

Grammar Logicised

Relativisation

Glyn Morrill

Received: date / Accepted: date

Abstract Many variants of categorial grammar assume an underlying logic which is associative and linear. In relation to left extraction, the former property is challenged by island domains, which involve nonassociativity, and the latter property is challenged by parasitic gaps, which involve nonlinearity. We present a version of type logical grammar including ‘structural inhibition’ for nonassociativity and ‘structural facilitation’ for nonlinearity and we give an account of relativisation including islands and parasitic gaps and their interaction.

Keywords islands · parasitic gaps · type logical categorial grammar · relativisation · structural facilitation · structural inhibition

1 Introduction

Today mainstream linguistics is largely informal and computational linguistics is largely statistical. The task of spelling out completely explicit fragments seems either forgotten or taken too lightly. But there is a tradition of logical categorial grammar dating back to Ajdukiewicz (1935[2]) which aims to practice linguistics as a branch of mathematical logic. For a while (especially around 1985–2000) this aspiration blended promisingly with the method of fragments, a methodology promoting the articulation of formal grammar fragments, such as the Montague fragment, and their combination and integration. But before major results were achieved in such comprehensive explicit integrated analysis, however, the field was overtaken by the aforementioned informal and statistical trends. Nevertheless, a small and committed community has remained. Now we are in a position to present a comprehensive explicit integrated analysis of relativisation as a show case example of deep and wide-coverage logical categorial grammar.¹

‘Categorial grammar’ refers to a family of approaches to syntax and semantics in which grammatical information is lexicalised and expressions are classified by recursively defined

¹ In relation to the question whether this contribution is linguistic theory or language engineering, we think that the approach stands as the former without precluding the latter. For example, throughout we seek to avoid lexical ambiguity, which is problematic for language engineering, but nor, on our view, is appeal to lexical ambiguity a good thing from the point of view of linguistic theory, since it typically renegades on the capture of generalisations.

syntactic types combinatorially governed by a type calculus, and in which semantic composition is driven by a structure preserving mapping from syntactic types to semantic types. In its type logical formulation the grammar is purely lexical and the type calculus is universal. There are a number of monographs, research monographs, reference works and textbooks on type logical grammar: Moortgat (1988[30]; 1997[32]), Morrill (1994[49]; 2011[50]; 2012[39]), Carpenter (1997[6]), Jäger (2005[16]), and Moot and Retoré (2012[33]).

A major challenge to categorial grammar, and indeed to all approaches to grammar, is left extraction such as relativisation in which a fronted filler appears ‘displaced’ from a gap extraction site. Relativisation is an unbounded dependency phenomenon: the distance between a relative pronoun and its extraction site can be indefinitely long:

- (1) a. the man that_{*t*}_{*i*} I know *t*_{*i*}
 b. the man that_{*t*}_{*i*} you know I know *t*_{*i*}
 c. the man that_{*t*}_{*i*} I know you know I know *t*_{*i*}
 ⋮

The treatment of relativisation in categorial grammar by means of assignment of higher-order functors to relative pronouns is well-established since Ades and Steedman (1982[1]) and yields the unboundedness property through associative assembly of the body of a relative clause.

However, although relativisation is unbounded it is not unconstrained. Various ‘islands’ can inhibit or block relativisation: weak islands such as subjects and adverbial phrases, from which extraction is mildly unacceptable, and strong islands such as coordinate structures and relative clauses themselves, from which extraction is completely unacceptable (see e.g. Szabolsci 2006[57]):

- (2) a. ?man who_{*t*}_{*i*} the friend of *t*_{*i*} laughed
 b. ?paper which_{*t*}_{*i*} John laughed before reading *t*_{*i*}
- (3) a. *man who_{*t*}_{*i*} John laughed and Mary likes *t*_{*i*}
 b. *man who_{*t*}_{*i*} John likes the woman that loves *t*_{*i*}

The conditions governing weak islands, especially, are subtle. For example, an indefinite appears to allow escape from a subject island:

- (4) man who_{*t*}_{*i*} a friend of *t*_{*i*} laughed

And Kluender (1998[21]) provides experimental evidence that there is a gradient effect on the acceptability of violations of the Complex Noun Phrase Constraint (CNPC) whereby increasing referential specificity of the extracted NP makes the sentence more acceptable, and so does decreasing the referential specificity of the extraction path. In general, under certain pragmatic or discourse-oriented conditions the processing of island violations is more acceptable (Lakoff 1986[23]; Deane 1991[8]; Kluender 1992[20], 1998[21]; Kehler 2002[19]; Hofmeister and Sag 2010[15]).

This state of affairs raises the question of how to interface grammatical coverage with explanations in terms of information structure or semantic information, or processing. The facts are interpreted by Kubota and Levine (2015[22], Section 4.6.2) to argue for a version of logical categorial grammar that freely overgenerates island constraint ‘violations’ in the syntax. But, for example, Sprouse et al. (2012[55], p.82) find “no evidence of a relationship between working memory capacity and island effects”; though see also the response of Hofmeister et al. (2013[14]). On the other hand, Newmeyer (2016[52], p.207) concedes that although the explanation of island phenomena has been a central feature of grammatical

theory since its inception, “more and more syntacticians have concluded that an exclusively syntactic approach to islands is overly ambitious”.

As we see it, the question of whether all, some or no island constraints should follow from syntactic theory is open, and while the jury is out on this issue it is acceptable, and we hope fruitful, to develop grammars as though islands were syntactic. Hence we shall adopt a conservative, syntactic, treatment of islands, while recognising that such ‘structural inhibition’ may need to be (re-)interpreted, for example, as a grammaticalisation of processing effects.²

Relativisation, furthermore, can also comprise ‘parasitic extraction’ in which a relative pronoun binds more than one extraction site (Taraldsen 1979[59]; Engdahl 1983[9]; Sag 1983[54]). There is a *single* ‘host’ gap which is not in an island, and according to the received wisdom, and according with the terminology ‘parasitic’, this may license a ‘parasitic’ gap in (any number of immediate weak) islands:

- (5) a. *the slave who_i John sold t_i t_i
 b. *the slave who_i John sold t_i to t_i
- (6) a. the man who_i the friends of t_i admire t_i
 b. the paper which_i John filed t_i without reading t_i
 c. the paper which_i the editor of t_i filed t_i without reading t_i

In addition, we observe here that these parasitic gaps may in turn function as host gaps licensing further parasitic gaps in (weak) subislands, and so on recursively:

- (7) a. man who_i the fact that the friends of t_i admire t_i surprises t_i
 b. man who_i the fact that the friends of t_i admire t_i without praising t_i offends t_i without surprising t_i

There are examples in which there appears to be a parasitic gap which is not in an island. The following is example (8a) from Postal (1993[53]):

- (8) man who_i Mary convinced t_i that John wanted to visit t_i

And an anonymous referee points out:

- (9) people whom_i you sent pictures of t_i to t_i

In respect of such examples we suggest that although there *seems* to be no island, there *could* be one.³ We present a tentative account along this line of ‘optional islands’ in Section 6.

Parasitic ‘structural facilitation’ represents a challenge to categorial grammar and all approaches to grammar. The above is the empirical analysis of islands and parasitic gaps and their interaction given a type logical, i.e. purely lexical, categorial account in Morrill (2011[50], Chapter 5). In this paper we give an account of the empirical analysis which improves on that account in the following respects:

- Multimodality and associated multimodal structural postulates are removed.⁴

² See also Morrill (2000[37]) for a way to link the binarity of formal categorial grammars and the gradience of the object of study.

³ Tom Roeper, p.c.

⁴ Structural postulates increase both the derivation search space and derivation length. Of course ways may be found to ameliorate this, but that would be precisely to absorb the structural properties in the way that is already done here (Valentín 2014[61]). *Ceteris paribus*, given the choice between structural postulates or no structural postulates, the latter is to be preferred.

- The proposal is set in the context of the displacement calculus of Morrill and Valentín (2010[41]) and Morrill, Valentín and Fadda (2011[48]) which is an advance on the discontinuous Lambek calculus of Morrill (2011[50], Chapter 6).
- Nonlinearity (structural facilitation) is formalised by use of a ‘stoup’ (Girard 2011[12]) which reduces the size of the proof search space.
- The rule of contraction generating parasitic gaps is simplified.
- The account integrates other aspects of grammatical analysis such as polymorphism, features, and intensionality.
- The correct interaction of all the grammatical aspects is verified by computer-generation of the analyses.
- Various possible exceptions to the empirical analysis are addressed.

The result, we think, is a formal and mathematically principled empirically adequate formalisation of relativisation which is thorough, very high level (concise) and which is computer-verified.

The structure of the paper is as follows. In Section 2 we present and illustrate the sequent calculus for our displacement logic. In Section 3 we present initial examples of analysis. In Section 4 we discuss approaches to relativisation with which we differ, and in Section 5 we present our theoretical analysis of relativisation. In Section 6 we address possible exceptions to our account. We conclude in Section 7. The semantic representation language used here is defined in Appendix A, and a lexicon is given in Appendix B.

2 Framework

The formalism used comprises the connectives of Table 1. The heart of the logic is the displacement calculus of Morrill and Valentín (2010[41]) and Morrill, Valentín and Fadda (2011[48]) made up of twin continuous and discontinuous residuated families of connectives having a pure Gentzen sequent calculus—without labels and free of structural rules—and enjoying Cut-elimination (Valentín 2012[60]). Other primary connectives are additives, 1st order quantifiers, normal (i.e. distributive) modalities, bracket (i.e. nondistributive) modalities, exponentials, limited contraction and limited weakening, and difference.⁵

We can draw a clear distinction between the primary connectives and the semantically inactive connectives and the synthetic connectives which are abbreviatory and are there for convenience, and to simplify derivation. There are semantically inactive variants of the continuous and discontinuous multiplicatives, and semantically inactive variants of the additives, 1st order quantifiers, and normal modalities.⁶ Synthetic connectives (Girard 2011[12]) divide into the continuous and discontinuous deterministic (unary) synthetic connectives, and the continuous and discontinuous nondeterministic (binary) synthetic connectives.⁷

⁵ Once Cut-elimination is established, the only challenge to decidability comes from nonlinearity: the universal exponential. In this connection, Morrill and Valentín (2015[44]) introduced a displacement logic $\mathbf{Db!}?$ with a relevant modality $!$ without bracket conditioning, and another system $\mathbf{Db!}_b?$ with bracket conditioning, as here. Kanovich et al. (2016[28]) prove the undecidability of $\mathbf{Db!}?$ and in unpublished work announce the undecidability of $\mathbf{Db!}_b?$. But Morrill and Valentín (2015[44]) prove the decidability of a linguistically sufficient special case of constrained bracketing of contraction with bracket conditioning.

⁶ For example, the semantically inactive additive conjunction $A \sqcap B: \phi$ abbreviates $A \& B: (\phi, \phi)$.

⁷ For example, the nondeterministic continuous division $B \div A$ abbreviates $(A \setminus B) \sqcap (B/A)$.

	cont. mult.	disc. mult.	add.	qu.	norm. mod.	brack. mod.	exp.	lim. contr. & weak.
primary	/ \ • <i>I</i>	↑ ↓ ⊙ <i>J</i>	& ⊕	∧ ∨	□ ◇	[] ⁻¹ ⟨ ⟩	! ?	 <i>W</i>
sem. inactive variants	•—○ ○—• ◐ ◑	↑ ↓ ⊙ ◐ ◑	□ ⊐	∨ ∃	■ ◆			
det. synth.	◁ ⁻¹ ▷ ⁻¹ ◁ ▷	∨ ^						diff.
nondet. synth.	÷ ⊗	↑↑ ↓↓ ⊙						—

Table 1 Categorical connectives

2.1 Syntactic types

The syntactic types of displacement logic are sorted $\mathcal{F}_0, \mathcal{F}_1, \mathcal{F}_2, \dots$ according to the number of points of discontinuity $0, 1, 2, \dots$ their expressions contain. Each type predicate letter has a sort and an arity which are naturals, and a corresponding semantic type. Assuming ordinary terms to be already given, where P is a type predicate letter of sort i and arity n and t_1, \dots, t_n are terms, $Pt_1 \dots t_n$ is an (atomic) type of sort i of the corresponding semantic type. Compound types are formed by connectives as indicated in Table 2,⁸ and the structure preserving semantic type map T associates these with semantic types.

2.2 Gentzen sequent calculus

We use a Gentzen sequent presentation standard from Gentzen (1934[10]) and Lambek (1958[26]). (Labelled) natural deduction can proportion a congenial proof format for categorical logic because the compositional term-structure of Curry-Howard semantics follows the structure of natural deduction derivation. However, here we use the Gentzen sequent proof format because:

⁸ We list only connectives drawn from the first two rows of Table 1, and we omit some which are not central here.

1.	$\mathcal{F}_i ::= \mathcal{F}_{i+j}/\mathcal{F}_j$	$T(C/B) = T(B) \rightarrow T(C)$	over
2.	$\mathcal{F}_j ::= \mathcal{F}_i \setminus \mathcal{F}_{i+j}$	$T(A \setminus C) = T(A) \rightarrow T(C)$	under
3.	$\mathcal{F}_{i+j} ::= \mathcal{F}_i \bullet \mathcal{F}_j$	$T(A \bullet B) = T(A) \& T(B)$	continuous product
4.	$\mathcal{F}_0 ::= I$	$T(I) = \top$	continuous unit
5.	$\mathcal{F}_{i+1} ::= \mathcal{F}_{i+j} \uparrow_k \mathcal{F}_j, 1 \leq k \leq i+j$	$T(C \uparrow_k B) = T(B) \rightarrow T(C)$	circumfix
6.	$\mathcal{F}_j ::= \mathcal{F}_{i+1} \downarrow_k \mathcal{F}_{i+j}, 1 \leq k \leq i+1$	$T(A \downarrow_k C) = T(A) \rightarrow T(C)$	infix
7.	$\mathcal{F}_{i+j} ::= \mathcal{F}_{i+1} \circ_k \mathcal{F}_j, 1 \leq k \leq i+1$	$T(A \circ_k B) = T(A) \& T(B)$	discontinuous product
8.	$\mathcal{F}_1 ::= J$	$T(J) = \top$	discontinuous unit
9.	$\mathcal{F}_i ::= \mathcal{F}_i \& \mathcal{F}_i$	$T(A \& B) = T(A) \& T(B)$	additive conjunction
10.	$\mathcal{F}_i ::= \mathcal{F}_i \oplus \mathcal{F}_i$	$T(A \oplus B) = T(A) + T(B)$	additive disjunction
11.	$\mathcal{F}_i ::= \bigwedge \forall \mathcal{F}_i$	$T(\bigwedge \forall A) = F \rightarrow T(A)$	1st order univ. qu.
12.	$\mathcal{F}_i ::= \bigvee \forall \mathcal{F}_i$	$T(\bigvee \forall A) = F \& T(A)$	1st order exist. qu.
13.	$\mathcal{F}_i ::= \square \mathcal{F}_i$	$T(\square A) = \mathbf{L}T(A)$	universal modality
14.	$\mathcal{F}_i ::= \diamond \mathcal{F}_i$	$T(\diamond A) = \mathbf{M}T(A)$	existential modality
15.	$\mathcal{F}_i ::= [\]^{-1} \mathcal{F}_i$	$T([\]^{-1} A) = T(A)$	univ. bracket modality
16.	$\mathcal{F}_i ::= \langle \rangle \mathcal{F}_i$	$T(\langle \rangle A) = T(A)$	exist. bracket modality
17.	$\mathcal{F}_0 ::= !\mathcal{F}_0$	$T(!A) = T(A)$	universal exponential
33.	$\mathcal{F}_i ::= \mathcal{F}_i \sqcap \mathcal{F}_i$	$T(A \sqcap B) = T(A) = T(B)$	sem. inactive additive conjunction
34.	$\mathcal{F}_i ::= \mathcal{F}_i \sqcup \mathcal{F}_i$	$T(A \sqcup B) = T(A) = T(B)$	sem. inactive additive disjunction
35.	$\mathcal{F}_i ::= \forall \forall \mathcal{F}_i$	$T(\forall \forall A) = T(A)$	sem. inactive 1st order univ. qu.
36.	$\mathcal{F}_i ::= \exists \forall \mathcal{F}_i$	$T(\exists \forall A) = T(A)$	sem. inactive 1st order exist. qu.
37.	$\mathcal{F}_i ::= \blacksquare \mathcal{F}_i$	$T(\blacksquare A) = T(A)$	sem. inactive universal modality
38.	$\mathcal{F}_i ::= \blacklozenge \mathcal{F}_i$	$T(\blacklozenge A) = T(A)$	sem. inactive existential modality

Table 2 Syntactic types

- Natural deduction does not capture symmetries as satisfactorily as Gentzen sequent calculus. For example, while product right is easy to express in ND, product left is awkward (unnatural); but both are straightforwardly expressed in Gentzen sequent calculus.⁹
- The title of the paper is ‘Grammar logicised’, i.e. there is an emphasis on the thesis that grammar can be reduced to logic. To maintain this it is appropriate to pitch the logical aspects as closely as possible to the usual Gentzen format with the associated symmetries and metatheory.
- It enables uniform formulation of all of the rules of inference.
- It dispenses with phonological labels.
- It lends itself more transparently to standard proof of Cut-elimination and consequent decidability results.
- It lends itself more transparently to focalisation (Andreoli 1992[3]); Morrill and Valentín 2015[45]) and consequent efficient computer generation and verification of the analyses.

Although discontinuous types play a minor role here, our analysis is pitched in their context in order to show its consistency with displacement calculus. Crucially, in Gentzen sequent configurations (Γ, Δ) for displacement calculus a discontinuous type is a mother, rather than a leaf, and dominates its discontinuous components marked off by curly brackets and colons.

⁹ Thus, in Gentzen format product left is simply $\frac{A(A \bullet B) \Rightarrow D}{A(A \bullet B) \Rightarrow D}$ but unlabelled ND requires something like

$$(i) \quad \frac{\begin{array}{c} \vdots \\ A \bullet B \end{array}}{A \quad B}$$

which does not respect the single-conclusion condition and there are consequent complications regarding, for example, which hypotheses can be cancelled when.

In Gentzen sequent antecedents for displacement logic with bracket modalities (structural inhibition) and exponentials (structural facilitation) there is also a bracket constructor for the former and ‘stoups’ for the latter.

Stoups (cf. the linear logic of Girard 2011[12]) (ζ) are stores read as multisets for reusable (nonlinear) resources which appear at the left of a configuration marked off by a semicolon (when the stoup is empty the semicolon may be omitted). The stoup of linear logic is for resources which can be contracted (copied) or weakened (deleted). By contrast, our stoup is for a linguistically motivated variant of contraction, and does not allow weakening. Furthermore, whereas linear logic is commutative, our logic is in general noncommutative and the stoup is used for resources which are also commutative. To anticipate our analysis a little, a hypothetical subtype emitted by a relative pronoun corresponding to a long-distance dependency will enter a stoup, percolate in stoups, maybe contracting to create (parasitic) gaps, and finally permute into a (host) extraction site.

A configuration together with a stoup is a *zone* (\mathcal{E}). The bracket constructor applies not to a configuration alone but to a configuration with a stoup, i.e a zone: reusable resources are specific to their domain.

Stoups \mathcal{S} and configurations \mathcal{O} are defined by (\emptyset is the empty stoup; Λ is the empty configuration; the *separator* 1 marks points of discontinuity):¹⁰

$$(10) \quad \begin{aligned} \mathcal{S} &::= \emptyset \mid \mathcal{F}_0, \mathcal{S} \\ \mathcal{O} &::= \Lambda \mid \mathcal{T}, \mathcal{O} \\ \mathcal{T} &::= 1 \mid \mathcal{F}_0 \mid \mathcal{F}_{i>0} \{ \underbrace{\mathcal{O} : \dots : \mathcal{O}}_{i \mathcal{O}'s} \} \mid [\mathcal{S}; \mathcal{O}] \end{aligned}$$

For a type A , its sort $s(A)$ is the i such that $A \in \mathcal{F}_i$. For a configuration Γ , its sort $s(\Gamma)$ is $|\Gamma|_1$, i.e. the number of points of discontinuity 1 which it contains. Sequents are of the form:

$$(11) \quad \mathcal{S}; \mathcal{O} \Rightarrow \mathcal{F} \text{ such that } s(\mathcal{O}) = s(\mathcal{F})$$

The figure \vec{A} of a type A is defined by:

$$(12) \quad \vec{A} = \begin{cases} A & \text{if } s(A) = 0 \\ A \{ \underbrace{1 : \dots : 1}_{s(A) \text{ 1's}} \} & \text{if } s(A) > 0 \end{cases}$$

Where Γ is a configuration of sort i and $\Delta_1, \dots, \Delta_i$ are configurations, the *fold* $\Gamma \otimes \langle \Delta_1 : \dots : \Delta_i \rangle$ is the result of replacing the successive 1's in Γ by $\Delta_1, \dots, \Delta_i$ respectively. Where Γ is of sort i , the hyperoccurrence notation $\Delta \langle \Gamma \rangle$ abbreviates $\Delta_0(\Gamma \otimes \langle \Delta_1 : \dots : \Delta_i \rangle)$, i.e. a context configuration Δ (which is externally Δ_0 and internally $\Delta_1, \dots, \Delta_i$) with a potentially discontinuous distinguished subconfiguration Γ (continuous if $i = 0$, discontinuous if $i > 0$). Where Δ is a configuration of sort $i > 0$ and Γ is a configuration, the k th *metalinguistic intercalation* $\Delta|_k \Gamma$, $1 \leq k \leq i$, is given by:

$$(13) \quad \Delta|_k \Gamma =_{df} \Delta \otimes \langle \underbrace{1 : \dots : 1}_{k-1 \text{ 1's}} : \Gamma : \underbrace{1 : \dots : 1}_{i-k \text{ 1's}} \rangle$$

i.e. $\Delta|_k \Gamma$ is the configuration resulting from replacing by Γ the k th separator in Δ .

¹⁰ Note that only types of sort 0 can go into the stoup; reusable types of other sorts would not preserve the sequent antecedent-succedent sort equality under contraction: $0 + 0 = 0$, but $i + i \neq i$ for $i > 0$.

2.3 Rules and linguistic applications

A semantically labelled sequent is a sequent in which the antecedent type occurrences A_1, \dots, A_n are labelled by distinct variables x_1, \dots, x_n of types $T(A_1), \dots, T(A_n)$ respectively, and the succedent type A is labelled by a term of type $T(A)$ with free variables drawn from x_1, \dots, x_n . In this section we give the semantically labelled Gentzen sequent rules for some primary connectives, and indicate some linguistic applications.

$$\begin{array}{l}
1. \quad \frac{\zeta_1; \Gamma \Rightarrow B: \psi \quad \zeta_2; \Delta \langle \vec{C}: z \rangle \Rightarrow D: \omega}{\zeta_1 \uplus \zeta_2; \Delta \langle \vec{C}/\vec{B}: x, \Gamma \rangle \Rightarrow D: \omega\{x\psi/z\}} /L \quad \frac{\zeta; \Gamma, \vec{B}: y \Rightarrow C: \chi}{\zeta; \Gamma \Rightarrow C/B: \lambda y \chi} /R \\
2. \quad \frac{\zeta_1; \Gamma \Rightarrow A: \phi \quad \zeta_2; \Delta \langle \vec{C}: z \rangle \Rightarrow D: \omega}{\zeta_1 \uplus \zeta_2; \Delta \langle \Gamma, \vec{A}\vec{C}: y \rangle \Rightarrow D: \omega\{y\phi/z\}} \setminus L \quad \frac{\zeta; \vec{A}: x, \Gamma \Rightarrow C: \chi}{\zeta; \Gamma \Rightarrow A \setminus C: \lambda x \chi} \setminus R \\
3. \quad \frac{\zeta; \Delta \langle \vec{A}: x, \vec{B}: y \rangle \Rightarrow D: \omega}{\zeta; \Delta \langle \vec{A} \bullet \vec{B}: z \rangle \Rightarrow D: \omega\{\pi_1 z/x, \pi_2 z/y\}} \bullet L \quad \frac{\zeta_1; \Gamma_1 \Rightarrow A: \phi \quad \zeta_2; \Gamma_2 \Rightarrow B: \psi}{\zeta_1 \uplus \zeta_2; \Gamma_1, \Gamma_2 \Rightarrow A \bullet B: (\phi, \psi)} \bullet R \\
4. \quad \frac{\zeta; \Delta \langle \Lambda \rangle \Rightarrow A: \phi}{\zeta; \Delta \langle \vec{I}: x \rangle \Rightarrow A: \phi} IL \quad \frac{}{\emptyset; \Lambda \Rightarrow I: 0} IR
\end{array}$$

Fig. 1 Continuous multiplicatives

The continuous multiplicatives of Figure 1, the Lambek connectives, Lambek (1958[26]; 1988[25]), defined in relation to concatenation/ appending, are the basic means of categorial categorization and subcategorization. Note that here and throughout the active types in antecedents are figures (vectorial) whereas those in succedents are not; intuitively this is because antecedents are structured but succedents are not. The directional divisions over, /, and under, \, are exemplified by assignments such as *the*: N/CN for *the man*: N and *sings*: $N \setminus S$ for *John sings*: S , and *loves*: $(N \setminus S)/N$ for *John loves Mary*: S . The continuous product \bullet is exemplified by a ‘small clause’ assignment such as *considers*: $(N \setminus S)/(N \bullet (CN/CN))$ for say *John considers Mary socialist*: S .¹¹

The discontinuous multiplicatives of Figure 2, the displacement connectives, Morrill and Valentín (2010[41]), Morrill, Valentín and Fadda (2011[48]), are defined in relation to intercalation/plugging. When the value of the k subindex indicates the first (leftmost) point of discontinuity it may be omitted. Circumfixation, \uparrow , is exemplified by a discontinuous idiom assignment *gives+1+the+cold+shoulder*: $(N \setminus S) \uparrow N$ for *Mary gives John the cold shoulder*: S , and infixation, \downarrow , and circumfixation together are exemplified by a quantifier phrase assignment *everyone*: $(S \uparrow N) \downarrow S$ simulating Montague’s S14 treatment of quantifying in. Circumfixation and discontinuous product, \odot , are illustrated together with the continuous unit in an assignment to a relative pronoun *that*: $(CN \setminus CN)/((S \uparrow N) \odot I)$ allowing both peripheral and medial extraction: *that John likes*: $CN \setminus CN$ and *that John saw today*: $CN \setminus CN$, although we

¹¹ But this makes no different empirical predictions from the more standard type of analysis in categorial grammar which simply treats verbs like *consider* as taking a noun phrase and an infinitive. Products are more truly motivated by antecedent occurrences in the (continuous) analysis of past participles of Morrill (2011[50], pp.64–65), or the discontinuous generalisation of this for a past participle such as *loved* in Appendix B here.

$$\begin{array}{l}
5. \quad \frac{\zeta_1; \Gamma \Rightarrow B: \psi \quad \zeta_2; \Delta \langle \vec{C}: z \rangle \Rightarrow D: \omega}{\zeta_1 \uplus \zeta_2; \Delta \langle C \uparrow_k \vec{B}: x \mid_k \Gamma \rangle \Rightarrow D: \omega \{ (x \psi) / z \}} \uparrow_k L \quad \frac{\zeta; \Gamma \mid_k \vec{B}: y \Rightarrow C: \chi}{\zeta; \Gamma \Rightarrow C \uparrow_k B: \lambda y \chi} \uparrow_k R \\
6. \quad \frac{\zeta_1; \Gamma \Rightarrow A: \phi \quad \zeta_2; \Delta \langle \vec{C}: z \rangle \Rightarrow D: \omega}{\zeta_1 \uplus \zeta_2; \Delta \langle \Gamma \mid_k A \downarrow_k \vec{C}: y \rangle \Rightarrow D: \omega \{ (y \phi) / z \}} \downarrow_k L \quad \frac{\zeta; \vec{A}: x \mid_k \Gamma \Rightarrow C: \chi}{\zeta; \Gamma \Rightarrow A \downarrow_k C: \lambda x \chi} \downarrow_k R \\
7. \quad \frac{\zeta; \Delta \langle \vec{A}: x \mid_k \vec{B}: y \rangle \Rightarrow D: \omega}{\zeta; \Delta \langle A \circ_k \vec{B}: z \rangle \Rightarrow D: \omega \{ \pi_1 z / x, \pi_2 z / y \}} \circ_k L \quad \frac{\zeta_1; \Gamma_1 \Rightarrow A: \phi \quad \zeta_2; \Gamma_2 \Rightarrow B: \psi}{\zeta_1 \uplus \zeta_2; \Gamma_1 \mid_k \Gamma_2 \Rightarrow A \circ_k B: (\phi, \psi)} \circ_k R \\
8. \quad \frac{\zeta; \Delta \langle 1 \rangle \Rightarrow A: \phi}{\zeta; \Delta \langle \vec{J}: x \rangle \Rightarrow A: \phi} JL \quad \frac{}{\emptyset; 1 \Rightarrow J: 0} JR
\end{array}$$

Fig. 2 Discontinuous multiplicatives

will argue in Section 4 that this strategy is inadequate, and the main point of the present paper is to promote another approach to relativisation.

In relation to the multiplicative rules, notice how the stoup is distributed reading bottom-up from conclusions to premise: it is partitioned between the two premises in the case of binary rules, copied to the premise in the case of unary rules, and empty in the case of nullary rules (axioms).

$$\begin{array}{l}
9. \quad \frac{\frac{\frac{\Xi \langle \vec{A}: x \rangle \Rightarrow C: \chi}{\Xi \langle A \& \vec{B}: z \rangle \Rightarrow C: \chi \{ \pi_1 z / x \}} \&L_1 \quad \frac{\Xi \langle \vec{B}: y \rangle \Rightarrow C: \chi}{\Xi \langle A \& \vec{B}: z \rangle \Rightarrow C: \chi \{ \pi_2 z / y \}} \&L_2}{\Xi \Rightarrow A: \phi \quad \Xi \Rightarrow B: \psi} \&R}{\Xi \Rightarrow A \& B: (\phi, \psi)} \\
10. \quad \frac{\frac{\frac{\Xi \langle \vec{A}: x \rangle \Rightarrow C: \chi_1 \quad \Xi \langle \vec{B}: y \rangle \Rightarrow C: \chi_2}{\Xi \langle A \oplus \vec{B}: z \rangle \Rightarrow C: z \rightarrow x, \chi_1; y, \chi_2} \oplus L}{\Xi \Rightarrow A: \phi} \oplus R_1 \quad \frac{\Xi \Rightarrow B: \psi}{\Xi \Rightarrow A \oplus B: \iota_2 \psi} \oplus R_2}{\Xi \Rightarrow A \oplus B: \iota_1 \phi} \oplus R_1
\end{array}$$

Fig. 3 Additives

The additives of Figure 3, Lambek (1961[24]), Morrill (1990[34]), Kanazawa (1992[18]), have application to polymorphism. For example the additive conjunction $\&$ can be used for *rice*: $N \& CN$ as in *rice grows*: S and *the rice grows*: S ,¹² and the additive disjunction \oplus can be used for *is*: $(N \setminus S) / (N \oplus (CN / CN))$ as in *Tully is Cicero*: S and *Tully is humanist*: S . The

¹² Note the computational advantage of this approach over assuming an empty determiner: if empty operators were allowed they could occur any number of times in any positions.

additive disjunction can be used together with the continuous unit to express the optionality of a complement as in $eats: (N \setminus S) / (N \oplus I)$ for *John eats fish: S* and *John eats: S*.¹³

Notice how the stoup is identical in conclusions and premises of additive rules.

$$\begin{array}{l}
 11. \quad \frac{\overrightarrow{\Xi \langle \overline{A[t/v]}: x \rangle \Rightarrow B: \psi}}{\overrightarrow{\Xi \langle \bigwedge vA: z \rangle \Rightarrow B: \psi \{ (z \ t) / x \}}} \wedge L \quad \frac{\Xi \Rightarrow A[a/v]: \phi}{\Xi \Rightarrow \bigwedge vA: \lambda v \phi} \wedge R^\dagger \\
 12. \quad \frac{\overrightarrow{\Xi \langle \overline{A[a/v]}: x \rangle \Rightarrow B: \psi}}{\overrightarrow{\Xi \langle \bigvee vA: z \rangle \Rightarrow B: \psi \{ \pi_2 z / x \}}} \vee L^\dagger \quad \frac{\Xi \Rightarrow A[t/v]: \phi}{\Xi \Rightarrow \bigvee vA: (t, \phi)} \vee R
 \end{array}$$

Fig. 4 Quantifiers, where † indicates that there is no a in the conclusion

The quantifiers of Figure 4, Morrill (1994[49]), have application to features. For example, singular and plural number in $sheep: \bigwedge nCNn$ for *the sheep grazes: S* and *the sheep graze: S*. And for a past, present or future tense finite sentence complement we can have $said: (N \setminus S) / \bigvee tS f(t)$ in *John said Mary walked: S*, *John said Mary walks: S* and *John said Mary will walk: S*.

Notice how the stoup is identical in conclusion and premise in each quantifier rule.

$$\begin{array}{l}
 13. \quad \frac{\overrightarrow{\Xi \langle \overline{A}: x \rangle \Rightarrow B: \psi}}{\overrightarrow{\Xi \langle \overline{\square A}: z \rangle \Rightarrow B: \psi \{ \vee z / x \}}} \square L \quad \frac{\boxtimes \Xi \Rightarrow A: \phi}{\boxtimes \Xi \Rightarrow \square A: \wedge \phi} \square R \\
 14. \quad \frac{\overrightarrow{\boxtimes \Xi \langle \overline{A}: x \rangle \Rightarrow \diamond B: \psi}}{\overrightarrow{\boxtimes \Xi \langle \overline{\diamond A}: z \rangle \Rightarrow \diamond B: \psi \{ \cup z / x \}}} \diamond L \quad \frac{\Xi \Rightarrow A: \phi}{\Xi \Rightarrow \diamond A: \cap \phi} \diamond R
 \end{array}$$

Fig. 5 Normal modalities, where \boxtimes/\diamond marks a structure all the types of which have main connective a box/diamond

With respect to the normal modalities of Figure 5, the universal (Morrill 1990[35]) has application to intensionality. For example, for a propositional attitude verb such as *believes* we can assign type $\square((N \setminus S) / \square S)$ with a modality outermost since the word has a sense, and a modality on the first argument but not the second, since the sentential complement is an intensional domain, but the subject is not.

Notice how the stoup is identical in conclusion and premise in each normal modality rule.

The bracket modalities of Figure 6, Morrill (1992[36]) and Moortgat 1995[31]), have application to nonassociativity and syntactical domains such as prosodic phrases and extraction islands. For example, single bracketing for weak islands: $walks: \langle \rangle N \setminus S$ for the sub-

¹³ Note the advantage of this over simply listing intransitive and transitive lexical entries: empirically the latter does not capture the generalisation that in both cases the verb *eats* combines with a subject to the left, and computationally every lexical ambiguity doubles the lexical insertion search space. Appeal to lexical ambiguity constitutes resignation from the capture of generalisations and is at best a promissory solution, unless there is true ambiguity.

$$\begin{array}{l}
15. \quad \frac{\Xi \langle \vec{A}: x \rangle \Rightarrow B: \psi}{\Xi \langle [\]^{-1} \vec{A}: x \rangle \Rightarrow B: \psi} [\]^{-1}L \quad \frac{[\] \Xi \Rightarrow A: \phi}{\Xi \Rightarrow [\]^{-1}A: \phi} [\]^{-1}R \\
16. \quad \frac{\Xi \langle \vec{A}: x \rangle \Rightarrow B: \psi}{\Xi \langle \langle \vec{A} \rangle: x \rangle \Rightarrow B: \psi} \langle \rangle L \quad \frac{\Xi \Rightarrow A: \phi}{[\] \Xi \Rightarrow \langle \rangle A: \phi} \langle \rangle R
\end{array}$$

Fig. 6 Bracket modalities

ject condition, and *without*: $[\]^{-1}(VP \setminus VP)/VP$ for the adverbial island constraint; and double bracketing for strong islands such as *and*: $(S \setminus [\]^{-1}[\]^{-1}S)/S$ for the coordinate structure constraint.

Notice how the stoup is identical in conclusions and premises of bracket modality rules.

$$\begin{array}{l}
17. \quad \frac{\Xi(\zeta \uplus \{A: x\}; \Gamma_1, \Gamma_2) \Rightarrow B: \psi}{\Xi(\zeta; \Gamma_1, !A: x, \Gamma_2) \Rightarrow B: \psi} !L \quad \frac{\zeta; \Lambda \Rightarrow A: \phi}{\zeta; \Lambda \Rightarrow !A: \phi} !R \\
\frac{\Xi(\zeta; \Gamma_1, A: x, \Gamma_2) \Rightarrow B: \psi}{\Xi(\zeta \uplus \{A: x\}; \Gamma_1, \Gamma_2) \Rightarrow B: \psi} !P \\
\frac{\Xi(\zeta \uplus \{A: x\}; \Gamma_1, [\{A: y\}; \Gamma_2], \Gamma_3) \Rightarrow B: \psi}{\Xi(\zeta \uplus \{A: x\}; \Gamma_1, \Gamma_2, \Gamma_3) \Rightarrow B: \psi \{x/y\}} !C
\end{array}$$

Fig. 7 Universal exponential

Finally, there is nonlinearity. The universal exponential of Figure 7, Girard (1987[11]), Barry, Hepple, Leslie and Morrill (1991[5]), Morrill (1994[49]), and Morrill and Valentín (2015[44]) has application to parasitic extraction. In the formulation here $!L$ moves the operand of a universal exponential (e.g. the hypothetical subtype of relativisation) into the stoup, where it will percolate as commented for the above rules. From there it can be copied into the stoup of a newly-created bracketed domain by the contraction rule $!C$ (producing a parasitic gap), and it can be moved into any position in the matrix configuration of its zone by $!P$ (producing a normal nonparasitic or host gap). For example:

$$(14) \quad \frac{\frac{\frac{\dots, A, \dots, [\dots, A, \dots], \dots, \dots \Rightarrow D}{\dots, A, \dots, [A; \dots, \dots], \dots, \dots \Rightarrow D} !P}{A; \dots, \dots, [A; \dots, \dots], \dots, \dots \Rightarrow D} !P}{\vdots} !C \\
\frac{\frac{A; \dots, \dots, \dots, \dots, \dots \Rightarrow B}{\dots, \dots, \dots, \dots, !A, \dots \Rightarrow B} !L}{}$$

Reading upwards, first the $!A$ is moved into the stoup by $!L$ and the exponential modality is removed (being in the stoup *means* that the type is under the associated resource management regime). We assume some derivation steps, indicated by vertical dots, and then an application of contraction $!C$. A domain becomes bracketed, *and this domain contains A*

in its stoup. This would correspond to a weak island containing a parasitic gap. Finally the ‘host’ and ‘parasitic’ gaps are permuted into position by two applications of $!P$.

Using the universal exponential, $!$, for which contraction induces island brackets, we can assign a relative pronoun type *that*: $(CN \setminus CN)/(S/!N)$ allowing parasitic extraction such as *paper that John filed without reading*: CN , where parasitic gaps can appear only in (weak) islands, but can be iterated in subislands, for example, *man who the fact that the friends of admire without praising surprises*. See Section 5.

Crucially, in the linguistic formulation $!$ does not have weakening, i.e. deletion, since, e.g., the body of a relative clause *must* contain a gap: **man who John loves Mary*.

In relation to the rest of the primary connectives: the existential exponential $?$ has application to iterated coordination (Morrill 1994[49]; Morrill and Valentín 2015[44]) and (unboundedly iterated) *respectively* (Morrill and Valentín 2016[46]), the limited contraction $|$ of Jäger (2005[16]) has application to anaphora and the limited weakening W of Morrill and Valentín (2014[43]) has application to words as types. The remaining, semantically inactive, connectives listed here were introduced as follows. Semantically inactive multiplicatives $\{\bullet, \dashv, \circ, \rightarrow, \odot, \oplus, \uparrow, \downarrow, \updownarrow, \ominus, \otimes\}$: Morrill and Valentín (2014[43]). Semantically inactive additives $\{\sqcap, \sqcup\}$: Morrill (1994[49]). Semantically inactive first-order quantifiers $\{\forall, \exists\}$: Morrill (1994[49]). Semantically inactive normal modalities $\{\blacksquare, \blacklozenge\}$: Hepple (1990[13]), Morrill (1994[49]). The rules for semantically inactive variants are the same as those for the semantically active versions syntactically, but have the same label on premises and conclusions semantically; see for example Figures 8, 9 and 10.¹⁴

$$33. \frac{\frac{\mathcal{E}\langle \vec{A}: x \rangle \Rightarrow C: \chi}{\mathcal{E}\langle \vec{A} \sqcap \vec{B}: x \rangle \Rightarrow C: \chi} \sqcap L_1 \quad \frac{\mathcal{E}\langle \vec{B}: y \rangle \Rightarrow C: \chi}{\mathcal{E}\langle \vec{A} \sqcap \vec{B}: y \rangle \Rightarrow C: \chi} \sqcap L_2}{\frac{\mathcal{E} \Rightarrow A: \chi \quad \mathcal{E} \Rightarrow B: \chi}{\mathcal{E} \Rightarrow A \sqcap B: \chi} \sqcap R}$$

Fig. 8 Semantically inactive additive conjunction

$$35. \frac{\frac{\mathcal{E}\langle \vec{A}[t/v]: x \rangle \Rightarrow B: \psi}{\mathcal{E}\langle \forall v \vec{A}: x \rangle \Rightarrow B: \psi} \forall L \quad \frac{\mathcal{E} \Rightarrow A[a/v]: \phi}{\mathcal{E} \Rightarrow \forall v A: \phi} \forall R^\dagger}{\mathcal{E} \Rightarrow \forall v A: \phi}$$

Fig. 9 Semantically inactive universal quantifier, where † indicates that there is no a in the conclusion

¹⁴ The synthetic connectives are: left and right projection and injection $\{\leftarrow^{-1}, \rightarrow^{-1}, \leftarrow, \rightarrow\}$, Morrill, Fadda and Valentín (2009[47]); split and bridge $\{\dot{\cdot}, \dot{\wedge}\}$, Morrill and Merenciano (1996[40]); continuous and discontinuous nondeterministic multiplicatives $\{\div, \otimes, \uparrow, \downarrow, \otimes\}$, Morrill, Valentín and Fadda (2011[48]). The difference operator $-$ of Morrill and Valentín (2014[42]) has application to linguistic exceptions.

$$37. \frac{\frac{\mathcal{E}\langle \overline{A}: x \rangle \Rightarrow B: \psi}{\mathcal{E}\langle \overline{\blacksquare}A: x \rangle \Rightarrow B: \psi} \blacksquare L}{\frac{\frac{\square/\blacksquare\mathcal{E} \Rightarrow A: \phi}{\square/\blacksquare\mathcal{E} \Rightarrow \blacksquare A: \phi} \blacksquare R}}{\blacksquare R}$$

Fig. 10 Semantically inactive universal normal modality

3 Initial examples

The first example is as follows:¹⁵

(15) [john]+walks : Sf

Note that in our syntactical form the subject is a bracketed domain, and this will generally be the case — implementing that subjects are weak islands. Lookup in our lexicon yields the following semantically labelled sequent:

(16) [$\blacksquare Nt(s(m)) : j$], $\square(\langle \exists g Nt(s(g)) \rangle \backslash Sf) : \wedge \lambda A (Pres (\sim walk A)) \Rightarrow Sf$

The lexical types are semantically modalised outermost, and this will always be the case — implementing that word meanings are intensions/senses; the modality of the proper name subject is semantically inactive (proper names are rigid designators), while the modality of the tensed verb is semantically active (the interpretation of tensed verbs depends on the temporal reference points). The verb projects a finite sentence (feature f) when it combines with a third person singular (bracketed) subject of any gender g (the existential quantification); the actual subject is masculine (feature m).

The derivation is as follows:

$$(17) \frac{\frac{\frac{\boxed{Nt(s(m))} \Rightarrow Nt(s(m))}{\boxed{\blacksquare Nt(s(m))} \Rightarrow Nt(s(m))} \blacksquare L}{\boxed{\blacksquare Nt(s(m))} \Rightarrow \boxed{\exists g Nt(s(g))} \exists R}{\boxed{\blacksquare Nt(s(m))} \Rightarrow \boxed{\langle \exists g Nt(s(g)) \rangle} \langle \rangle R}{\frac{\boxed{\blacksquare Nt(s(m))} \Rightarrow \boxed{\langle \exists g Nt(s(g)) \rangle \backslash Sf} \backslash L}{\frac{\boxed{\blacksquare Nt(s(m))}, \boxed{\langle \exists g Nt(s(g)) \rangle \backslash Sf} \Rightarrow Sf}{\boxed{\blacksquare Nt(s(m))}, \boxed{\square(\langle \exists g Nt(s(g)) \rangle \backslash Sf)} \Rightarrow Sf} \square L$$

The flow of information in the semantic reading of derivations can be illustrated for the case in hand as follows; note that in practice the steps of this information flow are implemented by unification stepwise with derivation. First, variables for the antecedent semantics are added in the endsequent:

¹⁵ The derivations we give have been computer-generated from a lexicon (given in Appendix B) and parser CatLog2 for the categorial logic, available at <http://www.cs.upc.edu/~morrill>. There is no particular reason for the exact constitution of the lexicon; it exemplifies the state of experimentation at the time that the subfragment of derivations presented in this paper was generated. The implementation is a categorial parser/theorem-prover CatLog2 comprising 6000 lines of Prolog using backward chaining proof-search in the Gentzen sequent calculus (Morrill 2011[38]), and the focusing of Andreoli (1992[3]); see Morrill and Valentín (2015[45]). In addition to focusing, the implementation exploits count-invariance (van Benthem 1991[63]; Valentín, Serret and Morrill (2013[62]). In focusing, proofs are built in alternating phases of don't care nondeterministic invertible/asynchronous rule application and focused noninvertible/synchronous rule application. The boxes in our derivations mark the focused types, which are the active types of synchronous rule application. All the reader needs to keep in mind is that if there is a boxed type in the conclusion of an inference step then it is the active type of that inference step, i.e. the type which is decomposed reading from conclusion to premises.

$$(18) \boxed{\blacksquare Nt(s(m)) : x}, \boxed{\langle \exists g Nt(s(g)) \setminus S f \rangle} : y \Rightarrow S f$$

Reading bottom-up, at the lowest inference step ($\square L$) the verb semantics is replaced by the extension z and the subject semantics x is carried over:

$$(19) \frac{\boxed{\blacksquare Nt(s(m)) : x}, \boxed{\langle \exists g Nt(s(g)) \setminus S f \rangle} : z \Rightarrow S f}{\boxed{\blacksquare Nt(s(m)) : x}, \boxed{\langle \exists g Nt(s(g)) \setminus S f \rangle} : y \Rightarrow S f} \square L$$

At the second inference we propagate the subject semantics on the argument branch:

$$(20) \frac{\boxed{\blacksquare Nt(s(m)) : x} \Rightarrow \boxed{\langle \exists g Nt(s(g)) \rangle} \quad \boxed{S f} \Rightarrow S f}{\boxed{\blacksquare Nt(s(m)) : x}, \boxed{\langle \exists g Nt(s(g)) \setminus S f \rangle} : z \Rightarrow S f} \setminus L \\ \frac{\quad}{\boxed{\blacksquare Nt(s(m)) : x}, \boxed{\langle \exists g Nt(s(g)) \setminus S f \rangle} : y \Rightarrow S f} \square L$$

The next three inferences involve semantically transparent copying of the antecedent semantics:

$$(21) \frac{\boxed{Nt(s(m)) : x} \Rightarrow Nt(s(m))}{\boxed{\blacksquare Nt(s(m)) : x} \Rightarrow Nt(s(m))} \blacksquare L \\ \frac{\quad}{\boxed{\blacksquare Nt(s(m)) : x} \Rightarrow \boxed{\exists g Nt(s(g))}} \exists R \\ \frac{\boxed{\blacksquare Nt(s(m)) : x} \Rightarrow \boxed{\langle \exists g Nt(s(g)) \rangle} \quad \boxed{S f} \Rightarrow S f}{\boxed{\blacksquare Nt(s(m)) : x}, \boxed{\langle \exists g Nt(s(g)) \setminus S f \rangle} : z \Rightarrow S f} \setminus L \\ \frac{\quad}{\boxed{\blacksquare Nt(s(m)) : x}, \boxed{\langle \exists g Nt(s(g)) \setminus S f \rangle} : y \Rightarrow S f} \square L$$

At the identity axiom the antecedent semantics is copied to the succedent:

$$(22) \frac{\boxed{Nt(s(m)) : x} \Rightarrow Nt(s(m)) : x}{\boxed{\blacksquare Nt(s(m)) : x} \Rightarrow Nt(s(m))} \blacksquare L \\ \frac{\quad}{\boxed{\blacksquare Nt(s(m)) : x} \Rightarrow \boxed{\exists g Nt(s(g))}} \exists R \\ \frac{\boxed{\blacksquare Nt(s(m)) : x} \Rightarrow \boxed{\langle \exists g Nt(s(g)) \rangle} \quad \boxed{S f} \Rightarrow S f}{\boxed{\blacksquare Nt(s(m)) : x}, \boxed{\langle \exists g Nt(s(g)) \setminus S f \rangle} : z \Rightarrow S f} \setminus L \\ \frac{\quad}{\boxed{\blacksquare Nt(s(m)) : x}, \boxed{\langle \exists g Nt(s(g)) \setminus S f \rangle} : y \Rightarrow S f} \square L$$

In a following phase the succedent semantics is copied from premises to conclusions as far as the root of the argument branch:

$$(23) \frac{\boxed{Nt(s(m)) : x} \Rightarrow Nt(s(m)) : x}{\boxed{\blacksquare Nt(s(m)) : x} \Rightarrow Nt(s(m)) : x} \blacksquare L \\ \frac{\quad}{\boxed{\blacksquare Nt(s(m)) : x} \Rightarrow \boxed{\exists g Nt(s(g))} : x} \exists R \\ \frac{\boxed{\blacksquare Nt(s(m)) : x} \Rightarrow \boxed{\langle \exists g Nt(s(g)) : x \rangle} \quad \boxed{S f} \Rightarrow S f}{\boxed{\blacksquare Nt(s(m)) : x}, \boxed{\langle \exists g Nt(s(g)) \setminus S f \rangle} : z \Rightarrow S f} \setminus L \\ \frac{\quad}{\boxed{\blacksquare Nt(s(m)) : x}, \boxed{\langle \exists g Nt(s(g)) \setminus S f \rangle} : y \Rightarrow S f} \square L$$

Now the functor value semantics in the antecedent of the value branch is labelled with a new variable w :

$$(24) \frac{\frac{\frac{\overline{Nt(s(m)) : x} \Rightarrow Nt(s(m)) : x}{\blacksquare Nt(s(m)) : x} \blacksquare L}{\blacksquare Nt(s(m)) : x \Rightarrow \overline{\exists g Nt(s(g)) : x} \exists R}}{\blacksquare Nt(s(m)) : x \Rightarrow \langle \exists g Nt(s(g)) : x \rangle \langle R}} \frac{\overline{Sf : w} \Rightarrow Sf}{\blacksquare Nt(s(m)) : x, \langle \exists g Nt(s(g)) \setminus Sf \rangle : z \Rightarrow Sf} \setminus L}{\blacksquare Nt(s(m)) : x, \square(\langle \exists g Nt(s(g)) \setminus Sf \rangle) : y \Rightarrow Sf} \square L$$

At the id axiom this semantics is copied from antecedent to succedent:

$$(25) \frac{\frac{\frac{\overline{Nt(s(m)) : x} \Rightarrow Nt(s(m)) : x}{\blacksquare Nt(s(m)) : j} \blacksquare L}{\blacksquare Nt(s(m)) : j \Rightarrow Nt(s(m)) : j} \exists R}}{\blacksquare Nt(s(m)) : x \Rightarrow \overline{\exists g Nt(s(g)) : x} \langle R}} \frac{\overline{Sf : w} \Rightarrow Sf : w}{\blacksquare Nt(s(m)) : x, \langle \exists g Nt(s(g)) \setminus Sf \rangle : z \Rightarrow Sf} \setminus L}{\blacksquare Nt(s(m)) : x, \square(\langle \exists g Nt(s(g)) \setminus Sf \rangle) : y \Rightarrow Sf} \square L$$

In the $\setminus L$ conclusion succedent the semantics of the major premise is subject to the substitution of w by the functional application of the functor z to the argument x :

$$(26) \frac{\frac{\frac{\overline{Nt(s(m)) : x} \Rightarrow Nt(s(m)) : x}{\blacksquare Nt(s(m)) : x} \blacksquare L}{\blacksquare Nt(s(m)) : x \Rightarrow \overline{\exists g Nt(s(g)) : x} \exists R}}{\blacksquare Nt(s(m)) : x \Rightarrow \langle \exists g Nt(s(g)) : x \rangle \langle R}} \frac{\overline{Sf : w} \Rightarrow Sf : w}{\blacksquare Nt(s(m)) : x, \langle \exists g Nt(s(g)) \setminus Sf \rangle : z \Rightarrow Sf : w\{z x/w\} = (z x)} \setminus L}{\blacksquare Nt(s(m)) : x, \square(\langle \exists g Nt(s(g)) \setminus Sf \rangle) : y \Rightarrow Sf} \square L$$

And thence to the conclusion of the endsequent:

$$(27) \frac{\frac{\frac{\overline{Nt(s(m)) : x} \Rightarrow Nt(s(m)) : x}{\blacksquare Nt(s(m)) : x} \blacksquare L}{\blacksquare Nt(s(m)) : x \Rightarrow \overline{\exists g Nt(s(g)) : x} \exists R}}{\blacksquare Nt(s(m)) : x \Rightarrow \langle \exists g Nt(s(g)) : x \rangle \langle R}} \frac{\overline{Sf : w} \Rightarrow Sf : w}{\blacksquare Nt(s(m)) : x, \langle \exists g Nt(s(g)) \setminus Sf \rangle : z \Rightarrow Sf : (z x)} \setminus L}{\blacksquare Nt(s(m)) : x, \square(\langle \exists g Nt(s(g)) \setminus Sf \rangle) : y \Rightarrow Sf : (z x)\{y/z\} = (\sim y x)} \square L$$

Now we can substitute in the lexical semantics j for *John* (x) and $\sim \lambda A(\text{Pres } (\sim \text{walk } A))$ for *walks* (y) and evaluate:¹⁶

¹⁶ Montague's Intensional Logic assigned nonlogical constants of type τ a denotation in the intension of τ and then interpreted a constant with respect to a world as its extension in that world. By contrast our semantic representation language assigns constants denotations in their own type, so our semantic representations have explicit extensionalizations of intensional constants.

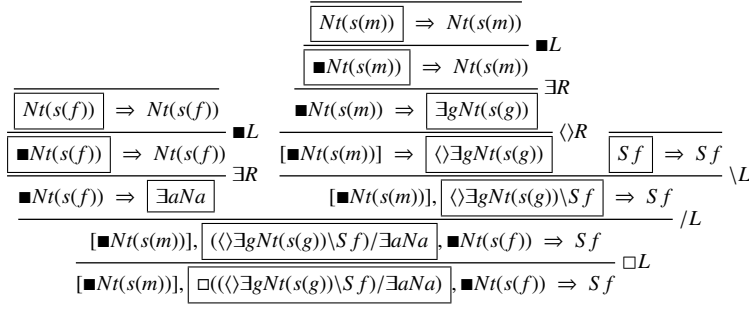


Fig. 11 Derivation for *John loves Mary*

$$\begin{aligned}
(28) \quad & (\sim \lambda A (Pres (\sim walk A)) j) = \\
& (\lambda A (Pres (\sim walk A)) j) = \\
& (Pres (\sim walk j))
\end{aligned}$$

(As we have said, this elucidation is not exactly how CatLog2 extracts semantics; CatLog2 uses unification and instantiation of metavariables to deliver in a single pass the unevaluated semantics of the upwards and downward phases, and then normalises.)

By way of a second example, the following is a simple transitive sentence:

$$(29) \quad [\mathbf{john}] + \mathbf{loves} + [\mathbf{mary}] : S f$$

Lexical lookup yields:

$$(30) \quad [\blacksquare Nt(s(m)) : j], \square(\langle \exists g Nt(s(g)) \rangle \backslash S f) / \exists a Na : \hat{\lambda} A \lambda B (Pres (\sim love A) B), \blacksquare Nt(s(f)) : m \Rightarrow S f$$

There is the derivation given in Figure 11. Reading upwards from the endsequent, the first inference removes the intensionality modality from the transitive verb, and then over left selects the object to analyse as the argument of the transitive verb; this is done by existential right instantiating the agreement feature to third person singular feminine, followed by (semantically inactive) intensionality modality left. The right hand branch is the same as for example (15) after the first inference. All this delivers semantics:

$$(31) \quad (Pres (\sim love m) j)$$

The next example has a subordinate clause:

$$(32) \quad [\mathbf{john}] + \mathbf{thinks} + [\mathbf{mary}] + \mathbf{walks} : S f$$

Lexical lookup yields the following; note that the propositional attitude verb is polymorphic with respect to a complementised or uncomplementised sentential argument, expressed with a semantically inactive additive disjunction:

$$(33) \quad [\blacksquare Nt(s(m)) : j], \square(\langle \exists g Nt(s(g)) \rangle \backslash S f) / (C P that \sqcup \square S f) : \hat{\lambda} A \lambda B (Pres (\sim think A) B), [\blacksquare Nt(s(f)) : m], \square(\langle \exists g Nt(s(g)) \rangle \backslash S f) : \hat{\lambda} C (Pres (\sim walk C)) \Rightarrow S f$$

This has the derivation given in Figure 12. Reading bottom-up, following elimination of the intensionality modality of the propositional attitude verb, over left partitions in such a way as to supply the subordinate clause as the propositional argument. Again, the righthand subtree is the same as for example (15) after the first inference. In the lefthand subtree semantically inactive additive conjunction right selects the modalised uncomplementized sentence type. The succedent modality is removed, this removal being licensed by the fact that all the antecedent types are modalised, and the remaining derivation is also like that for example (15). The derivation delivers semantics:

$$\begin{array}{c}
\boxed{Nt(s(f))} \Rightarrow Nt(s(f)) \\
\boxed{\blacksquare Nt(s(f))} \Rightarrow Nt(s(f)) \quad \blacksquare L \\
\boxed{\blacksquare Nt(s(f))} \Rightarrow \boxed{\exists g Nt(s(g))} \quad \exists R \\
\boxed{[\blacksquare Nt(s(f))] \Rightarrow \langle \exists g Nt(s(g)) \rangle} \quad \langle R \\
\boxed{[\blacksquare Nt(s(f))], \langle \exists g Nt(s(g)) \rangle \backslash S f} \Rightarrow S f \quad \backslash L \\
\boxed{[\blacksquare Nt(s(f))], \langle \exists g Nt(s(g)) \rangle \backslash S f} \Rightarrow S f \quad \square L \\
\boxed{[\blacksquare Nt(s(f))], \square(\langle \exists g Nt(s(g)) \rangle \backslash S f)} \Rightarrow \square S f \quad \square R \\
\boxed{[\blacksquare Nt(s(f))], \square(\langle \exists g Nt(s(g)) \rangle \backslash S f)} \Rightarrow \boxed{C P t h a t \sqcup \square S f} \quad \sqcup R \\
\boxed{[\blacksquare Nt(s(m))], \langle \exists g Nt(s(g)) \rangle \backslash S f} \Rightarrow S f \quad \blacksquare L \\
\boxed{\blacksquare Nt(s(m))} \Rightarrow Nt(s(m)) \quad \blacksquare L \\
\boxed{\blacksquare Nt(s(m))} \Rightarrow \boxed{\exists g Nt(s(g))} \quad \exists R \\
\boxed{[\blacksquare Nt(s(m))] \Rightarrow \langle \exists g Nt(s(g)) \rangle} \quad \langle R \\
\boxed{[\blacksquare Nt(s(m))], \langle \exists g Nt(s(g)) \rangle \backslash S f} \Rightarrow S f \quad \backslash L \\
\boxed{[\blacksquare Nt(s(m))], \langle \exists g Nt(s(g)) \rangle \backslash S f} \Rightarrow S f \quad / L \\
\boxed{[\blacksquare Nt(s(m))], \langle \exists g Nt(s(g)) \rangle \backslash S f} / (C P t h a t \sqcup \square S f), [\blacksquare Nt(s(f))], \square(\langle \exists g Nt(s(g)) \rangle \backslash S f)} \Rightarrow S f \\
\boxed{[\blacksquare Nt(s(m))], \square(\langle \exists g Nt(s(g)) \rangle \backslash S f) / (C P t h a t \sqcup \square S f)}, [\blacksquare Nt(s(f))], \square(\langle \exists g Nt(s(g)) \rangle \backslash S f)} \Rightarrow S f \quad \square L
\end{array}$$

Fig. 12 Derivation for *John thinks Mary walks*

$$\begin{array}{c}
\boxed{Nt(s(m))} \Rightarrow Nt(s(m)) \\
\boxed{\blacksquare Nt(s(m))} \Rightarrow Nt(s(m)) \quad \blacksquare L \\
\boxed{\blacksquare Nt(s(m))} \Rightarrow \boxed{\exists a N a} \quad \exists R \\
\boxed{[\blacksquare Nt(s(m))] \Rightarrow \exists a N a} \quad \exists R \\
\boxed{[\blacksquare Nt(s(m))], \exists a N a} \Rightarrow \boxed{\exists a N a \bullet \exists a N a} \quad \bullet R \\
\boxed{Nt(s(n))} \Rightarrow Nt(s(n)) \\
\boxed{Nt(s(n)) \& C N s(n)} \Rightarrow Nt(s(n)) \quad \& L \\
\boxed{\square(Nt(s(n)) \& C N s(n))} \Rightarrow Nt(s(n)) \quad \square L \\
\boxed{\square(Nt(s(n)) \& C N s(n))} \Rightarrow \boxed{\exists a N a} \quad \exists R \\
\boxed{[\square(Nt(s(n)) \& C N s(n))] \Rightarrow \exists a N a} \quad \exists R \\
\boxed{[\blacksquare Nt(s(f))], \langle \exists g Nt(s(g)) \rangle \backslash S f} \Rightarrow S f \quad \backslash L \\
\boxed{[\blacksquare Nt(s(f))], \langle \exists g Nt(s(g)) \rangle \backslash S f} \Rightarrow S f \quad / L \\
\boxed{[\blacksquare Nt(s(f))], \langle \exists g Nt(s(g)) \rangle \backslash S f} / (\exists a N a \bullet \exists a N a), [\blacksquare Nt(s(m))], \square(Nt(s(n)) \& C N s(n))} \Rightarrow S f \\
\boxed{[\blacksquare Nt(s(f))], \square(\langle \exists g Nt(s(g)) \rangle \backslash S f) / (\exists a N a \bullet \exists a N a)}, [\blacksquare Nt(s(m))], \square(Nt(s(n)) \& C N s(n))} \Rightarrow S f \quad \square L
\end{array}$$

Fig. 13 Derivation for *Mary buys John coffee*(34) $(Pres ((\sim think (\sim Pres (\sim walk m))) j))$

The following example involves a ditransitive verb:

(35) $[\mathbf{mary}] + \mathbf{buys} + \mathbf{john} + \mathbf{coffee} : S f$

Lexical lookup is as follows; note the use of (continuous) product (multiplicative conjunction) for the ditransitive verb, and the use of additive conjunction for the polymorphism of the mass noun *coffee* which can appear either as a bare nominal or with an article:

(36) $[\blacksquare Nt(s(f)) : m], \square(\langle \exists g Nt(s(g)) \rangle \backslash S f) / (\exists a N a \bullet \exists a N a) : \sim \lambda A \lambda B (Pres ((\sim buy \pi_1 A) \pi_2 A) B), \blacksquare Nt(s(m)) : j, \square(Nt(s(n)) \& C N s(n)) : \sim ((gen \sim coffee), \sim coffee) \Rightarrow S f$

There is the derivation given in Figure 13. After removal of the outer modality of the ditransitive verb, the partitioning of over left selects the two objects as the verb's product argument, partitioned in turn by product right. The indirect object *John* is analysed by existential right and inactive modality left inferences; the direct object *coffee* is analysed by existential right and (active) modality left inferences followed by selection of the bare noun type by additive conjunction left. The rightmost subtree is as usual for an intransitive sentence. This delivers semantics as follows in which a 'generic' operator applies to *coffee*:

(37) $(Pres (((\sim buy j) (gen \sim coffee)) m))$

The next example includes a definite article:

(38) $[\mathbf{the} + \mathbf{man}] + \mathbf{walks} : S f$

$$\begin{array}{c}
\boxed{CNs(m)} \Rightarrow CNs(m) \\
\boxed{\square CNs(m)} \Rightarrow CNs(m) \quad \square L \\
\boxed{Nt(s(m))} \Rightarrow Nt(s(m)) \\
\boxed{\square Nt(s(m))} \Rightarrow Nt(s(m)) \quad \square L \\
\boxed{Nt(s(m))/CNs(m)}, \square CNs(m) \Rightarrow Nt(s(m)) \quad \forall L \\
\boxed{\forall n(Nt(n)/CNn)}, \square CNs(m) \Rightarrow Nt(s(m)) \quad \forall L \\
\boxed{\blacksquare \forall n(Nt(n)/CNn)}, \square CNs(m) \Rightarrow Nt(s(m)) \quad \blacksquare L \\
\boxed{\blacksquare \forall n(Nt(n)/CNn)}, \square CNs(m) \Rightarrow \boxed{\exists g Nt(s(g))} \quad \exists R \\
\boxed{\blacksquare \forall n(Nt(n)/CNn)}, \square CNs(m) \Rightarrow \boxed{\langle \rangle \exists g Nt(s(g))} \quad \langle \rangle R \\
\boxed{Sf} \Rightarrow Sf \quad \backslash L \\
\boxed{\blacksquare \forall n(Nt(n)/CNn)}, \square CNs(m), \boxed{\langle \rangle \exists g Nt(s(g)) \setminus Sf} \Rightarrow Sf \quad \backslash L \\
\boxed{\blacksquare \forall n(Nt(n)/CNn)}, \square CNs(m), \boxed{\langle \rangle \exists g Nt(s(g)) \setminus Sf} \Rightarrow Sf \quad \square L
\end{array}$$

Fig. 14 Derivation for *The man walks*

We treat the definite article simply as an iota operator which returns the unique individual in the context of discourse satisfying its common noun argument (Carpenter 1997[6]); this unicity is presupposed by the use of the definite. Lexical lookup yields the semantically labelled sequent:

$$(39) [\blacksquare \forall n(Nt(n)/CNn) : \iota, \square CNs(m) : man], \square \langle \rangle \exists g Nt(s(g)) \setminus Sf : \hat{\lambda}A(Pres \tilde{c} walk A) \Rightarrow Sf$$

There is the derivation given in Figure 14. This is like the derivation of an intransitive sentence before, but with the analysis of the definite noun phrase subject at the top left. The derivation delivers semantics:

$$(40) (Pres \tilde{c} walk (\iota \tilde{c} man))$$

The next two examples have adverbial and adnominal prepositional modification respectively. We consider the adverbial case first:

$$(41) [\mathbf{john}] + \mathbf{walks} + \mathbf{from} + \mathbf{edinburgh} : Sf$$

Lexical lookup inserts a single value-polymorphic prepositional type, which uses semantically active additive conjunction:

$$(42) [\blacksquare Nt(s(m)) : j], \square \langle \rangle \exists g Nt(s(g)) \setminus Sf : \hat{\lambda}A(Pres \tilde{c} walk A), \square (\forall a \forall f (\langle \rangle Na \setminus Sf) \setminus (\langle \rangle Na \setminus Sf)) \& \forall n(CNn \setminus CNn) / \exists b Nb : \hat{\lambda}B(\tilde{c} fromadv B), (\tilde{c} fromadn B), \blacksquare Nt(s(n)) : e \Rightarrow Sf$$

There is the derivation given in Figure 15. After elimination of the outer modality of the preposition, over left selects as the prepositional argument the prepositional object, which is analysed in the leftmost subtree. In the sister subtree additive conjunction left selects the adverbial type for the prepositional phrase and for all left instantiates the subject agreement and verb form features to third person singular masculine, and finite. Following under left, in the middle subtree *walks* is analysed as the intransitive verb second argument of the adverbial preposition; note the analysis of the higher-order type by the under right rule, which lowers the conclusion succedent hypothetical subtype into the premise antecedent. The rightmost subtree is an intransitive sentence case again. All this delivers the semantics:

$$(43) ((\tilde{c} fromadv e) \lambda B(Pres \tilde{c} walk B)) j$$

$$\begin{array}{c}
\frac{\frac{\frac{Nt(s(A)) \Rightarrow Nt(s(A))}{\forall g Nt(s(g)) \Rightarrow Nt(s(A))} \forall L}{\blacksquare \forall g Nt(s(g)) \Rightarrow Nt(s(A))} \blacksquare L}{\blacksquare \forall g Nt(s(g)) \Rightarrow \exists a Na} \exists R \\
\frac{\frac{Nt(s(m)) \Rightarrow Nt(s(m))}{\blacksquare Nt(s(m)) \Rightarrow Nt(s(m))} \blacksquare L}{\blacksquare Nt(s(m)) \Rightarrow \exists g Nt(s(g))} \exists R}{\blacksquare Nt(s(m)) \Rightarrow \langle \exists g Nt(s(g)) \rangle} \langle R \\
\frac{\blacksquare \forall g Nt(s(g)) \Rightarrow \exists a Na \oplus (\exists g((CNg/CNg) \sqcup (CNg \setminus CNg)) - I)}{\blacksquare Nt(s(m)), \langle \exists g Nt(s(g)) \rangle \setminus S f \Rightarrow S f} \oplus R \quad \frac{\blacksquare Nt(s(m)), \langle \exists g Nt(s(g)) \rangle \setminus S f \Rightarrow S f}{\blacksquare Nt(s(m)), \langle \exists g Nt(s(g)) \rangle \setminus S f \Rightarrow S f} \setminus L}{\blacksquare Nt(s(m)), \langle \exists g Nt(s(g)) \rangle \setminus S f / (\exists a Na \oplus (\exists g((CNg/CNg) \sqcup (CNg \setminus CNg)) - I)), \blacksquare \forall g Nt(s(g)) \Rightarrow S f} / L \\
\blacksquare Nt(s(m)), \langle \exists g Nt(s(g)) \rangle \setminus S f / (\exists a Na \oplus (\exists g((CNg/CNg) \sqcup (CNg \setminus CNg)) - I)), \blacksquare \forall g Nt(s(g)) \Rightarrow S f} \blacksquare L \\
\blacksquare Nt(s(m)), \langle \exists g Nt(s(g)) \rangle \setminus S f / (\exists a Na \oplus (\exists g((CNg/CNg) \sqcup (CNg \setminus CNg)) - I)), \blacksquare \forall g Nt(s(g)) \Rightarrow S f} \blacksquare L
\end{array}$$

Fig. 17 Derivation for *Tully is Cicero*

The last two initial examples involve the copula with nominal and (intersective) adjectival complementation respectively. We consider first the nominal case:

(47) **[tully]+is+cicero** : Sf

Lexical lookup inserts a single argument-polymorphic copula type, which uses both semantically active and semantically inactive additive disjunction:¹⁷

(48) $\blacksquare Nt(s(m)) : t, \blacksquare (\langle \exists g Nt(s(g)) \rangle \setminus S f) / (\exists a Na \oplus (\exists g((CNg/CNg) \sqcup (CNg \setminus CNg)) - I)) : \lambda A \lambda B (Pres (A \rightarrow C.[B = C]; D.((D \lambda E[E = B]) B))), \blacksquare \forall g Nt(s(g)) : cicero \Rightarrow Sf$

There is the derivation given in Figure 17. After elimination of the outer copula modality the copula is applied to its nominal complement. Additive disjunction right selects the first, nominal, disjunct. The derivation delivers semantics:

(49) $(Pres [t = c])$

The (intersective) adjectival case is:

(50) **[tully]+is+humanist** : Sf

Lexical lookup yields:

(51) $\blacksquare Nt(s(m)) : t, \blacksquare (\langle \exists g Nt(s(g)) \rangle \setminus S f) / (\exists a Na \oplus (\exists g((CNg/CNg) \sqcup (CNg \setminus CNg)) - I)) : \lambda A \lambda B (Pres (A \rightarrow C.[B = C]; D.((D \lambda E[E = B]) B))), \square \forall n(CNn/CNn) : \sim \lambda F \lambda G [(F G) \wedge (\sim humanist G)] \Rightarrow Sf$

There is the derivation given in Figure 18. After elimination of its outer modality, the copula is applied to its adjectival complement. Semantically active additive disjunction right selects the second disjunct. The difference right rule checks that the antecedent is not empty, but this is not displayed. Exists right substitutes the existentially quantified variable for a metavariable A and semantically inactive additive disjunction right then selects the adjectival disjunct. The following semantics is delivered:

(52) $(Pres (\sim humanist t))$

4 Routes we do not take

Szaboloczi (1983[58]) and Steedman (1987[56]) aim to account for parasitic gaps in combinatory categorial grammar (CCG) by means of the combinator **S** such that $\mathbf{S} x y z = (x z) (y z)$, for example positing a combinatory schema:

¹⁷ The difference operator (Morrill and Valentin 2014[42]) for linguistic exceptions is also used. It involves negation as failure, which cannot easily be displayed. We do not dwell on this operator here.

and Vasishth (2002[17]) propose induction of parasitic gaps in adverbial clauses by a lexicalised gap-duplicating effect in the adverbial head. All contexts allowing parasitic gaps would require a corresponding gap-duplicating lexical ambiguity. The appeal to lexical ambiguity in lexical grammar formalisms is as frequent as it is untenable. Every ambiguity of every item doubles the lexical insertion search space. And in the case in hand there is to our knowledge no independent evidence, such as difference in meaning, for lexical ambiguity underlying parasitic extraction. We continue on the assumption that it is indeed a syntactic phenomenon.

5 Relativisation

Our account of relativisation rests on the lexical projection of islands by argument bracketing ($\langle \rangle$) and value antibracketing ($[]^{-1}$), and a single relative pronoun type of overall shape $R/(\langle \rangle N \square !N) \setminus S$ for both subject and object relativisation. Note that the two operands of the hypothetical gap subtype are conjoined because in analysis of the body of relative clauses the higher order succedent argument of form $\langle \rangle N \square !N$ is lowered into the antecedent according to the deduction theorem, where the conjunction left rules select the first or second operands to produce the subject/object relativisation alternation.

In subject relativisation $\langle \rangle N$ is selected by conjunction left, and satisfies the (bracketed) subject valency.

In object relativisation $!N$ is selected by conjunction left; when the $!L$ rule is applied to $!N$, the hypothetical subtype N moves into the stoup, from whence it can move by $!P$ to any (nonisland) position in its zone, realising nonparasitic extraction. However, in addition it can be copied by $!C$ to the stoup of a newly created weak island domain, realising parasitic extraction. The N in the outer stoup can be copied by $!C$ repeatedly, capturing that there may be parasitic gaps in any number of local weak islands; at the end of this process it moves by $!P$ to a host position in its zone. The N in an inner stoup can also be copied by $!C$ to the stoup of any number of newly created weak subislands, and so on recursively, capturing that parasitic gaps can also be hosts to further parasitic gaps; finally the stoup contents are copied by $!P$ to extraction sites in their zone.

In this section we analyse examples illustrating the account of relativisation. The first example is a minimal subject relativisation; note that the relative clause is doubly bracketed, corresponding to the fact that relative clauses are strong islands:¹⁹

(54) **man+**[[**that+walks**]] : $CNs(m)$

Lexical lookup yields the following, where there is semantically inactive additive conjunction of the hypothetical subtypes $\langle \rangle N$ for subject relativisation and $! \blacksquare N$ for object relativisation; the (semantically inactive) modality on the object gap subtype is to permit object relativisation from embedded modal/intensional domains:²⁰

¹⁹ As we will see relative clauses themselves, being doubly bracketed, will not allow parasitic gaps.

²⁰ The body of the relative clause is marked as a (semantically inactive) modal domain in order to make it a scope island. This account of relative clauses as scope islands operates on essentially the same lines as the capture of Principle A by modalities (see Morrill 1990[35]). Thus where, say, *everyone* has a type $\blacksquare((S \uparrow N) \downarrow S)$ the unmodalised hypothetical subtype N cannot be bound outside the modal domain of the body of a relative clause in which *everyone* occurs. To make the $\blacksquare R$ inference for the body of the relative clause, every antecedent type must be modalised. Thus if the body contains for example *everyone* where it is $\blacksquare((S \uparrow N) \downarrow S)$ this can scope clause-internally by decomposing after the $\blacksquare R$ inference, but if we try to decompose it before to make it scope clause-externally it will leave the hypothetical subtype N which is not modalised and so the $\blacksquare R$ inference is blocked.

subtree selects the body of the relative clause as the semantically inactive modalised higher-order subject-and-object polymorphic relative pronoun argument type. After (semantically inactive) modality right, licensed since the antecedent types are modalised, the conjoined hypothetical subject is lowered by $\backslash R$ into the antecedent. Observe how $\sqcap L$ selects the object relativisation hypothetical subtype $!\blacksquare Nt(s(m))$ and how this subsequently percolates in the stoup, passing in particular into the minor premise branch of the upper $/L$ inference and hence satisfying the object valency of the transitive verb *love*; subject and intransitive verb phrase are analysed as usual. This delivers the required semantics:

$$(59) (Pres \ (\sim walk \ (\iota \lambda D[(\sim man \ D) \wedge (Pres \ ((\sim love \ D) \ m))])))$$

An example with longer-distance object relativisation, in the context of an entire sentence, is:

$$(60) [\mathbf{the+man+}[[\mathbf{that+}[\mathbf{john+}[\mathbf{thinks+}[\mathbf{mary+}[\mathbf{loves}]]]+\mathbf{walks}]] : S f$$

Lexical lookup yields the following; note how the propositional attitude verb is polymorphic between a complementised and an uncomplementised sentential argument, expressed with a semantically inactive additive disjunction:

$$(61) [\blacksquare \forall n(Nt(n)/CNn) : \iota, \sqcap CNs(m) : man, [[\blacksquare \forall n([\]^{-1}[\]^{-1}(CNn \setminus CNn) / \blacksquare((\langle \rangle Nt(n) \sqcap !\blacksquare Nt(n)) \setminus S f)) : \lambda A \lambda B \lambda C[(B \ C) \wedge (A \ C)], [\blacksquare Nt(s(m)) : j], \sqcap((\langle \rangle \exists g Nt(s(g)) \setminus S f) / (C P that \sqcup \sqcap S f)) : \wedge \lambda D \lambda E (Pres \ ((\sim think \ D) \ E)), [\blacksquare Nt(s(f)) : m], \sqcap((\langle \rangle \exists g Nt(s(g)) \setminus S f) / \exists a Na) : \wedge \lambda F \lambda G (Pres \ ((\sim love \ F) \ G))], \sqcap(\langle \rangle \exists g Nt(s(g)) \setminus S f) : \wedge \lambda H (Pres \ (\sim walk \ H)) \Rightarrow S f$$

There is the derivation given in Figure 21. Inference up as far as $\textcircled{1}$ brings us to analysis of the complex common noun phrase in the lefthand subtree. The following preparation of the relative pronoun and double bracket cancellation of its value are as usual. After modality right and under right on the relative pronoun higher-order argument, $\sqcap L$ selects the object relativisation hypothetical subtype and $!L$ moves this into the stoup. In the stoup it percolates to the subordinate clause, (observe how $\sqcup R$ selects the uncomplementised sentential argument type of the propositional attitude verb) and there $!P$ moves it into position to satisfy the embedded clause object valency. This delivers the correct semantics:

$$(62) (Pres \ (\sim walk \ (\iota \lambda D[(\sim man \ D) \wedge (Pres \ ((\sim think \ (\sim (Pres \ ((\sim love \ D) \ m)) \ j))])))$$

There follows an example of medial object relativisation (the gap is in a nonperipheral position left of the adverb):

$$(63) \mathbf{man+}[[\mathbf{that+}[\mathbf{mary+}[\mathbf{likes+}[\mathbf{today}]]]] : CNs(m)$$

Appropriate lexical lookup yields:

$$(64) \sqcap CNs(m) : man, [[\blacksquare \forall n([\]^{-1}[\]^{-1}(CNn \setminus CNn) / \blacksquare((\langle \rangle Nt(n) \sqcap !\blacksquare Nt(n)) \setminus S f)) : \lambda A \lambda B \lambda C[(B \ C) \wedge (A \ C)], [\blacksquare Nt(s(f)) : m], \sqcap((\langle \rangle \exists g Nt(s(g)) \setminus S f) / \exists a Na) : \wedge \lambda D \lambda E (Pres \ ((\sim like \ D) \ E)), \sqcap \forall a \forall f((\langle \rangle Na \setminus S f) \setminus (\langle \rangle Na \setminus S f)) : \wedge \lambda F \lambda G (\sim today \ (F \ G))] \Rightarrow CNs(m)$$

There is the derivation in Figure 22. Analysis of the complex common noun phrase begins at the lefthand subtree $\textcircled{1}$. After modality right and conditionalisation of the conjoined hypothetical subtype, additive conjunction left applies to this latter to select the object relativisation subtype, which then moves into the stoup. After preparation of the adverb the stoup contents pass into its argument subbranch. Note how the object relativisation hypothetical gap subtype percolates in the stoup to satisfy the transitive verb object valency. The semantics delivered is:

(65) $\lambda C[(\sim man C) \wedge (\sim today (Pres ((\sim like C) m)))]$

As we remarked at the beginning of Section 3 subjects are weak islands (the Subject Condition of Chomsky 1973[7]); accordingly in our CatLog2 fragment there is no derivation of simple relativisation from a subject such as:

(66) **man+[[that+[the+friends+of]+walk]]** : $CNs(m)$

This is because *walk* projects brackets around its subject, but the permutation of the ! hypothetical gap subtype issued by the relative pronoun is limited to its zone and cannot penetrate a bracketed subzone. Roughly, the derivation blocks at * in:

$$(67) \frac{\frac{[N/CN, CN/PP, PP/N, N], N \setminus S \Rightarrow S}{N; [N/CN, CN/PP, PP/N], N \setminus S \Rightarrow S} *!P}{!N, [N/CN, CN/PP, PP/N], N \setminus S \Rightarrow S} !L}{[N/CN, CN/PP, PP/N], N \setminus S \Rightarrow !N \setminus S} \setminus R$$

However, a weak island ‘parasitic’ gap can be licensed by a host gap:

(68) **man+[[that+the+friends+of+admire]]** : $CNs(m)$

Lexical lookup yields:²¹

$$(69) \square CNs(m) : man, [[\blacksquare \forall n([\]^{-1}[\]^{-1}(CNn \setminus CNn) / \blacksquare ((\langle \rangle Nt(n) \square ! \blacksquare Nt(n)) \setminus S f)) : \lambda \lambda \lambda B \lambda C [(B C) \wedge (A C)], \blacksquare \forall n(Nt(n) / CNn) : \iota, \square (CNp / PPof) : friends, \square ((\forall n(CNn \setminus CNn) / \blacksquare \exists b N b) \& (PPof / \exists a N a)) : \sim of, \lambda DD), \square ((\langle \rangle (\exists a N a - \exists g N t(s(g))) \setminus S f) / \exists a N a) : \hat{\lambda} E \lambda F (Pres ((\sim admire E) F))] \Rightarrow CNs(m)$$

There is the derivation given in Figure 23, where the use of contraction !C, involving brackets and stoups, corresponds to generating the parasitic gap. The object relativisation hypothetical subtype moves into the stoup at depth seven in the lefthand subtree (before this the analysis is standard). Contraction then applies copying the gap type into the stoup of a newly created bracketed domain around the subordinate subject. Applications of !P then move the stoup contents into the object position of *admire* (host) and *of* (parasitic). This delivers the following semantics in which the gap variable is multiply bound:

(70) $\lambda C[(\sim man C) \wedge (Pres ((\sim admire C) (\iota (\sim friends C)))]$

Parasitic extraction from strong islands such as coordinate structures is not acceptable:

(71) *that_i Mary showed [[John and the friends of *t_i*]] to *t_i*

This is successfully blocked because strong islands are doubly bracketed. Although contraction could apply twice to introduce two bracketings, a copy of the hypothetical gap subtype would remain trapped in the stoup at the intermediate level of bracketing, blocking overall derivation. Likewise, as we remarked in footnote 19, parasitic extraction is not possible from relative clauses themselves, for the same reason: a superfluous gap subtype would remain trapped in between the double brackets required for the strong island.

A parasitic gap can also appear in an adverbial weak island:

(72) **paper+[[that+[john]+filed+without+reading]]** : $CNs(n)$

Lexical lookup for this example yields:

²¹ We gloss over the use of ‘difference’ here to mark non-third person singular; its use depends on *absence* of derivability (negation as failure) which of course cannot easily be displayed.

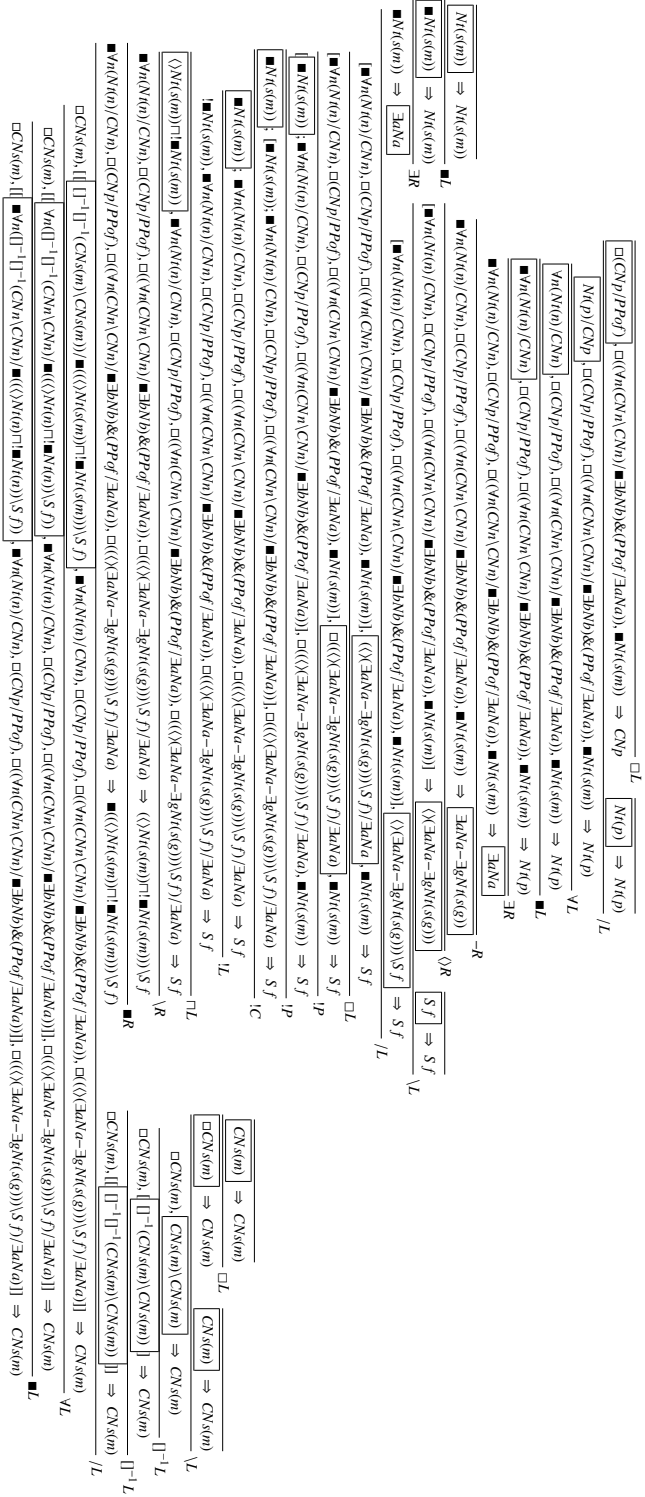


Fig. 23 Derivation of man that the friends of admire

$$(73) \quad \square CNs(n) : \text{paper}, [[\blacksquare \forall n([\]^{-1}[\]^{-1}(CNn \setminus CNn) / \blacksquare((\langle \rangle Nt(n) \sqcap ! \blacksquare Nt(n)) \setminus S f)) : \\ \lambda A \lambda B \lambda C[(B C) \wedge (A C)], [\blacksquare Nt(s(m)) : j], \square((\langle \rangle \exists g Nt(s(g)) \setminus S f) / \exists a Na) : \\ \wedge \lambda D \lambda E(\text{Past}(\langle \text{file } D \rangle E)), \blacksquare \forall a \forall f([\]^{-1}((\langle \rangle Na \setminus S f) \setminus (\langle \rangle Na \setminus S f)) / (\langle \rangle Na \setminus S psp)) : \\ \lambda F \lambda G \lambda H[(G H) \wedge \neg(F H)], \square((\langle \rangle \exists a Na \setminus S psp) / \exists a Na) : \wedge \lambda I \lambda J(\langle \text{read } I \rangle J)]] \Rightarrow CNs(n)$$

There is the derivation given in Figure 24. This time at depth eight contraction copies the host stoup gap into the stoup of a newly created bracketed domain around the subordinate adverbial phrase. This delivers semantics:

$$(74) \quad \lambda C[\langle \text{paper } C \rangle \wedge [(\text{Past}(\langle \text{file } C \rangle j)) \wedge \neg(\langle \text{read } C \rangle j)]]$$

In our final relativisation example the host gap licences two parasitic gaps, in the subject noun phrase and in an adverbial phrase:

$$(75) \quad \text{paper} + [[\text{that} + \text{the} + \text{editor} + \text{of} + \text{filed} + \text{without} + \text{reading}]] : CNs(n)$$

Lexical lookup yields:

$$(76) \quad \square CNs(n) : \text{paper}, [[\blacksquare \forall n([\]^{-1}[\]^{-1}(CNn \setminus CNn) / \blacksquare((\langle \rangle Nt(n) \sqcap ! \blacksquare Nt(n)) \setminus S f)) : \\ \lambda A \lambda B \lambda C[(B C) \wedge (A C)], \blacksquare \forall n(Nt(n) / CNn) : \iota, \square(\forall g CNs(g) / PPof) : \text{editor}, \\ \square((\forall n(CNn \setminus CNn) / \blacksquare \exists b Nb) \& (PPof / \exists a Na)) : \wedge \langle \text{of}, \lambda DD), \\ \square((\langle \rangle \exists g Nt(s(g)) \setminus S f) / \exists a Na) : \wedge \lambda E \lambda F(\text{Past}(\langle \text{file } E \rangle F)), \\ \blacksquare \forall a \forall f([\]^{-1}((\langle \rangle Na \setminus S f) \setminus (\langle \rangle Na \setminus S f)) / (\langle \rangle Na \setminus S psp)) : \lambda G \lambda H \lambda I[(H I) \wedge \neg(G I)], \\ \square((\langle \rangle \exists a Na \setminus S psp) / \exists a Na) : \wedge \lambda J \lambda K(\langle \text{read } J \rangle K)]] \Rightarrow CNs(n)$$

There is the derivation fragmented into Figures 25 and 26. There are two applications of contraction, at depth nine and ten, projecting brackets around the subordinate subject and adverbial phrase and giving rise to two parasitic gaps. This delivers the correct semantics:

$$(77) \quad \lambda C[\langle \text{paper } C \rangle \wedge [(\text{Past}(\langle \text{file } C \rangle (\iota \langle \text{editor } C \rangle))) \wedge \neg(\langle \text{read } C \rangle (\iota \langle \text{editor } C \rangle))]]$$

6 Possible exceptions

In this section we address three kinds of possible exceptions to the account given here, along the lines anticipated in the introduction.

First, there are examples in which there appears to be a parasitic gap which is not in an island. The following is example (8a) from Postal (1993[53]):

$$(78) \quad \text{man who}_i \text{ Mary convinced } t_i \text{ that John wanted to visit } t_i$$

And an anonymous referee points out:

$$(79) \quad \text{people whom}_i \text{ you sent pictures of } t_i \text{ to } t_i$$

In respect of such examples we have suggested that although there *seems* to be no island, there *could* be one. This is effected as follows for (78). Instead of a type of the form $((N \setminus S) / CP) / N$ for *convince* we assume $((N \setminus S) / CP) / (N \sqcup \langle \rangle N)$ where the semantically inactive additive disjunction disjunct N will be selected ordinarily, and $\langle \rangle N$ when there is parasitic extraction, as in (78). Similarly for (79) we assume for *picture* type $CN / (PP \sqcup \langle \rangle PP)$ where the second disjunct projects the brackets of a weak island.²² Thus in examples such as the following the semantically inactive additive disjunction inference for *convince* of type $((N \setminus S) / CP) / (N \sqcup \langle \rangle N)$ will select N :

²² The argument pattern $X \sqcup \langle \rangle X$ is a general mechanism for an argument optional island X . Likewise the dual value pattern $X \sqcap [\]^{-1} X$ is a general mechanism for a value optional island X . We could define synthetic connectives $(\langle \rangle) \sqcup X = X \sqcup \langle \rangle X$, $(\langle \rangle) \sqcap X = X \sqcap \langle \rangle X$, $([\]^{-1}) \sqcup X = X \sqcup [\]^{-1} X$, $([\]^{-1}) \sqcap X = X \sqcap [\]^{-1} X$. Then for example we would have the abbreviated forms: *convince*: $((N \setminus S) / CP) / (\langle \rangle) \sqcup N$ and *picture*: $CN / (\langle \rangle) \sqcup PP$.

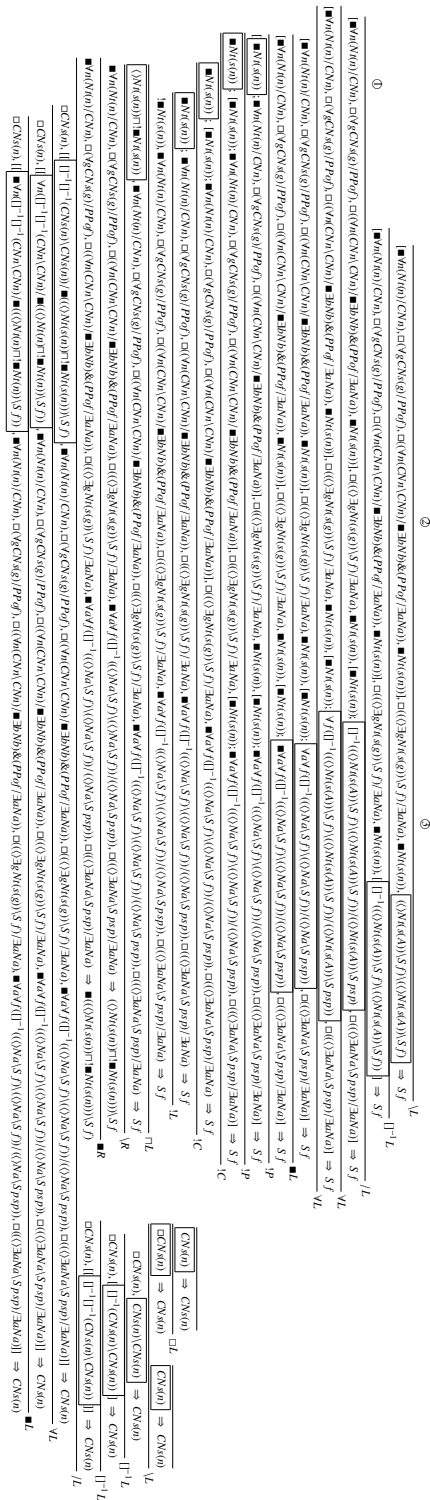


Fig. 26 Main derivation for paper that the editor of filed without reading

- (80) a. man who_{*i*} Mary convinced *t_i* that John wanted to visit Suzy
 b. man who_{*i*} Mary convinced the friends of *t_i* that John wanted to visit Suzy

But for (78) the semantically inactive additive disjunction inference for *convince* of type $((N \setminus S)/CP)/(N \sqcup \langle \rangle N)$ will select $\langle \rangle N$. Similarly for the picture noun case (79).

Secondly, recall the example (4) *man who a friend of laughed* of escape from an indefinite subject island. The standard type for an indefinite would be of the form $((S \uparrow N) \downarrow S)/CN$, but if the type were $((S \uparrow \langle \rangle N) \downarrow S)/CN$ the example (4) would be generated since the $\langle \rangle N$ hypothetical subtype will satisfy the bracketed subject valency without any input brackets blocking the extraction of the object of the preposition. The effect of the standard type is still required to satisfy valencies such as objects which are unbracketed; but we do not require lexical ambiguity: we can collapse the two cases into a single polymorphic indefinite type assignment: $((S \uparrow (N \sqcap \langle \rangle N)) \downarrow S)/CN$.²³

Thirdly, Levine and Hukari (2006[27]) cite an apparent example of ‘symbiotic’ extraction without a host gap:

- (81) people that_{*i*} fans of *t_i* gather from every continent just to listen to *t_i*

In such a case, if the nonspecific *fans* has a type $\langle \rangle N/PP$ the example is generated: the object of *of* acts as host to the parasitic gap in the *just to* adverbial clause. Again we can economise type assignment to bracket inducing and non-bracket inducing *fans* in a single polymorphic type $(N \sqcap \langle \rangle N)/PP$.

Finally, by the same token our response to the second issue predicts the possibility of symbiotic extraction with an indefinite subject host:

- (82) man that_{*i*} a friend of *t_i* went to Paris without e-mailing *t_i*

Thus the possible exceptions to our account receive at least tentative treatment. The data regarding when (parasitic) extraction is or is not possible are complex and perhaps better accounts of the possible exceptions can be found, but we have aimed to show how at least some of this additional complexity is already within the scope of type logical grammar.

7 Conclusion

We have illustrated, by reference to relativisation including islands and parasitic extraction, the thesis that grammar can be reduced to logic. Our type logical categorial grammar incorporating nonassociativity, nonlinearity, and their interaction is, we suggest, mathematically interesting, technically robust, and as empirically adequate and computationally advanced as other proposals.

Acknowledgements Research partially supported by an ICREA Acadèmia 2012, SGR2014-890 (MACDA) of the Generalitat de Catalunya and MINECO project APCOM (TIN2014-57226-P). Many thanks to Stefan Kaufmann for editorial guidance, and to *Linguistics and Philosophy* reviewing which has greatly improved the paper, and to Oriol Valentín for insight and support. All errors are my own.

Appendix A: Semantic Representation Language

Semantic types

Recall the following operations on sets:

²³ Abbreviated, according to the previous footnote, *a*: $((S \uparrow (\langle \rangle N) \downarrow S)/CN)$.

- (83) a. Functional exponentiation: $X^Y =$ the set of all total functions from Y to X
 b. Cartesian product: $X \times Y = \{\langle x, y \rangle \mid x \in X \ \& \ y \in Y\}$
 c. Disjoint union: $X \uplus Y = (\{1\} \times X) \cup (\{2\} \times Y)$

The set \mathcal{T} of *semantic types* of the semantic representation language is defined on the basis of a set δ of *basic semantic types* as follows:

$$(84) \ \mathcal{T} ::= \delta \mid \top \mid \mathcal{T} + \mathcal{T} \mid \mathcal{T} \& \mathcal{T} \mid \mathcal{T} \rightarrow \mathcal{T} \mid \mathbf{MT} \mid \mathbf{LT}$$

A *semantic frame* comprises a family $\{D_\tau\}_{\tau \in \delta}$ of nonempty *basic type domains* and a nonempty set W of worlds. This induces a nonempty *type domain* D_τ for each type τ as follows:

$$(85) \quad \begin{array}{ll} D_\top = \{\emptyset\} & \text{singleton set} \\ D_{\tau_1 + \tau_2} = D_{\tau_1} \uplus D_{\tau_2} & \text{disjoint union} \\ D_{\tau_1 \& \tau_2} = D_{\tau_1} \times D_{\tau_2} & \text{Cartesian product} \\ D_{\tau_1 \rightarrow \tau_2} = D_{\tau_2}^{D_{\tau_1}} & \text{functional exponentiation} \\ D_{\mathbf{MT}} = W \times D_\tau & \text{Cartesian product} \\ D_{\mathbf{LT}} = D_\tau^W & \text{functional exponentiation} \end{array}$$

Semantic terms

The sets Φ_τ of *terms* of type τ for each semantic type τ are defined on the basis of sets C_τ of constants of type τ and denumerably infinite sets V_τ of variables of type τ for each type τ as follows:

$$(86) \quad \begin{array}{ll} \Phi_\tau ::= C_\tau & \text{constants} \\ \Phi_\tau ::= V_\tau & \text{variables} \\ \Phi_\top ::= 0 & \text{dummy} \\ \Phi_\tau ::= \Phi_{\tau_1 + \tau_2} \rightarrow V_{\tau_1} \cdot \Phi_\tau; V_{\tau_2} \cdot \Phi_\tau & \text{case statement} \\ \Phi_{\tau + \tau'} ::= \iota_1 \Phi_\tau & \text{first injection} \\ \Phi_{\tau' + \tau} ::= \iota_2 \Phi_\tau & \text{second injection} \\ \Phi_\tau ::= \pi_1 \Phi_{\tau \& \tau'} & \text{first projection} \\ \Phi_\tau ::= \pi_2 \Phi_{\tau' \& \tau} & \text{second projection} \\ \Phi_{\tau \& \tau'} ::= (\Phi_\tau, \Phi_{\tau'}) & \text{ordered pair formation} \\ \Phi_\tau ::= (\Phi_{\tau' \rightarrow \tau} \Phi_{\tau'}) & \text{functional application} \\ \Phi_{\tau \rightarrow \tau'} ::= \lambda V_\tau \Phi_{\tau'} & \text{functional abstraction} \\ \Phi_\tau ::= \vee \Phi_{\mathbf{LT}} & \text{extensionalisation} \\ \Phi_{\mathbf{LT}} ::= \wedge \Phi_\tau & \text{intensionalisation} \\ \Phi_\tau ::= \cup \Phi_{\mathbf{MT}} & \text{projection} \\ \Phi_{\mathbf{MT}} ::= \cap \Phi_\tau & \text{injection} \end{array}$$

Given a semantic frame, a *valuation* f mapping each constant of type τ into an element of D_τ , an assignment g mapping each variable of type τ into an element of D_τ , and a world $i \in W$, each term ϕ of type τ receives an interpretation $[\phi]^{g,i} \in D_\tau$ as shown in Figure 27; the *update* $g[x := d]$ is $(g - \{(x, g(x))\}) \cup \{(x, d)\}$, i.e. the function which sends x to d and agrees with g elsewhere.

In $x.\phi$, $\lambda x\phi$ or $\wedge\phi$, ϕ is the *scope* of x , λx or \wedge . An occurrence of a variable x in a term is called *free* if and only if it does not fall within the scope of any x . or λx ; otherwise it is *bound* (by the closest x . or λx within the scope of which it falls). The result $\phi\{\psi_1/x_1, \dots, \psi_n/x_n\}$ of substituting terms ψ_1, \dots, ψ_n for variables x_1, \dots, x_n of the same types respectively in a term ϕ is the result of simultaneously replacing by ψ_i every free occurrence of x_i in ϕ . We

$$\begin{aligned}
[a]^{g,i} &= f(a) \text{ for constant } a \in C_\tau \\
[x]^{g,i} &= g(x) \text{ for variable } x \in V_\tau \\
[0]^{g,i} &= \emptyset \\
[\phi \rightarrow x.\psi; y.\chi]^{g,i} &= \begin{cases} [\psi]^{g[x:=d],i} & \text{if } [\phi]^{g,i} = \langle 1, d \rangle \\ [\chi]^{g[y:=d],i} & \text{if } [\phi]^{g,i} = \langle 2, d \rangle \end{cases} \\
[\iota_1 \phi]^{g,i} &= \langle 1, [\phi]^{g,i} \rangle \\
[\iota_2 \phi]^{g,i} &= \langle 2, [\phi]^{g,i} \rangle \\
[\pi_1 \phi]^{g,i} &= \mathbf{fst}([\phi]^{g,i}) \\
[\pi_2 \phi]^{g,i} &= \mathbf{snd}([\phi]^{g,i}) \\
[(\phi, \psi)]^{g,i} &= \langle [\phi]^{g,i}, [\psi]^{g,i} \rangle \\
[(\phi \psi)]^{g,i} &= [\phi]^{g,i}([\psi]^{g,i}) \\
[\lambda x \phi]^{g,i} &= d \mapsto [\phi]^{g[x:=d],i} \\
[\forall \phi]^{g,i} &= [\phi]^{g,i}(i) \\
[\wedge \phi]^{g,i} &= j \mapsto [\phi]^{g,j} \\
[\cup \phi]^{g,i} &= \mathbf{snd}([\phi]^{g,i}) \\
[\cap \phi]^{g,i} &= \langle i, [\phi]^{g,i} \rangle
\end{aligned}$$

Fig. 27 Interpretation of the semantic representation language

$$\begin{aligned}
\phi \rightarrow y.\psi; z.\chi &= \phi \rightarrow x.(\psi(x/y)); z.\chi \\
&\text{if } x \text{ is not free in } \psi \text{ and is free for } y \text{ in } \psi \\
\phi \rightarrow y.\psi; z.\chi &= \phi \rightarrow y.\psi; x.(\chi(x/z)) \\
&\text{if } x \text{ is not free in } \chi \text{ and is free for } z \text{ in } \chi \\
\lambda y \phi &= \lambda x(\phi(x/y)) \\
&\text{if } x \text{ is not free in } \phi \text{ and is free for } y \text{ in } \phi \\
&\alpha\text{-conversion} \\
\iota_1 \phi \rightarrow y.\psi; z.\chi &= \psi(\phi/y) \\
&\text{if } \phi \text{ is free for } y \text{ in } \psi \text{ and modally free for } y \text{ in } \psi \\
\iota_2 \phi \rightarrow y.\psi; z.\chi &= \chi(\phi/z) \\
&\text{if } \phi \text{ is free for } z \text{ in } \chi \text{ and modally free for } z \text{ in } \chi \\
\pi_1(\phi, \psi) &= \phi \\
\pi_2(\phi, \psi) &= \psi \\
(\lambda x \phi \psi) &= \phi\{\psi/x\} \\
&\text{if } \psi \text{ is free for } x \text{ in } \phi, \text{ and modally free for } x \text{ in } \phi \\
\vee \wedge \phi &= \phi \\
\cup \cap \phi &= \phi \\
&\beta\text{-conversion} \\
(\pi_1 \phi, \pi_2 \phi) &= \phi \\
\lambda x(\phi x) &= \phi \\
&\text{if } x \text{ is not free in } \phi \\
\wedge \vee \phi &= \phi \\
&\text{if } \phi \text{ is modally closed} \\
\cap \cup \phi &= \phi \\
&\eta\text{-conversion}
\end{aligned}$$

Fig. 28 Semantic conversion laws

say that ψ is *free for x in ϕ* if and only if no variable in ψ becomes bound in $\phi\{\psi/x\}$. We say that a term is *modally closed* if and only if every occurrence of \vee occurs within the scope of an \wedge . A modally closed term is denotationally invariant across worlds. We say that a term ψ is *modally free for x in ϕ* if and only if either ψ is modally closed, or no free occurrence of x in ϕ is within the scope of an \wedge . The laws of conversion in Figure 28 obtain. The so-called commuting conversions with respect to normalisation for the case statement are omitted.

Appendix B: A lexicon

a : $\blacksquare \forall g(\forall f((S f \uparrow \blacksquare Nt(s(g))) \downarrow S f) / CNs(g)) : \lambda A \lambda B \exists C[(A C) \wedge (B C)]$
admire : $\square((\langle \exists a Na - \exists g Nt(s(g)) \rangle \backslash S f) / \exists a Na) : \lambda A \lambda B (Pres (\sim \text{admire } A) B)$
And : $\blacksquare \forall f(S f / S f) : \lambda A A$
and : $\blacksquare \forall f((? \blacksquare S f \uparrow \square^{-1} \square^{-1} S f) / \blacksquare S f) : (\Phi^{n+} 0 \text{ and})$
and : $\blacksquare \forall a \forall f((? \blacksquare (\langle Na \backslash S f \rangle \uparrow \square^{-1} \square^{-1} (\langle Na \backslash S f \rangle) / \blacksquare (\langle Na \backslash S f \rangle)) : (\Phi^{n+} (s 0) \text{ and})$
and : $\blacksquare \forall a \forall f((? \blacksquare (S f / ! Na) \uparrow \square^{-1} \square^{-1} (S f / ! Na)) / \blacksquare (S f / ! Na)) : (\Phi^{n+} (s 0) \text{ and})$
and : $\blacksquare \forall f((? \blacksquare (S f / \exists a Na) \uparrow \square^{-1} \square^{-1} (S f / \exists a Na)) / \blacksquare (S f / \exists a Na)) : (\Phi^{n+} (s 0) \text{ and})$
and : $\blacksquare \forall w \forall a \forall b \forall f((\blacksquare ((S f \uparrow (\langle \langle Na \backslash S f \rangle \ominus Ww) / Nb)) \uparrow_2 Ww) \uparrow \square^{-1} \square^{-1} ((S f \uparrow (\langle \langle Na \backslash S f \rangle \ominus Ww) / Nb)) \uparrow_2 Ww) / \sim \blacksquare ((S f \uparrow (\langle \langle Na \backslash S f \rangle \ominus Ww) / Nb)) \uparrow_2 Ww) : \lambda A \lambda B \lambda C[(B C) \wedge (A C)]$
and : $\blacksquare \forall f \forall a((? \blacksquare (\langle Na \backslash S f \rangle / \exists b Nb) \uparrow \square^{-1} \square^{-1} (\langle Na \backslash S f \rangle / \exists b Nb)) / \blacksquare (\langle Na \backslash S f \rangle / \exists b Nb)) : (\Phi^{n+} (s (s 0)) \text{ and})$
and : $\blacksquare \forall f \forall a((? \blacksquare (\langle Na \backslash S f \rangle / \exists b Nb) \uparrow \square^{-1} \square^{-1} (\langle Na \backslash S f \rangle / \exists b Nb)) / \blacksquare (\langle Na \backslash S f \rangle / \exists b Nb)) : (\Phi^{n+} (s (s 0)) \text{ and})$
and : $\blacksquare \forall f \forall a((? \blacksquare (\langle Na \backslash S f \rangle / \exists b Nb) \uparrow \square^{-1} \square^{-1} (\langle Na \backslash S f \rangle / \exists b Nb)) / \blacksquare (\langle Na \backslash S f \rangle / \exists b Nb)) : (\Phi^{n+} (s (s 0)) \text{ and})$
and : $\blacksquare \forall f \forall a((\blacksquare (\langle Na \backslash S f \rangle / \exists b Nb) \uparrow \square^{-1} \square^{-1} (\langle Na \backslash S f \rangle / \exists b Nb)) / \blacksquare (\langle Na \backslash S f \rangle / \exists b Nb)) : (\Phi^{n+} (s (s 0)) \text{ and})$
and : $\blacksquare \forall f \forall a((\blacksquare (\langle Na \backslash S f \rangle / \exists b Nb) \uparrow \square^{-1} \square^{-1} (\langle Na \backslash S f \rangle / \exists b Nb)) / \blacksquare (\langle Na \backslash S f \rangle / \exists b Nb)) : (\Phi^{n+} (s (s 0)) \text{ and})$
and : $\blacksquare \forall f \forall a((\blacksquare (\langle Na \backslash S f \rangle / \exists b Nb) \uparrow \square^{-1} \square^{-1} (\langle Na \backslash S f \rangle / \exists b Nb)) / \blacksquare (\langle Na \backslash S f \rangle / \exists b Nb)) : (\Phi^{n+} (s (s 0)) \text{ and})$
and : $\blacksquare \forall f \forall a((\blacksquare (\langle Na \backslash S f \rangle / \exists b Nb) \uparrow \square^{-1} \square^{-1} (\langle Na \backslash S f \rangle / \exists b Nb)) / \blacksquare (\langle Na \backslash S f \rangle / \exists b Nb)) : (\Phi^{n+} (s (s 0)) \text{ and})$
and : $\blacksquare \forall f \forall a((\blacksquare (\langle Na \backslash S f \rangle / \exists b Nb) \uparrow \square^{-1} \square^{-1} (\langle Na \backslash S f \rangle / \exists b Nb)) / \blacksquare (\langle Na \backslash S f \rangle / \exists b Nb)) : (\Phi^{n+} (s (s 0)) \text{ and})$
and : $\blacksquare \forall a \forall b \forall f((? \blacksquare (\langle Na \backslash S f \rangle / PPb) \uparrow \square^{-1} \square^{-1} (\langle Na \backslash S f \rangle / PPb)) / \blacksquare (\langle Na \backslash S f \rangle / PPb)) : (\Phi^{n+} (s (s 0)) \text{ and})$
and : $\blacksquare \forall a \forall b \forall f((? \blacksquare (\langle Na \backslash S f \rangle / \exists c Nc \bullet PPb) \uparrow \square^{-1} \square^{-1} (\langle Na \backslash S f \rangle / \exists c Nc \bullet PPb)) / \blacksquare (\langle Na \backslash S f \rangle / \exists c Nc \bullet PPb)) : (\Phi^{n+} (s (s 0)) \text{ and})$
and : $\blacksquare \forall a \forall b \forall f((? \blacksquare (\langle Na \backslash S f \rangle / \exists c Nc \bullet PPb) \uparrow \square^{-1} \square^{-1} (\langle Na \backslash S f \rangle / \exists c Nc \bullet PPb)) / \blacksquare (\langle Na \backslash S f \rangle / \exists c Nc \bullet PPb)) : (\Phi^{n+} (s (s 0)) \text{ and})$
and : $\blacksquare \forall a \forall b \forall f((? \blacksquare (\langle Na \backslash S f \rangle / ! Nb) \uparrow \square^{-1} \square^{-1} (\langle Na \backslash S f \rangle / ! Nb)) / \blacksquare (\langle Na \backslash S f \rangle / ! Nb)) : (\Phi^{n+} (s (s 0)) \text{ and})$
ate : $\square((\langle \exists a Na \rangle \backslash S f) / \exists a Na) : \lambda A \lambda B (Past (\sim \text{eat } A) B)$
bagels : $\square(Nt(p(n)) \& CNp(n)) : \sim (\text{gen } \sim \text{bagels}), \sim \text{bagels}$
barn : $\square CNs(n) : \text{barn}$
be : $\square(\langle W[\text{there}] \rightarrow S b \rangle / \exists a Na) : \lambda A (\sim \text{be } A)$
before : $\blacksquare (\forall a \forall f(\langle \langle Na \backslash S f \rangle \rangle \langle \langle Na \backslash S f \rangle \rangle) / S f) : \lambda A \lambda B \lambda C((\text{before } A) (B C))$
beginning : $\square CNs(n) : \text{beginning}$
believes : $\square(\langle \langle \exists g Nt(s(g)) \rangle \backslash S f \rangle / (C P \text{that} \square \square S f)) : \lambda A \lambda B (Pres (\sim \text{believe } A) B)$
bill : $\blacksquare Nt(s(m)) : b$
book : $\square CNs(n) : \text{book}$
bought : $\square(\langle \langle \exists a Na \rangle \backslash S f \rangle / (\exists a Na \bullet \exists a Na)) : \lambda A \lambda B (Past ((\sim \text{buy } \pi_1 A) \pi_2 A) B)$
bought : $\square(\langle \langle \exists a Na \rangle \backslash S f \rangle / \exists a Na) : \lambda A \lambda B (Past (\sim \text{buy } A) B)$
by : $\blacksquare \forall a(\langle \langle Na \backslash S - \rangle \rangle \langle \langle Na \backslash S - \rangle \rangle) / Na) : \lambda A \lambda B \lambda C[[C = A] \wedge (B C)]$
by : $\square(\forall n(CNn \backslash CNn) / \exists a Na) : \lambda A \lambda B (\sim \text{by } A) B$
buys : $\square(\langle \langle \exists g Nt(s(g)) \rangle \backslash S f \rangle / (\exists a Na \bullet \exists a Na)) : \lambda A \lambda B (Pres ((\sim \text{buy } \pi_1 A) \pi_2 A) B)$
calls : $\square(\langle \langle \exists g Nt(s(g)) \rangle \backslash S f \rangle / \exists a(\langle W[\text{up}] \bullet Na \rangle \square (Na \bullet W[\text{up}])) : \lambda A \lambda B (\sim \text{phone } A) B)$
catch : $\square(\langle \langle \exists a Na \rangle \backslash S b \rangle / \exists a Na) : \lambda A \lambda B (\sim \text{catch } A) B$
cezanne : $\blacksquare Nt(s(m)) : c$
cd : $\square CNs(n) : cd$
charles : $\blacksquare Nt(s(m)) : c$
cicero : $\blacksquare \forall g Nt(s(g)) : c$
clark : $\blacksquare \forall g Nt(s(g)) : c$
coffee : $\square(Nt(s(n)) \& CNs(n)) : \sim (\text{gen } \sim \text{coffee}), \sim \text{coffee}$
created : $\square(\langle \langle \exists a Na \rangle \backslash S f \rangle / \exists a Na) : \lambda A \lambda B (Past (\sim \text{create } A) B)$
darkness : $\square(CNs(n) \& Nt(s(n))) : \sim (\sim \text{darkness}), (\text{gen } \sim \text{darkness})$
deep : $\square CNs(n) : \text{deep}$
did : $\blacksquare \forall a \forall g \forall b \forall h(\langle \langle \langle Na \backslash S g \rangle \rangle \langle \langle Nb \backslash S h \rangle \rangle) / (\exists c \langle Nc \backslash S f \rangle) \langle \langle \langle Na \backslash S g \rangle \rangle \langle \langle Nb \backslash S h \rangle \rangle) : \lambda A \lambda B ((A B) B)$
did+too : $(\langle \langle \langle Na \backslash S B \rangle \rangle \langle \langle NC \backslash S D \rangle \rangle) / (\langle \langle NE \backslash S F \rangle \rangle) \langle \langle \langle NG \backslash S H \rangle \rangle \langle \langle NI \backslash S J \rangle \rangle) : \lambda K \lambda L ((K L) L)$
doesnt : $\blacksquare \forall g \forall a((S g \uparrow (\langle \langle Na \backslash S f \rangle \rangle \langle \langle Na \backslash S b \rangle \rangle) \downarrow S g) : \lambda A - (A \lambda B \lambda C (B C))$
dog : $\square CNs(n) : \text{dog}$
donuts : $\square(Nt(p(n)) \& CNp(n)) : \sim (\text{gen } \sim \text{donuts}), \sim \text{donuts}$
earth : $\square CNs(n) : \text{earth}$
eat : $\square(\langle \langle \exists a Na \rangle \backslash S b \rangle / \exists a Na) : \lambda A \lambda B (\sim \text{eat } A) B$
edinburgh : $\blacksquare Nt(s(n)) : e$
editor : $\square(\forall g CNs(g) / PPof) : \text{editor}$
every : $\blacksquare \forall g(\forall f((S f \uparrow Nt(s(g))) \downarrow S f) / CNs(g)) : \lambda A \lambda B \forall C[(A C) \rightarrow (B C)]$
everyone : $\square \forall f((S f \uparrow \forall g Nt(g)) \downarrow S f) : \lambda A \forall B[(\sim \text{person } B) \rightarrow (A B)]$
face : $\square CNs(n) : \text{face}$

fell : $\Box(\exists a \langle \rangle Na \setminus S f) : \sim \lambda A(Past \text{ } (\sim fall A))$
filed : $\Box(\langle \rangle \exists g Nt(s(g)) \setminus S f) / \exists a Na : \sim \lambda A \lambda B(Past \text{ } (\sim file A) B)$
finds : $\Box(\langle \rangle \exists g Nt(s(g)) \setminus S f) / \exists a Na : \sim \lambda A \lambda B(Pres \text{ } (\sim find A) B)$
fish : $\Box CNs(n) : fish$
for : $\blacksquare(PPfor / \exists a Na) : \lambda AA$
form : $\Box(CNs(n) \& Nt(s(n))) : \sim \text{ } (\sim form, (gen \sim form))$
friends : $\Box(CNp / PPof) : friends$
from : $\Box((\forall a \forall f(\langle \rangle Na \setminus S f) \setminus (\langle \rangle Na \setminus S f)) \& \forall n(CNn \setminus CNn)) / \exists b Nb : \sim \lambda A(\sim fromadv A, (\sim fromadn A))$
gave : $\Box(\langle \rangle \exists a Na \setminus S f) / (\exists b Nb \bullet PPto) : \sim \lambda A \lambda B(Past \text{ } ((\sim give \pi_2 A) \pi_1 A) B)$
gave : $\Box(\langle \rangle \exists g Nt(s(g)) \setminus S f) / (\exists a Na \bullet W[the, cold, shoulder]) : \sim \lambda A \lambda B(Past \text{ } (\sim shun A) B)$
gave : $\Box(\langle \rangle \exists a Na \setminus S f) / \exists a Na / \exists a Na : \sim \lambda A \lambda B \lambda C(Past \text{ } ((\sim give A) B) C)$
girl : $\Box CNs(f) : girl$
gives : $\Box(\langle \rangle \exists g Nt(s(g)) \setminus S f) / (\exists a Na \bullet W[the, cold, shoulder]) : \sim \lambda A \lambda B(Pres \text{ } (\sim shun A) B)$
God : $\blacksquare Nt(s(m)) : God$
good : $\Box \forall n(CNn / CNn) : good$
has : $\Box(\langle \rangle \exists g Nt(s(g)) \setminus S f) / \exists a Na : \sim \lambda A \lambda B(Pres \text{ } (\sim have A) B)$
he : $\blacksquare \lceil^{-1} \forall g(\blacksquare S g \blacksquare Nt(s(m))) / (\langle \rangle Nt(s(m)) \setminus S g) : \lambda AA$
heaven : $\Box CNs(n) : heaven$
her : $\blacksquare \forall g \forall a(\langle \rangle Na \setminus S g) \uparrow \blacksquare Nt(s(f)) \downarrow (\blacksquare \langle \rangle Na \setminus S g) \blacksquare Nt(s(f)) : \lambda AA$
himself : $\blacksquare \forall f(\langle \rangle Nt(s(m)) \setminus S f) \uparrow Nt(s(m)) \downarrow (\langle \rangle Nt(s(m)) \setminus S f) : \lambda A \lambda B(A B) B$
horse : $\Box CNs(n) : horse$
humanist : $\Box \forall n(CNn / CNn) : \sim \lambda A \lambda B(A B) \wedge (\sim humanist B)$
in : $\Box(\forall a \forall f(\langle \rangle Na \setminus S f) \setminus (\langle \rangle Na \setminus S f)) / \exists a Na : \sim \lambda A \lambda B \lambda C(\sim in A) (B C)$
it : $\blacksquare W[it] : 0$
it : $\blacksquare \forall f \forall a(\langle \rangle Na \setminus S f) \uparrow \blacksquare Nt(s(n)) \downarrow (\blacksquare \langle \rangle Na \setminus S f) \blacksquare Nt(s(n)) : \lambda AA$
it : $\blacksquare \lceil^{-1} \forall f(\blacksquare S f \blacksquare Nt(s(n))) / (\langle \rangle Nt(s(n)) \setminus S f) : \lambda AA$
jogs : $\Box(\langle \rangle \exists g Nt(s(g)) \setminus S f) : \sim \lambda A(Pres \text{ } (\sim jog A))$
john : $\blacksquare Nt(s(m)) : j$
laughs : $\Box(\langle \rangle \exists g Nt(s(g)) \setminus S f) : \sim \lambda A(Pres \text{ } (\sim laugh A))$
left : $\Box(\langle \rangle \exists g Nt(s(g)) \setminus S f) : \sim \lambda A(Pres \text{ } (\sim leave A))$
let : $\Box(S im / S b) : let$
light : $\Box(CNs(n) \& Nt(s(n))) : \sim \text{ } (\sim light, (gen \sim light))$
likes : $\Box(\langle \rangle \exists g Nt(s(g)) \setminus S f) / \exists a Na : \sim \lambda A \lambda B(Pres \text{ } (\sim like A) B)$
logic : $\Box(Nt(s(n)) \& CNs(n)) : \sim \text{ } ((gen \sim logic), \sim logic)$
london : $\blacksquare Nt(s(n)) : l$
loses : $\Box(\langle \rangle \exists g Nt(s(g)) \setminus S f) / \exists a Na : \sim \lambda A \lambda B(Pres \text{ } (\sim lose A) B)$
love : $\Box(\langle \rangle \exists a Na \setminus S b) / \exists a Na : \sim \lambda A \lambda B(\sim love A) B$
loved : $\Box \forall a \forall b(\langle \rangle Na \setminus S -) \uparrow Nb \circ (\langle \rangle Na \setminus S -) \downarrow Nb \downarrow \forall g(CNg \setminus CNg)) : \sim \text{ } (\sim love, \lambda A \lambda B \lambda C[(B C) \wedge \exists D((A C) D)])$
loves : $\Box(\langle \rangle \exists g Nt(s(g)) \setminus S f) / \exists a Na : \sim \lambda A \lambda B(Pres \text{ } (\sim love A) B)$
man : $\Box CNs(m) : man$
mary : $\blacksquare Nt(s(f)) : m$
met : $\Box(\langle \rangle \exists a Na \setminus S f) / \exists a Na : \sim \lambda A \lambda B(Past \text{ } (\sim meet A) B)$
more : $\blacksquare \forall h \forall g \forall f((S f) \uparrow ((S h) \uparrow Nt(p(g))) \downarrow S h) / CNp(g)) \downarrow (S f) / (CPthan \uparrow \blacksquare((S h) \uparrow Nt(p(g))) \downarrow S h) / CNp(g)) : \lambda A \lambda B[\lambda C(A \lambda D \lambda E[(D C) \wedge (E C)]] > \lambda F(B \lambda G \lambda H[(G F) \wedge (H F)]]]$
mountain : $\Box CNs(n) : mountain$
moved : $\Box(\langle \rangle \exists a Na \setminus S f) : \sim \lambda A(Past \text{ } (\sim move A))$
necessarily : $\blacksquare(S A / \Box S A) : Nec$
of : $\Box((\forall n(CNn \setminus CNn)) / \blacksquare \exists b Nb) \& (PPof / \exists a Na) : \sim \text{ } (\sim of, \lambda AA)$
or : $\blacksquare \forall f(\langle \rangle \blacksquare S f \lceil^{-1} \lceil^{-1} S f) / \blacksquare S f : (\Phi^{n+} 0 \text{ } or)$
or : $\blacksquare \forall a \forall f(\langle \rangle \blacksquare Na \setminus S f) \lceil^{-1} \lceil^{-1} (\langle \rangle Na \setminus S f) / \blacksquare (\langle \rangle Na \setminus S f) : (\Phi^{n+} (s 0) \text{ } or)$
or : $\blacksquare \forall f(\langle \rangle \blacksquare S f / (\langle \rangle \exists g Nt(s(g)) \setminus S f) \lceil^{-1} \lceil^{-1} (S f / (\langle \rangle \exists g Nt(s(g)) \setminus S f)) / \blacksquare (S f / (\langle \rangle \exists g Nt(s(g)) \setminus S f)) : (\Phi^{n+} (s 0) \text{ } or)$
or : $\blacksquare \forall a \forall f(\langle \rangle \blacksquare Na \setminus S f) / \exists b Nb / \exists b Nb \lceil^{-1} \lceil^{-1} (\langle \rangle Na \setminus S f) / \exists b Nb / \exists b Nb / \blacksquare (\langle \rangle Na \setminus S f) / \exists b Nb / \exists b Nb) : (\Phi^{n+} (s (s 0)) \text{ } or)$
painting : $\Box(CNs(n) / PPof) : \sim \lambda A(\sim of A) \sim painting)$
paper : $\Box CNs(n) : paper$
park : $\Box CNs(n) : park$
past : $\Box \forall a \forall f(\langle \rangle Na \setminus S f) \setminus (\langle \rangle Na \setminus S f) / \exists b Nb : \sim \lambda A \lambda B \lambda C(\sim past A) (B C)$
perseverance : $\Box(Nt(s(n)) \& CNs(n)) : \sim \text{ } ((gen \sim perseverance), \sim perseverance)$
peter : $\blacksquare Nt(s(m)) : p$
phonetics : $\Box(Nt(s(n)) \& CNs(n)) : \sim \text{ } ((gen \sim phonetics), \sim phonetics)$

praises : $\square((\exists gNt(s(g))\backslash Sf)/\exists aNa) : \sim \lambda \lambda B(Pres(\sim \textit{praise } A) B)$
raced : $\square((\exists aNa\backslash Sf) : \sim \lambda A(Past(\sim \textit{race } A))$
raced : $\square \forall a \forall b((\exists Na\backslash S \rightarrow) \uparrow Nb) \circ ((\exists Na\backslash S \rightarrow) \uparrow Nb) \downarrow \forall g(CNg \backslash CNg)) : \sim \textit{race}2, \lambda \lambda B \lambda C[(B C) \wedge \exists D((A C) D)]$
rains : $\square((\exists W[it] \rightarrow Sf) : \sim (Pres \sim \textit{itrains})$
reading : $\square((\exists aNa\backslash S psp)/\exists aNa) : \sim \lambda \lambda B(\sim \textit{read } A) B$
robin : $\blacksquare \forall gNt(s(g)) : r$
said : $\square((\exists aNa\backslash Sf)/S im) : \sim \lambda \lambda B(Past(\sim \textit{say } A) B)$
saw : $\square((\exists aNa\backslash Sf)/(\exists aNa \oplus CPthat)) : \sim \lambda \lambda B(Past((A \rightarrow C.(\sim \textit{see } C); D.(\sim \textit{see } D)) B))$
seeks : $\square((\exists gNt(s(g))\backslash Sf)/\square \forall a \forall f((Na\backslash Sf)/\exists bNb)\backslash (Na\backslash Sf)) : \sim \lambda \lambda B(\sim \textit{tries}(\sim \textit{find } B)) B$
sees : $\square((\exists gNt(s(g))\backslash Sf)/\exists aNa) : \sim \lambda \lambda B(Pres(\sim \textit{see } A) B)$
sent : $\square((\exists aNa\backslash Sf)/(\exists bNb \bullet PPro)) : \sim \lambda \lambda B(Past((\sim \textit{sent } \pi_2 A) \pi_1 A) B)$
sent : $\square(((\exists aNa\backslash Sf)/\exists aNa)/\exists aNa) : \sim \lambda \lambda B \lambda C(Past((\sim \textit{send } A) B) C)$
she : $\blacksquare [^{-1} \forall g(\blacksquare S g \blacksquare Nt(s(f)))/(\exists Nt(s(f))\backslash Sg) : \lambda AA$
sings : $\square((\exists gNt(s(g))\backslash Sf) : \sim \lambda A(Pres(\sim \textit{sing } A))$
slept : $\square((\exists gNt(s(g))\backslash Sf) : \sim \lambda A(Past(\sim \textit{sleep } A))$
slowly : $\square \forall a \forall f(\square((\exists Na\backslash Sf)/(\exists Na\backslash Sf)) : \sim \lambda \lambda B(\sim \textit{slowly}(\sim \textit{A } \sim B))$
sneezed : $\square((\exists gNt(s(g))\backslash Sf) : \sim \lambda A(Past(\sim \textit{sneeze } A))$
sold : $\square((\exists aNa\backslash Sf)/(\exists bNb \bullet PPro)) : \sim \lambda \lambda B(Past(((\sim \textit{sell } \pi_2 A) \pi_1 A) B))$
someone : $\square \forall f((S f \blacksquare \forall gNt(g))\backslash Sf) : \sim \lambda A \exists B[(\sim \textit{person } B) \wedge (A B)]$
Spirit : $\square CNs(m) : \textit{Spirit}$
studies : $\square((\exists gNt(s(g))\backslash Sf)/\exists aNa) : \sim \lambda \lambda B(Pres(\sim \textit{study } A) B)$
such+that : $\blacksquare \forall n((CNn \backslash CNn)/(S f \blacksquare Nt(n))) : \lambda \lambda B \lambda C[(B C) \wedge (A C)]$
suzy : $\blacksquare Nt(s(f)) : s$
talks : $\square((\exists gNt(s(g))\backslash Sf) : \sim \lambda A(Pres(\sim \textit{talk } A))$
tall : $\square \forall g(CNg/CNg) : \textit{tall}$
tenmilliondollars : $\square Nt(s(n)) : \textit{tenmilliondollars}$
than : $\blacksquare (CPthan/\square Sf) : \lambda AA$
that : $\blacksquare (CPthat/\square Sf) : \lambda AA$
that : $\blacksquare \forall n([^{-1} [^{-1} (CNn \backslash CNn)/\blacksquare((\exists Nt(n) \blacksquare Nt(n))\backslash Sf)) : \lambda \lambda B \lambda C[(B C) \wedge (A C)]$
the : $\blacksquare \forall n(Nt(n)/CNn) : t$
the+cold+shoulder : $\blacksquare W[\textit{the, cold, shoulder}] : 0$
there : $\blacksquare W[\textit{there}] : 0$
thinks : $\square((\exists gNt(s(g))\backslash Sf)/(CPthat \sqcup \square Sf) : \sim \lambda \lambda B(Pres(\sim \textit{think } A) B))$
to : $\blacksquare ((PPro/\exists aNa) \blacksquare \forall n((\exists Nn\backslash Si)/(\exists Nn\backslash Sb))) : \lambda AA$
today : $\square \forall a \forall f((\exists Na\backslash Sf)/(\exists Na\backslash Sf)) : \sim \lambda \lambda B(\sim \textit{today } (A B))$
tries : $\square((\exists gNt(s(g))\backslash Sf)/\square(\exists gNt(s(g))\backslash Si)) : \sim \lambda \lambda B(\sim \textit{tries}(\sim \textit{A } B)) B$
tully : $\blacksquare Nt(s(m)) : t$
unicorn : $\square CNs(n) : \textit{unicorn}$
up : $\blacksquare W[\textit{up}] : 0$
upon : $\square((\forall b \forall f((\exists Nb\backslash Sf)/(\exists Nb\backslash Sf)) \& \forall g(CNg \backslash CNg)/\exists aNa) : \sim \lambda A((\sim \textit{uponadv } A), (\sim \textit{uponadn } A))$
void : $\square \forall g(CNg/CNg) : \textit{void}$
walk : $\square((\exists aNa \rightarrow \exists gNt(s(g))\backslash Sf) : \sim \lambda A(Pres(\sim \textit{walk } A))$
walk : $\square((\exists aNa\backslash Sb) : \sim \lambda A(\sim \textit{walk } A)$
walks : $\square((\exists gNt(s(g))\backslash Sf) : \sim \lambda A(Pres(\sim \textit{walk } A))$
was : $\blacksquare ((\exists gNt(s(g))\backslash Sf)/(\exists aNa \oplus (\exists g(CNg/CNg) \sqcup (CNg \backslash CNg) \rightarrow I))) : \lambda \lambda B(Past(A \rightarrow C.[B = C]; D.((D \lambda E[E = B]) B)))$
was : $\square((\exists W[\textit{there}] \rightarrow Sf)/\exists aNa) : \sim \lambda A(Past(\sim \textit{be } A))$
waters : $\square CNp(n) : \textit{waters}$
which : $\blacksquare \forall n \forall m((Nt(n) \uparrow Nt(m)) \downarrow ([^{-1} [^{-1} (CNm \backslash CNm)/\blacksquare((\exists Nt(n) \blacksquare Nt(n))\backslash Sf))) : \lambda \lambda B \lambda C \lambda D[(C D) \wedge (B (A D))]$
who : $\blacksquare \forall h \forall n([^{-1} [^{-1} (Nt(n)) \backslash ((S h \uparrow Nt(n)) \downarrow S h)]/\blacksquare((\exists Nt(n) \blacksquare Nt(n))\backslash Sf)) : \lambda \lambda B \lambda C[(A B) \wedge (C B)]$
will : $\blacksquare \forall a((\exists Na\backslash Sf)/(\exists Na\backslash Sb)) : \lambda \lambda B(Fut(A B))$
without : $\square(\forall g(CNg/CNg)/\exists aNa) : \sim \lambda \lambda B \lambda C[(B C) \wedge \neg(\sim \textit{with } A) C]$
without : $\blacksquare \forall a \forall f([^{-1} ((\exists Na\backslash Sf)/(\exists Na\backslash Sf)))/(\exists Na\backslash Spsp) : \lambda \lambda B \lambda C[(B C) \wedge \neg(A C)]$
woman : $\square CNs(f) : \textit{woman}$
yesterday : $\square \forall a \forall f((\exists Na\backslash Sf)/(\exists Na\backslash Sf)) : \sim \lambda \lambda B(\sim \textit{yesterday } (A B))$

References

1. Antony E. Ades and Mark J. Steedman. On the order of words. *Linguistics and Philosophy*, 4:517–558, 1982.
2. Kazimierz Ajdukiewicz. Die syntaktische Konnexität. *Studia Philosophica*, 1:1–27, 1935. Translated in Storrs McCall, editor, 1967, *Polish Logic: 1920–1939*, Oxford University Press, Oxford, 207–231.
3. J. M. Andreoli. Logic programming with focusing in linear logic. *Journal of Logic and Computation*, 2(3):297–347, 1992.
4. Chris Barker and Chung-chieh Shan. *Continuations and Natural Language*. Oxford University Press, New York and Oxford, 2015.
5. Guy Barry, Mark Hepple, Neil Leslie, and Glyn Morrill. Proof Figures and Structural Operators for Categorical Grammar. In *Proceedings of the Fifth Conference of the European Chapter of the Association for Computational Linguistics*, Berlin, 1991.
6. Bob Carpenter. *Type-Logical Semantics*. MIT Press, Cambridge, MA, 1997.
7. N. Chomsky. Conditions on transformations. In S. Anderson and P. Kiparsky, editors, *A Festschrift for Morris Halle*, pages 232–286. Holt, Rinehart and Winston, New York, 1973.
8. Paul Deane. Limits to attention: A cognitive theory of island phenomena. *Cognitive Linguistics*, 2(1):1–63, 1991.
9. E. Engdahl. Parasitic gaps. *Linguistics and Philosophy*, 6:5–34, 1983.
10. G. Gentzen. Untersuchungen über das logische Schliessen. *Mathematische Zeitschrift*, 39:176–210 and 405–431, 1934. Translated in M.E. Szabo, editor, 1969, *The Collected Papers of Gerhard Gentzen*, North-Holland, Amsterdam, 68–131.
11. Jean-Yves Girard. Linear logic. *Theoretical Computer Science*, 50:1–102, 1987.
12. Jean-Yves Girard. *The Blind Spot*. European Mathematical Society, Zürich, 2011.
13. Mark Hepple. *The Grammar and Processing of Order and Dependency*. PhD thesis, University of Edinburgh, 1990.
14. Philip Hofmeister, Laura Staum Casasanto, and Ivan Sag. Islands in the grammar? standards of evidence. In D. Forget, P. Hirschbuhler, F. Martineau, and M.-L. Rivero, editors, *Experimental Syntax and Island Effects*, pages 42–63. CUP, Cambridge, England, 2013.
15. Philip Hofmeister and Ivan A. Sag. Cognitive constraints and island effects. *Language*, 86(2):366–415, 2010.
16. Gerhard Jäger. *Anaphora and Type Logical Grammar*, volume 24 of *Trends in Logic – Studia Logica Library*. Springer, Dordrecht, 2005.
17. Martin Jansche and Shrawan Vasishth. Review of Mark Steedman, *The Syntactic Process*. *Journal of Linguistics*, 38:684–690, 2002.
18. M. Kanazawa. The Lambek calculus enriched with additional connectives. *Journal of Logic, Language and Information*, 1:141–171, 1992.
19. Andrew Kehler. *Coherence, Reference and the Theory of Grammar*. CSLI Publications, Stanford, California, 2002.
20. Robert Kluender. Deriving island constraints from principles of predication. In H. Goodluck and M. Rochemont, editors, *Island Constraints: Theory, Acquisition, and Processing*, pages 223–258. Kluwer, Dordrecht, 1992.
21. Robert Kluender. On the distinction between strong and weak islands: A processing perspective. In P. Culicover and L. McNally, editors, *The Limits of Syntax*, volume 29 of *Syntax and Semantics*, pages 241–279. Academic Press, San Diego, 1998.
22. Yusuke Kubota and Robert Levine. Against ellipsis: Arguments for the direct licensing of ‘non-canonical’ coordinations. *Linguistics and Philosophy*, 2015. to appear.
23. George Lakoff. Frame semantic control of the Coordinate Structure Constraint. In A. M. Farley, P. T. Farley, and K.-E. McCullough, editors, *CLS 22 PART 2: Papers from the Parasession on Pragmatics and Grammatical Theory, 152*. Chicago Linguistics Society, Chicago, 1986.
24. J. Lambek. On the Calculus of Syntactic Types. In Roman Jakobson, editor, *Structure of Language and its Mathematical Aspects, Proceedings of the Symposia in Applied Mathematics XII*, pages 166–178. American Mathematical Society, Providence, Rhode Island, 1961.
25. J. Lambek. Categorical and Categorical Grammars. In Richard T. Oehrle, Emmon Bach, and Deidre Wheeler, editors, *Categorical Grammars and Natural Language Structures*, volume 32 of *Studies in Linguistics and Philosophy*, pages 297–317. D. Reidel, Dordrecht, 1988.
26. Joachim Lambek. The mathematics of sentence structure. *American Mathematical Monthly*, 65:154–170, 1958.
27. Robert D. Levine and Thomas E. Hukari. *The Unity of Unbounded Dependency Constructions*. CSLI Publications, Stanford, California, 2006.

28. Stepan Kuznetsov Max Kanovich and Andre Scedrov. Undecidability of the Lambek calculus with a relevant modality. In Annie Foret, Glyn Morrill, Reinhard Muskens, Rainer Osswald, and Sylvain Pogodalla, editors, *Formal Grammar 2015: Revised Selected Papers. Formal Grammar 2016: Proceedings*, Berlin, 2016. Springer.
29. Vedrana Mihaliček and Carl Pollard. Distinguishing phenogrammar from tectogrammar simplifies the analysis of interrogatives. In P. de Groote and M.-J. Nederhof, editors, *Formal Grammar 2010/2011*, pages 130–145, Heidelberg, 2012. Springer.
30. Michael Moortgat. *Categorial Investigations: Logical and Linguistic Aspects of the Lambek Calculus*. Foris, Dordrecht, 1988. PhD thesis, Universiteit van Amsterdam.
31. Michael Moortgat. Multimodal linguistic inference. *Journal of Logic, Language and Information*, 5(3, 4):349–385, 1996. Also in *Bulletin of the IGPL*, 3(2,3):371–401, 1995.
32. Michael Moortgat. Categorial Type Logics. In Johan van Benthem and Alice ter Meulen, editors, *Handbook of Logic and Language*, pages 93–177. Elsevier Science B.V. and the MIT Press, Amsterdam and Cambridge, Massachusetts, 1997.
33. Richard Moot and Christian Retoré. *The Logic of Categorial Grammars: A Deductive Account of Natural Language Syntax and Semantics*. Springer, Heidelberg, 2012.
34. G. Morrill. Grammar and Logical Types. In Martin Stockhof and Leen Torenvliet, editors, *Proceedings of the Seventh Amsterdam Colloquium*, pages 429–450, 1990.
35. Glyn Morrill. Intensionality and Boundedness. *Linguistics and Philosophy*, 13(6):699–726, 1990.
36. Glyn Morrill. Categorial Formalisation of Relativisation: Pied Piping, Islands, and Extraction Sites. Technical Report LSI-92-23-R, Departament de Llenguatges i Sistemes Informàtics, Universitat Politècnica de Catalunya, 1992.
37. Glyn Morrill. Incremental Processing and Acceptability. *Computational Linguistics*, 26(3):319–338, 2000.
38. Glyn Morrill. Logic Programming of the Displacement Calculus. In Sylvain Pogodalla and Jean-Philippe Prost, editors, *Proceedings of Logical Aspects of Computational Linguistics 2011, LACL'11, Montpellier*, number LNAI 6736 in Springer Lecture Notes in AI, pages 175–189, Berlin, 2011. Springer.
39. Glyn Morrill. Logical grammar. In Ruth Kempson, Tim Fernando, and Nicholas Asher, editors, *The Handbook of Philosophy of Linguistics*, pages 63–92. Elsevier, Oxford and Amsterdam, 2012.
40. Glyn Morrill and Josep-Maria Merenciano. Generalising discontinuity. *Traitement automatique des langues*, 37(2):119–143, 1996.
41. Glyn Morrill and Oriol Valentín. Displacement Calculus. *Linguistic Analysis*, 36(1–4):167–192, 2010. Special issue Festschrift for Joachim Lambek, <http://arxiv.org/abs/1004.4181>.
42. Glyn Morrill and Oriol Valentín. Displacement Logic for Anaphora. *Journal of Computing and System Science*, 80:390–409, 2014. <http://dx.doi.org/10.1016/j.jcss.2013.05.006>.
43. Glyn Morrill and Oriol Valentín. Semantically Inactive Multiplicatives and Words as Types. In Nicholas Asher and Sergei Soloviev, editors, *Proceedings of Logical Aspects of Computational Linguistics, LACL'14, Toulouse*, number 8535 in LNCS, FoLLI Publications on Logic, Language and Information, pages 149–162, Berlin, 2014. Springer.
44. Glyn Morrill and Oriol Valentín. Computational Coverage of TLG: Nonlinearity. In M. Kanazawa, L.S. Moss, and V. de Paiva, editors, *Proceedings of NLCS'15. Third Workshop on Natural Language and Computer Science*, volume 32 of *EPIc*, pages 51–63, Kyoto, 2015. Workshop affiliated with Automata, Languages and Programming (ICALP) and Logic in Computer Science (LICS).
45. Glyn Morrill and Oriol Valentín. Multiplicative-Additive Focusing for Parsing as Deduction. In I. Cervésato and C. Schürmann, editors, *First International Workshop on Focusing, workshop affiliated with LPAR 2015*, number 197 in EPTCS, pages 29–54, Suva, Fiji, 2015.
46. Glyn Morrill and Oriol Valentín. On the logic of expansion in natural language. In Maxime Amblard, Philippe de Groote, Sylvain Pogodalla, and Christian Retoré, editors, *Proceedings of Logical Aspects of Computational Linguistics, LACL'16, Nancy*, LNCS, FoLLI Publications on Logic, Language and Information, Berlin, 2016. Springer.
47. Glyn Morrill, Oriol Valentín, and Mario Fadda. Dutch Grammar and Processing: A Case Study in TLG. In Peter Bosch, David Gabelaia, and Jérôme Lang, editors, *Logic, Language, and Computation: 7th International Tbilisi Symposium, Revised Selected Papers*, number 5422 in Lecture Notes in Artificial Intelligence, pages 272–286, Berlin, 2009. Springer.
48. Glyn Morrill, Oriol Valentín, and Mario Fadda. The Displacement Calculus. *Journal of Logic, Language and Information*, 20(1):1–48, 2011. Doi 10.1007/s10849-010-9129-2.
49. Glyn V. Morrill. *Type Logical Grammar: Categorial Logic of Signs*. Kluwer Academic Publishers, Dordrecht, 1994.
50. Glyn V. Morrill. *Categorial Grammar: Logical Syntax, Semantics, and Processing*. Oxford University Press, New York and Oxford, 2011.

51. Reinhard Muskens. Language, lambdas and logic. In Geert-Jan M. Kruijff and Richard T. Oehrle, editors, *Resource-Sensitivity in Binding and Anaphora*, volume 80 of *Studies in Linguistics and Philosophy*, pages 23–54. Kluwer Academic Press, Dordrecht, 2003.
52. Frederick J. Newmeyer. Nonsyntactic explanations of island constraints. *Annual Review of Linguistics*, 2:187–210, 2016.
53. Paul M. Postal. Parasitic Gaps and Across-the-Board Phenomenon. *Linguistic Inquiry*, 24(4):735–754, 1993.
54. I.A. Sag. On parasitic gaps. *Linguistics and Philosophy*, 6:35–45, 1983.
55. J. Sprouse, M. Wagers, and C. Phillips. A test of the relation between working memory capacity and syntactic island effects. *Language*, 88:82–123, 2012.
56. Mark Steedman. Combinatory Grammars and Parasitic Gaps. *Natural Language and Linguistic Theory*, 5:403–439, 1987.
57. A. Szabolcsi. Strong versus weak islands. In M. Everaert and H. van Riemsdijk, editors, *The Blackwell Companion to Syntax*, pages 479–531. Blackwell, Oxford, UK, 2006.
58. Anna Szabolcsi. ECP in categorial grammar. *Manuscript*, 1983, Max Planck Institute, Nijmegen.
59. T. Taraldsen. The theoretical interpretation of a class of marked extractions. In A. Belletti, L. Brandi, and L. Rizzi, editors, *Theory of Markedness in Generative Grammar*. Scuole Normal Superiore de Pisa, Pisa, 1979.
60. Oriol Valentín. *Theory of Discontinuous Lambek Calculus*. PhD thesis, Universitat Autònoma de Barcelona, Barcelona, 2012.
61. Oriol Valentín. The Hidden Structural Rules of the Discontinuous Lambek Calculus. In Claudia Casadio, Bob Coecke, Michael Moortgat, and Philip Scott, editors, *Categories and Types in Logic, Language and Physics: Essays Dedicated to Jim Lambek on the Occasion of His 90th Birthday*, volume 8222 of *LNCS, FoLLI Publications in Logic, Language and Information*, pages 402–420. Springer, Berlin, 2014.
62. Oriol Valentín, Daniel Serret, and Glyn Morrill. A Count Invariant for Lambek Calculus with Additives and Bracket Modalities. In Glyn Morrill and Mark-Jan Nederhof, editors, *Proceedings of Formal Grammar 2012 and 2013*, volume 8036 of *Springer LNCS, FoLLI Publications in Logic, Language and Information*, pages 263–276, Berlin, 2013. Springer.
63. J. van Benthem. *Language in Action: Categories, Lambdas, and Dynamic Logic*. Number 130 in *Studies in Logic and the Foundations of Mathematics*. North-Holland, Amsterdam, 1991. Revised student edition printed in 1995 by the MIT Press.