Grammatical Constants and Structural Variation*

1. Cognition as logical computation

Consider the question whether a certain configuration of words $w_1 \ldots w_n$ constitutes a wellformed expression of type B. Viewing natural language cognition as essentially *computational* in nature, we can reformulate this question as the deductive problem displayed in (1): Given A_i as the logical 'parts-of-speech' for the w_i, is the conclusion B derivable from Γ? Γ here would be some structural configuration of the A_i assumptions; the rules of inference for the grammar logic (the 'computational system') determine whether or not this configuration can be judged to be of type B. Considering the question from a semantic point of view, one obtains a deductive (or 'derivational') theory of grammatical meaning by reading the inference steps producing the conclusion B as instructions for building up a program t that computes the meaning of the conclusion B out of the meanings of the input parameters x_i — a program that can be applied to the actual lexical meanings of the w_i, in a concrete instance.

(1) PARSING *as* DEDUCTION

$$\begin{array}{ccc} w_1 & \cdots & w_n \\ \vdots & & \vdots \\ \underbrace{x_1 : A_1 & \cdots & x_n : A_n}_{\Gamma} \end{array} \vdash t : B$$

To work out this deductive perspective, a number of central questions have to be addressed.

- What are the CONSTANTS of grammatical reasoning? Can we provide an explanation for the *uniformity* of the form/meaning relationship across languages in terms of this vocabulary of logical constants, together with the deductive principles governing their use?
- How can we reconcile the idea of 'constants of grammatical reasoning' with the differences between languages, that is, with STRUCTURAL VARIATION in the realization of the form/meaning correspondence?

* The research reported on in this paper forms part of a joint project with R.T. Oehrle, supported by Grant SBR-9510706 of the National Science Foundation. A revised version of this paper appears in Bouma et al. (eds.) *Constraints and Resources in Natural Language Syntax and Semantics*, CSLI Publications, Stanford, 1999.

– What is the interaction between the 'derivational' meaning, read off from the inference steps constituting the process of grammatical deduction, and the 'lexical' meaning of the input assumptions w_i?

Our approach to these issues combines key themes of two lines of research: linear logic and categorial grammar.[1] Linear logic is a well-studied representative of what is known as a *resource-sensitive* system of inference. Technically, resource-sensitivity is obtained by the removal of the book-keeping rules of Contraction and Weakening. These 'structural' (as opposed to 'logical') rules allow the free multiplication ('cloning') or waste of assumptions in the process of deduction. Dropping Contraction and Weakening as 'hard-wired' components of an inference system, one introduces a clean separation between *logical* and *structural* aspects of reasoning. In the resulting logic assumptions have the status of finite, 'material' *resources*, and the rules of inference keep track of the production and consumption of these resources. The resource-sensitive style of inference is more fine-grained than the 'classical' style, which means that more logical constants become distinguishable. Specifically, it becomes possible to identify logical constants (the so-called modalities) for the explicit *control* over resource multiplicity: constants licensing multiplication or waste for modally marked assumptions. The linear style of inference, in other words, is more discriminating, but thanks to the modalities, not less expressive, than its coarser classical relative. We will show in Section 1.1 that resource-sensitivity provides the logical core of very fundamental grammaticality principles.

The second line of research has grown out of the work of Lambek. In a linguistic setting, the resources under consideration are natural language expressions: elementary form/meaning units ('words') and composite form/meaning configurations built out of these. Well-formedness, in this case, is determined not just by the multiplicity of the grammatical material, but also by its *structure*. The tradition of categorial type logics further refines the linear vocabulary, and introduces logical constants that are sensitive to linguistically relevant structural dimensions such as precedence (word order), dominance (constituency) and dependency. And parallel to the linear modalities controlling resource multiplicity, the categorial vocabulary can be enriched with control features providing deductive instruments for the fine-tuning of these structural aspects of grammatical resource management. The interplay between logical and structural aspects of grammatical composition is discussed in Section 1.2.[2]

Recently, the Minimalist Program has introduced a more computational perspective in

[1] See Girard (1987) and Lambek (1958, 1961) for the original papers. Van Benthem (1991, 1995) offers a panoramic perspective on categorial systems within the landscape of 'substructural' logics.

[2] Needless to say, our presentation here is not historical: the concept of a logic 'without structural rules' as introduced in Lambek's 1958 and 1961 papers antedates the introduction of linear logic by more than a quarter of a century. But the original Lambek systems had fixed resource management regimes: they lacked the vocabulary for structural *control*. The logic of categorial control features is developed in Moortgat (1996), Kurtonina and Moortgat (1997). An in-depth discussion of the connections between linear logic and categorial grammar can be found in Moortgat (1997).

the generative study of the language faculty. A number of authors have drawn attention to the non-trivial correspondences between minimalist grammar design and the deductive principles underlying resource-sensitive systems of inference.[3] Without attempting a full logical reconstruction of the Minimalist Program here, we will, where appropriate, provide suggestive pointers for readers who would be interested in exploring the logical basis for minimalist concepts.

1.1 *Grammatical resources*

The starred examples below show two ways of violating a fundamental grammaticality principle. Example (2a) fails to be a well-formed sentence because there is *not enough* grammatical material: the verb 'offer' requires a subject, a direct and an indirect object. In (2b) these three arguments are supplied, but in (2a) only one of them, leaving the sentence incomplete. The opposite is true when one compares (2c) and (2d). Here, (2c) is illformed because of a *surplus* of grammatical material: there is no way for a direct and an indirect object to enter into grammatical composition with the intransitive verb 'grin' which just requires a subject.

(2) a. *the Mad Hatter offered
b. the Mad Hatter offered Alice a cup of tea
c. *the Cheshire Cat grinned Alice a cup of tea
d. the Cheshire Cat grinned

Different linguistic theories have formulated a variety of principles to account for these basic facts of grammatical (in)completeness: the Subcategorization Principle in HPSG, the Theta Criterion in GB, the principles of Coherence and Completeness in LFG, to mention a few.[4] These principles, each stated in the theoretical vocabulary of the grammatical framework in question, have in common that they capture dependencies of a *local* nature, stateable within the subcategorizational domain of lexical items.

Comparing (2) with the examples in (3), one sees that dependencies of a potentially unbounded nature exhibit the same pattern of grammatical incompleteness and overcompleteness. In (3a) there is a correlation between the presence of the relative pronoun 'which' and the *absence* of an overtly realized direct object in the relative clause body: addition of the underlined phrase makes (3b) overcomplete, just like (2c). But now, as the examples (3c) and (3d) show, there is no guarantee that the relative pronoun which pre-empts the direct object slot can be found in the local subcategorizational domain of the verb selecting for that argument. So, typically, linguistic theories have come up with new sets of principles, interacting with but different from the ones governing local dependencies, to capture long-distance dependencies such as illustrated in (3): movement and empty category principles, slash feature percolation principles, etc.

[3] See Stabler (1997) for a computational interpretation of Chomsky (1995), and for example Cornell (1997) for the connection with resource-logical principles.
[4] See Pollard and Sag (1994), Chomsky (1981), Kaplan and Bresnan (1982).

(3) a. the tarts which the Hatter offered the March Hare
b. *the tarts which the Hatter offered the March Hare *a present*
c. the tarts which Alice thought the Hatter offered the March Hare
d. the tarts which the Dormouse said Alice thought ...

Such principles, in the overall design of linguistic theory, are irreducible primitives. As the above discussion shows, there is no unified 'principle-based' account of local and non-local dependencies. Our objective, in searching for the 'logic of grammar', is to present just such a unified account. We approach the problem in two stages: in a first approximation, we restrict our attention to 'multiplicity' issues (i.e. the 'occurrence' aspect of the grammatical resources); then we refine the picture by taking into account also the structural aspects of grammatical composition.

COMPOSITION: THE FORM DIMENSION. A resource-sensitive style of inference would seem to be a good starting point to come to grips with issues of grammatical multiplicity.[5] Consider the 'multiplicative' conjunction of linear logic, interpreted as the material composition of parts. The composition operation ($\circ$ in our notation) comes with an implication, which we write as $\multimap$, expressing incompleteness with respect to multiplicative composition. Using a linear implication $A \multimap B$ one actually 'consumes' a datum of type A in order to produce a B. The rules of inference in (4) state how one can *use* a resource implication and how one can *prove* an implicational goal, that is a claim of the form $A \multimap B$. We write $\Gamma \vdash A$ for the judgement that a structure Γ is a well-formed expression of type A. Notice that in the modus ponens rule, the use of the implication $\multimap$ goes hand in hand with the introduction of the structure building operation $\circ$ composing the structures Γ and Δ which the premises show to be of type A and $A \multimap B$ respectively. The rule of hypothetical reasoning *withdraws* a component A from the composition structure $A \circ \Gamma$ which the premise shows to be of type B, in order to prove that Γ is of type $A \multimap B$.

(4) ($\multimap$ E) from $\Gamma \vdash A$ and $\Delta \vdash A \multimap B$, conclude $\Gamma \circ \Delta \vdash B$ *modus ponens*
($\multimap$ I) from $A \circ \Gamma \vdash B$, conclude $\Gamma \vdash A \multimap B$ *hypothetical reasoning*

Using the linear implication to express grammatical incompleteness, we capture the resource-sensitive aspects of grammatical composition in deductive terms. Let us look first at local dependencies. In (5) we represent the type assignment to the lexical resources that would go into the composition of the sentence 'Alice talks to the Footman' (the 'numeration', in the vocabulary of the Minimalist Program). We number the lexical assumptions for future reference.

[5] See Morrill and Carpenter (1990) for an early assessment of the connection.

(5) 1. Alice $\vdash np$ *Lex*
2. talks $\vdash pp \multimap (np \multimap s)$ *Lex*
3. to $\vdash np \multimap pp$ *Lex*
4. the $\vdash n \multimap np$ *Lex*
5. footman $\vdash n$ *Lex*

The reasoning steps that lead from the lexical assumptions to the conclusion that 'Alice talks to the Footman' is indeed a datum of type s are given below. Each step of modus ponens is justified with a reference to the line which has the supporting judgements. Using Minimalist terminology, at each step of $\multimap$ elimination, the parts are 'merged' by means of the structure building operation $\circ$.

(6) 6. the $\circ$ footman $\vdash np$ $\multimap E(4,5)$
7. to $\circ$ the $\circ$ footman $\vdash pp$ $\multimap E(3,6)$
8. talks $\circ$ to $\circ$ the $\circ$ footman $\vdash np \multimap s$ $\multimap E(2,7)$
9. Alice $\circ$ talks $\circ$ to $\circ$ the $\circ$ footman $\vdash s$ $\multimap E(1,8)$

As desired, subcategorizational principles such as the ones mentioned above are 'encapsulated' into the deductive behaviour of a logical constant: the resource implication $\multimap$ expressing grammatical incompleteness. In the case of the local dependencies of (2), reasoning proceeds by modus ponens inferences. Moving on to unbounded dependencies such as (3), it turns out that the same constant $\multimap$ is expressive enough to establish the correlation between a relative pronoun and the absence of certain grammatical material in the relative clause body. This time, *hypothetical reasoning* for the resource implication provides the crucial inference steps.

(7) 0. whom $\vdash (np \multimap s) \multimap (n \multimap n)$ *Lex*

Lexical type assignment to the relative pronoun is given in (7). The formula expresses the fact that 'whom' will produce a relative clause ($n \multimap n$) when combined with the relative clause body of type $np \multimap s$. This nested implication launches a process of hypothetical reasoning: in order to establish the claim that the relative clause body is of type $np \multimap s$, we prove that with an extra np resource (line *6'*) the body would be of type s (line *9'*). At the point where this subproof is completed successfully, the $\multimap I$ inference withdraws the np hypothesis (line *10'*).

(8) 6'. | $x \vdash np$ *Hyp*
7'. | to $\circ$ $x \vdash pp$ $\multimap E(3,6')$
8'. | talks $\circ$ to $\circ$ $x \vdash np \multimap s$ $\multimap E(2,7')$
9'. | Alice $\circ$ talks $\circ$ to $\circ$ $x \vdash s$ $\multimap E(1,8')$
10'. Alice $\circ$ talks $\circ$ to $\vdash np \multimap s$ $\multimap I(6',9')$
11'. whom $\circ$ Alice $\circ$ talks $\circ$ to $\vdash n \multimap n$ $\multimap E(0,10')$
12'. footman $\circ$ whom $\circ$ Alice $\circ$ talks $\circ$ to $\vdash n$ $\multimap E(5,11')$
13'. the $\circ$ footman $\circ$ whom $\circ$ Alice $\circ$ talks $\circ$ to $\vdash np$ $\multimap E(4,12')$

COMPOSITION: THE MEANING DIMENSION. So far, we have limited our attention to the 'form' aspect of grammatical composition — to the way in which Introduction and Elimination of the $\multimap$ connective interacts with the structure-building operation $\circ$. But as announced at the beginning of this paper, the deductive perspective extends to the composition of grammatical 'meaning'.[6] In (9) we present the inference rules for $\multimap$ with a semantic annotation. The basic declarative units now are pairs $x{:}A$, where A is a formula and x a term of the simply typed lambda calculus — the representation language we use for grammatical meanings. Each inference rule is associated with an operation providing term decoration for the conclusion, given term decorations to the premises: function application, in the case of $\multimap E$, and function abstraction for $\multimap I$. Given a configuration Γ of assumptions $x_i{:}A_i$, the *process* of proving $\Gamma \vdash t{:}B$ produces a program t that specifies how to compute the meaning of the result B out of the input parameters x_i. This essentially *dynamic* (or: 'derivational', 'proof-theoretic') perspective on meaning composition is known as the Curry-Howard interpretation of proofs.

(9) $$\frac{\Gamma \vdash u:A \quad \Delta \vdash t:A \multimap B}{\Gamma \circ \Delta \vdash t\,u:B}\ (\multimap E) \qquad \frac{x:A \circ \Gamma \vdash t:B}{\Gamma \vdash \lambda x.t:A \multimap B}\ (\multimap I)$$

As an illustration, (10) gives the proofterms for some crucial stages in our earlier derivations. (We use boldface word forms as stand-ins for the unanalysed meanings of the lexical resources.) Line 9 is a pure application term, built up in the four $\multimap E$ steps of (6). Line 10' gives the proofterm for the relative clause body of (8), with abstraction over a variable x of type *np* as the correlate of the withdrawal of a hypothetical assumption in the $\multimap I$ step. Line 13', then, is the derivational meaning for the full noun phrase 'the footman whom Alice talks to'.

(10) 9. ((talk (**to** (**the footman**))) **Alice**)
10'. λx.((**talk** (**to** x)) **Alice**)
13'. (**the** ((**whom** λx.((**talk** (**to** x)) **Alice**)) **footman**))

The grammatical organization proposed here has a number of built-in proof-theoretic constraints with important consequences for 'interface' issues.

- PROOFS AS MEANING PROGRAMS. The composition of form and meaning proceeds in parallel, and is fully 'inference-driven'. There is no *structural* representation level of the grammatical resources (such as 'Logical Form') where meaning is read off. Instead, meaning is computed from the derivational process that puts the resources together.

[6] The use of resource-sensitive notions of meaning composition has become an important theme within the framework of LFG recently. See Dalrymple (1999) for a representative collection. But LFG 'syntax' is still put together by extra-logical means. We reject this dualism and advocate the stronger position that both grammatical form *and* meaning are put together in a process of resource-sensitive deduction.

- MEANING PARAMETRICITY.[7] The actual meanings of the resources that enter into the composition process are 'black boxes' for the Curry-Howard computation. No assumptions about the content of the actual meanings can be built into the meaning assembly process.
- RESOURCE SENSITIVITY. Because the grammar logic has a resource-sensitive notion of inference (each assumption is used exactly once), there is no need for 'syntactic' bookkeeping stipulations restricting variable occurrences: vacuous abstractions, closed subterms, multiple binding of variables, or unbound variables (other than the proof parameters) simply do not arise.

LEXICAL VERSUS DERIVATIONAL MEANING. The resource constraints impose severe limits on 'derivational' expressivity. But the grammar can overcome these limitations by means of *lexical* instructions for meaning composition. Consider the single-bind property of the λ abstractor — a consequence of resource-sensitivity. For the relative clause example of (8), we would like to associate the relative pronoun with an instruction to compute a property intersection semantics: intersection of the property obtained by abstracting over a *np* variable in the relative clause body, and the *n* property of the common noun which the relative clause combines with. Expressed as a lambda term, this means double binding of an entity type variable: $\lambda x.(\text{RELBODY}\ x) \wedge (\text{COMMONNOUN}\ x)$, a term which the derivational system cannot compute. However, we can 'push' the double bind term into the lexical semantics associated with 'whom', as shown in (11a).[8] Substituting the lexical program into the derivational proofterm for (10, line 13'), one obtains (11b) after simplification.

(11) a. whom: $(np \multimap s) \multimap (n \multimap n) - \lambda x_1 \lambda x_2 \lambda x_3.(x_1 x_3) \wedge (x_2 x_3)$
b. **the** $(\lambda x.((\textbf{talk}\ (\textbf{to}\ x))\ \textbf{Alice}) \wedge (\textbf{footman}\ x))$

1.2 *Grammatical reasoning: logic, structure and control*

In the preceding section we have ignored all structural aspects of grammatical composition. This was a deliberate move: we wanted to isolate the 'resource multiplicity' factor underlying both local and non-local dependencies in pure laboratory conditions, so to speak. As things stand, the $\circ$ operation of linear logic is insensitive to linear order (it does not discriminate between $\Delta_1 \circ \Delta_2$ and $\Delta_2 \circ \Delta_1$), and to hierarchical grouping (the structures $\Delta_1 \circ (\Delta_2 \circ \Delta_3)$ and $(\Delta_1 \circ \Delta_2) \circ \Delta_3$ count as the same). Technically, the structural rules of Commutativity and Associativity are still built-in components of the multiplicative operators of linear logic. Obviously, such a notion of composition is too crude, if we want to take grammatically relevant aspects of linguistic *form* into account.

We refine the tools for grammatical analysis by pushing the strategy of separating 'logic' and 'structure' to its natural conclusion: we drop Associativity and Commutativity as hard-wired components of the grammatical constants, obtaining the truly 'minimal'

[7] The term is from Dalrymple et al. (1999).

[8] The format for lexical entries is *word form: type formula – lexical recipe.*

logic of composition; then we bring these structural options back under explicit logical control. Dropping Associativity, the 'constituent structure' configuration of the resources becomes relevant for grammaticality judgements. Let f and g be resources with types $A \multimap B$ and $B \multimap C$ respectively. In an associative regime, f and g can be put together, and $f \circ g$ yields a conclusion of type $A \multimap C$, with the derivational meaning of function composition $\lambda x.g(fx)$. In a non-associative setting, this inference no longer goes through: the hypothetical A assumption and the implication $A \multimap B$ that would have to consume it, are not within the same constituency domain. Dropping also Commutativity, the resource implication $A \multimap B$ splits up into a left-handed $A\backslash B$ and a right-handed B/A, implications that insist on following or preceding the A resource they are consuming.

THE BASE LOGIC: RESIDUATION. We are in a position now to introduce the minimal logic of grammatical composition. For an easy presentation format, it is handy to introduce in the formula language a connective $\bullet$, corresponding to the structure building operations $\circ$: whereas $\circ$ puts together *structures* Γ and Δ into the composition structure $\Gamma \circ \Delta$, the $\bullet$ connective puts together *formulas* A and B into the product formula $A \bullet B$. With the explicit product connective, we can express deducibility judgements as statement of the form $A \rightarrow B$, where A is the $\bullet$ formula-equivalent of the $\circ$ structure Γ in our earlier formulation $\Gamma \vdash B$. The essential deductive principles of the base logic, then, are given by the so-called RESIDUATION laws of Fig. 1, which establish the correlation between grammatical incompleteness (as expressed by / and\) and composition ($\bullet$). Together with reflexivity ($A \rightarrow A$) and transitivity of the derivability relation (from $A \rightarrow B$ and $B \rightarrow C$, conclude $A \rightarrow C$) the residuation laws fully characterize the valid inferences of the base logic.[9]

$$A \rightarrow C/B \text{ iff } A \bullet B \rightarrow C \text{ iff } B \rightarrow A\backslash C$$

Figure 1. The base logic: residuation

Some familiar theorems and derived inference rules of the base logic are given below. It is important to keep in mind that these are 'universal' principles of grammatical composition, in the sense that they hold no matter what the structural properties of the composition relation may be. There is no option for cross-linguistic variation with respect

[9] There is a precise technical sense in which we are dealing with the truly 'minimal' logic here. The models for the base logic are specified with respect to *frames* $<W, R>$ (as for modal logic), where W is a set of grammatical resources, structured by the composition relation R ('Merge'). Formulas are interpreted as subsets of W. The constant $\bullet$ has the following interpretation: $x \in \|A \bullet B\|$ iff there exist grammatical parts y,z such that $y \in \|A\|$, $z \in \|B\|$ and $Rxyz$. (The implications / and \ are interpreted as the residuation duals.) The basic completeness result then says that $A \rightarrow B$ is provable iff, for every valuation on every frame, we have $\|A\| \subseteq \|B\|$. The laws of the base logic, in other words, do not impose any restrictions on the interpretation of the composition relation. But the addition of *structural postulates* does indeed restrict the interpretation of R to meet certain structural conditions. The 'modal' perspective on grammatical logics is worked out in depth in Kurtonina (1995).

to the principles in (12), in other words. But languages can vary with respect to a principle such as $(A\backslash B) \bullet (B\backslash C) \rightarrow A\backslash C$, which is not available in the base logic, but dependent on associativity assumptions, as we saw above.

(12) *application:* $(A/B) \bullet B \rightarrow A,\ B \bullet (B\backslash A) \rightarrow A$
lifting: $A \rightarrow B/(A\backslash B),\ A \rightarrow (B/A)\backslash B$
monotonicity: from $A \rightarrow B$ and $C \rightarrow D$, conclude $\left\{ \begin{array}{l} A/D \rightarrow B/C \\ A \bullet C \rightarrow B \bullet D \\ D\backslash A \rightarrow C\backslash B \end{array} \right.$

STRUCTURE AND CONTROL. From the base logic, we could recover the expressivity of the linear logic multiplicatives by adding postulates of Commutativity and Associativity.

(13) $A \bullet B \rightarrow B \bullet A$
$(A \bullet B) \bullet C \leftrightarrow A \bullet (B \bullet C)$

But what we said above about unrestricted use of waste and duplication of assumptions (Weakening, Contraction) applies to structural resource management as well: instead of global hard-wired settings, we need lexical *control* over resource management. Consider the Commutativity option. Example (14) gives some alternative ways of rendering a well-known Latin phrase. Although Latin has much greater flexibility with respect to linear order than, say, Dutch or English, it would be wrong to assume that Latin composition obeys a globally commutative regime: as the (c) example shows, a preposition like 'cum' has to precede its nominal complement. The challenge here is to reconcile the structural freedom of, for example, adjectival modifiers, with the rigid order requirements of prepositions.

(14) a. cum magna laude
b. cum laude magna
c. magna cum laude
d. *magna laude cum

Associativity, i.e. flexibility of constituency, has often been called upon to derive instances of 'non-constituent' coordination, such as the Right Node Raising case below. Yet, as the contrast between (15b) and (15c) shows, a global regime allowing restructuring, such as implemented by a structural postulate of Associativity, overgenerates: an associative regime would judge both the transitive verb 'love' and the non-constituent cluster 'thinks Mary loves' to be resources of type $(np\backslash s)/np$, and hence indistinguishable as arguments of 'himself', which as a relation reducer could be typed as $((np\backslash s)/np)\backslash(np\backslash s)$. In this case, one would like to lexically control structural relaxation in such a way that it is only licensed in the presence of the coordination particles.

(15) a. $\underbrace{\text{the Lobster loves}}_{s/np}$ but $\underbrace{\text{the Gryphon hates}}_{s/np}$ Turtle Soup

b. the Mad Hatter $\underbrace{\text{loves}}_{(np\backslash s)/np}$ himself

c. *the Mad Hatter $\underbrace{\text{thinks Alice loves}}_{(np\backslash s)/np}$ himself

In order to gain logical control over the structural aspects of grammatical resource management, we extend the formula language of the grammar logic with a pair of constants, $\Diamond$ and $\Box$. These constants will play a role analogous to the linear logic modalities governing resource multiplicity. We study $\Diamond$ and $\Box$ in their 'logical' and their 'structural' parts, as we did with the binary connectives. As for the logical part, the relation between $\Diamond$ and $\Box$ is the same as that between product and slash: they are residuation duals. In algebraic terms, we have the biconditional law of Fig. 2.

$$\Diamond A \rightarrow B \Leftrightarrow A \rightarrow \Box B$$

Figure 2. Residuation: unary connectives

Section 2 below is devoted to an illustration of the linguistic use of these new connectives. It will be useful here to prepare the ground and present some basic inferential patterns. Notice that the base logic allows neither $\Box A \rightarrow A$ nor $A \rightarrow \Diamond A$. Instead, the basic cancellation law is $\Diamond\Box A \rightarrow A$, with the dual pattern $A \rightarrow \Box\Diamond A$, as the reader can check in (16).[10]

(16)	from $\Box A \rightarrow \Box A$	*(Axiom),*	from $\Diamond A \rightarrow \Diamond A$	*(Axiom),*
	conclude $\Diamond\Box A \rightarrow A$	*(Res* $\Leftarrow$*)*	conclude $A \rightarrow \Box\Diamond A$	*(Res* $\Rightarrow$*)*

The constants $\Diamond$ and $\Box$ can play a central role in providing a logical basis for the 'control features' used within the Minimalist Program. The function of minimalist features is markedly different from the role they play in unification-based grammar architectures: minimalist features are not just 'there', they are active control devices that have to be explicitly 'checked' in the process of derivation. This resource-sensitive character of the control devices is captured exactly by the logic of $\Diamond$ and $\Box$: $\Box$ expresses the request for a feature to be checked, and composition with $\Diamond$ satisfies that request, 'checking' the feature.

Let us turn then to the *structural* component of the unary connectives, and see how

[10] The logic of grammar, in other words, is not a modal logic with principle **T**. Rather, the $\Diamond,\Box$ modalities are related like the inverse duals of *temporal* logic ('will be the case', 'has always been the case'): $x \in \|\Diamond A\|$ iff there exists y such that Rxy and $y \in \|A\|$ versus $x \in \|\Box A\|$ iff for all y, Ryx implies $y \in \|A\|$. Here R is a binary relation interpreting the unary $\Diamond,\Box$, cf. the ternary composition relation interpreting $\bullet$ and its residuals.

they can be used to control the composition process. In (17) we present some examples of modally restricted versions of structural postulates that, in the global form of (13), would destroy structural discrimination with respect to linear order or constituency, as we have seen above. Reordering or restructuring, in (17), has to be explicitly *licensed* by the presence of a $\Diamond$ decorated formula. And we'll see in Section 2 that this modal marking can be 'projected' from lexical type assignment, the way the structure building operation $\circ$ is driven by the $/,\backslash$ implications in the typing of lexical resources.

(17) $A \bullet \Diamond B \rightarrow \Diamond B \bullet A$

$(A \bullet B) \bullet \Diamond C \rightarrow A \bullet (B \bullet \Diamond C)$

To close this section we present the Natural Deduction format for the grammar logic — our display format for grammatical analysis in Section 2. A proof proceeds from axioms $x{:}A \vdash x{:}A$, where A is a formula, and x a variable of that type for the construction of the Curry-Howard proof term. Rules of inference for the binary vocabulary are given in Fig. 3. We distinguish the two resource implications, and add the inference rules for the $\bullet$ connective. In the absence of Commutativity/Associativity, the structure building operation $\circ$ now configures the resources as a *tree* (bracketed string). Notice that $/$ and $\backslash$ introduce refinement in the *form* dimension: with respect to the Curry-Howard derivational meaning, they are both interpreted in terms of function application and abstraction. Term decoration for the $\bullet$ connective associates introduction of this connective with pairing $\langle\cdot,\cdot\rangle$, and elimination with (left $(\cdot)_0$ and right $(\cdot)_1$) projection.

$$[/I]\frac{\Gamma \circ x : B \vdash t : A}{\Gamma \vdash \lambda x.t : A/B} \qquad \frac{\Gamma \vdash t : A/B \quad \Delta \vdash u : B}{\Gamma \circ \Delta \vdash (t\ u) : A}[/E]$$

$$[\backslash I]\frac{x : B \circ \Gamma \vdash t : A}{\Gamma \vdash \lambda x.t : B\backslash A} \qquad \frac{\Gamma \vdash u : B \quad \Delta \vdash t : B\backslash A}{\Gamma \circ \Delta \vdash (t\ u) : A}[\backslash E]$$

$$[\bullet E]\frac{\Delta \vdash u : A \bullet B \quad \Gamma[x : A \circ y : B] \vdash t : C}{\Gamma[\Delta] \vdash t[(u)_0/x, (u)_1/y] : C} \qquad \frac{\Gamma \vdash t : A \quad \Delta \vdash u : B}{\Gamma \circ \Delta \vdash \langle t, u\rangle : A \bullet B}[\bullet I]$$

Figure 3. Grammatical composition: $/$, $\bullet$, $\backslash$

In the natural deduction format, the residuation laws for $\Diamond$ and $\Box$ turn up as the Introduction and Elimination rules of Fig. 4. We use $\langle\cdot\rangle$ as the structure building operation corresponding to the logical constant $\Diamond$. In the term language for derivational semantics, we have constructors (the 'cap' operators) and destructors (the 'cup' operators) for the Introduction and Elimination inferences respectively.

$$\frac{\Gamma \vdash t : \Box A}{\langle \Gamma \rangle \vdash {}^{\vee}t : A}\ (\Box E) \qquad \frac{\langle \Gamma \rangle \vdash t : A}{\Gamma \vdash {}^{\wedge}t : \Box A}\ (\Box I)$$

$$\frac{\Gamma \vdash t : A}{\langle \Gamma \rangle \vdash {}^{\cap}t : \Diamond A}\ (\Diamond I) \qquad \frac{\Delta \vdash u : \Diamond A \quad \Gamma[\langle x : A \rangle] \vdash t : B}{\Gamma[\Delta] \vdash t[{}^{\cup}u/x] : B}\ (\Diamond E)$$

Figure 4. Structural control modalities.

Figures 3 and 4 cover the grammatical base logic. The translation between structural postulates, in the algebraic presentation, and structural rules for the N.D. format is straightforward. A postulate $A \rightarrow B$ corresponds to a rule of inference licensing replacement of a substructure Δ' in the premise by Δ in the conclusion, where Δ and Δ' are the structure equivalents of the (product) formulas A and B respectively.[11]

(18) $A \rightarrow B$ *(postulate)* $\qquad \dfrac{\Gamma[\Delta'] \vdash C}{\Gamma[\Delta] \vdash C}$ (N.D. rule)

2. Fine-tuning grammatical resource management

The components of the grammatical architecture proposed in the previous section are summarized below.

Logic. The core logical notions of grammatical composition ('Merge') are characterized in terms of universal laws, independent of the structural properties of the composition relation. The operations of the base logic (introduction/elimination of the grammatical constants) provide the interface to a derivational theory of meaning via the Curry-Howard interpretation of proofs.

Structure. Packages of resource-management postulates function as 'plug-in' modules with respect to the base logic. They offer a logical perspective on structural variation, within languages and cross-linguistically.

Control. A vocabulary of control operators provides explicit means to fine-tune grammatical resource management, by imposing structural constraints or by licensing structural relaxation.

To illustrate the interplay of these three components we return to *wh* dependencies in relativization. In Section 1.1, we concentrated on the 'multiplicity' aspect of these dependencies, and abstracted from structural factors. Our strategy for uncovering the fine-structure of grammatical resource management will be to make a *minimal* use of

[11] This back-and-forth translation between structures and formulas works on the assumption that we write postulates purely in terms of the connectives $\Diamond$ and $\bullet$ (and formula variables), as indeed we will.

structural postulates, thus exploiting the inferential capacity of the base logic to the full. As a case study, we contrast relativization in English and Dutch.

2.1 *English: right branch extraction*

Consider the English case first. As remarked above, binding of the subject of the relative clause body is structurally free: the examples in (20) are derivable in the base logic from the lexical assignments shown in (19) for relative pronouns 'who', 'that'. (As (20c) indicates, this type assignment is not appropriate for 'whom'.)

(19) who, that : $(n\backslash n)/(np\backslash s) - \lambda x\lambda y\lambda z.(x\ z) \wedge (y\ z)$

(20) a. (the song) that irritated the Gryphon
b. (the girl) who irritated the Duchess
c. (the girl) *whom irritated the Duchess

$$\dfrac{\mathrm{that} \vdash (n\backslash n)/(np\backslash s) \qquad \dfrac{\dfrac{[\mathrm{x}_1 \vdash np]^1 \qquad \dfrac{\mathrm{irritated} \vdash (np\backslash s)/np \qquad \dfrac{\mathrm{the} \vdash np/n \qquad \mathrm{gryphon} \vdash n}{\mathrm{the} \circ \mathrm{gryphon} \vdash np}[/E]}{\mathrm{irritated} \circ (\mathrm{the} \circ \mathrm{gryphon}) \vdash np\backslash s}[/E]}{\mathrm{x}_1 \circ (\mathrm{irritated} \circ (\mathrm{the} \circ \mathrm{gryphon})) \vdash s}[\backslash E]}{\mathrm{irritated} \circ (\mathrm{the} \circ \mathrm{gryphon}) \vdash np\backslash s}[\backslash I]^1}{\mathrm{that} \circ (\mathrm{irritated} \circ (\mathrm{the} \circ \mathrm{gryphon})) \vdash n\backslash n}[/E]$$

Figure 5. 'that irritated the Gryphon'

Consider now non-subject cases of relativization, such as the binding of the direct object role in 'the book that Dodgson wrote'. We have seen above that implication introduction in the base logic is restricted to the immediate (left or right) daughter of the structural configuration from which the hypothetical resource is withdrawn: the subject is thus accessible in Fig. 5,[12] but the direct object, as a daughter of the verb phrase, cannot be reached with a non-subject extraction assignment as given in (21). Under what *structural* assumptions can we make the appropriate set of non-subject positions accessible for relativization?

[12] In using the N.D. format of Figures 3 and 4, we stick to the handy 'sugared' presentation of Section 1.1: we omit the formula part on the left of $\vdash$, and use the word forms of the lexical assumptions as the term 'variables'.

$$\dfrac{\mathrm{the} \vdash np/n \quad \dfrac{\mathrm{book}\vdash n \quad \dfrac{\mathrm{that}\vdash (n\backslash n)/(s/np) \quad \dfrac{\dfrac{\dfrac{\mathrm{D}\vdash np \quad \dfrac{\mathrm{wrote}\vdash (np\backslash s)/np \quad [\mathrm{x}_1\vdash np]^1}{\mathrm{wrote}\circ \mathrm{x}_1 \vdash np\backslash s}[/E]}{\mathrm{D}\circ(\mathrm{wrote}\circ \mathrm{x}_1)\vdash s}[\backslash E]}{(\mathrm{D}\circ \mathrm{wrote})\circ \mathrm{x}_1\vdash s}[A1]}{\mathrm{D}\circ\mathrm{wrote}\vdash s/np}[/I]^1}{\mathrm{that}\circ(\mathrm{D}\circ\mathrm{wrote})\vdash n\backslash n}[/E]}{\mathrm{book}\circ(\mathrm{that}\circ(\mathrm{D}\circ\mathrm{wrote}))\vdash n}[\backslash E]}{\mathrm{the}\circ(\mathrm{book}\circ(\mathrm{that}\circ(\mathrm{D}\circ\mathrm{wrote})))\vdash np}[/E]$$

Figure 6. 'the book that Dodgson wrote'

(21) that, whom : $(n\backslash n)/(s/np) - \lambda x\lambda y\lambda z.x(z) \wedge y(z)$

(22) $(A \bullet B)\bullet C \rightarrow A \bullet(B \bullet C)$ $\quad A1$

Figure 6 shows that the associativity postulate *A1* realizes a restructuring that does make the direct object accessible. But does this postulate express the proper structural generalization? The answer must be negative — both on grounds of overgeneration and of undergeneration. As to the latter: *A1* (in combination with the type assignment in (21)) makes accessible only *right-peripheral* positions in the relative clause body. A relative clause such as 'the book that Dodgson dedicated to Alice Liddell', for example, would still be underivable. As to overgeneration, we have seen in our discussion of (15) that global availability of restructuring destroys constituent information that may be grammatically relevant.

The control operators provide the logical vocabulary to implement a more delicate resource management regime. In (23), the type assignment to non-subject relative pronouns is refined by adding a modal decoration $\Diamond\Box$ to the hypothetical *np* subtype. The postulate package of Fig. 7, keyed to the $\Diamond$ modality, then licenses structural access to non-subject positions. As this section proceeds, we will gradually accumulate motivation for the specific formulation of *P1* and *P2*. Let it suffice for now to remark that we have not introduced any *global* loss of structure-sensitivity (as an Associativity postulate for $\bullet$ would do); instead, we have narrowed down the structurally 'special' behavior to the 'gap' resource. Moreover, the postulates of Fig. 7 do not license access to *arbitrary* positions within the relative clause body: they only allow $\Diamond$ marked resources to communicate recursively with *right* branches of $\bullet$ structures.[13]

(23) that, whom: $(n\backslash n)/(s/\Diamond\Box np) -$ for semantics, cf. (25b)

[13] With the structural postulates of (17), arbitrary positions would indeed be accessible, making $rel/(s/\Diamond\Box np)$ and $rel/(\Diamond\Box np\backslash s)$ indistinguishable. In the text we pursue a more discriminating alternative, exploiting the left/right asymmetry.

$$(A \bullet B) \bullet \Diamond C \rightarrow (A \bullet \Diamond C) \bullet B \qquad P1$$
$$(A \bullet B) \bullet \Diamond C \rightarrow A \bullet (B \bullet \Diamond C) \qquad P2$$

Figure 7. Right branch extraction

A derivation for the relative clause '(the book) that Dodgson dedicated to Liddell' is presented in Fig. 8. Notice carefully how the structural control inferences interact with the purely logical steps.

- At a certain point in the derivation, the hypothetical $\Diamond\Box np$ resource will have to play the structural role of a simple (non-modalized) np, as a result of the reduction law $\Diamond\Box np \rightarrow np$. In the example of Fig. 8, the np is consumed in the direct object position.
- As long as this reduction has not applied, i.e. as long as the modal prefix $\Diamond\Box$ is intact, the leading $\Diamond$ licenses structural inferences $P1$ and $P2$. These inferences establish the communication between the clause peripheral position, where $\Diamond\Box np$ can be withdrawn in the Introduction step, and the structural position where np is actually consumed.

$$\dfrac{\text{that} \vdash relpro \qquad \dfrac{\dfrac{[\mathrm{x}_1 \vdash \Diamond\Box np]^1 \qquad \dfrac{\dfrac{\dfrac{\text{D} \vdash np \qquad \dfrac{\dfrac{\text{dedicated} \vdash ((np\backslash s)/pp)/np \qquad \dfrac{[\mathrm{x}_2 \vdash \Box np]^2}{\langle \mathrm{x}_2 \rangle \vdash np}\,[\Box E]}{\text{dedicated} \circ \langle \mathrm{x}_2 \rangle \vdash (np\backslash s)/pp}\,[/E] \qquad \dfrac{\text{to} \vdash pp/np \quad \text{L} \vdash np}{\text{to} \circ \text{L} \vdash pp}\,[/E]}{(\text{dedicated} \circ \langle \mathrm{x}_2 \rangle) \circ (\text{to} \circ \text{L}) \vdash np\backslash s}\,[/E]}{\text{D} \circ ((\text{dedicated} \circ \langle \mathrm{x}_2 \rangle) \circ (\text{to} \circ \text{L})) \vdash s}\,[\backslash E]}{\text{D} \circ ((\text{dedicated} \circ (\text{to} \circ \text{L})) \circ \langle \mathrm{x}_2 \rangle) \vdash s}\,[P1]}{(\text{D} \circ (\text{dedicated} \circ (\text{to} \circ \text{L}))) \circ \langle \mathrm{x}_2 \rangle \vdash s}\,[P2]}{(\text{D} \circ (\text{dedicated} \circ (\text{to} \circ \text{L}))) \circ \mathrm{x}_1 \vdash s}\,[\Diamond E]^2}{\text{D} \circ (\text{dedicated} \circ (\text{to} \circ \text{L})) \vdash s/\Diamond\Box np}\,[/I]^1}{\text{that} \circ (\text{D} \circ (\text{dedicated} \circ (\text{to} \circ \text{L}))) \vdash n\backslash n}\,[/E]$$

Figure 8. 'that Dodgson dedicated to Liddell' (*relpro* is $(n\backslash n)/(s/\Diamond\Box np)$)

The lexical semantics presented above for the non-modalized relative pronouns has to be refined to take the added structure-sensitivity into account. Consider what happens at the end of the conditional subproof for the relative clause body: the $/I$ rule withdraws a $\Diamond\Box np$ hypothetical resource, semantically binding a variable of that type (x_1). But in order to supply the appropriate type for the direct object np argument of 'dedicated' in the body of the relative clause, the $\Diamond$ and $\Box$ Elimination inferences have to 'lower' x_1 to ${}^{\vee\cup}x_1$.

(24) [/I] $\lambda x_1.(((\textbf{dedicated}\ {}^{\vee\cup}\mathrm{x}_1)\ (\textbf{to Liddell}))\ \textbf{Dodgson})$

Now compare the 'property intersection' lexical semantics for the non-modalized relative pronouns in (25a) with the refined meaning recipe for the modalized assignment in (25b). In the modal case, the operations ${}^{\cap\wedge}$ lift the entity-type variable z to the appropriate level to serve as an argument to the x variable, for the relative clause body which is now of type $s/\Diamond\Box np$.

(25) a. $(n\backslash n)/(np\backslash s) - \lambda x\lambda y\lambda z.(x\ z) \wedge (y\ z)$
b. $(n\backslash n)/(s/\Diamond\Box np) - \lambda x\lambda y\lambda z.(x^{\cap\wedge}z) \wedge (y\ z)$

The derivational meaning for the complete relative clause 'that Dodgson dedicated to Liddell' is given in (26). Substitution of the lexical semantics (25b) for 'that' leads to a term that can be simplified ('cup-cap' cancellation, twice: ${}^{\vee\cup\cap\wedge}x_1 = x_1$), which ultimately produces the desired property intersection semantics, when combined with a common noun meaning for the abstraction over y.

(26) **that** $(\lambda x_1.(((\mathbf{dedicated}\ {}^{\vee\cup}x_1)\ (\mathbf{to\ Liddell}))\ \mathbf{Dodgson}))$
$[\lambda x\lambda y\lambda z.(x^{\cap\wedge}z) \wedge (y\,z)]\ (\lambda x_1.(((\mathbf{dedicated}\ {}^{\vee\cup}x_1)\ (\mathbf{to\ Liddell}))\ \mathbf{Dodgson})) \rightarrow$
$\lambda y.(\lambda z.((((\mathbf{dedicated}\ z)\ (\mathbf{to\ Liddell}))\ \mathbf{Dodgson}) \wedge (y\ z)))$

2.2 *Dutch: left branch extraction*

The observant reader will have noticed that the postulate package of Fig. 7 is sensitive to the branching configuration of the structure it interacts with: $\Diamond$ marked material is accessible on right branches, but not on left branches. This choice limits the structural positions which the modalized relative pronoun type $(n\backslash n)/(s/\Diamond\Box np)$ can establish communication with. Some empirical consequences are illustrated below. Prepositions (*pp/np*) in English can be stranded as in (27a); (embedded[14]) subjects are inaccessible, leading to the so-called 'that-trace' effect of (27b); but the 'that-trace' violation can be avoided as in (27d) via a (complementizer-less) type assignment to 'think' which makes the 'embedded' subject a direct argument of the higher predicate, and realizes the required semantic composition via the associated lexical meaning recipe of (27e.)

(27) a. the girl whom Carroll dedicated his book to
b. *the footman whom Alice thinks that stole the tarts
c. thinks: $(np\backslash s)/cs$, that: cs/s
d. the footman whom Alice thinks stole the tarts
e. thinks: $((np\backslash s)/(np\backslash s))/np - \lambda x\lambda y\lambda z.((\mathbf{think}(y\ x))z)$

For an SVO language like English, where heads select their complements to the right, the right-branch extraction package of Fig. 7 has pleasant consequences. The distinct type-assignment to subject and non-subject cases of relativization correlates with the morphological *who whom* alternatives. But what about an SOV language, where complement selection is (predominantly) to the left? Considerations of symmetry would suggest the mirror-image left-branch extraction package of Fig. 9 here, together with type assignment to the relative pronouns launching the hypothetical 'gap' resource at the *left* periphery of the relative clause body.

(28) die, dat: $(n\backslash n)/(\Diamond\Box np\backslash s)$

[14] Remember we have the type assignment $(n\backslash n)/(np\backslash s)$ for relativization of the main subject of the relative clause body.

$$\Diamond A \bullet (B \bullet C) \rightarrow B \bullet (\Diamond A \bullet C) \qquad P1'$$
$$\Diamond A \bullet (B \bullet C) \rightarrow (\Diamond A \bullet B) \bullet C \qquad P2'$$

Figure 9. Left branch extraction

Let us contrast the empirical consequences of the type assignment (23) and the structural package in Fig. 9 with what we found above for Fig. 7 and the relative pronoun type-assignments for English. First of all, the asymmetry between subject and non-subject relativization (which in English gives rise to the 'that-trace' effect) disappears with the left-branch extraction package. As the reader can check, the subject-extraction case of (19) which was posited as a separate type-assignment in English, is *derivable* from (23). As a matter of fact, the type transition of (29) is valid already in the base logic — it does not depend on structural assumptions.

(29) $(n\backslash n)/(\Diamond\Box np\backslash s) \rightarrow (n\backslash n)/(np\backslash s)$

As a result of (29), a Dutch relative clause like (30a) has two possible derivations, paraphrased in (30b) and (30d). The derivation of Fig. 10 is obtained by simply reducing the $\Diamond\Box$ prefix, without accessing the structural package. It produces the proofterm (30c) where the relative pronoun binds the subject argument of 'vindt'. Communication between the relative pronoun and the direct object position is obtained by means of a $\Diamond$ licensed structural inference $P1'$. See Fig. 11 with proofterm (30e).

(30) a. (de lakei) die Alice gek vindt
b. (the footman) who considers Alice mad
c. **(who** λx_0.**(((considers mad) Alice)** x_0**))**
d. (the footman) who(*m*) Alice considers mad
e. **(who** λx_0.**(((considers mad)** x_0**) Alice))**

$$\dfrac{\text{die} \vdash (n\backslash n)/(\Diamond\Box np\backslash s) \qquad \dfrac{\dfrac{[\mathrm{x}_0 \vdash \Diamond\Box np]^1 \qquad \dfrac{\dfrac{[\mathrm{x}_1 \vdash \Box np]^2}{\langle \mathrm{x}_1\rangle \vdash np}[\Box E] \qquad \dfrac{\text{Alice} \vdash np \qquad \dfrac{\text{gek} \vdash ap \quad \text{vindt} \vdash ap\backslash(np\backslash(np\backslash s))}{\text{gek} \circ \text{vindt} \vdash np\backslash(np\backslash s)}[\backslash E]}{\text{Alice} \circ (\text{gek} \circ \text{vindt}) \vdash np\backslash s}[\backslash E]}{\langle \mathrm{x}_1\rangle \circ (\text{Alice} \circ (\text{gek} \circ \text{vindt})) \vdash s}[\backslash E]}{\mathrm{x}_0 \circ (\text{Alice} \circ (\text{gek} \circ \text{vindt})) \vdash s}[\Diamond E]^2}{\text{Alice} \circ (\text{gek} \circ \text{vindt}) \vdash \Diamond\Box np\backslash s}[\backslash I]^1}{\text{die} \circ (\text{Alice} \circ (\text{gek} \circ \text{vindt})) \vdash n\backslash n}[/E]$$

Figure 10. 'die Alice gek vindt': subject reading

$$\dfrac{\mathrm{die} \vdash (n\backslash n)/(\Diamond\Box np\backslash s) \qquad \dfrac{\dfrac{[\mathrm{x}_0 \vdash \Diamond\Box np]^1 \qquad \dfrac{\dfrac{\mathrm{Alice} \vdash np \qquad \dfrac{\dfrac{[\mathrm{x}_1 \vdash \Box np]^2}{\langle \mathrm{x}_1 \rangle \vdash np}[\Box E] \qquad \dfrac{\mathrm{gek} \vdash ap \qquad \mathrm{vindt} \vdash ap\backslash(np\backslash(np\backslash s))}{\mathrm{gek} \circ \mathrm{vindt} \vdash np\backslash(np\backslash s)}[\backslash E]}{\langle \mathrm{x}_1 \rangle \circ (\mathrm{gek} \circ \mathrm{vindt}) \vdash np\backslash s}[\backslash E]}{\mathrm{Alice} \circ (\langle \mathrm{x}_1 \rangle \circ (\mathrm{gek} \circ \mathrm{vindt})) \vdash s}[\backslash E]}{\langle \mathrm{x}_1 \rangle \circ (\mathrm{Alice} \circ (\mathrm{gek} \circ \mathrm{vindt})) \vdash s}[P1']}{\mathrm{x}_0 \circ (\mathrm{Alice} \circ (\mathrm{gek} \circ \mathrm{vindt})) \vdash s}[\Diamond E]^2}{\mathrm{Alice} \circ (\mathrm{gek} \circ \mathrm{vindt}) \vdash \Diamond\Box np\backslash s}[\backslash I]^1}{\mathrm{die} \circ (\mathrm{Alice} \circ (\mathrm{gek} \circ \mathrm{vindt})) \vdash n\backslash n}[/E]$$

Figure 11. 'die Alice gek vindt': direct object reading

Secondly, whereas Dutch verbal heads select to the left, prepositional phrases are head-initial. Prepositional complements, in other words, are inaccessible for the left-branch extraction structural package of Fig. 9. And indeed, we do not find stranded prepositions with the regular relative pronouns, witness the ungrammaticality of (31a) as compared to the English counterpart (31b).

(31) a. *(de uitkomst) die de Koningin op rekent
b. (the outcome) which the Queen counts on
c. *op het *versus* er op
d. er:$pp/(pp/np)$ – $\lambda z.(z\ \mathbf{it})$

As we have seen before, given the grammatical architecture proposed here, the only way to overcome the expressive limitations of the derivational system for a language is to use *lexical* resources. For the relativization of prepositional complements, the Dutch relative pronoun 'waar' provides such a lexical device. Dutch has a class of (neuter) personal pronouns, the so-called R-pronouns 'er', 'daar', for which the canonical structural position is to the *left* of the preposition they depend on semantically: see the contrast in (31c). The reader will have understood by now that the grammar doesn't have to rely on structural inferences to realize the required form/meaning composition: the lexical type assignment and meaning recipe of (31d) will do the job in the base logic.

Suppose now we treat 'waar' as a relativizer with respect to an R personal pronoun. Given the type assignment of (32a) (where we use *rpro* as an abbreviation for *pp (pp np))*, the relative pronoun 'waar' can establish communication with the left branch home position of an R pronoun by means of the structural inferences of the Dutch extraction package in Fig. 9, as the derivation in Fig. 12 shows. The derivational meaning for the relative clause body is given in (32b), with an abstraction over a variable x_0 of type $\Diamond\Box rpro$. After the application of the lexical program for 'waar' to (32b), we obtain the property intersection semantics of (32c) for the complete relative clause, with the required binding of the prepositional object.

(32) a. waar: $(n\backslash n)/(\Diamond\Box rpro\backslash s) - \lambda x.(\lambda y.(\lambda z.((x^{\cap\wedge}\ (\lambda w.(w\ z))) \wedge (y\ z))))$
b. $\Diamond\Box rpro\backslash s - \lambda x_0.((\mathbf{counts}(^{\vee\cup}x_0\ \mathbf{on}))(\mathbf{the\ Queen}))$
c. $n\backslash n - \lambda y.(\lambda z.(((\mathbf{counts}(\mathbf{on}\ z))(\mathbf{the\ Queen})) \wedge (y\ z)))$

$$
\dfrac{\mathrm{waar} \vdash r\text{-}relpro \qquad \dfrac{\dfrac{[\mathrm{x}_0 \vdash \Diamond\Box rpro]^1 \qquad \dfrac{\dfrac{\dfrac{\mathrm{de\ K} \vdash np \qquad \dfrac{\dfrac{\dfrac{[\mathrm{x}_1 \vdash \Box rpro]^2}{\langle \mathrm{x}_1 \rangle \vdash pp/(pp/np)}[\Box E] \qquad \mathrm{op} \vdash pp/np}{\langle \mathrm{x}_1 \rangle \circ \mathrm{op} \vdash pp}[/E] \qquad \mathrm{rekent} \vdash pp\backslash(np\backslash s)}{(\langle \mathrm{x}_1 \rangle \circ \mathrm{op}) \circ \mathrm{rekent} \vdash np\backslash s}[\backslash E]}{\mathrm{de\ K} \circ ((\langle \mathrm{x}_1 \rangle \circ \mathrm{op}) \circ \mathrm{rekent}) \vdash s}[\backslash E]}{\mathrm{de\ K} \circ (\langle \mathrm{x}_1 \rangle \circ (\mathrm{op} \circ \mathrm{rekent})) \vdash s}[P2']}{\langle \mathrm{x}_1 \rangle \circ (\mathrm{de\ K} \circ (\mathrm{op} \circ \mathrm{rekent})) \vdash s}[P1']}{\mathrm{x}_0 \circ (\mathrm{de\ K} \circ (\mathrm{op} \circ \mathrm{rekent})) \vdash s}[\Diamond E]^2}{\mathrm{de\ K} \circ (\mathrm{op} \circ \mathrm{rekent}) \vdash \Diamond\Box rpro\backslash s}[\backslash I]^1}{\mathrm{waar} \circ (\mathrm{de\ K} \circ (\mathrm{op} \circ \mathrm{rekent})) \vdash n\backslash n}[/E]
$$

Figure 12. 'waar de Koningin op rekent' ($r\text{-}relpro$ is $(n\backslash n)/(\Diamond\Box rpro\backslash s)$)

3. Conclusion

Assessments of the categorial contribution to linguistics tend to be strongly polarized. On the one hand, the categorial approach has been praised for its mathematical elegance and for the transparent design of the syntax-semantics interface. On the other hand, classical categorial systems have been judged to be of limited use for realistic grammar development because of the coarse granularity of the notion of grammatical inference they offer. The criticism is justified, we think, for systems with a fixed resource management regime. However, the enrichment of the type-logical language with an explicit *control* vocabulary changes the black-and-white picture: we hope to have shown that linguistic discrimination is indeed compatible with mathematical sophistication.

This paper adheres to the standard categorial view that macro-grammatical organization, both at the form level and at the meaning level, is fully determined by a deductive process of type inference over lexical assignments. But this standard view has been further articulated in a novel way: by factoring out the structural aspects of grammatical composition from the logical core, we have been able to reconcile the cross-linguistic uniformity in the build-up of the form-meaning correspondence with structural variation in the realization of this correspondence. The basic deductive operations of elimination and introduction of the grammatical constants are semantically interpreted in a uniform way; packages of structural inferences, triggered by lexically-anchored control features, determine how the form-meaning correspondence finds actual expression.

References

Benthem, J. van (1991, 1995) *Language in Action. Categories, Lambdas, and Dynamic Logic*. Studies in Logic, North-Holland, Amsterdam. (Student edition: MIT Press (1995), Cambridge, MA.).

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

Chomsky, N. (1995) *The Minimalist Program*. MIT Press, Cambridge, MA.

Cornell, T.L. (1997) 'A type-logical perspective on minimalist derivations'. In G. Morrill and R.T. Oehrle (eds), *Formal Grammar 1997*. Aix-en-Provence.

Dalrymple, M. (ed.) (1999) *Semantics and Syntax in Lexical Functional Grammar. The Resource Logic Approach*. MIT Press, Cambridge, MA.

Dalrymple, M. et al. (1999) 'Relating resource-based semantics to categorial semantics'. In Dalrymple, M. (ed.), *Semantics and Syntax in Lexical Functional Grammar. The Resource Logic Approach*. MIT Press, Cambridge, MA.

Girard, J.-Y. (1987) 'Linear logic', *Theoretical Computer Science* 50, 1–102.

Kaplan, R. M. and J. Bresnan (1982) 'Lexical-Functional Grammar: a formal system for grammatical representation'. Chapter 4 in J. Bresnan (ed.), *The Mental Representation of Grammatical Relations*. Cambridge, MA, 173–281

Kurtonina, N. (1995) *Frames and Labels. A Modal Analysis of Categorial Inference*. Ph. D. Dissertation, OTS Utrecht, ILLC Amsterdam.

Kurtonina, N. and M. Moortgat (1997) 'Structural Control'. In P. Blackburn and M. de Rijke (eds), *Specifying Syntactic Structures*. CSLI, Stanford, 1997, 75–113.

Lambek, J. (1958) 'The Mathematics of Sentence Structure', *American Mathematical Monthly* 65, 154–170.

Lambek, J. (1961) 'On the calculus of syntactic types'. In Jakobson, R. (ed.), *Structure of Language and Its Mathematical Aspects*. Proceedings of the Twelfth Symposium in Applied Mathematics. Providence, Rhode Island.

Moortgat, M. (1996) 'Multimodal linguistic inference', *Journal of Logic, Language and Information* 5(3,4), 349–385. Special issue on proof theory and natural language. Guest editors: D. Gabbay and R. Kempson.

Moortgat, M. (1997) 'Categorial Type Logics'. Chapter 2 in J. van Benthem and A. ter Meulen (eds), *Handbook of Logic and Language*. Elsevier, Amsterdam and MIT Press, Cambridge MA, 1997, 93–177.

Morrill, G. and B. Carpenter (1990) 'Compositionality, implicational logics, and theories of grammar', *Linguistics & Philosophy* 13, 383–392.

Pollard, C. and I. Sag (1994) *Head-Driven Phrase Structure Grammar*. Chicago.

Stabler, E. (1997) 'Derivational minimalism'. In Ch. Retoré (ed.), *Logical Aspects of Computational Linguistics*. Lecture Notes in AI, Springer, 68–95.