Meaningful patterns

Michael Moortgat

Contents

1 Introduction

Natural languages, in their individual ways, exploit patterns of grammatical form to convey the assembly of meaning. A central goal for linguistics is to find out in what respects languages are alike in this, and in what respects they can differ: in other words, to identify *invariants* in the composition of grammatical form and meaning, and to delineate the space for *variation*, allowing for the realization of the same meaning assembly through different structural patterns. Lambek's categorial framework addresses the first of these questions: it provides a vocabulary of type-logical *constants* together with a proof-theoretic modeling of 'grammatical computation'. But the original Lambek framework offers no tools for the analysis of structural variation: the structural properties of the constants are hard-wired. In this contribution, we explore two enhancements of the categorial architecture that bring the interplay between uniformity and variation into focus. First, we move from the one-dimensional world of Lambek calculus to a *multimodal* setting, where different composition operations live together and interact. The various modes of composition share the same base logic, but they can differ in their structural properties. Second, we introduce logical constants to *control* structural reasoning, thus replacing global structural regimes by lexically anchored local fine-tuning.[1]

Johan van Benthem's work has been vital in turning categorial grammar into a flexible tool for the analysis of linguistic form and meaning. With this contribution I would like to acknowledge a tremendous debt to his inspiring work. The material in §2 is based on joint work with Dick Oehrle.[2] I have taken the presentation format from Lakatos' engaging treatment of

[1] See Chapter 2, Section 4 of the *Handbook of Logic and Language* (Van Benthem and ter Meulen (eds), Elsevier Science, 1997), and Kurtonina and Moortgat 'Structural Control' in Blackburn and de Rijke (eds.) *Specifying Syntactic Structures*. CSLI, Stanford, 1997, 75–113.

[2] See our course at the ESSLLI99 summer school.

the 'logic of mathematical discovery'.[3] The methodological point of that book — that the field develops not through a monotonous accumulation of established truths, but through 'the incessant improvement of guesses by speculation and criticism' is very much in tune with Johan's view on logical research[4], and very commendable when it comes to investigating the 'logic of grammar'.

2 Linguistics 210

The discussion is set in a classroom. LINGUISTICS 210 *is an obligatory course in the Adult Education Programme. It is a mixed class. Half of the pupils are generative linguists. They talk a lot about grammar as a 'computational system'. But when asked to actually* compute *something, i.e. use an algorithm, they become evasive. The other half of the class consists of logicians. They have all heard about 'Lambek calculus'. But their exposure to grammatical patterns of any significance has been limited. Not surprisingly, they feel highly reluctant to extend the analytical vocabulary beyond 'slash' and 'backslash'. Fortunately for the teacher, the course comes with a nice software package.*[5]*. While the discussion is going on, both the linguists and the logicians have a chance to actually gain some hands-on experience with grammatical modeling. The students are all actively participating, with the exception of* EPSILON*, who has been complaining he does not see the point of taking the course since 'everybody knows these categorial systems are just clumsy ways of doing context-free grammar'. He has been asked to take some screen shots of the computer lab exercises, and make himself useful that way.*

TEACHER: Last week, we have been looking at the logical structure of linguistic theory. Here is a brief recap. The basic statements of our grammar logic are sequents $\Gamma \vdash A$, asserting that the structure Γ is of type A. Formulas (types) are built from atoms $p_1, p_2, \ldots$ with the binary connectives $/_i, \bullet_i, \backslash_i$ and the unary connectives $\diamondsuit_j, \Box_j$. Structures are built out of formulas with the operations $(\cdot \circ_i \cdot)$ and $\langle\cdot\rangle^j$, structural counterparts of $\bullet_i$ and $\diamondsuit_j$ respectively. The indices i and j are taken from given, finite sets I, J which we refer to as composition *modes*. The composition modes all share the same *base logic*, given by the residuation inferences of Fig 1.

$$\diamondsuit_j A \vdash B \quad \text{iff} \quad A \vdash \Box_j B$$
$$A \vdash C/_i B \quad \text{iff} \quad A \bullet_i B \vdash C \quad \text{iff} \quad B \vdash A \backslash_i C$$

Figure 1: The base logic: residuation

But they can differ in the *structural* rules they allow. Structural rules take the form of axiom schemata $A \vdash B$, where the input A is built out of distinct formula variables $A_1, \ldots, A_n$ purely in terms of $\diamondsuit_j$ and $\bullet_i$, and where the output B is a $\diamondsuit_{j'}/\bullet_{i'}$ formula constructed out of exactly *the same* $A_1, \ldots, A_n$. The postulates are *linear*, in other words: they can rearrange grammatical material, but they cannot duplicate or waste it. Specifying a grammar, in the present setting, means giving a lexicon, i.e. an assignment of type formulas to the words, and specifying a set of structural rules, both relative to a selection of modes from I, J.

Today, to give you a better understanding of the interplay between logical and structural reasoning, I'd like to discuss a puzzle in the grammar of Dutch with you: the positioning of

[3] Imre Lakatos *Proofs and Refutations. The Logic of Mathematical Discovery*, Cambridge, 1976.

[4] See his 'Logical semantics as an empirical science' in *Essays on Logical Semantics*, Reidel, Dordrecht, 1986.

[5] Judging from the pictures we find in § 4, the class is using Richard Moot's theorem prover Grail — no doubt an acronym for GRAMMAR = LOGIC.

the verbal head for different clause types.

HALF OF THE PUPILS: Oh no! Is this really necessary?

THE OTHER HALF: It's about time we get some data to look at!

TEACHER: Alpha, you always have these impressive displays of example sentences in your term papers. Maybe you could give your colleagues a gentle introduction to the problem. Please keep it simple — just stick to the facts and give us some handy descriptive terminology that will facilitate the discussion.

ALPHA [*to his neighbour*]: 'Stick to the facts' — what a perverse attitude! The Teacher does not seem to understand that Progress replaces naive classification by theory-generated classification! But I want to get my grade for this course, so let me do what I am asked . . . [*At the blackboard, addressing the class*] Look at the following groups of examples. In the (*a*) set, we have examples of subordinate clauses. I have underlined the verbal heads — they occur in clause-final position, following the complements. When you read in the literature on language typology that Dutch is an OV language ('Objects-Verb'), this is the clause type one is referring to. Note also that the verbal head can be a simple tensed verb (the first two examples), or a cluster of verbs (the last two). These verbal clusters, arising in the so-called Verb Raising construction, play a crucial role in the argument that not all natural languages are context free.[6]

EPSILON [*sneering*]: Too bad for the Lambekians!

ALPHA [*undisturbed*]: Now in non-subordinate clauses, the (tensed) verb occupies another position: compare (*a*) with (*b*) and (*c*). In the yes/no questions of (*b*), the tensed verb occurs clause-initially; in the declarative main clause (*c*), it takes the 'second position' — the position after the first major constituent, the subject, in a regular main clause.

a	(. . . als) Alice de Soepschildpad plaagt	*subordinate clauses*
	(. . . als) Alice de koningin gek vindt	
	(. . . of) Alice de Soepschildpad zou willen plagen	
	(. . . of) Alice de Soepschildpad probeert te plagen	
b	Plaagt Alice de Soepschildpad?	*polar interrogatives*
	Vindt Alice de koningin gek?	
	Zou Alice de Soepschildpad willen plagen?	
	Probeert Alice de Soepschildpad te plagen?	
c	Alice plaagt de Soepschildpad.	*declarative main clauses*
	Alice vindt de koningin gek.	
	Alice zal de Soepschildpad willen plagen.	
	Alice probeert de Soepschildpad te plagen.	

TEACHER: So we have three types of clauses— they are all assembled out of the same pieces, but the pieces are put together in different structural configurations. The challenge for our session today is to build a fragment that derives the different clausal types from *one* type assignment to the verbs, using structural reasoning. The approach I would like to explore with you can be called a 'key-and-lock' strategy. The strategy rests on two ideas. First, we distinguish the various clausal types by putting a control feature on the goal formula. In the table below, I

[6] Recommended reading for the class on this subject are Chapters 16 (Footloose and context-free) and 17 (Nobody goes around at LSA meetings offering odds) in Geoff Pullum's *The Great Eskimo Vocabulary Hoax*, University of Chicago Press, 1991.

have chosen mnemonic labels for the mode indices (e, i, 2 for end, initial and second position, respectively).

$\Gamma \vdash \Box_e s$	*subordinate clause (verb final)*
$\Gamma \vdash \Box_i s$	*main clause, interrogative (verb initial)*
$\Gamma \vdash \Box_2 s$	*main clause, declarative (verb second)*

Figure 2: Goal formulas for main and subordinate clauses

Second, we assign the verbal heads in these clauses a type with a control box as the main connective — say $\Box_0$ (The verbs, in their lexical type assignment, have no information as to what clause type they will be used in, so we need a fresh mode index here.) The verbs will have subcategorizational requirements, expressed in terms of the left and right implication connectives. But the $\Box_0$ decoration acts as a lock: as long as the $\Box_0$ connective has not been eliminated, the implications cannot be used. Unlocking $\Box_0$ requires a matching $\Diamond_0$ key, to be provided by (the residuals of) the control features on the goal formula. In order to establish the communication between $\Diamond_e, \Diamond_i, \Diamond_2$ and the verbal lock, we will formulate postulates that inspect the different clausal patterns, making sure that the verb indeed occupies the required position.

BETA: 'Keys and locks' — enough vague metaphors! Give us a derivation!

2.1 Verb final subordinate clauses

TEACHER: Before I can do that, we have to decide which position we take to be the canonical one for the basic lexical type assignment to the verbs. In your GIL course[7], you have been reading Jan Koster's classical paper[8] arguing forcibly that the subordinate clause is the canonical pattern: I suggest we take this as our starting position, and have verbs select their complements to the left, accordingly.

ZETA: But, Sir! You cannot be serious...

TEACHER [*interrupting*]: Yes, Zeta, I know: current teaching has it that Koster's arguments do not count anymore, and that the English SVO pattern is universal. But for this to work, you need to introduce a generous supply of abstract syntactic positions[9] for complements to move to. I think we agreed at the start of this course that this is not a legal move for the game we are playing: we do not have any autonomous 'syntax' generating 'positions' — structural patterns arise in the deductive process of deriving a goal formula from the type formulas of the lexical assumptions.

ZETA [*aside*]: No functional projections? They seem to be taking minimalism a bit too seriously here.

TEACHER: Let us see now how we can implement the idea of checking the (clause final) position of the verb explicitly. Our goal here is to derive a sequent $\Gamma \vdash \Box_e s$, in other words, to determine whether the structure Γ is indeed a verb final clause. Somewhere within Γ, we have the head verb, locked by the $\Box_0$ feature. The postulate package in Fig 3 establishes the communication between $\Box_e$ on the goal formula, and the lexical $\Box_0$ decoration. $P1$ allows the messenger $\Diamond_e$ to recursively traverse phrasal structure, looking for the verbal head on the right

[7] 'Great Ideas in Linguistics'

[8] 'Dutch as an SOV language', *Linguistic Analysis* **1** (1975), 111–136. Given the example sentences the class is using, the Dutch version may be more appropriate: 'Het werkwoord als spiegelcentrum', *Spektator* **3** (1974), 601–619.

[9] A conservative estimate for the structure of the functional domain would have the following spine of functional projections: CP-C'-AgrSP-TP-T-AgrOP(for IO)-AgrO'-AgrOP(for DO)-AgrO'-VP.

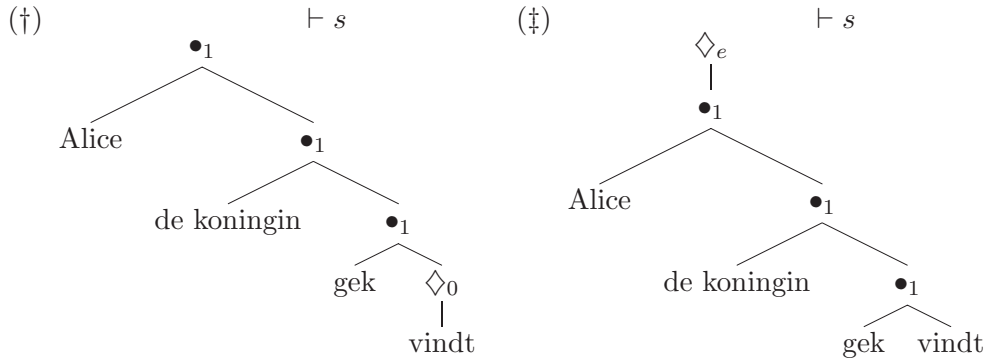
branch. When the recursion bottoms out, the control switches from $\Diamond_e$ to $\Diamond_0$ by means of $P2$. You can find a derivation in Fig 9.

$$\Diamond_e(A \bullet_1 B) \vdash A \bullet_1 \Diamond_e B \quad P1$$
$$\Diamond_e A \vdash \Diamond_0 A \quad P2$$

Figure 3: Verb final: subordinate clauses

BETA: I think I understand what you meant with the 'key-and-lock' strategy. Provided that the messenger $\Diamond_e$ indeed finds the head verb in clause final position, application of $P2$ gives rise to a reduction $\Diamond_0 \Box_0 A \vdash A$, which then unleashes the subcategorizational requirements of the verb.

TEACHER: Right. The way you formulate it, I see you're really visualizing the derivation process in terms of a top-down, backward chaining Gentzen computation. But you could also take the bottom-up Natural Deduction perspective of Fig 9. Here you start the computation from the lexical assumptions — the leaves of the derivation tree. The part of the proof above the line (†)



shows how far you can get with just the *logical* rules of inference: you eliminate the constants in the lexical type assignments. The sequence of *structural* inferences that follows then establishes the communication with (‡), the structure that yields the goal formula $\Box_e s$ after one further logical step of $[\Box I]$.

GAMMA: Before you all get carried away with this 'key-and-lock' game, could I bring it to your attention that the subordinate clause order is already derivable from the simple lexical assignment $\mathsf{vindt} \vdash ap\backslash(np\backslash(np\backslash s))$, without any $\Box$ decoration or pushing around of $\Diamond$ marks: just Functional Application!

TEACHER: Looking just at the examples in this section (where the clause is projected from a simple tensed head verb), this is true. We decided to take the subordinate clause order as canonical, and the type assignment, with verbal complements under a left implication, reflects that. I wonder whether you will maintain your sceptical attitude *vis à vis* $\Diamond, \Box$ when we broaden the discussion to include subordinate clauses headed by verb *clusters* rather than simple verbal heads. Alpha, why don't you...

2.2 Head adjunction: verb clusters

ALPHA: I take it you want me to give our logical friends here a little briefing as to what the relevant data are? [*Sarcastically*] In plain observational terms, of course! To keep things manageable, I suggest we restrict our attention to modal auxiliaries (wil, zal, moet ...) and

perception verbs (hoort, ziet, ...) for a start. They subcategorize for a bare infinitival complement. Now consider the following examples.[10]

dat Alice de Soepschildpad wil plagen
... wants to tease the Mock Turtle
dat Alice de Soepschildpad wil kunnen plagen
... wants to be able to tease the Mock Turtle
dat Alice de Soepschildpad een droevig lied hoort zingen
... hears the Mock Turtle sing a sad song

There are a number of things one can 'observe' about this composition pattern:

- Rather than combining with the complete infinitival complement, the modal axiliaries combine with the verbal *head* of the complement, passing over whatever complements this infinitival head itself may have. The resulting combination of verbal heads is what one usually refers to as the 'verb cluster' — I have underlined it in the examples above.
- The order of the verbs in a cluster gives rise to 'crossing dependencies', as you can see when you connect wil with its subject Alice, and plagen with its direct object de Soepschildpad.
- The formation of the verb cluster, with modal auxiliaries such as wil, ..., is obligatory. Compare the ungrammatical example below with its English counterpart.

  *dat Alice wil de Soepschilpad plagen
  that Alice wants to tease the Mock Turtle

TEACHER: Maybe we could challenge Gamma a little here, and see how far we can get without structural reasoning or control — just by solving plain Lambek type equations. Let's run the Buszkowski-Penn typing algorithm[11] on a sequence of verb phrases of growing complexity. In the input structures, the head of a configuration is the component pointed at by $\triangleleft/\triangleright$. You find the results in Fig 10. Any comments on the solutions for the *aux* element?

wil $\triangleleft$ slapen $\vdash vp$
de Soepschildpad $\triangleright$ (wil $\triangleleft$ plagen) $\vdash vp$
Alice $\triangleright$ (een taart $\triangleright$ (wil $\triangleleft$ overhandigen))) $\vdash vp$

BETA: From a semantic point of view, one could say that the first verb-phrase displayed is optimal, in the sense that structural and semantic composition coincide. But in the input data that follow, there is a discrepancy between semantic composition (requiring the assembly of the infinitive and its complements, to produce the saturated infinitival phrase which *aux* selects for) and structural composition (requiring the adjunction of *aux* to the head of its infinitival complement). Our plain Lambek typing algorithm cannot resolve this discrepancy — the types assigned to *aux* do not unify, so one ends up with irreducible lexical polymorphism, it would seem.

ALPHA: Let me add that cluster formation is a recursive process. As the example with horen already suggests, there is in principle no bound on the number of complements a cluster may require: the perception verbs *increment* the arity of the cluster. Lexical ambiguity, in

[10] The discussion of verb clusters usually involves kraanvogels and nijlpaarden. We stick to the Lewis Carroll cast.

[11] Buszkowski and Penn, 'Categorial grammars determined from linguistic data by unification', *Studia Logica*, **49**, 431–454.

other words, would mean an infinite lexicon, in the limiting case. Now I thought you were so keen on characterizing a grammar as a (finite!) lexicon, closed under a number of generating functions — the rules of inference, in the case at hand.

GAMMA: OK — so I'd need some inductive type assignment schema for the triggers of verb cluster formation, in the sense of

$$(\star) \qquad \text{if } aux \vdash A/B \text{ then also } aux \vdash (C\backslash A)/(C\backslash B)$$

with $aux \vdash vp/inf$ as the boundary case. I see that a naive lexical look-up procedure leads to non-termination with such a schema. But I'm sure I could come up with a more sophisticated call-by-need technique that would keep the parsing algorithm decidable.[12]

ETA: I'm sorry to interrupt you. You are talking about ways of overcoming the expressive limitations of plain Lambek calculus in the light of verb cluster formation. But I'm afraid you'll have to deal with a much more ugly problem first. I ran our theorem prover in the 'Generate' mode with the type assignment we got from Fig 10. As things stand, there is nothing that blocks the derivation of the ungrammatical 'English' order

*dat Alice wil de Soepschilpad plagen
that Alice wants to tease the Mock Turtle
dat Alice wil slapen
that Alice wants to sleep

given the type assignment wil $\vdash vp/inf$. Certainly that is a type assignment we'll need for the combination with a simple infinitival complement!

TEACHER: Let me summarize this discussion. On the one hand, the plain Lambek grammar overgenerates, as Eta was careful to point out: blocking the combination of wil with an infinitive that is itself the result of a number of composition steps, as in the starred example above, requires *control*. On the other hand, with just logical rules of inference, our Lambek grammar is too weak to establish the proper form-meaning correspondence for the verb clusters: we'd like to license *structural reasoning*, in order to resolve the discrepancy between the composition of grammatical form and meaning assembly.

To start with the second point: Gamma just proposed an inductive type assignment schema $(\star)$ to obtain the infinite set of possible types for an *aux* element from a 'basic' assignment. Such a lexical type assignment schema, as it stands, is external to the grammar logic proper. What we are looking for here is the right mixture of logical and structural reasoning that would enhance our proof-theoretic engine in such a way that the type transitions licensed by $(\star)$ can be computed on-line. I am sure Beta has ideas about this.

BETA: Well, if I take the combined top-down/bottom-up approach we discussed a moment ago, the situation is as follows. The top line of the derivation below is obtained in terms of bottom-up Modus Ponens steps. Conditional reasoning, in backward chaining fashion, allows me to transform the complex goal formula into a structure term.

$$\dfrac{\dfrac{\dfrac{\dfrac{\vdots}{vp/inf \bullet (obj \bullet obj\backslash inf) \vdash vp}\ \backslash,/E}{obj \bullet (vp/inf \bullet obj\backslash inf) \vdash vp}\ \text{structural pattern matching?}}{vp/inf \bullet obj\backslash inf \vdash obj\backslash vp}\ \backslash I}{vp/inf \vdash (obj\backslash vp)/(obj\backslash inf)}\ /I$$

[12] Maybe Gamma is thinking of something like the techniques for delayed evaluation worked out in Bouma and Van Noord's 'Constraint-based categorial grammar'.

At the point where the logical inferences halt, I can now solve a *structural* equation (just as we were solving logical type equations before). The pattern matching that establishes the communication between the two terms involved, yields the structural postulate:

$$(\dagger) \quad A \bullet (B \bullet C) \vdash B \bullet (A \bullet C)$$

TEACHER: Now for some structural fine-tuning! You realize that in the one-dimensional Lambek world, with just a single composition operation $\bullet$, your postulate causes fatal damage to order-sensitivity. But in the multimodal setting, we can refine the signature, and formulate (†) as an *interaction* principle, relating two distinct composition modes: our familiar $\bullet_1$ for regular phrasal complementation, and a mode $\bullet_0$ which we'll use to type the triggers of verb cluster formation.

$$(\ddagger) \quad A \bullet_1 (B \bullet_0 C) \vdash B \bullet_0 (A \bullet_1 C)$$

ETA: I can see how this interaction postulate allows you to derive the cluster wil plagen below, from lexical assumptions wil $\vdash vp/_0 inf$ and plagen $\vdash np\backslash_1 inf$. And I also see how it avoids the unwanted scrambling effects of one-dimensional (†). But what about the problem I just brought up? With (‡), your grammar still accepts the ungrammatical 'English' order — as a matter of fact, the structural configuration of the starred example below serves as the *premise* for the (‡) step. Surely, there is nothing in the logic that forces you to make the structural move!

als Alice de Soepschildpad wil plagen
*als Alice wil de Soepschildpad plagen

TEACHER: I think we have reached the point where you'll come to appreciate the control devices $\Diamond, \Box$. With the help of these connectives we can achieve exactly what Eta is alluding to: they provide logical means of forcing the composition of the verb cluster, rather than just allowing it to come into being. Let us go back to our earlier decision to 'lock' verbal heads with a $\Box_0$ feature, and consider the structural configuration of the end sequent we want to derive:

$$\text{Alice } ((\text{de soepschildpad}) \ (\text{wil} \circ_0 \text{plagen})) \vdash \Box_e s$$

We have seen already how $P1$ and $P2$ localize the verbal head in its clause final position. But now, instead of finding a simple verb there, the $\Diamond_0$ messenger is confronted with a verbal cluster. By strongly distributing $\Diamond_0$ over $\bullet_0$, Postulate $P3$ checks whether indeed the components of such a cluster are both verbal heads, and whether they are put together in the $\bullet_0$ mode. And we can improve on (‡) too, by lexically anchoring this structural inference, so that it has to be explicitly licensed by the 'verbal head' feature $\Diamond_0$. This yields $P4$ as a controlled form of (‡). The postulate will be called now from a lexical type assignment wil $\vdash \Box_0(vp/_0 inf)$. Have a look at the derivation in Fig 11.

$$\begin{array}{rl} \Diamond_0(A \bullet_0 B) \vdash \Diamond_0 A \bullet_0 \Diamond_0 B & P3 \\ A \bullet_1 (\Diamond_0 B \bullet_0 C) \vdash \Diamond_0 B \bullet_0 (A \bullet_1 C) & P4 \end{array}$$

Figure 4: Verb cluster formation: clause union

ETA: So your claim is that these postulates actually force the formation of the verb cluster, making the ungrammatical 'English' order underivable, as we wanted? Let me think . . . [*Silence*] I see it! We have two verbal heads in this clause — two $\Box_0$ locks, in other words. The goal formula $\Box_e s$ provides only one key. The only way to duplicate the $\Diamond_0$ key is by means of $P3$.

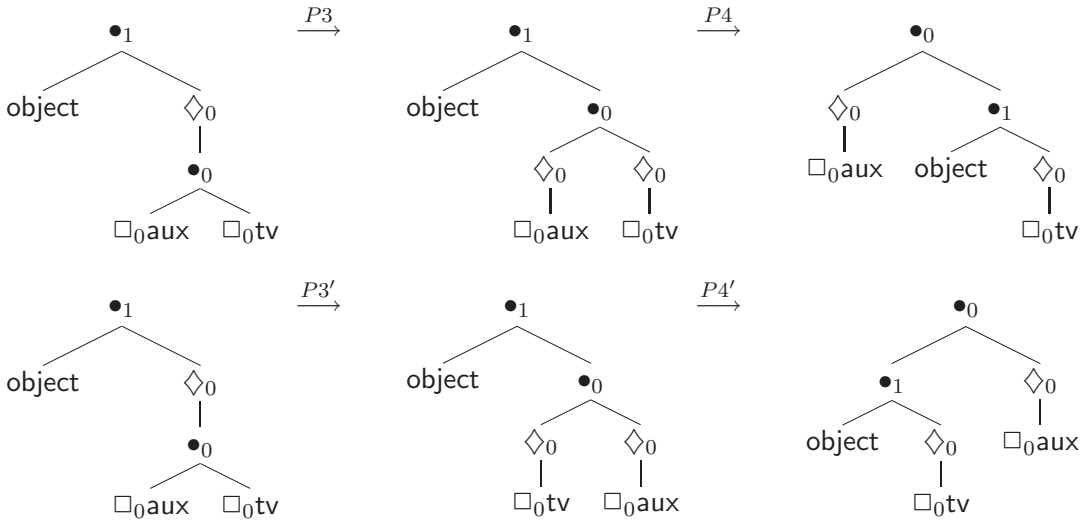
But that implies that $P2$ must already have switched the control from $\Diamond_e$ to $\Diamond_0$. The $P2$ move is irreversible, but once we have $\Diamond_0$, the control feature does not distribute through phrasal $\bullet_1$ structure any more. It's surprising, but it seems this utterly simple package actually gives us a grammar of clause union!

TEACHER [*aside*]: I am not sure everybody here shares Eta's enthousiasm about the simplicity. [*To the class again*] Well, utter simplicity... This was just a first stab. Maybe someone else has a more simple solution to our little structural puzzle?

DELTA: I am not sure my solution is more simple, but I think it is certainly not inferior to what you just presented. Your lexical assignment has the the key verbs (willen, horen, etc) select their complement to the right. Instead, in my fragment they select their complement infinitive to the left: schematically, my lexicon has $\Box_0(B\backslash_0 A)$ where you have $\Box_0(A/_0 B)$. The structural package, for my solution, replaces $P3$, $P4$ by the postulates below.

$$\begin{array}{ll} P3': & \Diamond_0(A \bullet_0 B) \vdash \Diamond_0 B \bullet_0 \Diamond_0 A \\ P4': & A \bullet_1 (B \bullet_0 \Diamond_0 C) \vdash (A \bullet_1 B) \bullet_0 \Diamond_0 C \end{array}$$

$P3'$ recursively 'unwinds' the crossing dependencies of the verbal cluster. The mixed associativity law $P4'$ changes the dominance relation between $\bullet_1$ and $\bullet_0$, giving the latter the appropriate scope for meaning assembly. Compare the derivation in Fig 14 with that of Fig 11. Below, you see the different structural rewritings in schematic form:



BETA: I don't understand how you arrive at the lexical type assignments $\Box_0(B\backslash_0 A)$. Structurally, the lexical items in question appear as *prefixes* to the infinitives they combine with — how would the Buszkowski-Penn typing algorithm ever come up with a *left* implication?

DELTA: I see what you mean. But I can streamline my lexicon, expressing the pleasant generalization that Dutch verbs select their complements uniformly to the left. Our discussion here has been focusing heavily on squeezing out structural generalizations at the level of the 'computational system' — translating Gamma's lexical type assignment schema ($\star$) into online structural reasoning, for example. But in the end, when it comes to measuring overall simplicity of a grammar, I think we should attach equal weight to generalizations at the level of the lexicon — constraints on the shape of lexical type assignments, for example, that restrict the space for such assignments to a proper subsystem of the full formula language.

ALPHA: That sounds like a valid methodological advice. But even at a more down-to-earth level, if you broaden the range of data under consideration a little, you will see that Delta has a point. Take the pronominal forms of infinitival complements: they are selected to the left.

omdat Alice de Soepschildpad wil plagen
omdat Alice dat (=de Soepschildpad plagen) wil

With the $P3/P4$ approach, not only would you have non-uniformity in the directionality of complement selection (leftward for some verbs, to the right for others), but the $\Box_0(A/_0B)$ type verbs that give rise to cluster formation need an extra assignment for pronominal realizations of their complements.

TEACHER: You seem to be preparing us for a choice between Delta's $P3'/P4'$ solution (and the lexical assignments that go with it) and the $P3/P4$ alternative, Delta coming out as the winner. But we could also take a more liberal attitude. I agree that the $P3/P4$ lexicon is slightly suboptimal as compared to Delta's. But this is an innocent form of lexical ambiguity — altogether different from Gamma's ($\star$) schema, where missing the structural generalization inflates the lexicon up to infinity. I'd say the two solutions are roughly of the same complexity. Now, suppose we say the goal of our game (formulating a fragment, in your case, or learning the language, for LAD) is to compute the appropriate form-meaning correspondences for Dutch. Then the two solutions can both count as successful. The grammars in Delta's head and in mine, in other words, could have different solutions for the logical type equations and the structural equations for the language in question — as long as these solutions associate the same forms with the same meanings, they would count as equally good.

But we've been talking about the subordinate clause long enough now. I think we should return to the main theme, and see how we can relate verb positioning for the different clausal types.

GAMMA: We are all ears.

2.3 Verb initial clauses: yes/no questions

TEACHER: Let us start with the *verb initial* main clause type $\Box_i s$. We saw that polar interrogatives (yes/no questions) exemplify $\Box_i s$ in its simplest form. Yes/no question have an intonational contour (marked by '?' in writing) that distinguishes them from declarative clauses. We could assign '?' the type $\Box_i s\backslash_1 q$, with an appropriate denotation type for question semantics for the basic type q. But today, we'll focus on the syntax of the $\Box_i s$ part.

Plaagt Alice de Soepschildpad?
Vindt Alice de Koningin gek?
Probeert Alice de Soepschildpad te kunnen vergeten?

The challenge in dealing with $\Box_i s$ is to establish a derivational relationship between verbal types in their canonical clause final position and in the fronted, clause initial position required for $\Box_i s$. I suggest we approach this problem in stages, as we did in discussing the subordinate clause. Let us look first at the fronting of simple tensed verbs, and then turn to verb fronting out of verb clusters. The control strategy can be essentially the same as for the treatment of subordinate clauses: we have to establish communication between $\Box_i$ on the goal formula, and the $\Box_0$ lock on the verbal head of the clause — in doing so, the diamond messenger has to make sure that the clause is indeed verb initial. Who wants to try?

KAPPA: Well, there IS a difference with the 'passive' inspection of structure of $P1/P2$: in the case of $\Box_i s$, the head verb cannot be actually used in its initial position: it has to be manoeuvred into the canonical clause final position. My solution is in Fig 5.

As you see, I haven't been able to manage just with $\Diamond_i$ and $\Diamond_0$: the signature of my fragment introduces an in-between unary mode j. Postulate $P5$ works on the assumption that the clause indeed starts with the verb, which it moves back to clause final position, marking it

$$\Diamond_i(A \bullet_1 B) \vdash B \bullet_1 \Diamond_j A \quad P5$$
$$(A \bullet_1 B) \bullet_1 \Diamond_j C \vdash A \bullet_1 (B \bullet_1 \Diamond_j C) \quad P6$$
$$\Diamond_j A \vdash \Diamond_0 A \quad P7$$

Figure 5: Verb initial. (to be cont'd)

with a $\Diamond_j$ control feature. Notice that $P5$ is a once-only, non-recursive structural inference. $P6$ then 'lowers' the verb into the verb phrase, to the appropriate level of embedding where, with the help of $P7$, it can be unlocked in its canonical position. You can view a sample derivation in Fig 15. The structural inferences, again, mediate between the canonical part of the proof (logical inferences only) and the structure required by the $\Box_i s$ goal formula.

TEACHER: Good. But what about the interrogatives below, where the fronted verb is part of a verb cluster in the subordinate counterpart?

Wil Alice de Soepschildpad plagen?
vs als Alice de Soepschildpad wil plagen
Probeert Alice de Soepschildpad te plagen?
vs als Alice de Soepschildpad probeert te plagen

BETA: Let me try to narrow down the problem via the combined top-down/bottom up strategy we used before. Reasoning backwards from the conclusion, postulates $P5, P6$ bring us as far as the bottom line of the derivation below. We have to formulate a structural inference that establishes the communication between the bottom line and the part of the derivation which, with the help of $P7$, feeds into a structure from which the canonical clause-final order of the verb cluster can be computed.

$$\dfrac{\dfrac{\text{(cf Fig 11)}}{\text{Alice} \circ_1 (\text{de Soepschildpad} \circ_1 \langle \text{wil} \circ_0 \text{plagen} \rangle^0) \vdash s}}{\dfrac{\text{Alice} \circ_1 (\text{de Soepschildpad} \circ_1 \langle \text{wil} \circ_0 \text{plagen} \rangle^j) \vdash s}{\text{Alice} \circ_1 (\text{de Soepschildpad} \circ_1 (\text{plagen} \circ_1 \langle \text{wil} \rangle^j)) \vdash s}\ P7}\ \text{structural inference?}$$

GAMMA: The straightforward answer is $P7'$: it introduces $\bullet_0$, the mode for verb cluster formation, and it feeds into $P7$, which in turn communicates with the strong distributivity postulate $P3$.

$$P7' \quad A \bullet_1 \Diamond_j B \vdash \Diamond_j (B \bullet_0 A)$$

If you agree that the rewriting sequence $P7' \rightsquigarrow P7 \rightsquigarrow P3$ will always occur *en bloc*, we can speed up the derivation by using a well-known program transformation technique: we can 'telescope' this sequence into a one-step structural inference, $P7''$. A derivation which the compiled structural inference is displayed in Fig 16.

$$A \bullet_1 \Diamond_j B \vdash \Diamond_0 A \bullet_0 \Diamond_0 B \quad P7''$$

Figure 6: Fronting out of verb cluster. (Add to Fig 5)

ETA: I hate to interrupt you again. But running our theorem prover in 'Generate' mode, I see that nothing blocks fronting of the complete verb cluster, rather than just the finite head:

*Wil plagen Alice de Soepschildpad?
vs Wil Alice de Soepschildpad plagen?

TEACHER: Now this problem is very similar to what we saw before, when we had to block the derivation of the ungrammatical 'English' order in subordinate clauses. I think adjusting the package $P5 - P8$ in the required way will make a very nice question for the written exam!

2.4 Verb second: declarative main clauses

TEACHER: That brings us to the final major clausal type on our agenda: the declarative main clause, with goal type $\Box_2 s$. The tensed head verb of a declarative sentence occupies second position, following the subject.[13] Compare the examples below with what we have seen before.

Alice plaagt de Soepschildpad.
Alice vindt de koningin gek.
Alice zal de Soepschildpad willen plagen.
Alice probeert de Soepschildpad te plagen.

KAPPA: After the work I've done for $\Box_i$, the structural control package for verb second is really very simple: in the nonrecursive postulate $P8$, the control operator $\diamondsuit_2$ skips the first constituent of the clause it is in construction with, transmitting $\diamondsuit_i$ (verb initial) control to the remainder of the clause. See Fig 17 for an example.

$$\diamondsuit_2(A \bullet_1 B) \vdash A \bullet_1 \diamondsuit_i B \quad P8$$

Figure 7: Verb second: main clauses

2.5 Extraction: take-home assignment

TEACHER: At this point, I would have liked to investigate with you how verb positioning interacts with the formation of relative clauses, and with subordinate and main clause questions. But I see it is time already for your next class. As a take-home assignment, I propose you extend the fragments we have been discussing with the following clause types.

a	de Soepschildpad die Alice probeert te plagen	(*relative clause*)
b	(Alice weet) wie de koningin gek vindt	(*subordinate question*)
c	Wie wil de taarten stelen?	(*main clause question*)

Your task is twofold: provide type assignments for the new lexical items (underlined in the examples above): relative pronouns like die, subordinate or main clause question pronouns like wie, and specify the new structural inferences (if any) keyed to these type assignments. Here are some facts your type assignments will have to take into account, and some hints. (I am calling the material following these pronouns the BODY — of the relative clause, subordinate or main clause question, respectively).

[13] What we say here does not apply to 'topicalized' sentences, such as 'De SOEPSCHILDPAD zal Alice willen plagen', which are distinguished from declaratives by their prosodic realisation, and which would thus receive their own clausal type.

- With respect to verb positioning, the body of relative clauses and subordinate questions is of the $\Box_e s$ type: the verb (cluster) occurs in clause-final position; the main clause question body is verb initial — type $\Box_i s$.

- In (a), (b) and (c) above, the body is not a *complete* clause (of type $\Box_e s$ or $\Box_i s$), but a clause from which a noun phrase is missing. Use conditional reasoning (with respect to a *np* resource) to make sure the body constituents are not overcomplete.

- The pronouns are not specific as to *what* noun phrase hypothesis in the body they are withdrawing when reasoning: (a) and (b), for example, are ambiguous between a subject and an object hypothesis. Characterize the 'accessible' positions for hypothetical reasoning in terms of controlled structural inferences.

3 Conclusion

The slogans 'Parsing = Deduction, Grammar = Proof Theory'[14] rest on the assumption that mathematical logic offers appropriate tools for modeling the cognitive abilities underlying knowledge and use of language, and language acquisition. Can logic indeed provide an insightful *explanation* of linguistic cognition, or is its role limited to the level of 'mere descriptive adequacy'? The answer to this question depends very much on the kind of logic one uses in grammatical analysis. When in the middle of the 17th century, Newton and Leibniz invented the calculus, they introduced a 'new' mathematical language — a language with the right expressive power for the analysis of motion and change, dynamic processes that had come to occupy a central position on the agenda of physics. The framework of substructural logic, in a similar spirit, turns out to provide appropriate instruments for reasoning about grammatical resources, and the dynamics of the assembly of grammatical form and meaning. Cross-linguistic invariants in the form-meaning correspondence are captured in terms of the proof-theoretic interpretation of the grammatical constants. The dimensions for structural variation are charted by complementing the logical core of the computational system with controlled structural reasoning.

[14] See Van Benthem *Language in Action: Categories, Lambdas and Dynamic Logic* North-Holland, Amsterdam, (Studies in Logic, vol. 130). Paperback reprint with new Appendix, The MIT Press, 1995.

4 Screen shots from the computer lab

$\Diamond_0$	*verbal head*
$\Diamond_e$	*control: verb final*
$\Diamond_i, \Diamond_j$	*control: verb initial*
$\Diamond_2$	*control: verb second*
$\bullet_1$	*phrasal composition*
$\bullet_0$	*head adjunction (clause union)*

Figure 8: Signature: mode distinctions for major clausal types.

$$\dfrac{\text{of} \vdash x/\Box_e s \qquad \dfrac{\text{alice} \vdash np \qquad \dfrac{\dfrac{\text{de} \vdash np/n \quad \text{koningin} \vdash n}{\text{de koningin} \vdash np}\,[/E] \qquad \dfrac{\text{gek} \vdash ap \qquad \dfrac{\text{vindt} \vdash \Box_0(ap\backslash(np\backslash(np\backslash s)))}{\langle \text{vindt} \rangle^0 \vdash ap\backslash(np\backslash(np\backslash s))}\,[\Box E]}{\text{gek } \langle \text{vindt} \rangle^0 \vdash np\backslash(np\backslash s)}\,[\backslash E]}{(\text{de koningin})\ (\text{gek } \langle \text{vindt} \rangle^0) \vdash np\backslash s}\,[\backslash E]}{\text{alice } ((\text{de koningin})\ (\text{gek } \langle \text{vindt} \rangle^0)) \vdash s}\,[P2]}{\text{alice } ((\text{de koningin})\ (\text{gek } \langle \text{vindt} \rangle^e)) \vdash s}\,[P1]}{\text{alice } ((\text{de koningin})\ \langle \text{gek vindt} \rangle^e) \vdash s}\,[P1]}{\text{alice } \langle (\text{de koningin})\ (\text{gek vindt}) \rangle^e \vdash s}\,[P1]}{\langle \text{alice } ((\text{de koningin})\ (\text{gek vindt})) \rangle^e \vdash s}\,[\Box I]}{\text{alice } ((\text{de koningin})\ (\text{gek vindt})) \vdash \Box_e s}\,[/E]}{\text{of } (\text{alice } ((\text{de koningin})\ (\text{gek vindt}))) \vdash x}$$

Figure 9: Example: verb final. Here and below, for legibility the subscripts for the 'default' phrasal composition mode $\bullet_1$ in the type formulas are dropped. In the antecedent structure term, we omit $\circ_1$ altogether, and use minimal bracketing to indicate grouping. The scope of the structural $\langle \cdot \rangle$ operator is indicated by means of color coding.

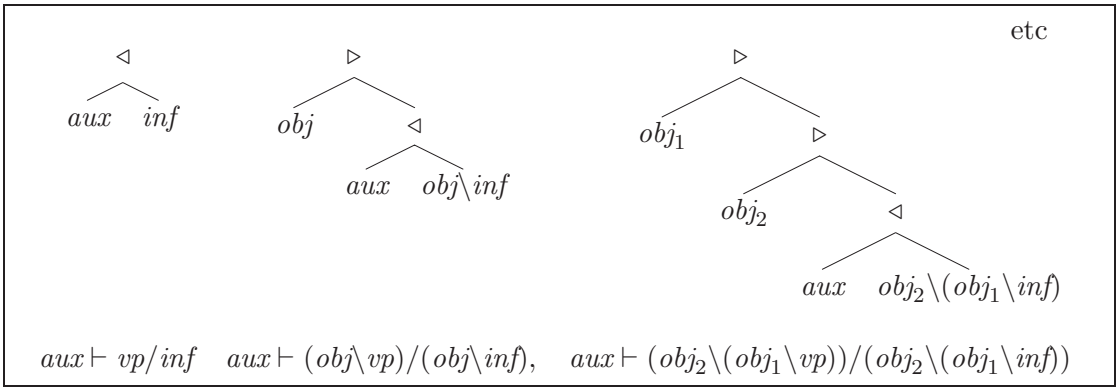


Figure 10: Solving type equations

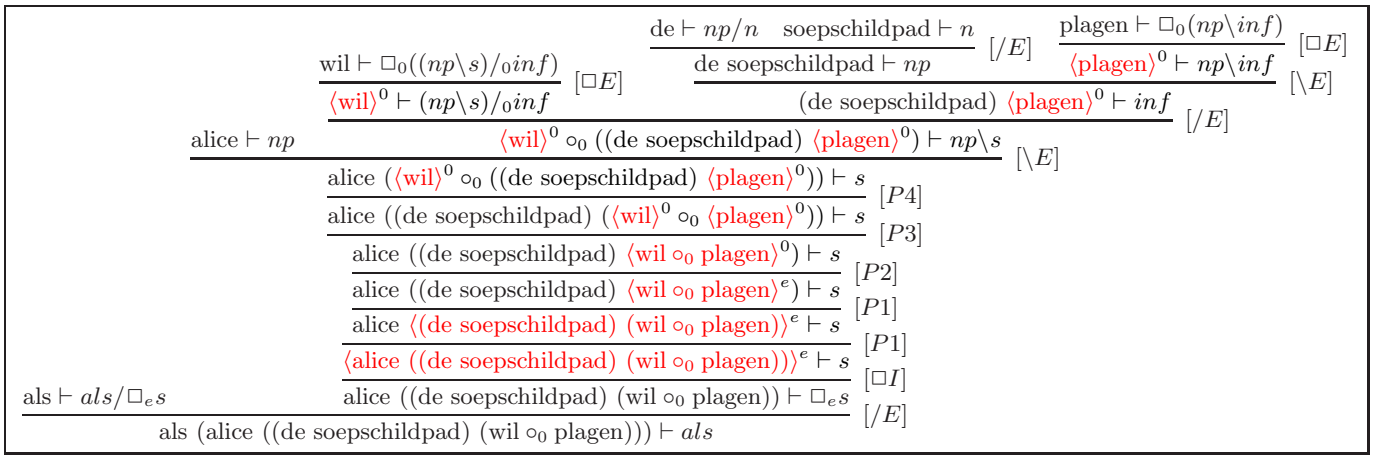


Figure 11: Example: verb cluster

1.	$np - y_0$	*Lex*
2.	$np/_1 n - z_0$	*Lex*
3.	$n - x_1$	*Lex*
4.	$np - z_0(x_1)$	$/E$ $(2,3)$
5.	$\Box_0(np\backslash_1 inf) - z_1$	*Lex*
6.	$np\backslash_1 inf - {}^{\vee} z_1$	$\Box E$ (5)
7.	$inf - ({}^{\vee} z_1\ z_0(x_1))$	$\backslash E$ $(4,6)$
8.	$\Box_0((np\backslash_1 s)/_0 inf) - y_1$	*Lex*
9.	$(np\backslash_1 s)/_0 inf - {}^{\vee} y_1$	$\Box E$ (8)
10.	$np\backslash_1 s - ({}^{\vee} y_1\ ({}^{\vee} z_1\ z_0(x_1)))$	$\backslash E$ $(7,9)$
11.	$s - (({}^{\vee} y_1\ ({}^{\vee} z_1\ z_0(x_1)))\ y_0)$	$\backslash E$ $(1,10)$
12.	$\Box_e s - {}^{\wedge}(({}^{\vee} y_1\ ({}^{\vee} z_1\ z_0(x_1)))\ y_0)$	$\Box I$ (11)

Figure 12: Meaning assembly. Proof term for the $\Box_e s$ subderivation. Term constructors and destructors for the logical Introduction and Elimination steps. The structural inferences leave the proof term unchanged.

1.	$np - \mathbf{alice}$	*Lex*
2.	$np/_1 n - \lambda x_2.(\iota y_2 x_2(y_2))$	*Lex*
3.	$n - \lambda z_2.\mathbf{mock}(\mathbf{turtle}(z_2))$	*Lex*
4.	$np - \iota y_2 \mathbf{mock}(\mathbf{turtle}(y_2))$	$/E\ (2,3)$
5.	$\Box_0(np\backslash_1 inf) - {}^\wedge\mathbf{tease}$	*Lex*
6.	$np\backslash_1 inf - \mathbf{tease}$	$\Box E\ (5)$
7.	$inf - \mathbf{tease}(\iota y_2 \mathbf{mock}(\mathbf{turtle}(y_2)))$	$\backslash E\ (4,6)$
8.	$\Box_0((np\backslash_1 s)/_0 inf) - {}^\wedge(\lambda x_3.(\lambda y_3.{}^\vee\mathbf{want}(y_3, x_3)))$	*Lex*
9.	$(np\backslash_1 s)/_0 inf - \lambda x_3.(\lambda y_3.{}^\vee\mathbf{want}(y_3, x_3))$	$\Box E\ (8)$
10.	$np\backslash_1 s - \lambda y_3.{}^\vee\mathbf{want}(y_3, \mathbf{tease}(\iota y_2 \mathbf{mock}(\mathbf{turtle}(y_2))))$	$\backslash E\ (7,9)$
11.	$s - {}^\vee\mathbf{want}(\mathbf{alice}, \mathbf{tease}(\iota y_2 \mathbf{mock}(\mathbf{turtle}(y_2))))$	$\backslash E\ (1,10)$
12.	$\Box_e s - \mathbf{want}(\mathbf{alice}, \mathbf{tease}(\iota y_2 \mathbf{mock}(\mathbf{turtle}(y_2))))$	$\Box I\ (11)$

Figure 13: Meaning assembly. Substitution of lexical semantics in the proof-term, with on-the-fly conversion. The lexical meaning programs for plagen (line 5) and wil (line 8) have a leading $^\wedge$ to 'erase' the Elimination trace of the $\Box_0$ control operators. The program for wil, moreover, has a leading $^\vee$ on the body of the term which cancels the Introduction trace of $\Box_e$.

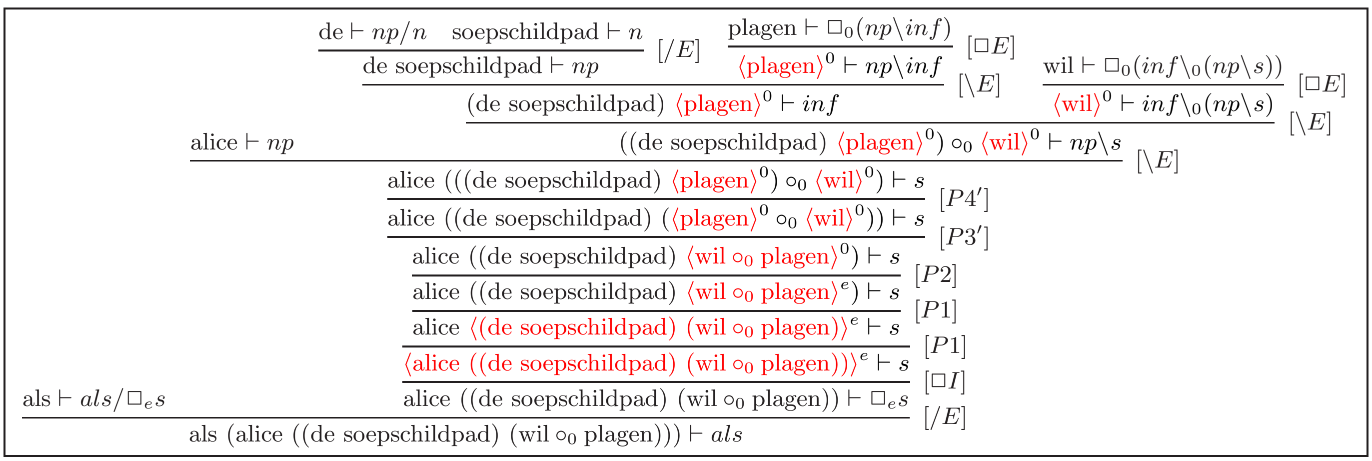


Figure 14: Example: verb cluster, Delta's solution

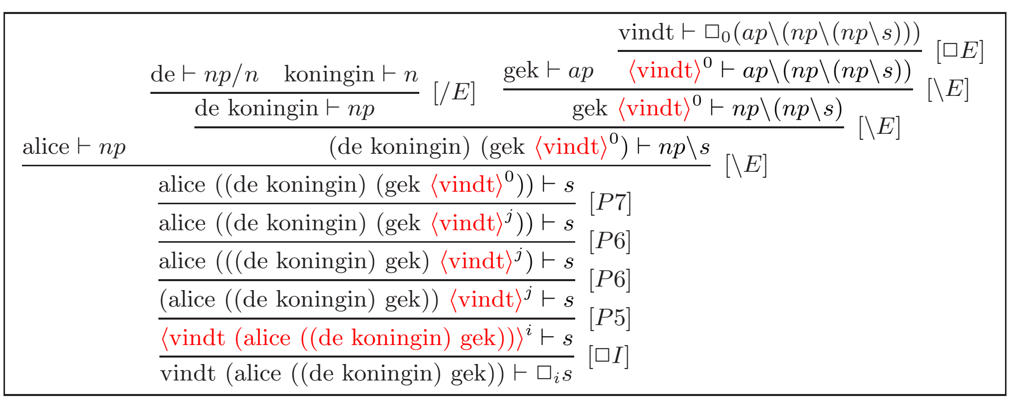


Figure 15: Verb fronting for polar interrogatives. Kappa's solution.

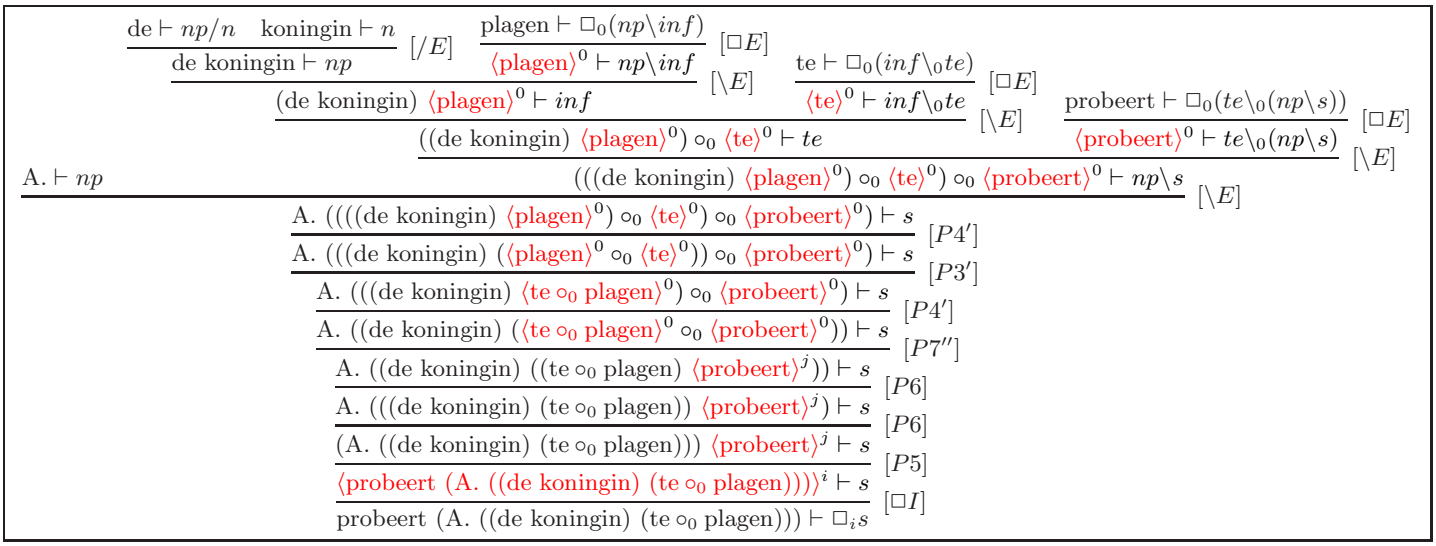


Figure 16: Fronting out of verb clusters. Gamma's $P7''$ as compilation-by-partial-execution.

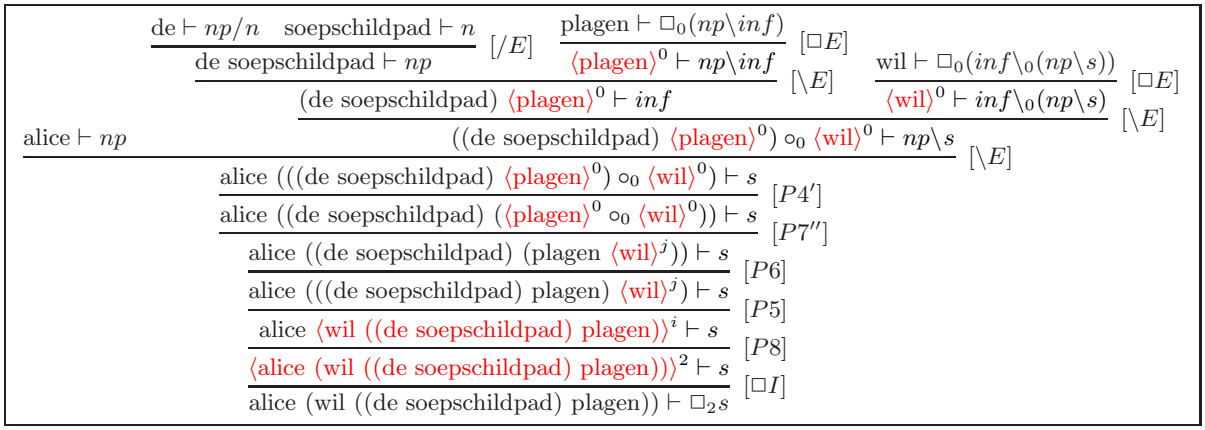


Figure 17: Verb second: main clauses