between left- and right-dislocation.

'null pronominal' at the 'gapsite' in each conjunct. General properties of conjoined expressions are responsible for ensuring that these pronominals are 'co-referential' and a general rule licenses right dislocated expressions whose dependency properties are determined by the verb in the final conjunct. Under these assumptions we show how the problematic properties noted above are straightforwardly explained.

2 Dynamic Syntax

The framework we adopt is that of *Dynamic Syntax* (Kempson et al. 2001). This theory models the process of natural language understanding as a monotonic tree growth process defined over the left-right sequence of words, with the goal of establishing some propositional formula as interpretation. Taking information from words, pragmatic processes and general rules, the theory derives partial tree structures that represent the content of a string as interpreted in context up to the current point in the parse. Intrinsic to this process are concepts of underspecification whose resolution is driven by requirements which determine the process of tree growth, because of the need to satisfy them in order for a parse to be successful.

To get the model of the PROCESS of establishing such a structure as interpretation, all nodes in the semantic trees constructed during a parse are introduced with requirements to be fulfilled, reflecting the idea that the tree is underspecified with respect to some property that needs to be specified as the parse proceeds. Requirements are shown as question marks before some annotation and may appear with any of the labels that decorate a node. They drive the parsing process because a string is defined as wellformed if (and only if) at least one logical form can be constructed from the words in sequence with no requirements outstanding. In consequence, as we shall see, the imposition of requirements and their subsequent satisfaction are central to explanations to be given. The exposition in this paper is not formal and technical details should be sought in Kempson et al. (2001) and Cann et al. (2002).[2] However, there are certain technical matters that must be illustrated here in order that the reader may follow the analysis in later sections.

As noted above, the structures that are built are representations of content, not of constituency or other structural characterisation of strings. The principal drivers of the parsing process are thus requirements to establish nodes of certain semantic types, starting from the initial (universal) requirement to build a representation of the propositional content expressed by a string in context: $?Ty(t)$ where t is the type of a proposition and Ty is its associated label. Unlike most Categorial Grammars only a restricted number of types are postulated: $Ty(e)$, the type of a term; $Ty(cn)$, the type of a common noun; $Ty(e \rightarrow t)$, the type of a one-place predicate; and types indicating the arities and argument types of different predicates.

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

Satisfaction of type requirements are achieved when *Formulae* of the appropriate type are constructed. These are given as expressions in some

---

[2] Significant rules are also provided in the Appendix.

Lambda Calculus labelled with the predicate Fo and are provided by parsing words in a string. Lexical entries in Dynamic Syntax are not simply some collection of properties that label terminal nodes in a tree, but instead define transitions between trees. They thus encode packages of *actions* which are initiated by some *trigger*, the condition that provides the context under which subsequent development takes place. These conditional actions may involve the building of nodes (using the `make`(α) action, α a tree relation - see below), movement of the pointer, (using the `go`(α) action, α a tree relation) and/or the annotation of a node with type and formula information (using the `put`(α) action, α a list of labels). There is, additionally, a failure statement that operates when some condition fails to be met. This is commonly an instruction to abort the parsing sequence. For example, parsing the word *John* gives rise to the set of actions in (**??**) which simply annotate the current node with formula and type values.

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

The lexical entries of verbs other than intransitives are more complex, containing sets of actions that build and annotate nodes and give rise to additional requirements to construct expressions of the types of non-subject arguments. The result of parsing the verb *upset*, for example, which is triggered by a predicate requirement $?Ty(e \to t)$[3], yields the sub-tree in Figure **??**. (See the lexical entry in Appendix **??**.)

[Figure 1 about here.]

An innovation of the current framework that allows the definition of

---

[3] Like all verbs in English but not necessarily in other languages, which may have propositional or more specific predicate triggers.

construction rules and lexical entries is the use of a modal logic over tree structures. The Logic of Finite Trees (LOFT, Blackburn and Meyer-Viol 1994) provides a means of referring to arbitrary nodes in a tree using modal operators over mother ($\uparrow$) and daughter ($\downarrow$) relations, possibly annotated with functor-argument information. So we have the operators (amongst others): $\langle\downarrow\rangle$ the general daughter relation; $\langle\downarrow_0\rangle$ and $\langle\downarrow_1\rangle$ the argument and functor daughter relations, respectively; $\langle\downarrow_*\rangle$ the dominance relation (the reflexive, transitive closure of the daughter relation); and the inverses of these using the mother relation, i.e. $\langle\uparrow\rangle$, $\langle\uparrow_1\rangle$, $\langle\uparrow_0\rangle$ and $\langle\uparrow_*\rangle$. Combinations of these modal operators allow reference to any node in a tree from any other node.

The specific and novel advantage of LOFT emerges from the use of the LOFT operators in combination with a generalization of the concept of requirement $?X$ to any decoration X. This combination makes it possible to describe partial trees which have requirements on a treenode that are modal in form. This means that they display requirements which will be fulfilled by some *other* node having a given annotation. Requirements are thus not restricted to nonmodal requirements such as $?Ty(e)$, or simple modal type requirements, such as $?\langle\downarrow_1\rangle Ty(e \rightarrow t)$. To the contrary, any formula may be used to express a requirement. So while $\langle\downarrow_*\rangle Fo(\alpha)$ holding at a node n implies that n dominates a node m where $Fo(\alpha)$ holds, $?\langle\downarrow_*\rangle Fo(\alpha)$ holding at n implies that $Fo(\alpha)$ is *required* to hold at some node m dominated by n. By this means, requirements may constrain subsequent development of a node in the tree at some *arbitrary* distance from the node on which the requirement is imposed; and this provides an additional mechanism for pairing noncontiguous expressions according as one expression imposes some *requirement* on a node which is secured by a decoration on some discrete node by the other expression.

Figure **??** shows five stages in parsing the string *John upset Mary.* A pair of construction rules derive the initial expansion in Figure **??**a, permitting the parse of the first word in the string *John* to annotate this node and move the pointer on to the predicate node, as shown in figure **??**b. At this point the verb *upset* is parsed to construct the object node and annotate the functor node as in figure **??**c. Finally, parsing *Mary* annotates the object node as in figure **??**d. [4] The remaining type requirements in Figure **??**d are satisfied by a rule that compiles and completes the tree through the operation of functional application over types to yield the complete propositional tree in Figure **??**e (which also shows treenode addresses as illustration, although these are elsewhere omitted).

[Figure 2 about here.]

2.1 Anaphora

Interacting with tree growth of this sort is the context-dependent processing of anaphoric expressions. This phenomenon of content underspecification, which we here take in a representationalist spirit (cf. Kempson et al. 1999, Kempson et al. 2001:ch.1 for arguments), involves lexical projection of a *metavariable* to be replaced by some selected term during the construction process. Such replacement is associated with a process of *Substitution* that is pragmatic, and system-external, restricted only in so far as locality considerations distinguishing individual anaphoric expressions preclude certain formulae as putative values of the projected metavariable (i.e. analogues of the Binding Principles, Chomsky 1981, etc.).

---

[4] Note in each partial tree the position of the "pointer", $\diamondsuit$, which identifies which particular node is under development: this is to reflect what point the construction process has reached in building up a structure.

(4) Q: Who upset Mary?

Ans: John upset her.

In processing the pronoun *her* in (**??**), the object node is first decorated with a metavariable $\mathbf{U}$, with an associated requirement, $?\exists\mathbf{x}.Fo(\mathbf{x})$ to find a contentful value for the formula label, as shown in the lexical entry in (**??**).[5]

$$\text{(5)}\quad \textit{her}\quad \begin{array}{ll} \textbf{IF} & ?Ty(e) \\ \textbf{THEN} & \texttt{put}(Fo(\mathbf{U}), Ty(e), ?\exists\mathbf{x}.Fo(\mathbf{x}), [\downarrow]\bot) \\ \textbf{ELSE} & \textbf{ABORT} \end{array}$$

Construed in the context provided, *Substitution* will determine that the formula $Fo(\mathbf{U})$ is replaced by $Fo(Mary)$ which satisfies the imposed requirement.

Note the 'bottom restriction' in (**??**), $[\downarrow]\bot$, which prevents further elaboration of the node it decorates (because it requires that necessarily nothing holds of any node that it dominates). This means that pronouns behave, in English, like contentive expressions in that they must decorate a 'terminal node' on a tree. This has an effect in preventing dislocated expressions from being associated with a position labelled with a pronoun by the process of *Merge*, to which we now turn.

2.2 Left Dislocation

A third sort of underspecification concerns positions within trees. All treenodes have *addresses* which encode their status as functor or argument nodes and their distance from the topnode as signalled by the value of the *treenode* label Tn. The details are not important here (see Kempson et al. 2001:51-53), but the use of this label enables a treenode to be underspecified with

---

[5] A more detailed specification of *her* would include a condition that causes the update sequence of actions to abort in an environment in which the node to be decorated was a subject node, but we ignore this complexity here.

respect to its position in a tree in relation to some other node. Such nodes are marked with an underspecified dominance relation with respect to some other node, shown by the modality $\langle\uparrow_*\rangle Tn(a)$, where a is some given address, and a requirement to find a fixed position within a tree, $?\exists\mathbf{x}.Tn(\mathbf{x})$. This allows an expression to be parsed without it having a fixed position at that point in the parse of a string but ensures that it acquire some determinate position at some point in the parsing process. Such positional underspecification is used to account for long distance dependencies which are analysed in terms of initially unfixed nodes whose position in the emergent tree structure is fixed at some later stage in the parsing process. A construction rule of **Adjunction* introduces an unfixed node of Type e, just in case there is an incomplete tree of $Ty(t)$ that dominates no other material, thus ensuring that the unfixed node appears only at the left periphery of a clause (see Appendix **??**).

As an illustration of the effect of this rule, consider the analysis of the string *That man, John dislikes.* This is illustrated in Figure **??** which shows an initially projected unfixed node, with the characteristic modality $\langle\uparrow_*\rangle Tn(a)$ showing only that the node in question has at some point in the parse process to be fixed at some node dominated by a node with address $Tn(a)$. The parse proceeds as illustrated in Figure **??** up to the string final verb. At this juncture, the pointer, $\diamondsuit$, is at the node of the internal argument and there is a type requirement outstanding to construct a node of type e. In this environment, the unfixed node may *Merge* with the node hosting the pointer, a process that unifies the information of the unfixed and fixed nodes, so satisfying the oustanding requirements to find a fixed position for the unfixed node and a formula of the appropriate type for the internal argument node. Ultimately, completion of the tree yields a

$Ty(t)$ Formula value, $Dislike(That, x, Man(x))(John)$ decorating the topnode, with all requirements fulfilled.[6]

[Figure 3 about here.]

3 Linked Structures and Relative Clauses

We have so far seen how individual trees can be built up following information provided by both general rules and lexical instructions. However, the more general perspective is to model how multiple structures are built up in context. One of the innovative aspects of Dynamic Syntax is that it allows for the building of structures in tandem, constructing first one partial structure, and then another which uses the first as its context. This process is displayed in particular by relative clauses. The characteristic property of what we shall call "linked" structures is that they typically share a common term, and furthermore, the process of inducing the second of such a pair of structures involves a transition from the one tree to the other which itself imposes a constraint for a second occurrence of the term to be shared in that second "LINKed" tree.

Consider, as the simplest case, the analysis of a non-restrictive relative clause like that in (**??**).

(6) That man, who John detests, teaches formal semantics.

The intuition is that the word *who*, correctly described by Jespersen (1927) as a relative *pronoun*, provides the means of copying information from one structure to the other. Having processed the phrase *That man* to yield a partial tree in which the formula $Fo(That, x, Man(x))$ annotates the subject

---

[6] We leave on one side a discussion of the analysis of noun phrases with determiners, for which see Kempson et al. 2001 chapter 7 and Kempson and Meyer-Viol forthcoming.

node (the 'head' node), a transition is licensed by a rule of *LINK Adjunction* (Appendix **??**) which introduces a new tree with a topnode decorated with a requirement to build a propositional tree containing an occurrence of the formula $Fo(That, x, Man(x))$ at some node, without further specification as to where in the newly introduced tree that might be.[7] The new tree is related to the first by the LINK modalities $\langle L \rangle$ from the head node to the LINKed tree and its inverse $\langle L^{-1} \rangle$ from the LINKed tree back to the head. This modal relation provides a type of relation between nodes that is additional to the normal ones of dominance and command, familiar from all tree-theoretic approaches to syntax: one that loosely relates the content of two independent trees. The effect of applying this rule in parsing (**??**) is shown in Figure **??** with the LINK relation shown by the inverse LINK operator on the topnode of the second tree.

[Figure 4 about here.]

Having parsed *That man* and projected the top node of the new LINKed tree, a step of **Adjunction* introduces an unfixed node and the relative pronoun *who* provides the necessary copy of the formula decorating the head for the linked tree according to the set of lexical actions shown in (**??**) which decorates an unfixed node with the formula value of the head of the relative clause.

$$
(7) \quad who_{rel} \quad \begin{array}{ll}
\textbf{IF} & \{?Ty(e), \langle \uparrow_* \rangle \langle L^{-1} \rangle Fo(\alpha)\} \\
\textbf{THEN} & \texttt{put}(Fo(\alpha), Ty(e), [\downarrow]\bot) \\
\textbf{ELSE} & \textbf{ABORT}
\end{array}
$$

The process of tree construction then proceeds as in the simpler case of left dislocation, such as *That man, John dislikes*, with the initially unfixed node

---

[7] The two occurrences get essentially bound together as a consequence of a later LINK evaluation rule. See Kempson et al. forthcoming.

having its position in that tree established in due course through the process of *Merge*, as illustrated by the dotted arrow in Figure **??**.

[Figure 5 about here.]

Completing the parse yields two propositional structures with an interpretation that John detests that man and that man teaches formal semantics.[8] The two rules of *LINK Adjunction* and **Adjunction* thus jointly provide, in conjunction with the lexical actions defining the relative pronoun, a formal reflex of how paired structures can be built subject to a requirement of overlap of content, with a formula in one tree being required to be found within a second.

Notice here that the locality of attachment is determined by the modality associated with the required unfixed node: $?\langle\downarrow_*\rangle X$ from some node n indicates that X appears dominated by n within the current tree and not in some LINKed tree. *LINK Adjunction* thus requires the shared formula in a relative clause to be internal to the current structure and not within some other LINKed tree. This adequately accounts for the effects of strong islands and the example in (**??**) is correctly predicted to be ungrammatical.[9]

(8) *That man who the student who detests thinks is no good teaches semantics.

The same combination of *LINK Adjunction* and **Adjunction* can account also for restrictive relative clauses which involve the projection of a linked propositional tree from a node of type e, albeit this time internal to any quantifier (see Kempson et al. 2001 ch. 4 for details). However, there is

---

[8] See Kempson, Meyer-Viol and Otsuka forthcoming for discussion of the interpretation of linked structures in nteraction with quantification.

[9] Note also that this is subject to cross-linguistic variation (see Kempson et al. 2001:121-130). We leave on one side a discussion of other island restrictions such as the Sentential Subject Constraint as not centrally relevant to the discussion.

no reason to assume that the type of the head need be restricted to nodes of this type and LINKed structures may be constructed from one top node of type t to a new top node of the same type, forcing the outcome to be a common term in both. Such an analysis seems correct for correlative constructions such as are found in Hindi and elsewhere (**??**). It is possible also that extraposition from NP (**??**) could be analysed in the same way.

(9) a. *ve* *do* *laRkiyaan* *Lambii* *naiN* ***jo*** ***khaRii*** ***haiN***
those two girls tall be-PR who standing be-PR
[Hindi]
Those two girls who are standing are tall

b. ***jo*** ***laRkiyaaN*** ***khaRii*** ***haiN*** *ve* *do* *lambii*
Which girls standing be-PR those two tall
haiN
be-PR
Which girls are standing, those two are tall

(10) A woman entered, who Bill said teaches semantics in London.

The examples in (**??**) and (**??**) can both be analysed by a Correlative LINK Rule that induces structures like that shown schematically in Figure **??** which shows a LINK relation built from a completed and compiled propositional tree with a requirement for a copy of one of its subterms to be found in the LINKed structure. Although we do not go into details here, notice what the correlative structure provides: two trees of the same type linked by a shared term. This generalisation of the LINK mechanism for relative clauses is what is required for an analysis of Right Node Raising. However, before this can be given, we need to look at co-ordination and right dislocation within Dynamic Syntax.

[Figure 6 about here.]

4 Co-ordination and LINK

As we have seen, the LINK construction is built up by introducing a structural relation between two trees, with the first partial tree providing the context in which the second is to be processed. In relative clauses, the construction of this LINK relation involves introducing an explicit requirement for such context dependence, with the first tree (parsing the head) providing a *Fo* value that is required to be found in the LINKed tree projected from the relative itself. The LINK mechanism can also be used to analyse other constructions that show semantically weak connections between expressions that require pragmatic inference to be interpreted in context such as gapless topic (**??**a) and afterthought (**??**b) constructions where a peripheral expression provides a term that is found within the main proposition, via a metavariable provided by an anaphoric expression (see Cann et al. 2002 for more discussion).

(11) a. As for John, Mary intensely dislikes him.
b. She talks too fast, Ruth Kempson.

Both of these constructions may be analysed as involving a LINK relation between the primary propositional tree and that analysing the peripheral term. The interpretive effect of the LINK relation, however, remains to be established by the hearer.

While relative clauses, gapless topic and afterthought constructions in English all achieve their somewhat different effect through a shared term, there are syntactically analogous constructions in other languages where there is no obviously shared term, but where the semantic effect that is achieved without explicit presence of a copy requires pragmatic enrichment for interpretation. This appears in certain topic constructions in languages like Japanese and Korean where an initial expression provides information that the hearer needs to use to establish the intended interpretation of the principal proposition (the ‘aboutness’ effect, see Kuno 1973).

(12) a. *haru-wa* *sakura-ga* *ii.* [Japanese]
spring$_{TOP}$ cherryblossom$_{NOM}$ good
‘As for Spring, cherryblossom is beautiful.’

b. *sakana-wa* *tai-ga* *oisii.*
fish$_{TOP}$ red-snapper$_{NOM}$ delicious
‘As for fish, redsnapper is delicious’

Such constructions indicate that the requirement of linked structures to share a term is independent of the building of a LINK relation as such. While a shared term provides one way of construing the relatedness of the content of two trees, this relatedness may be established *pragmatically* in different ways where no term is shared.[10]

This interaction of term sharing and the building of linked structures provides the basis of a principled account of co-ordination within Dynamic Syntax and its use in RNR constructions. As an initial attempt at characterising the effect of parsing a conjunction like *and*, consider (**??**) where the actions induced by parsing the word simply launch a LINK relation with a requirement to construct an expression of the same type as the triggering node.[11] The effect of (**??**) is illustrated in Figure **??** which results from parsing the first four words of *Jane came in and Mary fainted.*

---

[10] For a proposal to separate the construction of LINK structures from the requirement of term sharing, see Otsuka 1999.

[11] Notice that the trigger is not a requirement for a type but an assertion that some node is type-complete. This prevents the acceptance of such strings as **Jane and fainted came in.* The definition of the LINK relation imposes a general requirement of type identity on *and* conjuncts. In this paper, we restrict our attention to conjunction of formulae of type t. For an analysis of conjunctions of type e see Marten (this volume).

$$(13)\quad \textit{and}\quad \begin{array}{ll} \textbf{IF} & Ty(X)\\ \textbf{THEN} & \texttt{make}(\langle L\rangle),\texttt{go}(\langle L\rangle),\texttt{put}(?Ty(X))\\ \textbf{ELSE} & \textbf{ABORT}\end{array}$$

[Figure 7 about here.]

Completing the parse of *Jane came in and Mary fainted* yields two linked propositional trees, the first with formula value *Come-in(Jane)* and the second with $Faint(Mary)$. However, we need a way to semantically evaluate LINK structures, which, as indicated above, needs to be semantically weak to be compatible with a number of different effects. In Kempson et al. (2001), the relation between linked structures is given the weakest possible interpretation, that of conjunction, all that is necessary for the interpretation of restrictive and non-restrictive relative clauses. Assuming that conjunction is the correct way to interpret linked trees in order to allow pragmatic enrichment of content, the completion of the construction process from parsing *Jane came in and Mary fainted* yields a propositional formula for the whole structure: $Come - in(Jane) \wedge Faint(Mary)$.

However, while this is an appropriate interpretation for clauses conjoined by *and*, other conjunctions, such as *or* require different means of interpreting the two linked structures. We thus generalise the process of interpreting linked structures, introducing an $EVAL$ predicate which takes as value some (possibly logical) connective. The rule of *LINK Evaluation*, given formally in Appendix **??**, uses this predicate to determine the appropriate semantic relation between the linked trees. Thus, for example, the lexical entry for *or* is given in (**??**), where the actions induced by parsing the word not only build a LINKed structure but also annotate it with the predicate $EVAL(\vee)$.

$$\text{(14)} \quad \textit{or} \quad \begin{array}{ll} \textbf{IF} & Ty(X) \\ \textbf{THEN} & \texttt{make}(\langle L\rangle), \texttt{go}(\langle L\rangle), \texttt{put}(?Ty(X), EVAL(\vee)) \\ \textbf{ELSE} & \textbf{ABORT} \end{array}$$

The result of parsing a sentence like *Jane jumped or Lou skipped* and applying *LINK Evaluation* is shown in Figure **??** where $f_\vee(Fo(Jump(Jane)), Fo(Skip(Lou)))$ is interpreted as $Fo(Jump(Jane) \vee Skip(Lou))$.[12]

[Figure 8 about here.]

In this section, we have set up a very general means of parsing and evaluating co-ordinate constructions which involves the use of linked structures. Although the output formulae are given in normal propositional logic form with standard connectives, this approach actually provides an asymmetric account of co-ordination. With its emphasis on the process of establishing propositional content, the analysis ensures that an initial conjunct provides the context in which to construe later ones. If we grant the hypothesis that context determines how a string is interpreted and nest this framework within a larger pragmatic perspective, we have a basis for explaining that $p \wedge q$ may not be interpreted the same as $q \wedge p$, because q may induce different inferential effects over p than p does over q. We thus have a direct means of accounting for the difference in interpretation between *Jane broke her leg and fell over* and *Jane fell over and broke her leg*, despite the use of the simple propositional connective.

---

[12] The reason for expressing the evaluation predicate on the second structure and then propagating that information back up through the LINK relations is to account for multiple co-ordinations where the middle conjuncts may be unmarked by a conjunction and the whole is interpreted according to that preceding the final conjunct, e.g. *Jane's at home, in the library or getting drunk with her flatmates.*

5 Right Dislocation

Adopting a left-right parsing perspective on syntactic analysis provides a natural way of accounting for asymmetries between left and right peripheral phenomena. For example, in analysing right dislocated expressions, there is no general 'wait-and-see' mechanism provided by the theory, and no free shift of the pointer to allow parsing to proceed over a gap. Hence, right dislocated expressions cannot be straightforwardly analysed in terms of *Merge* at a point in a parse where there is a type requirement but no lexical input to satisfy it. A consequence of this is that there is no right-peripheral equivalent of a left dislocated gapped topic:

(15) *Kim understood was the new principal, a well-known scientist from London.

(**??**) is ruled out because, as English is a non-pro-drop language, the pointer is placed at the subject node of the embedded propositional structure after parsing the main verb, *understood.* Since the actions of verbs are triggered by a requirement of type $e \rightarrow t$, not $Ty(e)$, the parse necessarily aborts.

In pro-drop structures, however, the actions induced by a verb are triggered by a requirement of type t and give rise to a subject node that is decorated by a metavariable. This satisfies the type requirement of the subject and permits the parse to proceed. Hence, examples like that in (**??**) are well-formed because there is a pronominal type element in subject position that is interpreted as the formula value of the right-dislocated noun phrase.

(16) *poyii Kannan* [Malayalam]
went Kannan
He went, Kannan.

The association between the right dislocated subject expression in (**??**) and the subject treenode is achieved through a rule that projects unfixed nodes at the right periphery. This rule of *Final-*Adjunction* (given in Appendix **??**) differs in a number of respects from that for unfixed nodes at the left periphery (Appendix **??**). In the first place, the unfixed node is projected from a *compiled* tree of type t, rather than from a tree consisting only of node with the requirement to build such a tree. This is a necessary consequence of the parsing perspective of Dynamic Syntax, not an arbitrary condition, for the reason given above: there is no free movement of the pointer that guides the parsing process. Hence, a tree must be type-complete before an unfixed node can be licensed. Secondly, the type of the unfixed node is free to allow for right dislocation of expressions of any type, and finally, a locality condition is imposed on the unfixed node to capture the Right Roof Constraint (Ross 1967). The requirement $?\langle\uparrow_0\rangle\langle\uparrow_*^1\rangle Tn(a)$ requires that the unfixed node be fixed as an argument along the functor spine of the tree to which it is attached[13] which prevents such examples as:

(17) *That a review came out last week is embarrassing of my latest novel.

(18) *That it is likely is certain that I am wrong.

Consider, then, the analysis of (**??**). Parsing the verb *poyii* provides a metavariable in subject position, in line with other pro-drop languages as noted above. At this point, *Substitution*, the regular process for interpreting anaphoric expressions, may apply to provide a value for the metavariable. However, it need not. If no substitution is made the predicate can still combine with the metavariable to yield a type-complete tree that nevertheless is not fully complete because it contains an outstanding requirement: to find a formula value for the metavariable. To fulfil this requirement a substituend

---

[13] We assume a theory such as Marten (2002) in which prepositional adjuncts are treated as arguments of the main predicate. Preposition stranding is permitted just in case prepositions do not build structure but simply annotate nodes of type e.

must be found but as the pointer is no longer at the subject node this cannot be through *Substitution*. However, the value can be provided by the right peripheral expression *Kannan* if this is taken to decorate an unfixed node, as in Figure **??**. The right unfixed node requires a fixed address in the tree and the metavariable still needs a value both of which requirements can be satisfied if (and only if) the former is merged with the latter node.[14]

[Figure 9 about here.]

Note what this analysis entails: expressions can be analysed as decorating an unfixed node introduced late in the parsing process if there is some expression providing a metavariable decorating the node with which a right peripheral expression is to be associated. In pro-drop languages, this is provided for subject (and perhaps other) positions by the predicate, but in non-pro-drop languages, there may exist specialised pronouns, which lack a terminal-node restriction and fulfil the same function. An example of such a construction in English is 'it-extraposition', illustrated in (**??**) which can be analysed by allowing *it* to project a metavariable of type t whose value is provided by the right extraposed clause (see Cann 2001 and Cann et al. 2002 for some discussion).

(19) It is likely that I am wrong.

6 Analysing Right Node Raising

With the assumption of the applicability of both LINK transitions and **Adjunction* at the right periphery in processing an individual clause, the

---

[14] The analysis is somewhat reminiscent of the analysis of subject inversion in Italian proposed in Rizzi (1982) except that the inverted subject is analysed as projected from a clausal, not predicate, node.

challenge now is whether the combination of a LINK relation and *Final-*Adjunction* can be used to reflect the notoriously problematic properties of Right Node Raising.

There is in fact a straightforward account following the dynamics of the parse process. Assuming that parsing can involve the projection of a metavariable as an interim formula value, this metavariable can be taken as the shared term in the linked structures. Once its presence in the second conjunct is secured, an unfixed node is introduced as a late construction step, which merges with the node in the second structure decorated with the metavariable. In virtue of there being a copy of this metavariable in both conjuncts introduced at an earlier stage in the construction process, the dislocated expression is thus interpreted as contributing to the interpretation of both conjuncts.

Two additional assumptions are needed to make this account possible. The first, an extension of the current framework, is that intonation can give clues as to what structure is to be built. This is an aspect of the input which we have so far ignored altogether and indeed the analysis of prosodic information within the DS system remains an open question. However, in such a system, with an explicit parsing-oriented perspective, sensitivity to intonation is entirely expected: intonation forms part of the phonetic signal and is thus available to induce procedures of interpretation during the course of a parse. We suppose, then, that intonation can have the effect within the predicate of signalling the ad hoc construction of a metavariable as an interim formula value, indicating that the containing structure remains incomplete at the current stage of the interpretation process.

In RNR, the distinctive intonational pattern makes manifest to the hearer that she must do something extra in order to successfully parse the string. What this extra effort entails is the decoration of a type-incomplete node with a metavariable of the appropriate type and subsequent compilation of the current tree, leaving open the formula requirement associated with the metavariable. This is achieved through the postulation of a 'lexical free ride' which is given in full in (**??**). Given a trigger of a (free) type requirement, the action checks to see that the current node is within a predicate domain (shown by *Condition* in (**??**)). If this condition is satisfied then the current node is decorated with a metavariable, a formula requirement and a requirement that there be somewhere above the current node at some (possibly subsequent) point in the construction process, a node labelled with some evaluation value.[15] Finally, the pointer moves away from the current node to ensure that the rule of *Substitution* cannot apply at this point to replace the metavariable with a formula value from context.

(20) *Lexical Metavariable Insertion*

```
IF     $?Ty(X)$                                                   Trigger
THEN   IF     $\langle\uparrow_0\rangle\langle\uparrow_*^1\rangle?Ty(e \to t)$,   Condition
       THEN   $\mathtt{put}(Fo(\mathbf{U}), Ty(X)$,                 Metavariable and Type
              $?\exists\mathbf{x}.Fo(\mathbf{x})$,                  Formula Requirement
              $?\langle U\rangle\exists\mathbf{x}.EVAL(\mathbf{x}))$;   Evaluation requirement
              $\mathtt{go}(\uparrow_0)$                             Pointer movement
       ELSE   ABORT
ELSE   ABORT
```

There are a number of things to note about the rule in (**??**). In the first place, it is a lexical rule without lexical input which means that it could

---

[15] Note the modality $\langle U\rangle$, a weak dominance relation that ranges over $\uparrow$ and L^{-1} relations. This modality (and its inverse $\langle D\rangle$ ranging over $\downarrow$ and L) does not respect strong islands.

overgenerate wildly. However, it is not completely unrestricted. In the first place, the trigger constraint $\langle\uparrow_0\rangle\langle\uparrow^1_*\rangle?Ty(e \rightarrow t)$, restricts the application of the actions to predicate internal positions, thus disallowing the parsing of such strings as:

(21) *Yesterday, fell over and hurt his back, Mrs. M's new gardener.

Secondly, the complex requirement of there being an evaluation label, $?\langle U\rangle\exists\mathbf{x}.EVAL(\mathbf{x})$, ensures that *Lexical Metavariable Insertion* only applies in co-ordinate constructions since it is only these that project an annotation with some $EVAL$ statement.[16]

Despite these restrictions, the rule remains dangerously liberal. Nevertheless, there is reason to think that pragmatic restrictions otherwise provide the appropriate constraint on its applicability. Putting relevance theoretic assumptions (see Sperber and Wilson 1995 and Carston 2002, inter al.),[17] together with the parsing perspective of Dynamic Syntax would lead us to expect that an option such as this should not be taken by the hearer unless it is made manifest, e.g. by intonation, that the normal parsing processes will not produce the intended result. Using such a strategy as a regular parsing choice would constitute a violation of the general constraint of minimising cognitive cost in establishing any given effect on two counts for the following reasons. First, the indirect route of projecting a variable only to provide it by a step of *Final-*Adjunction* will always be cognitively costly since it extends the number of steps that have to be taken to yield the more direct result. Secondly, the existence of (**??**) multiplies the parsing possibilities

---

[16] The requirement for an $EVAL$ statement prevents an analysis of *Heavy NP Shift* using the same lexical process. Given the strong differences between Heavy NP Shift and Right Node Raising (not least the lack of 'incomplete' intonation to signal the former construction and its stricter locality requirements), it is probably correct to assume that different processes of right dislocation are instantiated by the two constructions.

[17] Sperber and Wilson claim that all cognitive processing is driven by the balancing of cognitive effort and effect. See Sperber and Wilson 1995.

that need to be entertained by the hearer at every point in the parse, thus threatening to very considerably increase cognitive effort. Unless there are clear signals, then, that the more indirect route is essential to achieving the intended interpretation, this strategy will not meet optimal relevance considerations, and will be avoided.[18] Hence the universally distinctive intonation for Right Node Raising constructions.[19]

We are now in a position to show how the parse of a RNR sentence proceeds, taking (**??**) as the example.

(22) Kim dislikes, but Sandy really admires, the new professor of rhetoric.

The first conjunct is parsed normally up to the main verb where, signalled by intonation, an application of (**??**) licenses the introduction of a metavariable as an interim object value, enabling the tree to be compiled. This tree is type-complete but not fully complete as an interpretation as the formula requirement for the metavariable is not yet satisfied.

From this complete propositional node, parsing *but* provides a LINK transition (with an $EVAL$ statement whose value we take to be the same as for *and*, i.e. $\wedge$) to another tree with a propositional type requirerment, as discussed in section 4. As noted there, the connective only imposes a requirement of type identity plus the label $EVAL(\alpha)$, without any requirement of a shared term between the two conjuncts. This is the general case, but provision for a shared term may be provided separately and this consti-

---

[18] A pragmatic analysis of RNR constructions remains outstanding, but we note here that additional effects associated with RNR may be stylistic, or may involve the use of the second conjunct to project a constraint/hedge upon the construal of the first conjunct, eg as introduced by *but* or *or*.

[19] It is notable that when Right Node Raising data such as (i) are presented as data for visual acceptability judgements, without any punctuation clue as to the intonation that needs to be imposed on the sequence of words, it is common for them to be rejected as not wellformed, only to have that judgement reversed when the same data are read with the intonation characteristic of right-node raising constructions:
(i) John interviewed and Harry made notes on that new student who is in trouble.

tutes the second of our two ancillary assumptions, yielding a generalisation of the correlative construction to allow a shared term in any LINKed, type-identical constructions. The rule of *LINK Dependency*, stated formally in the Appendix **??**, targets a tree containing an incomplete LINKed tree of the same type and copies one of the formula values within the primary tree as a requirement to be found somewhere in the LINKed tree. This rule imposes the weakest of conditions on where in the two substructures the shared term may appear, and is in effect nothing more than a condition on the output formulae decorating the two linked structures that they share a common subterm. The application of this rule targeting the object node in the first conjunct induces the tree in figure **??** .

[Figure 10 about here.]

Intonation again licenses the insertion of a metavariable into the second conjunct which is identical to that introduced into the first conjunct – indeed it must be identical to the first occurrence, in order to meet the imposed requirement $?\langle D\rangle Fo(\mathbf{U})$. The second conjunct is compiled, again to give a tree which is type-complete but whose formula isn't complete. At this point, *Final-*Adjunction* is used to project an unfixed node allowing the parse of the right dislocated expression, as shown in Figure **??**.

[Figure 11 about here.]

The unfixed node *Merges* with the node decorated by the metavariable in the second conjunct, thus satisfying both the former's tree node requirement and providing the value for the outstanding metavariable. The metavariable in the first conjunct is then updated with the same formula value to satisfy its formula requirement and the LINK structure is evaluated to give the tree in Figure **??**, now with all requirements (including the $EVAL$ requirements) satisfied and yielding a final formula value $Dislike(The, x, Professor(x))(Kim) \wedge Admire(The, x, Professor(x))(Sandy)$.

[Figure 12 about here.]

In this analysis all tree development, except that induced by parsing overt lexical items, is optional. However, any other choice of action would lead to the parse aborting. In particular, failure either to identify the metavariable in the second conjunct with that in the first conjunct or to apply the LINK Dependency rule will lead to its formula requirement not being satisfied.

Of the many consequences of this analysis, there are two that we wish to highlight, both of which stem from the fact we characterize the right-peripheral constituent as unfixed locally within the structure projected from the *second* conjunct, while the occurrence of the same formula decorating a node within the structure projected from the *first* conjunct is secured solely through the anaphoric properties of the metavariable. This striking difference between our analysis and all others is a consequence of building semantic trees, and not trees defined over structural properties of strings.

In the first place, the account captures exactly the tension between the locality imposed by the Right Roof Constraint and the apparently conflicting potential for such dependencies to hold across strong islands, as in (1d). The fact that such apparent island violations occur is the result of the very weak modality associated with the *LINK Dependency* rule, enabling the relation between the shared term in both conjuncts to appear anywhere in those or any structure LINKed to them. *Final-*Adjunction*, on the other hand, applies only locally and requires a merge of a right unfixed node within a local tree.

The second consequence of our analysis leads us, unlike all other analyses, to expect an asymmetry between the two conjuncts. Context-sensitive conditions may be satisfied in the second conjunct without them necessarily having to be met in the first.[20] Such asymmetries are duly manifested, for example in the appearance of negative polarity items, as in the English and Hindi examples in (**??**), where such items are only licensed by a negative element in the second conjunct.

(23) a. John read but he hasn't understood any of my books.

b. *John hasn't understood but he has read any of my books.

c. *John-ne parhaa lekin voh samjhaa nahĩ meri*
John-Erg read but he understand-past not my
koi kitaabẽ
any books
John read but hasn't understood any of my books.

d. **John-ne samjhaa nahĩ lekin voh parhaa meri koi*
John-Erg understood not but he read-past my any
kitaabẽ
books
John has not understood but has read any of my books.

This is easily explained in a parsing account in which NPIs require a local environment in which a negative element appears. Assuming that negative elements annotate their most local propositional mother with a distinctive label such as $NEG(+)$, lexical entries for NPIs may be constructed that are sensitive to this label and will abort the parsing process if no such annotation is already established within the tree (see Appendix **??** for an illustration of the lexical entry for *any*). Because the condition for a negative

---

[20] This analysis notably sidesteps the problem faced by all movement analyses of Right Node Raising (eg Postal 1998) in which the right-dislocated element c-commands all other expressions in the string. It also sidesteps the problem confronting in-situ analyses such as Hartmann 1998 (with deletion in the first conjunct) which would preclude any such asymmetry.

marker refers only to local (non-LINK) mother nodes, it follows that only a negative element in the second conjunct will ever license an NPI in Right Node Raising. The same considerations explain why it is the selectional properties of the verb in the final conjunct that must be satisfied rather than those of the first as illustrated in (**??**f). Other such effects can be found in instances of case-mismatch on the right periphery (see Cann et al. 2002 for more discussion).

The final point to be made is that the differences between left and right dislocation follow directly from the parsing perspective taken here. Left dislocation involves 'gaps', in Dynamic Syntax interpreted as points at which an unfixed node *Merges* with a node decorated only by a type requirement. In the analysis given here, such an operation is constrained to be local to a construction and so (**??**c) is impossible, while (**??**b) is ungrammatical because only a single left dislocated element is permitted (in English). Right dislocation, however, necessarily involves a 'pronominal', rather than a 'gap', strategy, that is the use of a metavariable as a placeholder for the formula provided by the right unfixed node. This is not constrained by strong islands (there is no modality associated with the metavariable), as noted above, and is not limited in the number of such metavariables.

We have thus shown in this paper how a parsing perspective gives a natural characterisation of the differences between left and right dislocation and, in particular, that concepts of building partial trees in tandem and unfixed nodes within individual trees combine with a process of tree growth to provide a general account of co-ordination and right dislocation. The tools needed to characterise these two phenomena were shown to provide the basis for modelling the complex properties of Right Node Raising. We suggest that the success in accomplishing this task – where other approaches uniformly fail – signals the need of a change of theoretical direction to one in which grammar formalisms for natural languages are defined to reflect the dynamics of left to right processing.

References

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

Cann, R. 2001. How pronominal are expletive pronouns? In Dekker, P. and van Rooy, R. (eds.) *Proceedings of the 13th Amsterdam Colloquium*, 55-60.

Cann, R., Kempson, R. and Otsuka, M. 2002. 'On left and right dislocation: a dynamic perspective'. MS University of Edinburgh/Kings College London.

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

Chomsky, N. 1981. *Lectures on Government and Binding.* Dordrecht: Foris Publications.

Hartmann, K. 1998. Right Node Raising and Gapping: Interface conditions on prosodic deletion. Ph.D Thesis Frankfurt.

Jespersen, O. 1927. *A Modern English Grammar on Historical Principles.* London: George Allen and Unwin.

Kempson, R. and Meyer-Viol, W. forthcoming. Indefinites and Scope Choice. In Bezuidenhout, A. and Reimer, M. (eds.) *Descriptions and Beyond: an interdisciplinary collection of essays on definite and indefinite descriptions and other related phenomena.* Oxford: Oxford University Press.

Kempson, R., and Meyer-Viol, W. and Gabbay, D. 1999. Towards a dynamic account of VP Ellipsis. In Lappin, S. and Benmamoun, E. (eds.) *Fragments and Ellipsis Construal.* Oxford: Oxford University Press, 227-290.

Kempson, R. Meyer-Viol, W. and Gabbay, D. 2001. *Dynamic Syntax: The Flow of Language Understanding.* Oxford: Blackwell.

Kempson, R. Meyer-Viol, W., and Otsuka, M. forthcoming. Growth of logical form: the dynamics of syntax. In Peregrin, J. (ed.) *Meaning in the*

Dynamic Turn. Oxford: Elsevier.

Kuno, S. 1973. *The Structure of the Japanese Language.* Cambridge MA: MIT Press.

Levine, R. 2001. The extraction riddle: just what are we missing? *Journal of Linguistics* 37, 145-76.

McCawley, J. 1988. *The Syntactic Phenomena of English.* Chicago: Chicago University Press.

McCloskey, J. 1986. Right Node Raising and preposition stranding. *Linguistic Inquiry* 17, 183-186.

Marten, L. 2002. *At the Syntax-Pragmatics Interface.* Oxford: Oxford University Press.

Marten, L. this volume. The dynamics of asymmetrical conjunction.

Postal, P. 1998. *Three Investigations of Extraction.* Cambridge MA: MIT Press.

Otsuka, M. 1999. The Head-Internal Relative in Japanese from an LDS perspective. *SOAS Working Papers in Linguistics* 9, 39-73.

Rizzi, L. 1982 *Issues in Italian Syntax.* Dordrecht: Foris Publications.

Ross, J. R. 1967. Constraints on Variables in syntax. Ph.D thesis. MIT.

Sperber, D. and Wilson, D. 1995. *Relevance: communication and cognition* (2nd edition). Oxford: Basil Blackwell.

Steedman, M. 1996. *Surface Structure and Interpretation.* Cambridge, MA: MIT Press.

7 Appendix

A–1. *upset*

$$
\begin{array}{ll}
\textbf{IF} & ?Ty(e \to t) \\
\textbf{THEN} & \texttt{go}(\langle\uparrow_1\rangle); \texttt{put}(PAST(S_i, \mathbf{U})); \texttt{go}(\langle\downarrow_1\rangle); \\
 & \texttt{make}(\langle\downarrow_1\rangle); go(\langle\downarrow_1\rangle); \\
 & \texttt{put}(Ty(e \to e \to t), Fo(Upset), [\downarrow]\bot); \\
 & \texttt{go}(\langle\uparrow_1\rangle); \texttt{make}(\langle\downarrow_0\rangle); \texttt{go}(\langle\downarrow_0\rangle); \\
 & \texttt{put}(?Ty(e)) \\
\textbf{ELSE} & \textbf{ABORT}
\end{array}
$$

A–2. **Adjunction*

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

A–3. *LINK Adjunction*

$$
\frac{\{..\overbrace{\{X, Fo(\alpha), Ty(e), \diamondsuit\}}^{Head}\}}{\{..\underbrace{\{X, Fo(\alpha), Ty(e)\}}_{Head}, \underbrace{\{\langle L^{-1}\rangle X, ?Ty(t), ?\langle\downarrow_*\rangle Fo(\alpha), \diamondsuit\}}_{Linked\ node}\}}
$$

A–4. *LINK Evaluation*

$$
\frac{\{..\{Tn(a), Fo(\alpha), Ty(X)\}\langle L^{-1}\rangle Tn(a), Fo(\beta), EVAL(\phi), Ty(X), \diamondsuit\}}{\begin{array}{c}\{..\{Tn(a), EVAL(\phi), f_\phi(Fo(\alpha), Fo(\beta)), Ty(X), \diamondsuit\}, \\ \{\langle L^{-1}\rangle Tn(a), EVAL(\phi), Ty(X), Fo(\beta)\}\end{array}}
$$

A–5. *Final-*Adjunction*

$$
\frac{\{..., \qquad \overbrace{\{Tn(a), Ty(t), \diamondsuit\}}^{Type-compiled\ propositional\ tree} \qquad \}}{\begin{array}{c}\{..., \{Tn(a), Ty(t)\}, \\ \underbrace{\{\langle\uparrow_*\rangle Tn(a), ?Ty(X), ?\exists\mathbf{x}Tn(\mathbf{x})}_{Unfixed\ Node}, \quad \underbrace{?\langle\uparrow_0\rangle\langle\uparrow_*^1\rangle Tn(a),}_{Right\ Roof\ Constraint} \quad \diamondsuit\}\}\end{array}}
$$

A–6. *LINK Dependency*

$$\frac{\begin{array}{c}\{Tn(a), Ty(t), \ldots \{(MOD)Tn(a), \ldots Ty(X), Fo(\alpha), \ldots\} \ldots\},\\ \{\langle L^{-1}\rangle Tn(a), ...?Ty(t), \Diamond\}),\end{array}}{\begin{array}{c}\{Tn(a), Ty(t), \ldots \{(MOD)Tn(a), \ldots Ty(X), Fo(\alpha), \ldots\} \ldots\},\\ \{\langle L^{-1}\rangle Tn(a), ...?Ty(t), ?\langle D\rangle Fo(\alpha), \Diamond\}\end{array}}$$

$$MOD \in \{\langle\uparrow_0\rangle, \langle\uparrow_1\rangle, \langle L^{-1}\rangle\}^*$$

A–7. *any*

```
IF      $?Ty(e)$
THEN    IF      $\uparrow_* NEG$
        THEN    make($\langle\downarrow_1\rangle$), go($\langle\downarrow_1\rangle$),
                put($Fo(\lambda P.(\epsilon, P)), Ty(cn \rightarrow e)$), go($\langle\uparrow_1\rangle$),
                make($\langle\downarrow_0\rangle$), go($\langle\downarrow_0\rangle$), put($?Ty(cn)$)
        ELSE    ABORT
ELSE    ABORT
```

List of Figures

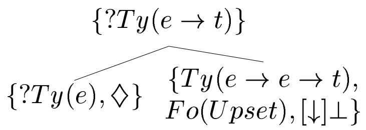

Figure 1: The result of parsing *upset*

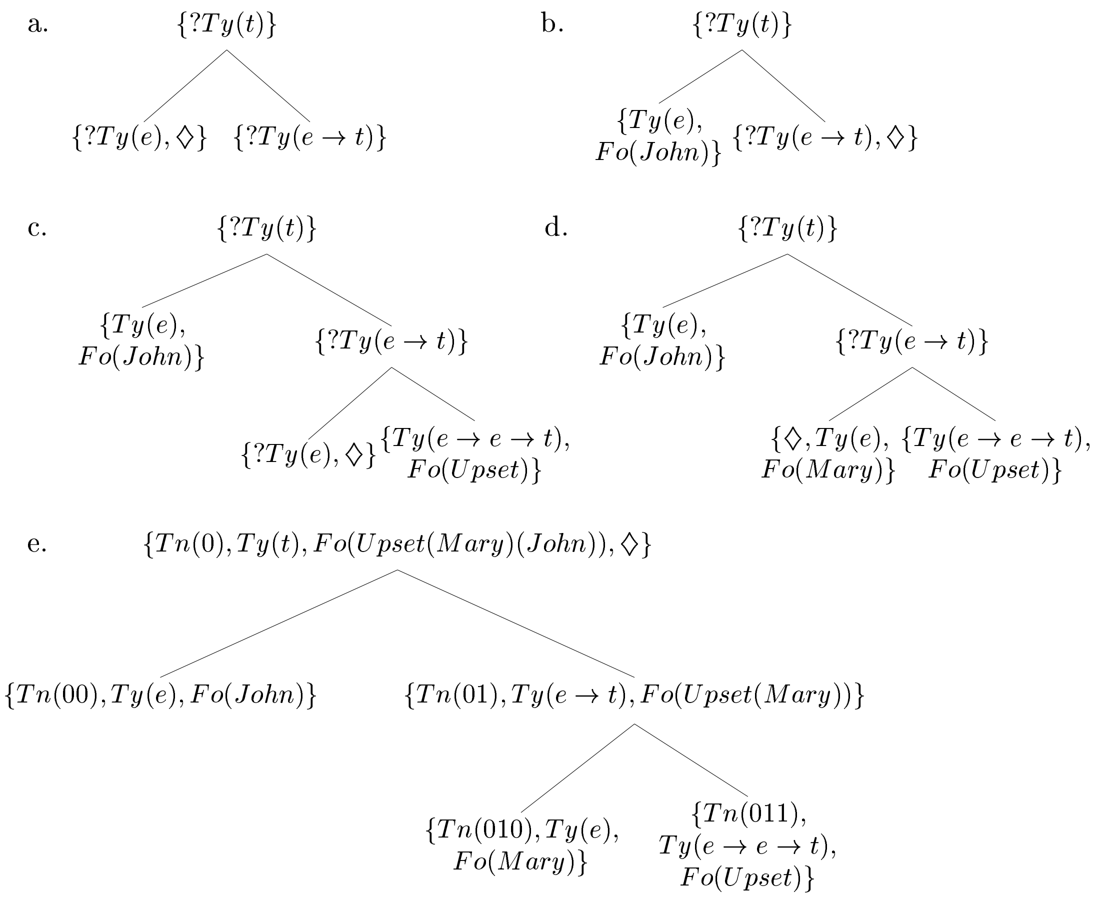

Figure 2: Five stages in parsing *John upset Mary*

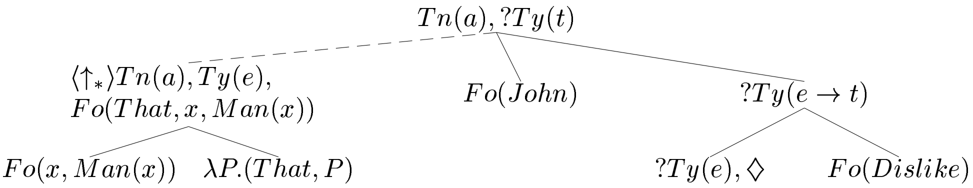

Figure 3: Parsing *That man, John dislikes*

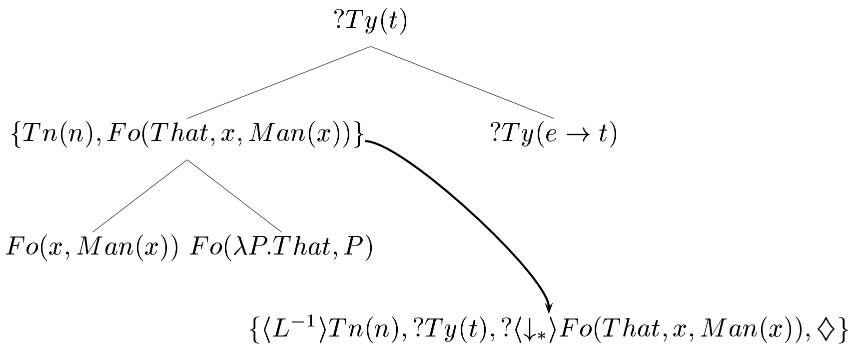

Figure 4: Parsing *That man*

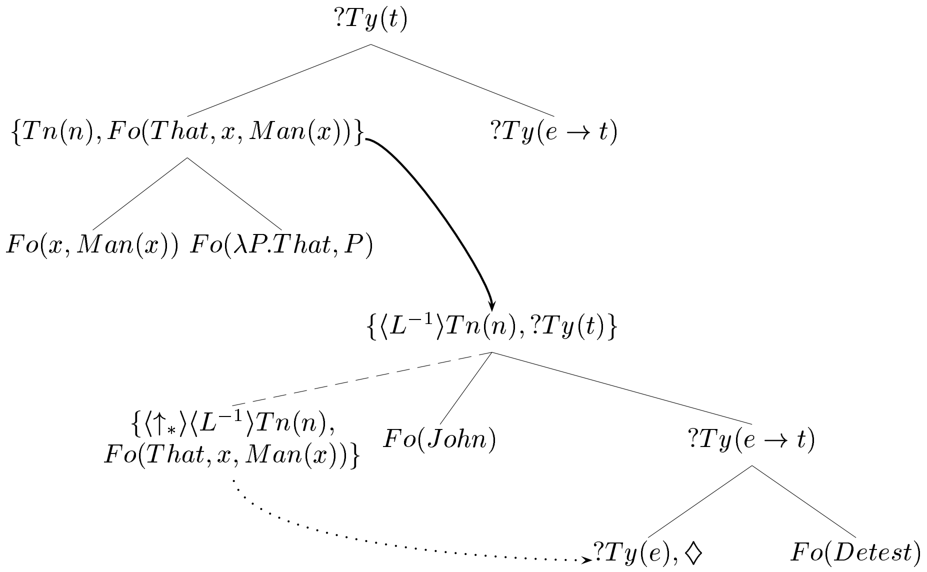

Figure 5: Parsing *That man, who John detests*

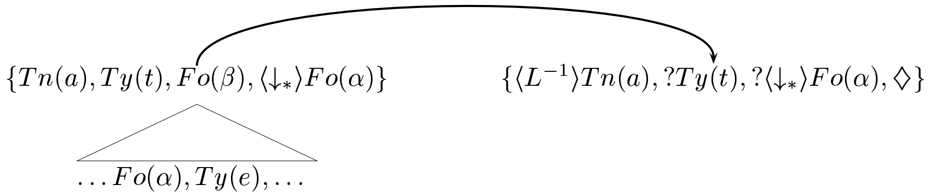

Figure 6: Correlative Structures

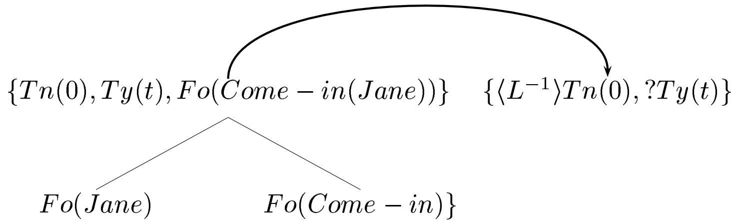

Figure 7: Parsing *Jane came in and*

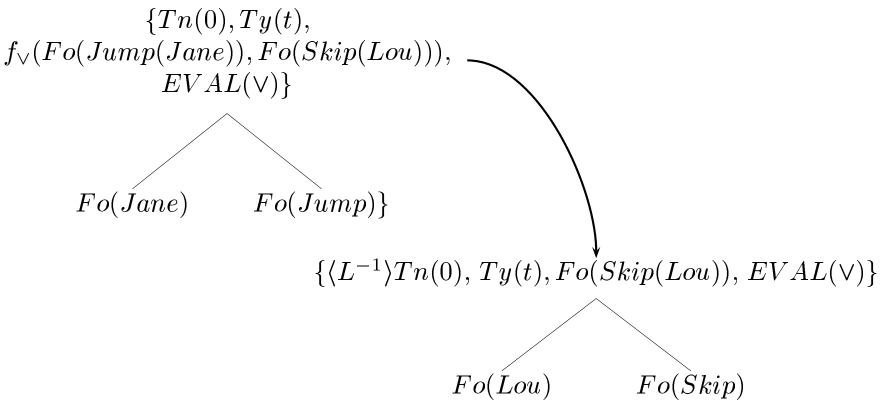

Figure 8: Parsing *Jane jumped or Lou skipped*

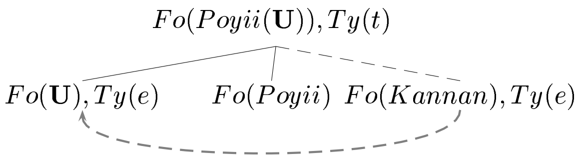

Figure 9: Parsing *poyii Kannan*

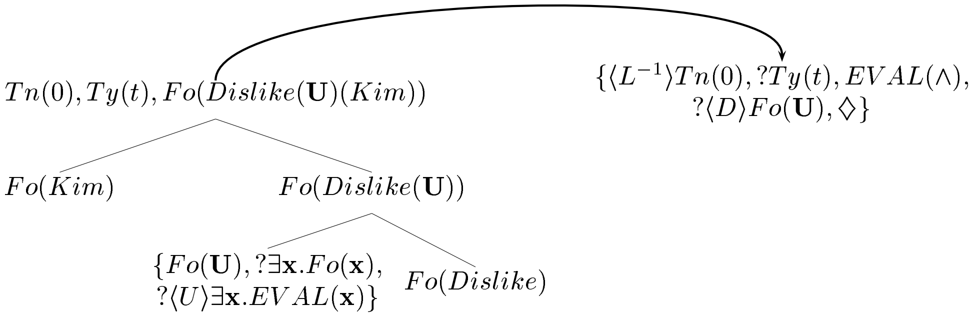

Figure 10: Parsing *Kim dislikes but*

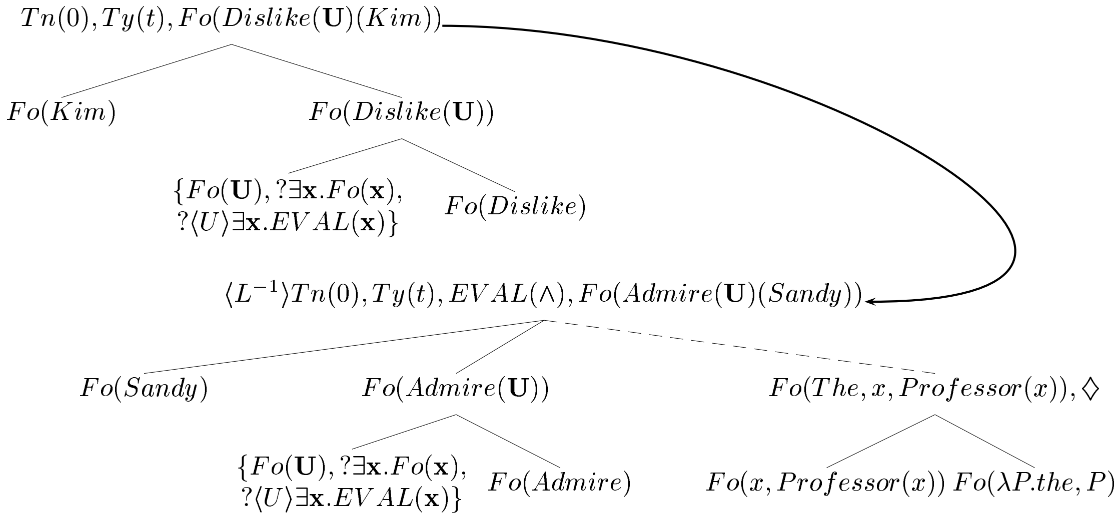

Figure 11: Parsing the second conjunct

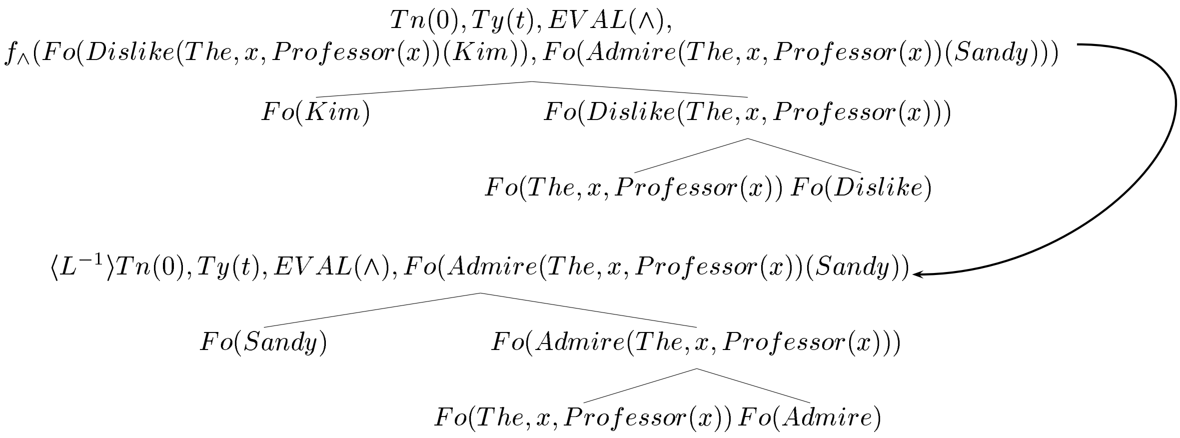

Figure 12: Completing the tree