Syntactic Computation as Labelled Deduction : WH a case study

RUTH KEMPSON, WILFRIED MEYER-VIOL, DOV GABBAY

November 18, 1996

1) The Question

Over the past thirty years, the phenomenon of long-distance dependence has become one of the most well studied phenomena. Requiring as it does correlation between some position in a string and the c-commanding operator which determines its interpretation, it is uncontroversially assumed across different theoretical frameworks to involve an operator-variable binding phenomenon as in standard predicate logics (cf. Chomsky 1981, Morrill 1994, Pollard & Sag 1991, Lappin & Johnson 1996). However, it is known to display a number of properties which distinguish it from the logical operation of quantifier variable binding, and these discrepancies are taken to be indicative of the syntactic idiosyncracy of natural language formalisms. Investigation of these properties has led to the postulation of increasing numbers of discrete phenomena. There has been little attempt until recently to ask the question as to why the overall cluster of *wh* processes exist (for recent partial attempts, cf Cheng 1991, Müller & Sternefeld 1994, 1996, and Cole 1996).[1] The primary purpose of this paper is to propose an answer to this question. Having set out an array of largely familiar data in Section 1, in Section 2 of the paper we develop the LDS-NL framework within which the analysis is set. This is a formal deductive framework being established as a model of the process of utterance interpretation. Then in Section 3 we present a unified account of the crossover phenomenon, and in Sections 4-5 we briefly indicate analyses of reconstruction, *wh*-in situ and multiple *wh* questions, and partial *wh* movement phenomena, showing how a typology of *wh* variation emerges. In all cases, the solution will make explicit reference to the discrete stages whereby interpretation is incrementally built up in moving on a left-right basis from the initial empty state to the completed specification of a logical form corresponding to the interpretation of the string in context. The account is thus essentially procedural. In closing we reflect on the direction which this conclusion suggests - that the boundaries between syntax, semantics and pragmatics need to be redrawn, with syntax redefined as the dynamic projection of structure within an abstract parsing schema.

1.1 Failure to Display Scopal Properties in Parallel with Quantifying Expressions

As is well known, *wh*-expressions fail to display scopal properties in parallel with quantifying expressions. Initial *wh* expressions may take narrow scope with respect to any operator following it as long as that operator precedes the position of the gap. Hence (1) allows as answers to this question, ones in which the *wh* has been construed as having scope relative to the expression *every British farmer* in the subordinate clause.

(1) What is the Union insisting that every British farmer should get rid of?
Answer : At least 1,000 cattle.
Answer: His cattle.

On the assumption that scope is displayed in the syntactic structure assigned to the string, questions such as these appear to require an LF specification which displays the relative scope of the two expressions in contravention of the structure associated with the surface string. This phenomenon is quite unlike quantifiers in logical systems. A given quantifier may bind free variables if and only if these variables are within its scope where this is defined by the rule of syntax which introduces that quantifier, hence by definition guaranteeing a configuration equivalent to c-command. Furthermore, other natural language quantifiers behave much more like logical quantifiers, and in the main must be interpreted internally to the clause in which they are contained.[2] Thus (2)-(3) are unambiguous. Neither can be interpreted with the quantified expression in the subordinate clause taking scope over the matrix subject:

(2) Every British farmer is complaining that most countries of the EU fail to appreciate the problem
≠ 'For most countries of the EU_x, every British farmer is complaining that x fails to appreciate the problem'

(3) Most countries of the EU are responding that every British farmer fails to appreciate the seriousness of the problem.
≠ 'Of every British $farmer_y$ most countries of the EU are responding that y fails to appreciate the seriousness of the problem'.

This phenomenon can be analysed by defining *wh* expressions to be a complex higher-type quantifier simultaneously binding two positions, one of which is an invisible pronominal element (Chierchia 1992), but this technical solution fails to provide any basis for explaining other phenomena associated with *wh* expressions. Crossover phenomena in particular, though an essential piece of supporting evidence for this analysis, become a mere syntactic stipulation.

1.2 Crossover

Pretheoretically, the crossover phenomena is simply the interaction between WH and anaphora construal. Within the GB paradigm, this has been seen as dividing into at least three discrete phenomena (Chomsky 1981, Lasnik & Stowell 1991, Postal 1993). The data are as follows:

(4) *Who_i does Joan think that he_i worries e_i is sick?
(5) *Who_i does Joan think that his_i mother worries e_i is sick?
(6) *$Whose_i$ exam $results_j$ was he_i certain e_j would be better than anyone else's?
(7) Who_i does John think e_i worries his_i mother is sick?
(8) Who_i does Joan think e_i worries that he_i is sick?
(9) $Whose_i$ exam $results_j$ e_j were so striking that he_i was suspected of cheating?
(10) *$John_i$, who Sue thinks that he_i worries e_i is sick unecessarily, was at the lecture.
(11) John, who_i his_i mother had ignored e_i, fell ill during the exam period.
(12) John, $whose_i$ exam results he_i had been certain e_i would be better than anyone else's, failed dismally.

The need to distinguish discrete sub-classes of phenomena arises from the analysis of the gap as a name subject to Principle C of the A-binding principles (Chomsky 1981). A strong crossover principle is said to preclude a gap (as a name) being coindexed with any c-commanding argument expression, hence precluding (4), (10), and possibly (6), while licensing (7)-(9) on the grounds that the relation between gap and *wh* operator is a relation of A'binding and not of A-binding. ((6) has been dubbed "extended strong crossover" because the *wh* expression, being a possessive determiner, doesn't, strictly speaking, bind the gap, but a subexpression within it.) Such a restriction however fails to preclude (5) and (6), for which a separate restriction of weak crossover is set up. There are several versions of this principle (Higginbotham 1981, Koopman & Sportiche 1982, Chomsky 1981, Lasnik & Stowell 1991) - the simplest is that a pronoun which does not c-command a given trace may nevertheless not be coindexed with it if it is to the left of the trace (and right of the binding operator). This restriction in its turn however fails to predict that in some circumstances this restriction may get suspended as in (11)-(12), and an alternative analysis in which the traces in this position are not names but prominal-like "epithet" expression, is advocated. The phenomenon is thereby seen as a cluster of heterogeneous data, not amenable to a unified analysis. No explanation is proffered for why the data should be as they are, and Postal 1993 describes the phenomenon as a mystery.

1.3 Reconstruction:

Wh reconstruction is a process whereby at least part of some "moved" *wh* expression has to be moved back to the initial position. Moreover it seems that this movement has to be relative to intervening complementizer positions, since the *wh* and its containing parts appear to have to be interpreted/licensed relative to each intervening clause between the initial *wh* position and the gap. The data which specifically display this are:

(13) Which pictures of each other$_i$ did Bill say [Sue and Mary]$_i$ were worried that the press were planning to publish e?

In (13) the expression *each other*, despite not being c-commanded by *Sue and Mary* in their string position, and not being local to them in their initial d-structure position, nevertheless allows an interpretation in which *each other* is construed as dependent on and identified by *Sue and Mary*, hence requiring some point in the interpretation at which the anaphor is suitably local to that expression as antecedent. It may be argued that anaphors are insufficient evidence for the need to refer to sites intermediate between the gap and its WH operator, since there is a whole array of discourse-related anaphor-dependence not all of which falls within the purview of a reconstruction explanation (cf. Pollard and Sag 1991 who analyse them as separate logophoric anaphors, and hence not subject to any locality restriction). However there is a wide range of independent evidence that information from some initial *wh* position has to be associated with each of the intervening complementizer sites between *wh* expression and the gap it binds (reported in Hukari and Levine 1995).Here I shall take this evidence on trust, but cf. in particular the complementizer alternation in Irish reported by McCloskey 1979).

1.4 *Wh* in situ

There are of course also the familiar island restrictions associated with *wh*-initial expressions, which are entirely alien to quantification in formal systems, and so, like the other data, differentiate long-distance dependency effects from regular operator-variable binding. However, more striking is that *wh* in situ expressions, despite commonly being said to be subject to the same movement as *wh* initial expressions but at the level of LF (Reinhart 1994, Aoun & Li 1991, Huang 1982) are characteristically not subject to these same restrictions. So, unlike (14), (15) allows an interpretation in which the *wh* expression is construed, so to speak, externally to the domain within which the *wh* expression is situated:

(14) *Which document did the journalist that leaked to the press apologise ?
(15) The journalist that leaked which document to the press became famous overnight?

The phenomenon of *wh*-in-situ is arguably peripheral in English, but in languages where this is the standard form of *wh* question, the distribution of *wh* in situ, unless independently restricted (cf. the data of Iraqi Arabic below) is characteristically not subject to the same constraints as *wh* movement (Chinese, Japanese, Malay) (data from Simpson 1995):

(16) Ni bijiao xihuan [[ta zenmeyang zhu] de cai] ? CHINESE
You more like how cook REL food
What is the means x such that you prefer the dishes which he cooks by x?

.

1.5 Multiple *Wh* Structures

Paired with this phenomenon are multiple *wh* questions, of which the initial *wh* expression is subject to island restrictions, but the *wh* expression in situ is not:

(17) Who do you think should review which book?
(18) *Who$_i$ did the journalist leak the document in which Sue had criticised e$_i$ to which press?
(19) Who reported the journalist that leaked which document to the press?

1.6 Partial WH movement

Finally, there is the phenomena in German dubbed "partial *wh* movement" in which apparently expletive *wh* elements anticipate full *wh* expressions later in the string, but are not themselves binders of any gapped position.

(20) Was glaubst du was Hans meint mit wem Jakob gesprochen hat ?
With whom do you think Hans thought/said Jakob had spoken?

Such expletive elements must invariably take the form *was* in all complementizer positions between the initial position and the *wh* expression they anticipate, but subsequent to that full *wh* expression, the complementizer selected must be *dass*. This gives rise to a number of discrete forms, with identical interpretation :

(21) Was glaubst du mit wem Hans meint dass Jakob gesprochen hat ?
(22) Mit wem glaubst du dass Hans meint dass Jakob gesprochen hat ?
With whom do you think Hans thought/said Jakob had spoken?

This phenomenon, with minor variations, is widespread in languages in which the primary structure is the *wh* in situ form. Iraqi Arabic for example has a reduced *wh* expression which is suffixed to the verb indicating the presence of a *wh* expression in a subordinate clause. However, unlike German, the subordinate clause contains the full *wh* expression in situ. Also unlike German, this suffix *sh-*, a reduced form of *sheno* (='what'), must precede the verb in each clause between the initial clause carrying the first instance of the expletive and the clause within which the full *wh* expression itself occurs. Without *sh-*, the presence of the *wh* in situ in a tensed clause is ungrammatical (data from Simpson 1995):

(23) Mona raadat [riijbir Su'ad tisa'ad *meno*]
Mona wanted to force Suad to help who
Who did Mona want to force Suad to help?
(24) *Mona tsawwarat [Ali ishtara *sheno*]
Mona thought Ali bought what
(Intended: What did Mona think that Ali bought?)
(25) Sheno$_i$ tsawwarit Mona [Ali ishtara t$_i$]
What thought Mona Ali bought
What did Mona think Ali bought ?
(26) *sh-*'tsawwarit Mona [Ali raah *weyn*] ?

Q-thought Mona Ali went where
Where did Mona think that Ali went ?
(27) *sh*-'tsawwarit Mona [Ali ishtara sheno ?
Q-thought Mona Ali bought

These phenomena have only recently been subject to serious study, but their analysis in all frameworks remains controversial (McDaniel 1988, Srivistav Dayal 1993, Simpson 1995, Johnson & Lappin 1996, Müller & Sternfeld 1996).

Faced with this apparent heterogeneity, it is perhaps not surprising that these phenomena are generally taken in isolation from each other, requiring additional principles. Of those who provide a general account, Lappin & Johnson articulate an account within the HPSG framework which involves three discrete operators: - a binding operator for *wh* expressions, a discrete operator for *wh* in situ, and yet a further operator to express the expletive phenomena. The primary task in the various theoretical paradigms seems to have been that of advocating sufficient richness within independently motivated frameworks to be able to describe the data. Little or no attention has been paid to why *wh* expressions display this puzzling array of data.

2. The Proposed Answer

The answer we propose demands a different, and more dynamic, perspective. Linguistic expressions will be seen to project not merely structure over which semantic interpretation can be projected, but also structure pertinent to the incremental meta-level process of establishing that structure. This dynamic projection of structure is set within a framework for modelling the process of utterance interpretation, which is defined as a goal-directed task of establishing a propositional formula. Two concepts of content are defined within the framework - the content associated with the structure which results from the output of the structure-building process, and the content associated with the process itself - both reflected in lexical specifications. A primary commitment is to provide a representational account of the often observed asymmetry between the content encoded in some given linguistic input and its interpretation in context, an interpretation which is not fully specified by the lexical expression itself (cf. Sperber & Wilson 1986, whose insights about context dependence this framework is designed to reflect). *Wh* expressions will be seen as displaying such asymmetry. Unlike straightforward quantifying expressions, they underspecify both the content and the structural configuration which they are taken as projecting. As clause-initial expressions they do NOT project a uniquely determined tree relation with what follows, and this poses a problem to be resolved during the interpretation process. Hence their content is projected through the structure as it is incrementally projected until able to establish the node at which their interpretation is projected. *Wh* in situ constructions are the mirror image to *wh* initial constructions, their tree relation with their sister nodes being uniquely determined, but not the projection of the +Q feature determining questionhood. Parametric variation will be seen to arise from different locality effects associated with the correspondence between the *wh* expression in situ and this Q feature. Finally, partial movement constructions will emerge as a direct consequence of combining this underspecification analysis of the characterisation of *wh* with the dynamics of the goal-directed parsing task. The consequence of this shift to a more dynamic syntactic perspective is a much closer relation between formal properties of grammars and parsers, a consequence which we shall reflect on briefly in closing.

2.1 The point of departure: goal-directed proofs

The point of departure for this analysis is a display of a natural deduction proof, annotated with a set of meta-level statements about the goals and subgoals imposed on successive steps of the proof:

(28) $P\to(Q\to R) \vdash Q\to(P\to R)$

step			
1	$P\to(Q\to R)$		GOAL $Q\to(P\to R)$
2	Q		GOAL $P\to R$
3	P		GOAL R
4	$Q\to R$	Modus Ponens	
5	R	Modus Ponens	
6	$P\to R$	→Intro.	
7	$Q\to(P\to R)$	→Intro.	

The proof involves the construction of two ancillary assumptions, and two steps each, of Modus Ponens and → Introduction. Viewed as a piece of dynamic syntax, two properties of goal-directed reasoning are brought out by this display:

(A) To specify some goal and elements to be contained in that goal does not constitute achieving it or having those elements. P, Q, and R are not elements of the proof in (28) in virtue of being listed elements of the goals (as at steps 1,2,3). Such elements only become parts of the proof as they are assumed or derived.

(B) The properties of the goal, together with the rules of the system, determine the form of subordinate routines to be set up in achieving that goal. Hence the goal listed at step 1 is reflected in the subordinate goals within the tree structure that direct the path to be followed in getting to that goal. There is an apparent percolation of information "down" the proof structure from goal to subgoal. These properties are not confined to goal-driven natural deduction systems. It is a general property of tasks with a given target that (i) specifying the target to be achieved does not constitute meeting the target; (ii) the form of the target dictates the form of subgoals to be set up in reaching that target. The parallelism with *wh* expressions is intriguing, given the percolation of information from an initial *wh* expression down the tree from clause to clause, and suggests that *wh* initial strings might revealingly be analysed as imposing a task whose resolution depends in some sense on the percolation of this information through the structural process of building up interpretation. It is this form of solution that we shall seek to articulate.

2.2 The Framework: Labelled Deductive System for Natural Language - LDS$_{NL}$

The general framework within which the analysis is set is a model of the process of natural language interpretation, where the goal is to model the pragmatic process of incrementally building up propositional structure from only partially specified input. The underlying aim is to model the process of understanding reflecting at each step the partial nature of the information encoded in the string and the ways in which this information is enriched by choice mechanisms which fix some particular interpretation.

The process is driven by a mixed deductive system - type-deduction is used to project intra-clausal structure (much as in Categorial Grammar - cf. in particular Morrill

1994, Oehrle 1995), but there is in addition inference defined over databases as units for projecting inter-clausal (adjunct) structure (cf. Joshi and Kulick 1995 for a simple composite type-deduction system). The background methodology assumed is that of Labelled Deductive Systems (Gabbay forthcoming). According to this methodology, mixed logical systems can be defined, allowing systematically related phenomena to be defined together while keeping their discrete identity. A simple example is the correlation between the functional calculus and conditional logic (known as the Curry-Howard isomorphism) with functional application corresponding to Modus Ponens, Lambda abstraction corresponding to Conditional Introduction. Thus we might define Modus Ponens for labelled formulae as:

(29) Modus Ponens for labelled formulae

$$\frac{\begin{array}{l}\alpha : P\\ \beta : P \rightarrow Q\end{array}}{\beta(\alpha) : Q}$$

In the system we adopt here, intra-clausal structure is built up by steps of type deduction, much as in Categorial grammar, but, since the primary task is that of building up a propositional structure, we define the formula to be the expression being established, and the labelling algebra to be the set of specifications/constraints which drive that process. (30) provides the simplest type of example, displaying how type deduction (and its twinned operation of function application) drives the process of projecting a representation of propositional content which duly reflects the internal mode of combination:

(30) Type : Formula
e : John
$e \rightarrow t$:smokes
t : smokes(John)

The interpretation process is formalised in an LDS framework in which labels guide the parser in the goal-directed process of constructing labelled formulas, in a language in which these are defined together. Declarative units consists of pairs of sequences of labels followed by a content formula. The formula of a declarative unit is the side representing the content of the words supplied in the course of a parse. The labels annotate this content with linguistic features and control information guiding the direction of the parse process. Since the aim is to model the incremental way information is built up through a sequence of linguistic expressions, we shall need a vocabulary which enables us to describe how a label-formula constellation is progressively built up. With this in mind, declarative units are represented as finite sets of formulas:

$$\{Lab_1(l_1),...Lab_n(l_n), FO(\phi)\}.$$

In the course of a parse these feature sets grow incrementally.

The model is a state transition system which defines the steps which license movement from state to state. Each state is seen as a partial tree anticipating steps of deduction to be realised in establishing the required propositional structure, with specifications of what has been done at each stage, and what remains to be done. The task is goal-driven, the goal to establish a database of type t using information as incrementally provided on a left-right basis by a given input string. The process is a syntactic deductive system licencing a sequence of task states. Each state has a database and a header. The database indicates the information established so far. The header indicates the overall goal of that task - SHOW X for some type X; the tree node of the particular task being built up; the subgoal of the given task - what remains TO DO in the current task; and a specification of which task state it is. (31) displays the general format:

(31)

Tree node	GOAL	SUBGOAL	Task number

m SHOW t TODO Type(t) 0

m SHOW t TODO <d>Type(e), <d>Type($e \to t$) 0

.

.

n SHOW e TODO $\varnothing$ 1.
Formula(α) & Type(e)

n TODO $\varnothing$ 1
Formula(α) & Type(e)

m SHOW t TODO <d>Type($e \to t$) 0
<d>(Formula(α) & Type(e))

p SHOW $e \to t$ TODO<$e \to t$> 2

.

.

n TODO$\varnothing$1 .... p TODO$\varnothing$ 2
Formula(α) Formula(β)
& Type(e) & Type ($e \to t$)

m SHOW t TODO $\varnothing$ 0
<d>(Formula (α) & Type(e))
<d>(Formula (β) & Type($e \to t$))

n 1 .... p 2
Formula(α) Formula(β)
& Type(e) & Type ($e \to t$)

m SHOW t TODO $\varnothing$ 0
Formula ($\beta(\alpha)$) &Type(t)

(<d>P = "P holds at a daughter of me")

In the initial state, the goal is to show **t**, with the subgoal TODO also specifying that as a task to be carried out. The node is m, the initial state. The final state is when that same node is arrived at as the last of a sequence of states with no subgoals remaining, and with the goal of the task fulfilled, and a propositional formula established. Each subtask set in fulfilling that task is assigned a task number, and described according to a tree-node vocabulary which enables trees to be defined in terms of properties holding at its nodes, and relations between them. Successive steps of introduction rules introduce subtasks, which once completed combine in steps of elimination to get back to the initial task and its successful completion. Thus for example, the opening sequence of states given presentation of a subject NP introduces the subtasks, TODO, of building first a formula of type **e**, and then a formula of type **e→t** (enabling a formula of type **t** to be derived. And correspondingly, the last step of the derivation is a step at which, both these subtasks having been completed, a step of Modus Ponens applying to the completed tasks establishes the goal of deducing a formula of type **t** at the initial task state, with no subgoals remaining (TODO is empty). The result is a sequence of tasks each completed. Notice that these task states progressively completed reflect the anticipation of structure corresponding to the semantic interpretation of the string, and they are not a semantically-blind assignment of syntactic structure. As the proto-type sketch in (31) suggests, the system is an inferential building of a feature-annotated tree structure. One of its distinguishing properties is that this articulation of the process of building a tree structure is itself the syntactic engine, as driven by lexical specifications. There is no externally defined syntactic mechanism over and above this. Unlike other syntactic models, the system combines object level information pertaining to the structure and interpretation of the resulting formula with metalevel information about the process of establishing it. So there is DECLARATIVE structure which indicates what the content is (type plus formula plus tree node). And there is IMPERATIVE structure which indicates what remains to be done.

2.3. The logical languages: Formula-language, Label-language,Tree-language, language of Declarative Units

To express this degree of richness, we need a number of correlated languages, the language of the formulae, the language of the labels, the language of the tree-nodes, and the composite language which combines these various sub-languages in defining the structure within which the annotated tree node is built up.

2.3.1 The language of the formulae

The expressions of the formula language are expressions of an extended quantifier-free lambda calculus .

Terms are predicate constants 'sing', 'see', 'smile', etc and a range of lambda expressions; individual constants 'John', 'Mary', etc.[3] The quantifier-variable notation of predicate logic is replaced by epsilon (equivalent to $\exists$) and tau (equivalent to $\forall$) terms, each of type **e**, and each with a restrictive clause nested within the term itself. For example, *a man* is projected as 'ϵx(x, man x))' of type **e**. In addition, there are a range of specialised "m-variables" which act as metavariables, and have to be replaced during the computation of the propositional formula. These are annotated to indicate the expression from which they are projected: eg. WH (to be read as 'gap'), u-pro. In all cases such expressions are taken as placeholders and operations map these expressions onto some expression of the formula language which replaces them. More formally:

Definition of the Language of Content Formulas:

Terms and Formulas of the language L_C for a non-empty set C of quantifier operators are built from individual variables in V, predicate variables in P, individual constants in A, m-variables in M, and predicate constants in P as follows

1. the set T_C of L_C-terms is defined by
    (a) all elements of A are in T_C
    (b) all elements of M are in T_C
    (b) if $x \epsilon$ V, $a \epsilon A$, $c \epsilon C$ and $\phi[a/x]$ in $FORM_C$. then $(cx,\phi) \epsilon T_C$

2. The set Λ_V of lambda terms of L_C is defined by

    if $X \epsilon V$, $a \epsilon A$, $\phi[a/x] \epsilon FORM_C \cup T_C$ then $\lambda X.\phi \epsilon \Lambda_C$, if $X \in P$, $a \in A$, $\phi[a/x] \in FORM_C \cup T_C$ then $\lambda X.\phi \in \Lambda_C$

3. The set $FORM_C$ of L_C formulas is defined by

(a) if $t_1,....t_n \in T_C$ and P an n-place predicate of L, then $Pt_1...Pt_n \in FORM_C$
(b) if ϕ, $\phi' \in FORM_C$ then $\phi\#_i\phi' \in FORM_C$, where $\#_i \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$.

Notice that λ may bind variables within the scope of elements of C but not vice versa. Moreover in elements of T_C and $FORM_C$, there occur no free variables.

Example:We will represent the content of a word like *some* by

$\lambda P(\epsilon x, P)$

It is a function requiring an instance of (the type of) P (for instance *man*) to become a complete object (of type e), i.e., epsilon terms.

$(\epsilon x, Man)$

2.3.2 **The language of the labels**

Labels present all information that drives the combinatorial process. These include:
(i) the logical types e, t, <e,t>, <e,<e,t>>, etc. These are represented as type-logical formulae **e, t, e→t, e→(e→t)**,.., corresponding to the syntactic categories DP, IP, intransitive verb, transitive verb, and so on, and are displayed as:
Typ(e), Typ(e→t), etc. We may also allow <cn> as a type to distinguish nouns from intransitive verbs.
(ii) Tree node identifiers.
These define the tree position of the declarative unit under construction from which its combinatorial role is determined (see below).
(iii) Features distinguishing discrete sentence types such as +Q for questionhood
We might also add a further range of features such as

case features defining the combinatorial role of the formula directly, eg a diacritic "Use me last" defining subject through an ordering on steps of Modus Ponens which itemises one premise of type e as to be used only when the output is of type t (cf. Gabbay & Kempson 1992).
tense features defining tense as a label to a formula of type t (following Gabbay 1994, Finger & Gabbay 1993).

All issues of case and tense we leave to a later occasion, here allowing the set of label-types to be open-ended.

2.3.2.1 Tree Language

The tree node identifiers and other properties of the nodes are described by a tree-node logic, LOFT (Logic of Finite Trees), a logic for a propositional modal language with 8 modalities (Blackburn & Meyer-Viol 1995):
<u> P P holds at my mother
<d> P P holds at a daughter of current node
<l> P P holds of a left-sister of current node
<r> P P holds of a right-sister of current node
In addition to the operator <x>, x ranging over u, d, l, r, its dual [x] is defined:
$[x]P \models \neg\langle x\rangle\neg P$
The connectives are standard: &, V, →, ¬, and there is also ⊤ (an arbitrary tautology), and ⊥

(an arbitrary falsum). This is a modal logic which allows nodes of a tree to be defined in terms of the relations that hold between them. For purposes of this paper, the system can be displayed by example. In (32) for example, from the standpoint of node n, where Q holds, <d>S and >d>(T & V) hold; from the standpoint of n', where S holds, <u>Q and <r>(T & V) hold; and from the standpoint of n", where (T & V) holds, <u>Q and <l>S hold:

(32)

```
                n      Q
              /   \    <d>S
            /       \  <d>(T & V)
          /           \
S      n'              n"     T & V
<u>Q                          <u>Q
<r>(T&V)                      <l>S
```

The language can have constants which may be defined as required:

Constants, eg.

m	[u]⊥	root node ("nothing is above me")
L	<u>m & [l]⊥	left-most daughter

Note the use of the arbitrary falsum. The arbitrary tautology may also be used as in:

<d>⊤ "I have a daughter"

Also manipulated are Kleene star operators for defining the reflexive transitive closure of the basic relations:

<x>*P := P v <x><x>*P

[x]*P := ¬<x>*¬P

For example:

<d>*X	Some property X holds either at some node n or at <d>n or at <d><d>n etc. "Some property X holds either here or at a node somewhere below here"
<u>*Tn(m)	Either here or somewhere above me is the rootnode m. This property is true of all nodes in a tree dominated by a node identified as having m as root

This use of the Kleene star operator, which emerges as a simple consequence of developing the language of tree node features as a logic, provides a richness to syntactic description not hitherto exploited (though cf. Kaplan & Zaenen 1988 for its use in defining a concept of functional uncertainty defined over LFG f-structures): it provides the capacity to specify a property as holding either at some given node or at some node elsewhere in the tree in which it is contained. It is this relatively weak disjunctive specification which we shall use to characterise *wh* or other expression initial in a string, whose properties and their projection within the string are not dictated by its immediate neighbours. The effect will be that not all expressions in a sequence fully determine their structural role in interpreting the string from that position in the string.

2.3.4 Declarative Units - The language of database entries

The combination of these separate languages can now be defined. Words are defined as providing all the building blocks, the initial premises and extra annotations or projection of tree structure as indicated - in the form of "declarative unit formulas" (DU formulas).

These specify annotations on tree nodes of the form
<d>(Typ(e) & Form(John))
"AT my daughter is a formula 'John' of type e"
Task states contain sets of DU formulas.

Each word w is associated with a set Y of DU-formulas.
Lex(w) = Y
eg Lex(*John*) = {Typ(e), Form(John)}
Through such specifications as these, words project meta-level instructions on the building and derivation of the required configuration.

Definition 2 (Language of Declarative Units)

The language of declarative units is a first-order language with

Non-logical Vocabulary:

(1) a denumerable number of sorted constants from Lab$_i$ for $i \leq m$, where $L_i = < \mathrm{Lab}_i, R_i^1, \ldots R_i^m >$ structures the set of feature values in Lab$_i$

(2) monadic predicates Form ('Formula'), Typ ('Type'),Tn('tree-node'), $C_i, i \leq n$ and identity ' =',

(3) modalities},u., (up), <d> (down), <l>(left, <r> (right) and their starred versions <d>*, <u>*, <r>*, <l>*. We will use the abbreviation $<\#>^n$, $n \in IN$ for <#>,....<#> n times.

Formulas:

1. If $j \in L_C$ then Form(j) is an (atomic) DU-formula.
   If $k \in \mathrm{Lab}_1$ then Typ(k) is an (atomic) DU formula.
   If $k \in \mathrm{Lab}_2$ then Tn(k) is an (atomic) DU-formula.
   If $k \in \mathrm{Lab}_i$, $2 < i \leq n$, then $C_i(k)$ is an (atomic) DU-formula.
   If t,t' are variables or individual constants, then t=t' is an (atomic) DU-formula.
   $\top$, $\perp$ are also formulae.
2. If ϕ and ψ are DU-formulae, then $\phi \#_i \psi$ is a DU-formula for $\#_i \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$.
   If x is a variable and ϕ a DU-formula, then $\forall x \phi$ and $\exists x \phi$ are DU-formulas.
   If M is a modality and ϕ a DU-formula, then $M\phi$ is a DU-formula.

With this composite language system, inference rules which characterise the transition between input state and final outcome can now be set out. All Inference operations are defined as meta-level statements - in terms of DU formulas and relations between them. For example, applications of the rule of Modus Ponens

(33)
$$\frac{<e, \ldots>: \phi \qquad\qquad <e \rightarrow t, \ldots> : \psi}{<t, \ldots> : \psi(\phi)}$$

for declarative units become a metalevel statement licensing the accumulation of information at nodes in a tree structure, represented as:

(34) Modus Ponens for DU-Formulas

$$\frac{\langle d\rangle(\mathrm{Typ}(e) \wedge \mathrm{Form}(\phi)) \wedge \langle d\rangle(\mathrm{Typ}(e\to t) \wedge \mathrm{Form}(\psi))}{\mathrm{Typ}(t) \wedge \mathrm{Form}(\psi(\phi))}$$

An item has Typ Feature t and Form Feature $\psi(\phi)$ if it has daughters with Typ Features $e\to t$ and e and Form Features ψ and ϕ respectively. Controlled Modus Ponens is then a straightforward generalisation. For instance,

(35) $$\frac{\langle d\rangle(X(l) \wedge \mathrm{Typ}(e) \wedge \mathrm{Form}(\phi)) \wedge \langle d\rangle(\mathrm{Typ}(e\to t) \wedge \mathrm{Form}(\psi))}{\mathrm{Typ}(t) \wedge \mathrm{Form}(\psi(\phi))}$$

where Modus Ponens is restricted to a daughter with the feature 'X(l)'. In general in this modal logic, a rewrite rule $Y_1,\ldots, Y_n \Rightarrow X$ gets the form

(36) $\langle d\rangle Y_0 \wedge \ldots \wedge \langle d\rangle Y_n \to X$

2.4 Task States:

A Task state is a description of the state of a task.A task is completely described by determining the goal, what has been constructed, achieved, what still has to be done, and the location in the tree. So the four feature dimensions of a task state are

Goal (G). Values on this dimension are the semantic types in the label set Ty.
This feature indicates which semantic object is under construction.

Tree Node (TN). Values are elements of the label set Tn. The 'top-node' in Tn will be denoted by 1.This feature fixes the location of the task in question within a tree structure.

Discrepancy (TODO). Values are (finite sequences of) DU-formulas. This dimension tells us what has to be found/constructed before the goal object can be constructed.

Result (DONE). Values are lists, sequences, of DU-formulas. These values will be subsets of a set of DU-formulas corresponding to a typed expression under interpretation.

We will represent the task state TS(i) by

TN show G TODO
DONE

Example

(37) TN show e TODO <d>(Typ(e→t))
<d>(Form(λP.(ϵx, P)) & Typ((e→t)→e)))

We can distinguish three kinds of task states

(i) Task Declarations

TN show G TODO G
DONE $\emptyset$

The Task Declaration. Nothing has yet been achieved with respect to the goal G. Everything is still to be done. Analogously to the description of declarative units we can represent the above task state as a list of feature-value statements as follows.

{Goal(G), Tn(TN), Todo(G), Done(0)}

(ii) Tasks in Progress

TN show G TODO β
α

In the middle of a task. If things are set up right, then $\alpha\beta \Rightarrow G$.

{Goal(G), Tn(TN), Todo(β), Done(α)}

The value of DONE gives an element of the incremental model, possibly incompletely specified. The value of TODO gives a demand associated with this element that still has to be satisfied

(iii) Satisfied Tasks

TN show G TODO $\emptyset$
α

A Satisfied Task. There is nothing left to be done. This can be represented as DU(i) for some declarative unit functions DU. Soundness of the deductive system amounts to the fact that the goal G can be computed, derived, from α in case TODO is empty.

{Goal(G), Tn(TN), TODO($\emptyset$), DONE(α)}

From a different perspective we can consider the state

TN show G TODO $\emptyset$
DONE

as constituting an association between a node in a tree and a Labelled object decorating that node.

Tree node n ← Task i → Task State

Notice that this can be seen as a tree node decorated by some feature structure plus an unsatisfied demand.

In the course of a parse we might have a Task State with the Tree Node feature undefined. Given that the tree node feature identifies the function-argument structure of the declarative unit formula in question, such Task State with no Tree Node feature would have no specified functional role in the compilation of propositional content. as for example in topic structures:

$\emptyset$ show G TODO β

DONE α

Notice that this is not equivalent to the specification of a task as

|<u>*m show G TODO β|
DONE α

In the latter case, the task state specifies that DU formula α contributes at some fixed, albeit unspecified, point within a tree of root node m.

Parse States

A parse state consist of a pair : a bookkeeping device which gives a value for the parsing pointer, the string counter which represents its location in the string, and some length l of the string of tasks completed; and a sequence S of tasks states of length l. Details of this indexing system we take to be central to any parsing implementation, but in this paper, we specify such details only where necessary.

2.5 Dynamics: The Basic Transition Rules

The dynamics of the parse process consists then of reaching a final parse state starting from a begin state where the transitions in the process are licensed, driven, by the words in a string. Concretely, the dynamics of the parsing process, is the dynamics of demand satisfaction. This sequence of parse states can be seen as a tree setting out the steps by introduction that will be necessary to complete in order to derive a formula corresponding to an interpretation of the string. The tree corresponds to a skeletal anticipation of the internal semantic structure of the resulting propositional formula, and not to a tree-structure for the input sequence of words. Indeed there is no necessary one-to-one correspondence between the individual linguistic expressions in the string and nodes of the tree.

Basic Transition Rules

In the following the symbols X,Y,Z,.. will range over individual DU-formulas, the symbols α, β,... will range over (possibly empty) sequences of such formulas, D, D',... will range over (possibly empty) sequences of tasks, and w_i, w_{i+1},... will range over words.

The start of a parsing sequence is a single task state, the Axiom state. The last element of such a sequence is the Goal state. The number of task states in a parse state grows by applications of the Subgoal Rule. Tasks become satisfied by applications of the Scanning and the Completion Rules.

The building up of a total interpretation is the incremental building of a tree through a succession of task states from the initial Axiom state to the Goal state.

(38) AXIOM for a string of length k, with string counter 1,

|0 SHOW t TODO Typ(t) 0|

(39) GOAL for string of length k, and string counter k,

|0 SHOW t ∅ j| . D
Typ(t)

where Elements of D are task states with fully specified DU formulas.
In other words all task states are satisfied, with empty TODO, for all expressions k and task states 0-j, with the final state a satisfied task state of type t. Intermediate steps involve paired rules of Introduction/Elimination, and Subordination /Completion, with a rule of scanning which introduces specifications from the lexicon under a matching condition. We give SCANNING first:

(40) SCANNING

For some string position s, task state p, and tree node i

$$\begin{array}{c} D \quad \boxed{i \quad \text{SHOW } X \quad \text{TODO } U,\beta \quad p} \quad D' \\ \alpha \\[1em] D \quad \boxed{i \quad \text{SHOW } X \quad \text{TODO } \beta \quad p} \quad D' \\ \alpha, Y \end{array}$$

if $\text{LEX}(w_{s+1}) = Y, U \in Y$

This rule licenses the introduction of material from the lexicon as long as it satisfies the condition specified in TODO. These conditions are established by the rule of INTRODUCTION.

(41) INTRODUCTION

For some string position s, task state p, and tree node i

$$\begin{array}{c} D \quad \boxed{i \quad \text{SHOW } X \quad \text{TODO } Z,\beta \quad p} \quad D' \\ \alpha \\[1em] D \quad \boxed{i \quad \text{SHOW } X \quad \text{TODO } \langle d\rangle Y_0, \ldots.\langle d\rangle Y_n., \beta \quad p} \quad D' \\ \alpha \end{array}$$

where $Y_0, \ldots. Y_n \Rightarrow Z$

This rule licenses the introduction of a set of new subgoals as long as there is a corresponding elimination rule which takes the parse back to the subgoal which constitutes the input to the present task. Introduction itself sets up the need for a rule transferring the requirement on the daughter to a TODO at which the requirement has to be met:

(42) SUBORDINATION

For some string position s, task state p, and tree node i

$$\begin{array}{c} D \quad \boxed{i \quad \text{SHOW } X \quad \text{TODO}(\langle d\rangle(\text{Typ}(Y)), \beta) \quad p} \quad D' \\ \alpha \\[1em] D \quad \boxed{i \ \text{SHOW } X \ \text{TODO}\langle d\rangle(\text{Typ}(Y)), \beta) \ p} \quad \boxed{k \ \text{SHOW } Y \ \ \text{TODO}(\text{Typ}(Y)) \quad p+1} \quad D' \\ \alpha \end{array}$$

where $R_d(i,k)$ (R_d is the relation R holding between a node i and a daughter k)

Subordination is one of a pair of rules which license the transition from a task to be carried out on a daughter to the creation of a task corresponding to that daughter. Like Introduction, it relies on a converse rule, COMPLETION, which defines the transition back to the mother:

(43) COMPLETION

For some string position s, task state p, and tree node i,

D   i SHOW X TODO Y,β   D'   k SHOW Y TODO ∅   D''
α $U_0,...,U_n$
D   i SHOW X TODO β   D'   k SHOW Y TODO ∅   D''
$\alpha(<d>(U_0,...,U_n))$ $U_0,...,U_n$
where R_d (i,k)

COMPLETION then feeds ELIMINATION, the rule twinned with INTRODUCTION:

(44) ELIMINATION

For some string position s, task state p, and tree node i

D   i   SHOW X   TODO β   p   D'
α , $<d>Y_0,.....<d>Y_n$
D
i   SHOW X   TODO β   p   D'
α , Z_k
where $Y_0,......Yn \Rightarrow Z$

This rule effects the converse of Introduction, so that type specifications of subtasks are progressively derived. This rule is the generalisation of the meta-statement form of Modus Ponens (cf. 36).

These rules together project a succession of task states as set out schematically in (31). The overall task is thus to SHOW t by successive steps of Introduction, Subordination, Scanning or Introduction, Completion, Elimination. This is the minimal set of rules for licensing parse states within which deduction will take place (cf. Milward 1993).

Definition: Executions

An execution d(s) with respect to a string of words s is a sequence d(s) = $<d_1, ..., d_k>$ of parse states such that

1. d_1 = a task state with the string counter at 0, for string of length l

Tn(0)   SHOW t   TODO(Typ(t))   1

2. for all i : $0 < i \leq k$, d_{i+1} is the conclusion of an application of one of the rules with premise i.

An execution $d_1, ...,d_k$ is successful for string s if d_k contains only satisfied task states.

If d_1, ...,d_k is a successful execution then because d_1 started as the Axiom (TODO(Tn(0)) $\in d_1(0)$), the parse state d_k must contain

Tn(0)	SHOW t Typ(t)	TODO(∅) k

That is, DONE(Typ(0)) $\in d_k(0)$.

These are the basic set of general rules driving the parsing process. There are other rules to add to this set. In particular there are the rules associated with subordination, and rules specifically associated with *wh* expressions. We also presume on rules which relate sequences of tasks states as units. These give rise to linked task-sequences, to which we shall return, in considering adjunction.

2.6 Modelling the partial nature of Natural Language Content

Before increasing the complexity with such additional rules, we indicate how we model the gap between lexically specified content and its assigned interpretation within the context of some given string, as this is the heart of any account of how utterances are interpreted in context.[4] The most extensively studied phenomenon involving such asymmetry is anaphora. In this model, we take the underspecification of the input specification associated with pronominal anaphora a step further than in many other analyses. We assume that pronouns are invariably inserted from the lexicon with a single specification of content, and that any bifurcation into bound-variable pronoun, indexical pronoun, E-type etc, is solely a matter of the nature of the contextually made choice, context here being taken to include logical expressions already established within the string under interpretation. Accordingly, pronouns are projected as meta-variables with an associated procedure which imposes limits on the pragmatic choice of establishing an antecedent expression from which to select the form which the pronoun is to be taken as projecting:

Lex(*he*) = {Typ(e), Form(u-pro), Gender(male),<u>Typ(t), ...}

(Notice how the condition on the mother of the formula projected from *he* is in effect a feature-checking device, licensing the occurrence of the pronoun within a particular frame.)

Instantiation of the m-variable 'u-pro' is on-line. It must be selected only from identified formulae, where an identified formula is either a formula in some satisfied task (i.e. in the done box of a task with empty TODO and identified tree node) or a formula ϕ which has been derived elsewhere in the discourse sequence. For a pronominal of type e, there is a further restriction that the formula selected as providing its value may not occur within the same t-domain within which u-pro is projected. This is expressed as a side-condition, given here only informally. Note the meta-level status of this characterisation of pronouns. Anaphora resolution is defined not as a task of reference assignment defined in semantic terms, but as a selection process defined over available representations. The nature of this choice will determine whether the denotative content of the pronoun relative to the assigned interpretation is that of a variable, a referential term, etc.

2.6.1 Underspecification of tree configuration

More unorthodox than the recognition that a single pronoun has a single lexical specification which by enrichment of its input specification becomes a bound-variable, constant, etc, is the claim that expressions in a string may also underspecify the role the expression is to play in the compilation of interpretation for the string. This is the substance of the analysis of initial *wh* expressions. *Wh* expressions, we claim, project a specification which is not merely incomplete in content, but also structurally incomplete, lacking any indication of how the expression is structurally related to the the sequence of expressions that follow. We reflect this with the specification of the initial position as (45):

(45)

Tree node	GOAL	SUBGOAL	Task		Tree node	GOAL	SUBGOAL
m	SHOW t Label_S(+Q)	TODO Typ(t)	j	...	<u>*m	SHOW e Form(WH) & Typ(e)	TODO ∅

(45) displays the projection of an initial t task at some node m identified as a question, but with everything still to do, except that a completed e task has been added, lacking merely the specification of where in the tree it holds. This unidentified task contains the formula WH ("the gap"). WH is an m-variable, either retained as a primitive term to be resolved by the hearer and so incomplete in content, or in relative clauses resolved by unification with the formula projected by the adjoined head. "<u>*m" is an abbreviation for

<u>*(Tn(m))   The root node is either here or somewhere above me

In other words, the structural position of 'Form(WH) & Typ(e)', hence its function-argument role in the propositional formula under construction, is not fixed at this juncture in the parsing process. Seen as an on-line target-driven parsing task, by a single step of inference we can shift to the perspective provided from the current node m, and display the same information as:

(46)

Tree node	GOAL	SUBGOAL	Task		Tree node	GOAL	SUBGOAL
m	SHOW t <d>*(Form(WH) & Typ(e)) Label_S(+Q)	TODO Typ(t)	0	...	<u>*m	SHOW e Form(WH) & Typ(e)	TODO ∅

The <d>* form of specification also projects the information that somewhere in the tree needed by the node m is a node but does so from the perspective seen from node m. Note the consequence that though the node m itself stands in the relation of mother to its daughters the node at which 'Form(WH) & Typ(e))' holds does not stand in any such relation to daughters of m. The lack of any such relation between the initial *wh* expression and other expressions dominated by the sentence-node analogous to the structural notion of c-command (a relation which is intrinsic to the relation between binding operator and expressions falling within its scope), automatically follows, for the node at which the properties projected by the *wh* expression hold is not established from that initial position. So already we have a way to predict the scope idiosyncrasy of initial *wh* expressions that they freely allow narrow scope effects with respect to expressions which follow them (listed as problem (1) above).

We have not yet added any account of why the properties of *wh* might get carried down from one clausal domain to another. But consider by way of example the evaluation of '<d>*(Form(WH) & Typ(e)) in the DONE compartment of some node whose TODO lists 'typ(t)' (as at the point of projection of the <d>* feature. The definition of '<d>*X' recall is:

'X v <d><d>*X'
(="X holds either here or at some daughter")
On the mere assumption that information at nodes is invariably locally checked for consistency, the mismatch between TODO typ(t) and the first disjunct in
(Form(WH) & Typ(e)) v <d><d>*(Form(WH) & Typ(e))
will lead to the deduction that at that node what holds is only the second disjunct, viz:
<d><d>*(Form(WH) & Typ(e))
The attribution of this feature to that node will have to be checked for consistency with what is entered into its daughter node. Hence at its daughters the feature '<d>*(Form(WH) & Typ(e))' will duly be checked. But this guarantees that the attribution of the feature '<d>*(WH & Typ(e))' is checked for consistency at each node between the node at which it is introduced (by the initial *wh* expression) and the node at which the first disjunct is matched by some given TODO and the expression resolved.

The resolution of this disjunctive specification and an unfixed node <u>*j is achieved at node i by some TODO specification associated with a task state being taken as satisfied by the presented floating constituent, whose node characterisation is thereby identified (WH RESOLUTION):

(48) WH RESOLUTION
For some string position s, tree node i, $R_{<u>*}$m and task state p,

D | i SHOW e TODO(Typ(e)), j | | <u>*m SHOW e TODO0 | D'
<d>*(Form(α) & Typ(e) FORM(α) & Typ(e)

D | i SHOW e TODO 0 j | D'
Form(α) & Typ(e)

where Typ(x) ∈ Lex(s) & x≠ type...→e & x≠ e

The side condition is a restriction that the type of the current word in the string must neither meet the TODO specification directly, nor set up a type specification which a sequence of Introduction and Elimination steps would satisfy. This guarantees that such resolution only takes place when faced with a clash between the TODO specification and the lexical specification of the next word in the sequence.

We now set out two examples. First is the specification of input state and output state for the string *Who does John like?*

(49) Who does John like?

INPUT TASK STATE

m SHOW t TODO Typ(t) 1

SHOW t TODO Typ(t) 1 <u>*m SHOW e TODO∅

Who Label$_S$(+Q) & Typ(t) Form(WH) & Typ(e)

<d>*(Form(WH) & Typ(e))

OUTPUT TASK STATE

.... m SHOW t TODO 0 1

Form(like(WH)(John) & Label$_S$(+Q) & Typ(t))

Notice how the lexical specification of *who* simultaneously projects information both about its mother node (that it is a question) and about some unplaced constituent, presenting a disjunction to be resolved via consistency checks with a sequence of daughter nodes. The finally derived state with the t target duly completed has this input disjunction resolved.

(50), our second example derivation, specifies a characterisation of the task state following the projection of information following *think*, displaying the disjunctive specification associated with *who* being carried down to information projected by the string making up the subordinate clause through inconsistency between the type of the *wh* element and that assigned to each intermediate right-branching daughter, with the point at which the information projected by *who* still not fixed:

(50) Who do you think Bill likes?:

TASK STATE following entry of *think*:

m SHOW t TODO Typ(t) j <u>*m SHOW e TODO∅

Who Label$_S$(+Q) & Typ(t) Form(WH) & Typ(e)

<d>*(Form(WH) & Typ(e)

n=<u>m n SHOW e TODO ∅ j+1

you Form(you) & Typ(e)

.

.

p SHOW e→t TODO <d>Typ(t) j+2

<d>*(Form(WH) & Typ(e)

think <d>(Form(think) & Typ(t→(e→t))

q SHOW t TODO Typ(t) j+1

<d>*(Form(WH) & Typ(e)

3. Crossover: The Basic Restriction

We are now in a position to present the basic crossover restriction, to wit that in questions, pronouns can never be interpreted as dependent on the preceding WH expression unless they also follow the gap. This restriction is uniform, and runs across strong and weak crossover configurations (cf. examples (4)-(9) repeated here):

(4) *Who$_i$ does Joan think that he$_i$ worries e$_i$ is sick?
(5) *Who$_i$ does Joan think that his$_i$ mother worries e$_i$ is sick?
(6) *Whose$_i$ exam results$_j$ was he$_i$ certain e$_j$ would be better than anyone else's?
(7) Who$_i$ does John think e$_i$ worries his$_i$ mother is sick?
(8) Who$_i$ does Joan think e$_i$ worries that he$_i$ is sick?
(9) Whose$_i$ exam results$_j$ e$_j$ were so striking that he$_i$ was suspected of cheating?

This is directly predictable from the <x>* form of characterisation of initial *wh* expressions. Formulas characterised as holding at some node <u>*m ("somewhere above me is the node identified as m") do not have a fixed position in the configuration, and so are not identified unless there is some independent means of identification. In particular, since any such formulas are not identified as holding at the position m at which the information is projected, they are not available from that position to serve as an antecedent for the purpose of pronominal resolution. The effect of WH resolution when it later applies is indeed to determine the position within the configuration at which the properties projected by the *wh* expression should be taken to hold. Such features thus become available for pronominal resolution only after the gap has been projected. In this way, the system is able to predict that *wh*-expressions in questions do not provide an antecedent for a following pronominal until the gap (=WH) is constructed. Hence the primary crossover restriction *WH-Q..pronoun..gap (for both weak, strong, and extended strong crossover data).

3.1 Crossover and Relative Clauses

When we turn to relative clauses, we have data which are problematic for all accounts in terms of operator-gap binding, as these demonstrate that the phenomenon is context-sensitive (Lasnik & Stowell 1991, Postal 1993). In some contexts, the primary crossover restriction is suspended altogether, in others it remains in force. This is inexplicable given an analysis of the primary crossover restriction solely in terms of the relative positions of the three elements WH, pronominal, and gap, as any binding precluded by one such configuration should continue to be excluded no matter what environment the configuration is incorporated into as a subpart. On the other hand, if the lack of availability of the information projected by the *wh* expression is due to its lack of identification from the position in the string in which the expression occurs, then should there be some other means of identifying that information, this externally projected information may provide a means of resolving the anaphoric under-specification. This is the approach we shall take, following the account of relative clause interpretation.

3.1.1 Relative Clauses as Linked Databases

As so far set out, the system is a pure type-deduction system, with information projected solely through the paired occurrence of introduction and elimination rules. When we turn to relative clauses, a different system is proposed. For this purpose we introduce the concept of linked database. Linked databases are arbitrary pairs of databases which overlap with respect to a single formula, defined within one database to be identified with a formula in the second (host) database. The LINKED databases are two independent domains linked through identification of a variable. Schematically, in (51), B_1 is unified with A_1, C_3 unified with A_3, and so s is LINKED to both s' and s":

(51) s: t_1:A_1, t_2:A_2, t_3:A_3

Unify B_1/A_1 Unify C_3/A_3

↑ ↑

s': t_1':B_1, t_2':B_2 t_3':B_3 s": t_1":C_1, t_2":C_2 t_3":C_3

It is this pattern of linked databases that we adopt for relative clause modification. Following the same schematic display, we want:

(52) John who adores Mary left Sue.

s_1: **e**:John **e→(e→t)**:leave **e:**Sue

↑ LINK(s_1(John), s_2(WH), WH/John)

s_2: **e**:WH/John **e→(e→t)**:adore **e**:Mary

The relative clause is interpreted entirely independently of the host sentence, *John left Sue*, within which it is inserted, except that the interpretation of *who* is fixed as identical to whatever is the interpretation of *John*. Our databases here will be sequences of task states, each deriving a DU formula of the form 'Form(P), Typ(t)', which we call task sequences.

The relation LINK itself is defined for a pair of task sequences one of which contains a DU-formula 'form(WH)':

(53)

LINK(Δ({X, Typ(e), Form(α)}) , Δ'({Y, Typ(e), Form(WH)}), WH/α) ⇔

For a task sequence Δ containing at least 1DU-formula of the form

{X, Typ(e), Form(α)}

Δ' a distinct task sequence containing at least 1 DU-formula of the form

{Y, Typ(e), Form (WH)}

α unifies with WH, α as the most general unifier replacing all occurrences of WH**.**

This definition of a LINKED database is static, and cannot be the sole basis for accounting for relative clause construal. We define a lexical entry for WH as a relativiser which adds merely the information that WH is to be the basis for LINKage between the database it initiates and some host database

(54)
INPUT TASK STATE
D D'
k show e TODO<ADV>Typ(t)) m SHOW t TODO Typ(t)
α

OUTPUT TASK STATE
D
D'
k show e TODO<ADV>Typ(t)) m SHOW t TODO Typ(t) <u>*m SHOW e TODO∅
α Form(α) & Typ(e)

Where R_{LINK}(k,m)

In the process defined as (54), the content α is carried from the host task-state p into the independent task sequence initiated by the task state p+1 with its goal of SHOW t. The value of *wh* is therefore identified as identical with that of its head by definition. However the specification is nevertheless characterised as an unfixed task state, which leads by inference to the disjunctive specification <d>*(Form(α) & Typ(e)) holding at the new task state of type t, and this form will as before feed into a system of local consistency checking giving rise to percolation of the specification <d>*(Form(α) & Typ(e)) down through the tree. The difference is that in this case, the value α is no longer that of WH but some substituend α.

This gives rise to the following prediction. With WH identified independently by a given antecedent α, and this information identifying it then carried down through the structure, we predict locality clashes with the restriction on pronoun construal (principle B effects). The locality restriction on pronominal, recall, was:
Any pronominal u-pro associated with type e, may not take as value for its formula any identified formula of type e within the same t-domain within which u-pro is contained. This restriction has to be met at the point when the information from the pronominal is entered into the tree configuration, for this is an online choice needed to complete the identified task state. [5] Given the identification of the unfixed node as 'Form(John) & Typ(e)' in (55)-(57), the information carried down the structure through the <d>* specification to be checked at each sub-task is the information that either the DU-formula itself holds at the sub-task or it holds at some daughter. It is this information to which the locality restriction on the pronominal is sensitive. If both the carried down <d>* specification and the specification of pronoun are of type e, then given the restriction on pronominal construal that the pronoun may not identify with a premise of type e in the same set of taskstates yielding some conclusion of Typ(t), an interpretation of the pronoun on the basis of that disjunctive specification is precluded. If either the pronominal being projected or this specification is within a determiner and hence not functioning as a minor premise of type e, then resolution of the pronominal using that disjunctive specification is not precluded. Hence the asymmetry between (55) and (56)-(57):

(55) *John$_i$ who$_i$ Sue thinks he$_i$ worries e$_i$ will fail, left.
(56) John, who$_i$ his$_i$ mother had ignored e$_i$, fell ill during the exam period.
(57) John, whose$_i$ exam results he$_i$ had been certain e$_j$ would be better than anyone else's, got a nasty shock when they came out.

We give (58) as an intermediate characterisation of (55) displaying the information made available at the onset of the clause initiated by *he*. In (58), as the projection of *he* indicates, both pronominal and the information from the identified *wh* expression are of type e. Moreover, by inference, at each node m, n, q, r, the property '<d>*(Form(John) & Typ(e))' holds, guaranteeing direct local availability of the information 'Form(John) & Typ(e)'. Hence the pronoun, given the locality restriction on its resolution, will not be identified using the formula 'John'. However,with either the pronominal or the *wh* expression as determiner, though the locality constraint is checked, there will be no such violation:

(58) Intermediate characterisation of (55):
John

i SHOW e TODO Typ(e) 1
Form(John) & Typ(e)

who
m SHOW t TODO <d>e, <d>(Typ(e→t)) 2
<d>*(Form(John) & Typ(e))

<u>*m SHOW e TODO∅
Form(John) & Typ(e)

Sue
n SHOW e TODO ∅ 3
Form(Sue) & Typ(e))

.
.

thinks
q SHOW e→t TODO Typ<d>(t) 4
<d>(Form(think) & Typ(t→(e→t))
<d>*(Form(John) & Typ(e)

r SHOW t TODO Typ(t) p+3
<d>*(Form(John) & Typ(e))
he <d>(Form(u-pro) & Typ(e))

Choice of value for u-pro ≠ John.

At node r, the DU-formula 'Form(John) & Typ(e)' is declared to hold either at r or somewhere below r. This projection of 'Form(John) & Typ(e)' as either holding at r or at some node below r is sufficient to preclude the identification of the metavariable 'u-pro' as 'John', for such an identification would violate the locality restriction on identifying 'u-pro' that it may not be identified with any other premise of type e within the same local domain of type t. Such violation will not occur if either the pronominal of the WH is nested within a determiner, as in these cases, ei˝ther pronominal or antecedent will not be functioning within the relevant domain as projecting the type e.

The general pattern predicted by the present analysis is that anaphora resolution in nonrestrictive relatives will pattern in parallel with sequences in which the order of name plus pronominal parallels the order *wh*-pronoun. So (59) is predicted to parallel (60) (both *wh* and name precede the pronominal *his*), (57) with (61) (both *John's exam* *results* and *whose exam results* precede the pronominal *him)*, and similarly (62) with (63):

(59) The bastard$_i$ who$_i$ his$_i$ mother worshipped...
(60) The bastard$_i$ worships his$_i$ mother.
(61) John's exam results gave him a nasty shock.
(62) *Joan$_i$, behind whom$_i$ she$_i$ looked e$_i$ nervously, coughed.
(63) *Behind Joan$_i$. she$_i$ looked nervously.

The one further type of case is that of restrictive relatives, which, apparently puzzlingly, behave more like questions than their nonrestrictive counterpart. The data here are somewhat unclear (hence the term "weak crossover"). Some examples seem to display a weak crossover effect: others do not. (64)-(65) seem to preclude an identification of the pronominal as dependent on the nominal head:

(64) Every child$_i$ who$_i$ his$_i$ brother ignored e$_i$ throughout his party was clearly ill at ease.
(65) At least one woman$_i$ who$_i$ her$_i$ cousin refused to cooperate with e$_i$ reported her to the authorities.

Judgments are much less clear in:

(66) Noone$_i$ who$_i$ his$_i$ mother systematically ignores e$_i$ makes a good parent
(67) Every patient who$_i$ their$_i$ doctor cannot make time to see e$_i$ tends to get sick more often.
(68) If a snooker team$_i$ who$_i$ their$_i$ coach may have bullied e$_i$ into playing with us turns up, please be nice to them.

In (67) and (68) it is unclear whether *their* is genuinely a bound-variable construal (rather than some group construal (*their* construed as 'the patients' in (67), as 'the members of the team' in (68)). In the face of this somewhat unclear data we adopt the view that restrictive relatives display crossover effects much as in questions. The explanation of this correlation between one type of relative and questions rests on the account within the LDS$_{NL}$ analysis of quantification, and we cannot do more than allude to it here. Determiners in this framework are not analysed as themselves projecting fully specified quantifying terms, but merely input procedures for creating such terms. Thus, for example, the determiner *a* is characterised as projecting the weakly specified content formula Form($\lambda P.\epsilon x\ Px\ \&\ Ryx$), where x is construed as dependent on some additional entity y, and the value of both y and R have to be chosen as an anaphoric-like choice, in the light of the full specification of the nominal head and any other restrictive predicates. In this form, the determiner+nominal sequence provided does not provide the fully specified formula sufficient to transmit as an identified term through the RELATIVE CLAUSE ADJUNCTION rule (54). Hence the (weaker) crossover effects in restrictive relatives. Unlike nonrestrictive relatives, where the preceding determine+nominal sequence is taken to project a fully identified term, there is no such term to carry across in restrictive relatives, and these accordingly display a pattern similar to the crossover effects displayed in WH questions. Since the particularities of the LDS$_{NL}$ account of quantification are not central to this paper, we shall leave this aspect of the analysis in this promissory form.

The significance of this account of crossover phenomena is that the explanation is not stateable either over the input to interpretation or over its output, for the explanation turns on progressive identification of values DURING the process of tree incrementation. It is sensitive to linear order, partiality of information at intermediate steps in interpretation process, and the way information is accumulated through the interpretation process. In particular, the dynamics involved in interpretation of *wh* expressions follows from the goal of seeking to resolve the disjunctive input initially projected. Furthermore, it is the availability of information presented at intermediate steps which allowed us to reduce crossover facts in relatives to a principle B form of explanation, rather than the problematic principle C explanation which had led to the bifurcation into numbers of discrete subcases. The context sensitivity of "weak" and "extended strong" crossover but not "strong" crossover is thus predicted without any stipulation over and above independent characterisations of *wh* expressions, pronouns, and relative clauses.

4. Reconstruction

Confirmation of the need to refer to intermediate stages of the process of interpretation, and not merely the input state and output result comes from reconstruction phenomena, and the apparent satisfaction of the locality condition associated with the anaphor *each other* in (69), giving rise to the interpretation (70):

(69) Which pictures of each other did John think that Sue and Mary were hoping that Bill would not sell ?
(70) Which picture of Sue did John think that Mary was hoping that Bill would not sell, and which picture of Mary did John that Sue was hoping that Bill would not sell?

This interpretation is, apparently, available despite the fact that the locality condition is not met either over the input structure to the process of reconstruction (upon orthodox assumptions of syntactic structure) or over the structure which results from that process. It has been suggested to us that these cases are not very strong, since there is independent evidence for treating these anaphors as a separate "logophoric" case as there are other instances where any treatment in terms of a standard concept of locality seems not to be available:

(71) The pictures of each other that John and Mary really liked were the pictures of each other in the bath.

Pending an account of how information is projected across the predicate *be* and an extension of LINKing of relative clauses to application to these data, the evidence presented by (71) remains somewhat unclear. However, there are a number of quite different phenomena which buttress the case that reconstruction must have the effect of recording information that "passes through" AT the node as it passes through (cf. Hukari and Levine 1995). Amongst these are the partial movement phenomena, to which we return in section 5.3.[6]

5. A Typology for Wh-Construal

In the face of the presented evidence, one might grant the need for some form of incrementality, but nevertheless argue that the slash mechanism of HPSG, with its percolation of WH features progressively up a tree through feature unification captures just the right dynamic element, without abandoning the overall declarative formalism. What, one might ask, does this disjunctive specification approach have to offer, over and above that, more conservative, form of specification? It is furthermore extremely close to the functional uncertainty analysis of LFG (Kaplan & Zaenen 1988). Kaplan & Zaenen indeed analyse long-distance dependencies in terms of the Kleene * operator, and so constitute a genuine precursor of the present analysis. However, in that case, the disjunction is defined over string sets and the f-structure specifications, not, as here, over structural specifications.

The advantage specific to this account is the dynamic parsing perspective within which the account is set. This dynamic perspective provides the basis for a general typology of WH constructions, explaining why they occur as they do, rather than simply defining discrete mechanisms for each new set of data. This unifying form of explanation is not available to the more orthodox frameworks, in which syntax is defined purely statically. We take in order *wh* in situ constructions, multiple *wh* constructions, and partial movement constructions. (In all cases we shall restrict attention to full NP *wh* expressions such as *who*, *what*)

5.1 *Wh* in situ Constructions

In the framework adopted, a formal symmetry holds between *wh* initial and *wh* in situ constructions: *wh* in situ constructions are the converse of *wh*-initial constuctions. We specify together the result of processing a *wh* initial expression, and the effect of processing a *wh* in situ expression.

(72) *Wh*-initial:

INPUT	m	SHOW t	TODO Typ(t)	0	
OUTPUT	m	SHOW t Label_S(+Q)	TODO Typ(t)	0	<u>*m SHOW e TODO ∅ k Form(WH) & Typ(e)

(73) *Wh*-in-situ:

INPUT	n	SHOW e	TODO Typ(e)	i
OUTPUT	n	SHOW e Form(WH) & Typ(e) <u>*(Label_S(+Q))	TODO ∅	i

The *wh* initial expression encodes an instruction that its formula and type are satisfied at some lower point in the tree, together with the specification that the node currently under construction has the property of being a question. The *wh* in situ expression, conversely, encodes an instruction that it is the premise 'Form(WH) & Typ(e)' which is projected into the current task state, and it is the instruction pertaining to the feature +Q which is disjunctively specified as holding somewhere higher in the tree. There is no unidentified task state as the *wh* in situ projects information to a node of the tree, and the node invoked in the specification involving the Kleene* operator is already in the tree.

In languages which freely allow WH either in situ or initially, with a free process of NP "scrambling", this characterisation of *wh*-initial expressions needs to be generalised, for all NPs can occur in the initial position. We have two options for this initial position - either it is associated with some as yet unfixed node in the tree, <u>*m for some root node m, or it is not associated with a tree node at all. In either case, any NP will be able to occur in this position, so a single lexical entry for WH can be defined, for its position either in situ or initially will be guaranteed by whatever general rule characterises this position:

(74)WH
INPUT | i SHOW e TODO Typ(e) j |

OUPUT | i SHOW e TODO ∅ j |
Form(WH) & Typ(e)
<u>*(Label_S(+Q))

This position is variously construed as a topic, and associated with a following clause, which must contain some element construed as a pronominal identified with this topic; or it is construed like a WH expression, and associated with a following sequence missing one expression. To characterise this variability, we propose the following AXIOM EXTENSION rule, and FOCUS RULE:

(75) AXIOM EXTENSION RULE (TOPICALISATION):

INPUT

| m SHOW t TODO Typ(t) 1 |

OUTPUT

| m SHOW t TODO Typ(t) 1 | | ∅ SHOW e TODO Typ(e) 2 |

This rule provides the possibility for the initial position in a clause to be associated with a free-floating expression, not fixed at any node in the tree.[7] The following rule narrows down this specificaton as an anticipation that the constituent in question is to be fixed somewhere in the root tree as yet unidentified:

(76) FOCUS RULE:

INPUT

| m SHOW t TODO Typ(t) 1 | | ∅ SHOW e TODO ∅ Typ(e) 2 |
Form(α) & Typ(e)

OUTPUT

| m SHOW t TODO Typ(t) 1 | |<u>*m SHOW e TODO Typ(e) 2 |
Form(α) & Typ(e)

Though very similar in effect to the Topic rule, the FOCUS rule treats this initial position as associated with some e-task requirement somewhere lower down the tree under construction. The applicability of the rules of subordination and resolution duly follow. In effect these rules are nothing more than a statement of the fact that the initial NP constituent projected in a string in these languages may be analysed as not having a fixed tree node, and may be construed either as a topic, associated anaphorically with some site within the following sequence, or construed as a focus, associated with some empty site within the following sequence. These rules are all inference rules, by definition optional, and accordingly three positions are predicted for *wh* expressions. We predict both WH initial configurations and unrestricted occurrence of WH in situ configurations(observed in Chinese, and also in Japanese, Malay):[8]

(77) Ni bijiao xihuan [[ta zenmeyang zhu] de cai] ? CHINESE
You more like how cook REL food
What is the means x such that you prefer the dishes which he cooks by x?

Both rules (75) and (76) can apply recursively, allowing a sequence of Topic-Topic constructions, Topic-Focus constructions, or even Focus-Focus constructions (cf. the case of Bulgarian below).

Given the symmetry in the characterisation of *wh* initial constructions and *wh* in situ constructions, it might seem that no asymmetry between these two types of *wh* constructions could be predicted. It is however a widely known observation that, while *wh* initial constructions impose island conditions, such as the preclusion of dependency of a *wh* expression into a relative clause, *wh* in situ expressions can be freely construed with scope wider than that of the relative clause within which they occur (eg Aoun & Li 1991). This prediction is especially problematic for analyses which involve overt *wh* movement at s-structure (pre-SPELLOUT), and covert *wh* movement at LF (post-SPELLOUT) such as advocated in Reinhart 1995 (cf. Simpson 1995 for detailed discussion). Nevertheless, the mirror-image characterisation of WH-initial and WH-in-situ, perhaps surprisingly, provides a natural basis for predicting asymmetry between these two forms of dependence - on the basis that only the former is part of a search through a domain as yet unbuilt. We take the preclusion of dependency into relative clauses. WH initial sets up a disjunctive specification to be resolved within a certain task-domain (the clause it fronts) (cf. (72) with its explicit specification that the disjunctive specification has to be resolved within the given task-domain). The problem with examples such as (78), is that this resolution task cannot be achieved within the domain defined. The presence of the head of the relative clause (*the man* in (78)) to the contrary guarantees that all requirements of the presented task, viz. the presence of a major premise (of the form X→Y) and some appropriate constellation of minor premises (of the form X) - are satisfied without having been able to resolve the configurational position of the unfixed task state assigned <u>*m tree node characterisation. It thus remains unresolved in the task assigned to the node m, leading to lack of wellformedness:
.

(78) Who did John see the man who likes e
↑

The string is illformed because an assigned completion task on structure within the domain of that task cannot be satisfied given information from other expressions in the string. In WH in situ configurations to the contrary, there will never be such an unresolvable disjunction. The projection of the gap formula (Form(WH) & Typ(e)) immediately meets whatever restrictions are imposed by other expressions in the environment, so there is never a node with no identifier. And the <u>* characterisation accompanying the +Q feature merely adds a feature to the node 'm' independently projected - there is no unresolved disjunction, to be satisfied within a given task state. This latter specification is thus nothing more than a feature-checking requirement.

Given the characterisation of the +Q feature associated with the *wh* in situ expression as holding at some node dominating the *wh* expression, a natural typology emerges according as (i) this characterisation is otherwise unrestricted, (ii) it is restricted to some subset of the total domain, or (iii) it is required to hold at the node immediately dominating the *wh* expression:
(i) <u>*(Label$_S$)Q
(ii) <u>*(Label$_S$)Q subject to condition Y
(iii) <u>(Label$_S$)Q
The first case is that of Chinese, already observed. *Wh* expressions occur freely in situ without restriction, as long as the overall structure is interpreted as a question. (If it is not, the expression gets interpreted as an indefinite.)

The intermediate case is that of Iraqi Arabic. Iraqi Arabic is a mixed *wh*-initial and *wh* in situ language. In sentences with an initial tensed verb, but otherwise a sequence of nontensed verbs, the *wh* expression may occur EITHER in situ or in the fronted position:

(79) Mona raadat [riijbir Su'ad tisa'ad *meno*]
Mona wanted to force Suad to help who
Who did Mona want to force Suad to help?
(80) Meno Mona raadat [riijbir Su'ad tisa'ad]

If however the subordinate clause is tensed, then (without some special ancillary device - see section 5.3.1), the *wh* expression may not occur in situ and must be preposed:

(81) *Mona tsawwarat [Ali ishtara *sheno*]
Mona thought Ali bought what
(Intended: What did Mona think that Ali bought?)
(82) Sheno$_i$ tsawwarit Mona [Ali ishtara t$_i$]
What thought Mona Ali bought
What did Mona think Ali bought ?

The restriction (also displayed in other languages - Hindi, and some dialects of German) involves the concept of a tense domain. The tense-marked verb defines a domain within which *wh* expressions are licensed, but any second intervening tense-marked verb breaks the domain, and the *wh* expression is not licensed to occur. The restriction is directly analogous to the locality restriction on anaphor resolution, which specifies that an antecedent must be selected within the domain defined by the most local tensed predicate. For this we require some restriction on the <u>* characterisation of the +Q feature, to guarantee that it be projected onto a node suitably local to the occurrence of the *wh* expression. Pending a detailed account of tense, we leave this as an informally described condition only, noting merely that the domain is exactly that of anaphors.

Finally, in the most restricted case (iii), the feature projected by a *wh* expression must be met at its immediate mother. Such a requirement predicts the facts of Bulgarian, Czech, Hungarian etc, assuming recursive application of rule (75)-(76). In this family of languages, *wh* expressions are all required to precede the verb:

(83) Koj kogo vizda? Bulgarian
Who whom sees
(84) *Koj vizda kogo ?
Who sees whom
The +Q specification takes the form:
<u>Label$_S$(Q)
The Topicalisation and Focus rules apply recursively, each addition of an unspecified node specification ensuring return to the initial task state. Hence the motherhood condition will be met by recursive applications of the rule, and a sequence of all *wh* expressions in a multiple *wh*-question predicted to occur at the beginning of the string

In sum, the richness of having disjunctive feature specifications provides the basis for a natural class of *wh* in situ languages, according as the restriction on the associated +Q feature is maximally weak and unspecified, imposes some restrictions on an otherwise free process, or imposes a restriction yielding a fully determined result.

5.2 **Multiple wh questions**

The two characterisations of *wh* initial and *wh* in situ constructions now combine together to provide the right set of predictions. Multiple *wh* constructions are simply *wh* initial constructions and *wh* in situ processes of construction in combination. We predict the effect in English that *wh* initial is subject to subjacency effects - its associated gap not being able to occur inside a relative clause, while the secondary *wh*, occurring in situ, is not subject to any such subjacency effects, and so may occur within a relative structure.
(85) *Who$_i$ did the journalist leak the document in which Sue had criticised e$_i$ to which press?
(86) Who reported the journalist that leaked which document to the press?
(87) Who reported the journalist that had leaked which document to which committee?

We predict the *Iraqi* data that the second *wh* in a multiple *wh* structure may only occur if all superordinate verbs except the clause containing the primary *wh* expression are nonfinite:

(88) Sheno ishtara Ali t [minshaan yenti li-meno]?
What bought Ali in order to give to-whom
What did Ali buy to give to whom ?
(89) *Meno tsawwar [Ali xaraj weyya meno]?
Who thought Ali left with whom
Who thought that Ali left with whom

And the multiple fronting of WH expressions (Bulglarian/Hungarian/Czech)is, as already demonstrated, predicted from the immediate motherhood restriction. As further predicted, no other structure is allowed.

5.3 **Partial WH movement**

The final part of the puzzle as presented is the partial *wh* movement, a strategy for overcoming what for a language with a tensed-clause restriction on *wh* in situ constructions might otherwise impose an unacceptably low limit on the expressiveness of the language. If a language such as German had no means of ensuring that the *wh* expression is interpreted from within a tensed clause as asking a matrix form of question, how would the content projected by the English question *Who do you think Bill likes?* be conveyed? The answer is through a device which overcomes the tense restriction, by anticipating the *wh* expression without actually producing it. And this is the datum presented by (90)-(91):

(90) Was glaubst du was Hans meint mit wem Jakob gesprochen hat ?
(91) Was glaubst du mit wem Hans meint dass Jakob gesprochen hat

Seen from an orthodox quantifier-variable binding perspective for analysing *wh* expressions, the puzzle about these anticipatory *was* particles in German is that they do not constitute any indication of a specific question posed - merely a cautionary advance notice that such a *wh* question will be posed at some later point in the on-line interpretation process. How then can their so-called "expletive" properties be related to those of the *wh* operator with which they are paired? (cf. Cheng 1991, Dayal 1994, Brandner 1996) - They are not operators, and not variables, either. A natural means of capturing this phenomenon is suggested by this framework. Given the manipulation of <d>*X as a database entry, and given the specification of Declarative Unit formulas in the TODO box, there is no reason to preclude a <d>*X specification from occurring in the TODO box also. As long as this involves the addition of a feature to a type independently introduced by an introduction rule, this feature addition will be harmless, and will not give rise to a proliferation of extra feature-specific introduction and elimination rules.What the specification of <d>*Form(WH) in the TODO box would mean was that ahead in the projection of structure for the purpose of building an interpretation is the need to enter a <d>*Form(WH) in the done box, this in its turn indicating that later in the interpretation process, a GAP will be postulated. We thus have a natural formal analogue to the informal intuition that these expressions indicate what sort of interpretation task lies ahead, which is but a simple extension of the analysis provided for *wh* expressions themselves. Accordingly, we define an extra lexical definition of *was*, via the projection of <d>*Form(WH) in TODO:

(92) *was* (non-expletive)

INPUT | m SHOW t TODO Typ(t) 0 |

OUTPUT | m SHOW t TODO Typ(t) 0 | | <u>*m SHOW e TODO ⌀ k

Label_S(+Q) Form(WH) & Typ(e)

(93) *was* (expletive)

INPUT
i SHOW t TODO Typ(t) j

OUTPUT
i SHOW t TODO Typ(t) & <d><d>*Form(WH) j
Label_S(+Q) & Typ(t)

The target instruction to construct an expression <d><d>*Formula(WH) in TODO is essentially anticipatory, inducing a WH expression lower down in the configuration. - the transfer of information from TODO into the DONE compartment is only relative to independent provision of the task TODO. There is no unattached task state in this structure, so the only means of provision is from lexical input - hence the encoding of the necessity for a *wh* expression at some daughter lower in the tree . The '<d><d>*' form of specification ensures minimally daughterhood. Since *wh* expressions in German themselves project the specification that the immediate mother node of the point at which they are introduced be of type t bearing the +Q attribute, the partial movement effect of requiring the *wh* expression not to be in situ is enforced, and the *wh* expression predictably obligatorily occurs at the front of the subordinate clause in which they occur. The only modification of the specification of the *wh* expression, other than its characterisation as part of the TODO configuration, is the requirement that it must hold at a daughter node and may not hold at the node at which it is inserted. This ensures that such an "expletive" *was* never occur in the same clause as the *wh* expression it anticipates, given that full *wh* expressions in German, as in English, are restricted to occurring in a position where their mother is +Q and hence a t task. The sole functional purpose indeed of this expletive is to indicate that a full *wh* form will be in some clause LOWER than the one it itself initiates. Hence we predict the data (94)-(99):

(94) *Was glaubst er was ?
(to mean "What does he believe?)
(95) *Was mit wem hat Jakob gesprochen?
(96) Was glaubst du was Hans meint mit wem Jakob gesprochen hat ?
(97) Was glaubst du mit wem Hans meint dass Jakob gesprochen hat ?
(98) *Mit wem glaubst du was Hans meint dass Jakob gesprochen hat ?
(99) Was glaubst du was/dass Hans meint mit wem Jakob gesprochen hat ?

(94)-(95) both contain the expletive *was* with some full *wh* expression in the same clause (precluded by the clash between the <d><d>* form of specification and the lexical specification of full *wh* expressions). (96) correctly has a series of *was* particles followed by a *mit wem* initial to its own clause, allowing all lexical specifications to be satisfied. These "expletive" was expressions merely impose a target - they do not constitute a discrete DU-formula at an identifiable and hence discrete node. (97) also allows the lexical specifications both of the expletive *was* and *wem* to be satisfied - note there is nothing in the full *wh* form (as illustrated by the full form *was*) to guarantee that a clause initial *wh* form is satisfied by the presence of a gap in the very same clause. In (98), the requirement imposed by the expletive *was* is not satisfied, there being no lower clause-initial *wh* expression. Though reiterating expletive *was* guarantees at each clause boundary so marked that a full-form *wh* will not occur there, even a single instance of *was* is sufficient to induce the presence of a full *wh*-initial form lower in the structure. Hence *was* and *dass* may alternate in structures such as (99) (cf. Müller & Sternefeld 1996). Finally we predict that the two construction types may be combined as long as the gap triggered by some clause-initial full *wh* expression precedes a *was* expletive to be followed by a second full clause-initial *wh*. We thus predict the wellformedness of:

(100) Wer$_1$ e$_1$ glaubt was ich meinte mit wem$_2$ Jakob e$_2$ gesprochen hat ?

5.3.1 Cross clausal *Wh*-licensing: Iraqi Arabic

Iraqi Arabic (and Hindi) display a closely related phenomenon. These languages have a tense-domain restriction on the occurrence of the *wh* expression, and so *wh* in situ expressions cannot be construed as questions, without some further ancillary device. In order to project the required interpretations, a *wh* expression occurs in a tensed clause, apparently suspending its own locality restriction. The possibility of such an apparent violation of the restriction is dependent on the presence of a feature in each higher clause above indicating that there is some occurrence of a *wh* expression in the lower clause. Unlike German however, the *wh* expression occurs in situ in the lower clause. Moreover, the occurrence of the indicative particle is obligatory in each successive clause. In Iraqi Arabic, the prefix *sh*- (a reduced form of *sheno* (= *what*)) is prefixed on the higher verb, and ALL verbs superordinate to the *wh* in situ in the embedded tensed clause have to bear this prefix:

(101) *sh*-'tsawwarit Mona [Ali raah *weyn*] ?
Q-thought Mona Ali went where
Where did Mona think that Ali went ?

Since the *wh* in situ in embedded clauses is otherwise licensed only if the only verb which is tensed is the highest matrix clause, this will require a specific lexically triggered definition projecting the feature +Q onto the task state of type t which will ultimately contain the formula 'Form(WH) & Typ(e)':

(102) SH-(expletive)
INPUT
| i SHOW e→t TODO Typ(e→t) j |

OUTPUT | i SHOWe→t TODO<d>(Typ(t) & Label$_S$(+Q)) j |

It is of some interest that the projection of the form 'Form(WH) & Typ(e)' from the *wh* in situ form would in fact satisfy some imposed target in TODO of the disjunctive form '<d>*(Form(WH) & Typ(e)', suggesting a closer parallelism between Iraqi Arabic and German than the lexical specification (102) provides. However, unlike in German, the extension of the licensing domain by *sh*- is essential to the acceptability of the presence of the *wh* in situ expression in embedded tense clauses. So the imposition of a disjunctive TODO specification as in German would not be sufficient; and, given the requirement of lexically specifying the carry-down of the feature 'label$_S$(+Q)', not necessary either. So though the parallelism between Iraqi Arabic and German is a consequence of the current characterisation, the lexical specification of the prefix *sh* cannot take the same form as the expletive *was* in German.

6. Conclusion

The substance of this account of *wh* expressions has been the claim that the disjunctive specification made available by statements of the form <x>*P provides the basis for expressing natural linguistic generalisations - in particular not only explaining the structural properties of *wh* initial sentence forms, but also providing a principled basis from which to predict a whole family of generalisations about *wh* structures. *Wh*-initial effects, *wh* in situ effects and the required array of partial movement effects are correctly predicted, as are the array of otherwise puzzling crossover phenomenona. Phenomena thought to be entirely independent of one another have been explained on the same basis. And no vocabulary special to any one of these devices has been introduced.

Essential to the account have been two primary properties.
(i) the asymmetry between the input provided by any individual expression on the one hand and its interpretation/structural role in interpretation on the other;
(ii) a specification of how such encoded input information is incrementally enriched on a left-right basis from the initial state to the ultimate assignment of propositional form as output.

The novelty of the details of this account lies in the claim that natural-language expressions may not merely only relatively weakly specify the content to be assigned to them in context, but they may also fail to project a fully defined tree relation with the string in which they occur. Expressions in a string must therefore be subject to a process of enrichment which involves not merely fixing of the content of the expression relative to context, but also involves on-line decision as to the nature of the structure to be assigned, and the procedures to be following in building it up.

The significance of this analysis lies in the fact that it is couched within a formal framework for modelling the "pragmatic" process of utterance interpretation, while nevertheless purporting to be an explanation of an array of syntactic phenomena. And in this framework, there is no concept of syntactic structure over and above the structure in terms of which the incremental process of interpretation is modelled. There is the language of formula plus label, there is the language of tree nodes, and there is the language for defining the meta-level process of projecting individual lexical specifications as input on a left-right basis onto fully determined propositional formulae. There is however no external body of axioms to which this parsing model makes reference to determine individual moves to be made. The model of the parsing process itself provides the structural framework, indeed this is the syntax, and it is in terms of this framework that all linguistic explanations are couched. Furthermore the projection of structure from *wh* expressions is taken to be part of the process of resolving such relatively weak input specifications, defined, like anaphora resolution in terms of the (primarily) left-right projection of information from preceding linguistic input. The explanation therefore falls within a family of explanations which might loosely be called parsing explanations (cf. earlier attempts by Erteschek-Shir 1973, Marcus 1981, Berwick and Weinberg 1984).

It should however be stressed that this explanation of the data d˝eparts from earlier conceptions of the relation between competence and performance, or semantics and pragmatics, in which the competence model is defined in terms of an independently defined body of syntactic/semantic/phonological axioms which performance/pragmatic explanations take as input. We are not proposing a pragmatic model in terms that takes a fixed, semantically interpreted structural configuration as input with pragmatic principles applying to this input to yield a set of contextually fixed values. And we are not proposing an explanation of *wh* phenomena in terms of parsing strategies merely to come to the conclusion that *wh*-binding, crossover, reconstruction, *wh*-in-situ and partial *wh*-movement effects fall outside the remit of the natural-language computational system, leaving the assumption of a computational system specific to the language faculty reduced but intact. We are to the contrary defining a model of the parsing process which provides the total vocabulary for explaining structural (syntactic) properties of natural language. What we are proposing is that the human faculty for natural language is a capacity for parsing, a specialised deductive capacity for pairing linguistic expressions with logical forms which they are taken to express, these logical forms themselves being vehicles for inference of an orthodox sort.

Despite the procedural flavour of this framework, many commonalities with other frameworks remain. Together with all other linguistic frameworks, we are assuming that the lexicon provides the input on the basis of which interpretation is projected, and that such encoded information provides all the information needed to characterise idiosyncrasies of individual languages. Together with other frameworks (HPSG, Categorial Grammar) we assume that lexical specifications include type-logical information fixing the combinatorial properties of individual expressions. Together with others, we assume that these lexical specifications also include representation of concepts which in some cases fix the denotational content of some individual expressions. However, unlike other frameworks, we assume that such lexical specifications may include procedural instructions on the process of parsing itself, and that a unitary characterisation of lexical specifications requires the definition of all such specification as procedures which provide input to the incremental projection from a string onto some logical form. The overall framework in terms of which these lexical specifications are defined, is, then, the metalevel theory which defines the inferential, goal-directed process which constitutes the activity of parsing. This inferential activity has properties in common with other reasoning capacities and is not defined to be complementary to those more general capacities.

This opens up a new perspective for syntactic enquiry. Language-particular processes of interpretation can be seen as interacting freely with more generalised reasoning processes such as Modus Ponens, allowing some aspects of linguistic structure to be captured in these more general terms. This allows us to explore the boundary between language-internal processes and general reasoning processes without having to assume a fixed interface in which language-internal properties are defined independent of such more general processes, with any properties which are reducible to these more general processes having to be seen as falling outside the domain of enquiry narrowly construed (cf. Chomsky 1995). We might for example find evidence for retaining the view that some subjacency effects constitute language-internal restrictions entirely independent of any general property of the logic of the reasoning task underpinning the parsing process. However, with richer multi-dimensional logics to hand, some of them might turn out to be consequences of the human capacity for composite forms of reasoning (as argued above). On the view being proposed, we would not be forced to move from this latter conclusion to the view that all such subjacency effects fall outside the remit of the discipline devoted to articulating the language faculty. Similarly with crossover phenomrna, *wh*-in situ effects, etc. With this shift of perspective, properties of natural language may overlap with independently motivated reasoning systems, sharing some properties, diverging in others.

The shift of perspective has consequences. First it suggests that the study of syntax has, following the lead of semantics, to become dynamic, defined in terms of the ongoing projection of information on a left-right basis (cf. Johnson & Moss 1994 for the proposal that current models of grammar might revealingly be recast as dynamic algebras). Secondly, our concept of competence in oppos˝ition to some concept of performance has to be revised. We no longer envisage the systems underpinning natural language according to the static pattern imposed by classical Fregean logics, with strings assigned denotational content direct, and some ancillary and entirely separate largely unknown theory of performance explaining how these systems are manipulated in communication. Rather we envisage natural languages as systems specifically developed for the dynamic enterprise of projecting infinite variety of interpretative content from a finite lexicon. Seen in this light, the underspecification of natural language content is no longer an embarrassing divergence from formal language systems, to be patched up in the analysis to approximate as closely as possible to those systems. Such specifications are to the contrary indicative of the purpose for which natural languages are designed. Natural languages are metalevel devices for the projection of vehicles of thought/inference, encoding procedures whereby the intended content can most effectively be retrieved. There is no longer a dichotomy between the perspectives provided by theories of competence and theories of performance. Theories of linguistic competence are indeed theories about the language faculty, and these are theories about the abstract formal properties of the framework which we put to use in parsing. Such theories are complemented by theories of pragmatics. The burden of pragmatic theories, and, more generally, performance theories, is to articulate the general constraints imposed by the cognitive system which determine how the choices made available by the competence system are actually realised in context (Sperber and Wilson 1986). The two together combine together to yield a theory of linguistic knowledge and use.

Footnotes

1. This paper was stimulated by J. Aoun, who posed this question in a talk to the Linguistics Department SOAS February 1996. We are grateful to Andrew Simpson, Shalom Lappin and Abbas Benmamoun for conversations over many months, and to the audience at the Bangor conference on Syntactic Categories for comments.
2. Indefinites are a systematic exception to this. Cf Reinhart 1995, Winter 1996, Farkas forthcoming, Abusch 1994, Kempson and Meyer-Viol in preparation for recent attempts to account for this phenomenon.
3. We leave open the question of whether the adicity of predicates should include an argument for an event variable, but do not include such an argument position in what follows.
4. This point has been emphasised both in the semantic and pragmatic literature for over a decade now. Cf Kamp 1981, Kamp & Reyle 1994, Barwise & Perry 1983, Sperber and Wilson 1985, and the articles within these paradigms which have followed these.
5. In the case of adjuncts such as *With his hat on, John left the room*, it is arguably the lack of a specified treenode identifier for the initial adjunct which delays the process of

anaphora resolution.

6. As Andrew Simpson has pointed out to us, the detailed characterisation of these data will require an asymmetry between an initial *wh* expression, which is invisible for anaphora resoltuion in virtue of being unfixed and therefore unidentified, and the identification of an anaphor within the *wh* expression which CAN take place from this unfixed position. Though a stipulation, this is not unnatural given the asymmetry between being an anaphor looking for an identifier to resolve its intrinsic under-specification, and being an identified expression providing an antecedent.

7. As a means of resolving the contribution to be made by any free-floating initial constituent, we assume the task of identifying an anaphoric expression will involve directly assigning it the declarative unit associated with the initial constituent, thereby guaranteeing the contribution of that constituent to the overall interpretation. Though this process remains only informal, it is a mere reflection of the fact that topic expressions have no structural relation to other parts of the structure other than providing an antecedent form some pronominal contained therein.

8. An alternative characterisation of a language such as English, would be to enforce the application of this rule, leaving the lexical characterisation of *wh* expressions simple. However, given the obligatory accompanying process of auxiliary inversion in English, we prefer direct lexical characterisation of *wh* initial constructions in such cases.

.

References

Abusch, D. (1994). 'The scope of indefinites'. *Natural Language Semantics* 3, 88-135.

Aoun, J. & Li, A. (1991). 'WH in situ', *Linguistic Inquiry*

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

Berwick R. & Weinberg, A. (1984) . *The Grammatical Basis of Linguistic Performance,* MIT Press. Cambridge, Mass.

Blackburn, S. & Meyer-Viol, W. (1994) 'Linguistics, logic and finite trees' *Bulletin of Interest Group in Pure and Applied Logics* Vol.1, No.1, 3-29.

Blackburn, S. Meyer-Viol, W., & de Rijke, M (1995). 'A proof system for finite trees' Report CS-R9563. CWI. Amsterdam.

Brandner, E. (1996). 'The *was-w-* construction and clausal typing' in ed. E. Doron & S. Wintner, *Proceedings of the Eleventh Annual Conference of Israel Association for Theoretical Linguistics , IATL 3*, 25-55.

Cheng, L. 1991) *On the Typology of WH Questions*. MIT Ph.D.

Chomsky, N. (1981) *Lectures on Government and Binding.* Foris, Dordrecht.

Chomsky, N. (1994). *On Minimalism.* MIT Press.

Chierchia, G. (1984). *Topics in the Syntax and Semantics of Infinitives and Gerunds*. Ph.D UMASS Amherst.

Chierchia, G. (1992) 'Questions with quantifiers' *Natural Language Semantics* 1, 181-234.

Culicover, P. (1996). 'On distinguishing A'-movements' *Linguistic Inquiry* 27, 445-63.

Dayal, V. (1994), 'Scope marking as indirect wh-dependency' *Natural Language Semantics* 2, 137-70.

Dowty, D. (1992). 'Variable-free' syntax, variable-binding syntax, the natural deduction Lambek calculus and the crossover constraint' *WCCFL 12*

Erteschek-Shir, N.(1973). *On the Nature of Island Constraints*. Ph.D MIT.

Farkas, D. (forthcoming). 'Indexical Scope', in Sczabolsci, A. (ed.) *Ways of Scope-Taking*. Kluwer.

Gabbay, D. (1990/fcmng). *Labelled Deductive Systems*. Oxford University Press. Oxford

Gabbay, D. (1994). 'Classical vs non-classical logics (the universality of classical logic)' in Gabbay, D. Hogger,C. Robinson,J. *Handbook of Logic in Artificial Intelligence and Logic Programming: Vol.2 Deductive Methodologies*, 359-500. Clarendon, Oxford.

Gabbay, D. & Kempson, R. (1992). 'Natural-language content: a proof-theoretic perspective', in ed. Dekker, P. & Stokhof, M. *Proceedings of 8th Amsterdam Semantics Colloquium*. Amsterdam.

Higginbotham, J. (1981). 'Pronouns as variables', *Linguistic Inquiry*

Huang, J. (1982). *Logical relations in Chinese and the theory of grammar*. Ph.D, MIT.

Hukari, T & Levine, D. (1995). 'Adjunct extraction', *Journal of Linguistics*. 31, 195-216.

Johnson, D. & Lappin, S. (forthcoming) 'A critique of the minimalist programme' *Linguistics & Philosophy.*

Johnson, D. & Moss, L. (1994). 'Grammar formalisms Viewed as Evolving Algebras' *Linguistics & Philosophy* 17, 537-60.

Joshi, A. & Kulick, S. (1995). 'Partial proof trees as building blocks for a categorial grammar' in Morrill, G. & Oehrle, R. (eds.) *Proceedings of Formal Grammar Conference,* Barcelona July 1995.

Kamp, H. (1981). 'A theory of truth and semantic representationi in Groenendijk, J. Janssen, R. Stokhof, M. (eds.) *Formal Methods in the Study of Language,* Mathematical Centre Tract 135, 277-322. Amsterdam.

Kamp, H. & Reyle, U. (1994). *From Discourse to Logic*. Kluwer, Dordrecht.

Kaplan, R. & Zaenen, A. (1988) 'Long-distance dependencies, constituent structure, and functional uncertainty' in Baltin, M. & Kroch, A. (eds.) *Alternative Conceptions of Phrase Structure*, 17-43. University of Chicago Press. Chicago.

Kempson, R. (1995a). 'Ellipsis: a natural deduction perspective' in Kempson, R. (ed.) *Bulletin of The Interest Group of Pure and Applied Logics: special edition on Language and Deduction* Vol.3, Number 2,3, 489-526.

Kempson, R. (1995b). 'Semantics, pragmatics, and natural-language interpretation' in Lappin, S. (ed.) *Handbook of Contemporary Semantic Theory*. Blackwell, Oxford.

Kempson, R. (1995c). Crossover: a dynamic perspective. In Jensen, S. (ed.) *SOAS Working Papers 6.*

Koopman, H. & Sportiche, D. (1982),'Variables and the bijection principl' *The Linguistic Review* 2, 139-60.

Lasnik, H. & Stowell, T. (1991). 'Weakest crossover'. *Linguistic Inquiry* 22:687-720.

McCloskey, J. (1979). *Transformational Syntax and Model-Theoretic Semantics: A Case Study in Modern Irish*. Reidel, Dordrecht.

McDaniel, D. (1989). 'Partial and multiple *wh* movement. *NLLT 7 565-604.*

Marcus, M. (1980). *A Theory of Syntactic Recognition for Natural Languiage*. MIT Press. Cambridge, Mass.

Milward, D. (1993). Dynamics. dependency grammar and incremental interpretation. *COLING 14.* 1095-9

Morrill, G. (1994). *Type-logical Grammar*. Kluwer.

Müller G. & Sternefeld (1996) 'A' chain formation and economy of derivation' *LI* 27.3, 580-512.

Oehrle, D. (1995). 'Term-labelled categorial type system' *Linguistics and Philosophy* 18.

Ouhalla, J. (1994) Remarks on the binding properties of WH-in situ. ms. London.

Pollard, C. & Sag, I. (1991), *Head-Driven Phrase Structure Grammar*. University of Chicago Press, Chicago.

Pollard, C. & Yoo, E-J (1996) 'Quantifiers, *wh*-phrases, and a theory of argument selection'. ms Ohio.

Postal, P.(1972). *Crossover Phenomena*. Holt, Rinehart, Winston, New York.

Postal, P. (1993). 'Remarks on weak crossover effects' *Linguistic Inquiry* 24, 539-56.

Reinhart, T. (1994) WH in situ. Proceeding of 9th Amsterdam. Colloquium

Reinhart, T. (1995) Interface Strategies. OTS ms.

Simpson, A. (1995). *Wh-movement, Licensing and the Locality of Feature Checking*.Ph.D. SOAS, London.

Sperber, D. & Wilson, D. (1986). *Relevance: Communication & Cognition*. Blackwell.

Winter, Y. (forthcoming) 'Choice functions and the scopal semantics of indefinites'. *Linguistics & Philosophy*.