

Semantically Inactive Multiplicatives and Words as Types[★]

Glyn Morrill & Oriol Valentín

Universitat Politècnica
de Catalunya
morrill@lsi.upc.edu & oriol.valentin@gmail.com

Abstract. The literature on categorial type logic includes proposals for semantically inactive additives, quantifiers, and modalities Morrill (1994[17]), Hepple (1990[2]), Moortgat (1997[9]), but to our knowledge there has been no proposal for semantically inactive multiplicatives. In this paper we formulate such a proposal (thus filling a gap in the typology of categorial connectives) in the context of the displacement calculus Morrill et al. (2011[16]), and we give a formulation of words as types whereby for every expression w there is a corresponding type $W(w)$. We show how this machinery can treat the syntax and semantics of collocations involving apparently contentless words such as expletives, particle verbs, and (discontinuous) idioms. In addition, we give an account in these terms of the only known examples treated by Hybrid Type Logical Grammar (HTLG henceforth; Kubota and Levine 2012[4]) beyond the scope of unaugmented displacement calculus: gapping of particle verbs and discontinuous idioms.

1 Introduction

Examples of collocations are as follows. Expletives:

- (1) a. *It* rains.
- b. *There* came a man.

Discontinuous idiom:

- (2) *Mary gives Peter the cold shoulder.*

Particle verb:

- (3) a. *Mary calls Peter up.*
- b. *Mary calls up Peter.*

(Observe in (3) the non-determinism of the position of the object *Peter* with respect to the conjunct *calls up*.) We give an account of such collocations which respects the intuition that the constructions derive from content words (*rains, came, gives, calls*) which require certain other words without content in their grammatical context.¹ Furthermore we do this in such a way that it is possible to give an account of gapping as (almost) like type coordination which includes coverage of examples such as (4c, d), for which Kubota and Levine (2012[4]) invoke Hybrid Type Logical Grammar.

[★] Research partially supported by an ICREA Acadèmia 2012 to the alphabetically first author, and by BASMATI MICINN project (TIN2011-27479-C04-03) and SGR2009-1428 (LARCA). Many thanks to three anonymous LACL referees for valuable comments and suggestions.

¹ As a referee points out, *gives . . . the cold shoulder* and *calls . . . up* are not completely frozen expressions: *Mary gives the same cold shoulder to Peter, Mary calls Peter right up*. Based on such observations Wasow, Sag and Nunberg (1983[20]) claim that the elements of such constructions should be assigned semantic interpretations. If this is the case the present paper presents a solution to a problem which does not exist, but we think that although the present paper is not the last word, such observations do not dissolve the reality of the qualitative difference between normal compositional semantics and idiomatic semantics with reduced content.

- (4) a. Peter sees Mary and Robin, Clark.
 b. Peter calls up Mary and Robin, Clark.
 c. Peter calls Mary up and Robin, Clark.
 d. Peter gives Mary the cold shoulder and Robin, Clark.

Thus the technically and conceptually independently motivated machinery of semantically inactive multiplicatives and words as types accounts for the only known application of HTLG beyond the scope of displacement calculus.

In Section 2 we present the machinery. In Section 3 we give our empirical accounts. We conclude in Section 4.

2 Displacement Categorical Logic

The syntactic types of our categorial logic are sorted according to the number of points of discontinuity their expressions contain. Each *type predicate letter* will have 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 Figure 1, and the homomorphic semantic type map T associates these with semantic types. Note the new semantically inactive concatenative (continuous) and intercalative (discontinuous) multiplicatives in the middle of the table.

The set \mathcal{O} of *configurations* of (hyper)sequent calculus for our categorial logic is defined as follows, where Λ is the metalinguistic empty string and 1 is a metalinguistic placeholder:

$$(5) \mathcal{O} ::= \Lambda \mid 1 \mid \mathcal{F}_0 \mid \underbrace{\mathcal{F}_{i>0}\{\mathcal{O} : \dots : \mathcal{O}\}}_{i \text{ O's}} \mid \mathcal{O}, \mathcal{O} \mid [\mathcal{O}]$$

The sort sA of a type A is the $i \in \mathcal{N}$ such that $A \in \mathcal{F}_i$. The sort of a configuration Γ is the number of placeholders 1 that Γ contains.

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.

The usual configuration distinguished occurrence notation $\Delta(\Gamma)$ signifies a configuration Δ with a distinguished subconfiguration Γ , i.e. a configuration occurrence Γ with (external) context Δ . In the hypersequent calculus the distinguished *hyperoccurrence* notation $\Delta\langle \Gamma \rangle$ signifies a configuration *hyperoccurrence* Γ with external *and internal* context Δ as follows: the *distinguished hyperoccurrence* notation $\Delta\langle \Gamma \rangle$ abbreviates $\Delta_0(\Gamma \otimes \langle \Delta_1, \dots, \Delta_i \rangle)$.

Where Δ is a configuration of sort $i > 0$ and Γ is a configuration, the *leftmost* metalinguistic wrap is given by

$$\Delta \downarrow \Gamma =_{df} \Delta \otimes \underbrace{\langle \Gamma, 1, \dots, 1 \rangle}_{i-1 \text{ 1's}}$$

i.e. the configuration resulting from replacing by Γ the leftmost placeholder in Δ ; and the *rightmost* metalinguistic wrap is given by

$$\Delta \downarrow \Gamma =_{df} \Delta \otimes \underbrace{\langle 1, \dots, 1, \Gamma \rangle}_{i-1 \text{ 1's}}$$

i.e. the configuration resulting from replacing by Γ the rightmost placeholder in Δ .

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

$$(6) \vec{A} = \begin{cases} A & \text{if } sA = 0 \\ A \underbrace{\{1 : \dots : 1\}}_{sA \text{ 1's}} & \text{if } sA > 0 \end{cases}$$

$\mathcal{F}_0 ::= W(w)$	$T(W(w)) = \top$	words as types
$\mathcal{F}_j ::= \mathcal{F}_i \setminus \mathcal{F}_{i+j}$	$T(A \setminus C) = T(A) \rightarrow T(C)$	suffix [7]
$\mathcal{F}_i ::= \mathcal{F}_{i+j} / \mathcal{F}_j$	$T(C / B) = T(B) \rightarrow T(C)$	prefix [7]
$\mathcal{F}_{i+j} ::= \mathcal{F}_i \bullet \mathcal{F}_j$	$T(A \bullet B) = T(A) \& T(B)$	concatenation product [7]
$\mathcal{F}_0 ::= I$	$T(I) = \top$	concatenation product unit [6]
$\mathcal{F}_j ::= \mathcal{F}_{i+1} \downarrow \mathcal{F}_{i+j}$	$T(A \downarrow C) = T(A) \rightarrow T(C)$	infix [16]
$\mathcal{F}_{i+1} ::= \mathcal{F}_{i+j} \uparrow \mathcal{F}_j$	$T(C \uparrow B) = T(B) \rightarrow T(C)$	circumfix [16]
$\mathcal{F}_{i+j} ::= \mathcal{F}_{i+1} \circlearrowright \mathcal{F}_j$	$T(A \circlearrowright B) = T(A) \& T(B)$	intercalation product [16]
$\mathcal{F}_1 ::= J$	$T(J) = \top$	intercalation product unit [16]
$\mathcal{F}_i ::= \mathcal{F}_i \& \mathcal{F}_i$	$T(A \& B) = T(A) \& T(B)$	additive conjunction [5, 10]
$\mathcal{F}_i ::= \mathcal{F}_i \oplus \mathcal{F}_i$	$T(A \oplus B) = T(A) + T(B)$	additive disjunction [5, 10]
$\mathcal{F}_i ::= \bigwedge V \mathcal{F}_i$	$T(\bigwedge v A) = F \rightarrow T(A)$	1st order univ. qu. [17]
$\mathcal{F}_i ::= \bigvee V \mathcal{F}_i$	$T(\bigvee v A) = F \& T(A)$	1st order exist. qu. [17]
$\mathcal{F}_i ::= \Box \mathcal{F}_i$	$T(\Box A) = \mathbf{L}T(A)$	universal modality [11]
$\mathcal{F}_i ::= \Diamond \mathcal{F}_i$	$T(\Diamond A) = \mathbf{M}T(A)$	existential modality [9]
$\mathcal{F}_i ::= []^{-1} \mathcal{F}_i$	$T([]^{-1} A) = T(A)$	univ. bracket modality [12, 8]
$\mathcal{F}_i ::= \langle \rangle \mathcal{F}_i$	$T(\langle \rangle A) = T(A)$	exist. bracket modality [12, 8]
$\mathcal{F}_{i+j} ::= \mathcal{F}_{i+j} \mathcal{F}_j$	$T(B A) = T(A) \rightarrow T(B)$	contr. for anaph. [3]
$\mathcal{F}_j ::= \mathcal{F}_i \multimap \mathcal{F}_{i+j}$	$T(A \multimap C) = T(C) \text{ if } T(A) = \top$	left sem. inactive under
$\mathcal{F}_i ::= \mathcal{F}_{i+j} \multimap \mathcal{F}_j$	$T(C \multimap B) = T(B) \text{ if } T(C) = \top$	left sem. inactive over
$\mathcal{F}_j ::= \mathcal{F}_i \multimap \mathcal{F}_{i+j}$	$T(A \multimap C) = T(A) \text{ if } T(C) = \top$	right sem. inactive under
$\mathcal{F}_i ::= \mathcal{F}_{i+j} \multimap \mathcal{F}_j$	$T(C \multimap B) = T(C) \text{ if } T(B) = \top$	right sem. inactive over
$\mathcal{F}_{i+j} ::= \mathcal{F}_i \bullet \mathcal{F}_j$	$T(A \bullet B) = T(B) \text{ if } T(A) = \top$	right sem. inactive conc. product
$\mathcal{F}_{i+j} ::= \mathcal{F}_i \circlearrowleft \mathcal{F}_j$	$T(A \circlearrowleft B) = T(A) \text{ if } T(B) = \top$	right sem. inactive conc. product
$\mathcal{F}_j ::= \mathcal{F}_i \downarrow \mathcal{F}_{i+j}$	$T(A \downarrow C) = T(C) \text{ if } T(A) = \top$	lower sem. inactive infix
$\mathcal{F}_i ::= \mathcal{F}_{i+j} \circlearrowright \mathcal{F}_j$	$T(C \circlearrowright B) = T(B) \text{ if } T(C) = \top$	lower sem. inactive circumfix
$\mathcal{F}_j ::= \mathcal{F}_i \downarrow \mathcal{F}_{i+j}$	$T(A \downarrow C) = T(A) \text{ if } T(C) = \top$	upper sem. inactive infix
$\mathcal{F}_i ::= \mathcal{F}_{i+j} \circlearrowleft \mathcal{F}_j$	$T(C \circlearrowleft B) = T(C) \text{ if } T(B) = \top$	upper sem. inactive circumfix
$\mathcal{F}_{i+j} ::= \mathcal{F}_i \circlearrowright \mathcal{F}_j$	$T(A \circlearrowright B) = T(B) \text{ if } T(A) = \top$	lower sem. inactive interc. product
$\mathcal{F}_{i+j} ::= \mathcal{F}_i \circlearrowleft \mathcal{F}_j$	$T(A \circlearrowleft B) = T(A) \text{ if } T(B) = \top$	upper sem. inactive interc. product
$\mathcal{F}_i ::= \mathcal{F}_i \sqcap \mathcal{F}_i$	$T(A \sqcap B) = T(A) = T(B)$	sem. inactive additive conjunction [17]
$\mathcal{F}_i ::= \mathcal{F}_i \sqcup \mathcal{F}_i$	$T(A \sqcup B) = T(A) = T(B)$	sem. inactive additive disjunction [17]
$\mathcal{F}_i ::= \forall V \mathcal{F}_i$	$T(\forall v A) = T(A)$	sem. inactive 1st order univ. qu. [17]
$\mathcal{F}_i ::= \exists V \mathcal{F}_i$	$T(\exists v A) = T(A)$	sem. inactive 1st order exist. qu. [17]
$\mathcal{F}_i ::= \blacksquare \mathcal{F}_i$	$T(\blacksquare A) = T(A)$	sem. inactive universal modality [2]
$\mathcal{F}_i ::= \blacklozenge \mathcal{F}_i$	$T(\blacklozenge A) = T(A)$	sem. inactive existential modality
$\mathcal{F}_i ::= \leftarrow^{-1} \mathcal{F}_i$	$T(\leftarrow^{-1} A) = T(A)$	left projection [15]
$\mathcal{F}_i ::= \rightarrow^{-1} \mathcal{F}_i$	$T(\rightarrow^{-1} A) = T(A)$	right projection [15]
$\mathcal{F}_i ::= \leftarrow \mathcal{F}_i$	$T(\leftarrow A) = T(A)$	left injection [15]
$\mathcal{F}_i ::= \rightarrow \mathcal{F}_i$	$T(\rightarrow A) = T(A)$	right injection [15]
$\mathcal{F}_{i+1} ::= \underset{\sim}{\mathcal{F}}_i$	$T(\underset{\sim}{\mathcal{F}} A) = T(A)$	split [14]
$\mathcal{F}_i ::= \underset{\sim}{\mathcal{F}}_i$	$T(\underset{\sim}{\mathcal{F}} A) = T(A)$	bridge [14]
$\mathcal{F}_i ::= \mathcal{F}_{i+j} \div \mathcal{F}_j^{\bar{\bar{}}}$	$T(B \div A) = T(A) \rightarrow T(B)$	nondet. division [16]
$\mathcal{F}_{i+j} ::= \mathcal{F}_i \times \mathcal{F}_j$	$T(A \times B) = T(A) \& T(B)$	nondet. concatenation product [16]
$\mathcal{F}_j ::= \mathcal{F}_{i+1} \Downarrow \mathcal{F}_{i+j}$	$T(A \Downarrow C) = T(A) \rightarrow T(C)$	nondet. infix [16]
$\mathcal{F}_{i+1} ::= \mathcal{F}_{i+j} \Uparrow \mathcal{F}_j$	$T(C \Uparrow B) = T(B) \rightarrow T(C)$	nondet. circumfix [16]
$\mathcal{F}_{i+j} ::= \mathcal{F}_{i+1} \circlearrowright \mathcal{F}_j$	$T(A \circlearrowright B) = T(A) \& T(B)$	nondet. intercalation product [16]

Fig. 1. Categorical logic types

A *sequent* $\Gamma \Rightarrow A$ comprises an antecedent configuration Γ of sort i and a succedent type A of sort i . The hypersequent calculus for the categorial logic has the following identity axiom and Cut rule:

$$(7) \frac{}{\vec{A}:x \Rightarrow A:x} id \quad \frac{\Gamma \Rightarrow A:\phi \quad \Delta\langle \vec{A}:x \rangle \Rightarrow B:\psi}{\Delta\langle \Gamma \rangle \Rightarrow B:\psi\{\phi/x\}} Cut$$

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* and includes:

$$(8) \mathcal{T} ::= \delta \mid \top \mid \mathcal{T} \& \mathcal{T} \mid \mathcal{T} \rightarrow \mathcal{T}$$

A *semantic frame* comprises a family $\{D_\tau\}_{\tau \in \delta}$ of non-empty *basic type domains* and this induces a non-empty *type domain* D_τ for each type τ such that:

$$(9) \quad \begin{aligned} D_\top &= \{\emptyset\} \\ D_{\tau_1 \& \tau_2} &= D_{\tau_1} \times D_{\tau_2} \\ D_{\tau_1 \rightarrow \tau_2} &= D_{\tau_2}^{D_{\tau_1}} \end{aligned}$$

Semantically labelled semantically active multiplicative rules are given in Figure 2. The term 0 is interpreted as the unique inhabitant of type \top .

2.1 Semantically inactive multiplicatives

Semantically labelled semantically inactive concatenative and intercalative multiplicative rules are given in Figures 3 and 4 respectively.

2.2 Words as types

Let Σ be the lexical vocabulary. The semantic type for types $W(w)$ where $w \in \Sigma^*$ is \top ; the standard syntactical interpretation for types $W(w)$ is:

$$[[W(w)]] =_{\text{def}} \{w\}$$

Notice that in particular the interpretation of the continuous unit I coincides with the interpretation of $W(\epsilon)$, i.e.:

$$[[I]] = [[W(\epsilon)]] = \{\epsilon\}$$

Therefore we postulate the following logical rules related to $W(\epsilon)$:

$$\frac{\Delta\langle \Lambda \rangle \Rightarrow A:\phi}{\Delta\langle W(\epsilon):x \rangle \Rightarrow A:\phi} W(\epsilon)L \quad \frac{}{\Lambda \Rightarrow W(\epsilon):0} W(\epsilon)R$$

In addition:

$$\frac{\Delta\langle W(w_1):x, W(w_2):y \rangle \Rightarrow A:\phi}{\Delta\langle W(w_1+w_2):z \rangle \Rightarrow A:\phi} W(w)L \quad \frac{\Gamma_1 \Rightarrow W(w_1):\phi \quad \Gamma_2 \Rightarrow W(w_2):\psi}{\Gamma_1, \Gamma_2 \Rightarrow W(w_1+w_2):0} W(w)R$$

This added machinery is clearly sound with respect to the syntactical interpretation given to displacement calculus (Morrill et al. [16]). Cut-elimination applies as well.

$$\begin{array}{c}
\frac{\Gamma \Rightarrow A: \phi \quad \Delta \langle \vec{C}: z \rangle \Rightarrow D: \omega}{\Delta \langle \Gamma, \vec{A}: y \rangle \Rightarrow D: \omega \{(y \phi)/z\}} \setminus L \quad \frac{\vec{A}: x, \Gamma \Rightarrow C: \chi}{\Gamma \Rightarrow A \setminus C: \lambda x \chi} \setminus R \\
\\
\frac{\Gamma \Rightarrow B: \psi \quad \Delta \langle \vec{C}: z \rangle \Rightarrow D: \omega}{\Delta \langle \vec{C}/\vec{B}: x, \Gamma \rangle \Rightarrow D: \omega \{(x \psi)/z\}} /L \quad \frac{\Gamma, \vec{B}: y \Rightarrow C: \chi}{\Gamma \Rightarrow C/B: \lambda y \chi} /R \\
\\
\frac{\Delta \langle \vec{A}: x, \vec{B}: y \rangle \Rightarrow D: \omega}{\Delta \langle \vec{A} \bullet \vec{B}: z \rangle \Rightarrow D: \omega \{\pi_1 z/x, \pi_2 z/y\}} \bullet L \\
\\
\frac{\Gamma_1 \Rightarrow A: \phi \quad \Gamma_2 \Rightarrow B: \psi}{\Gamma_1, \Gamma_2 \Rightarrow A \bullet B: (\phi, \psi)} \bullet R \\
\\
\frac{\Delta \langle \Lambda \rangle \Rightarrow A: \phi}{\Delta \langle \vec{T}: x \rangle \Rightarrow A: \phi} IL \quad \frac{}{\Lambda \Rightarrow I: 0} IR \\
\\
\frac{\Gamma \Rightarrow A: \phi \quad \Delta \langle \vec{C}: z \rangle \Rightarrow D: \omega}{\Delta \langle \Gamma \downarrow \vec{A}: y \rangle \Rightarrow D: \omega \{(y \phi)/z\}} \downarrow L \quad \frac{\vec{A}: x \downarrow \Gamma \Rightarrow C: \chi}{\Gamma \Rightarrow A \downarrow C: \lambda x \chi} \downarrow R \\
\\
\frac{\Gamma \Rightarrow B: \psi \quad \Delta \langle \vec{C}: z \rangle \Rightarrow D: \omega}{\Delta \langle \vec{C} \uparrow \vec{B}: x \uparrow \Gamma \rangle \Rightarrow D: \omega \{(x \psi)/z\}} \uparrow L \quad \frac{\Gamma \uparrow \vec{B}: y \Rightarrow C: \chi}{\Gamma \Rightarrow C \uparrow B: \lambda y \chi} \uparrow R \\
\\
\frac{\Delta \langle \vec{A}: x \downarrow \vec{B}: y \rangle \Rightarrow D: \omega}{\Delta \langle \vec{A} \circ \vec{B}: z \rangle \Rightarrow D: \omega \{\pi_1 z/x, \pi_2 z/y\}} \circ L \\
\\
\frac{\Gamma_1 \Rightarrow A: \phi \quad \Gamma_2 \Rightarrow B: \psi}{\Gamma_1 \downarrow \Gamma_2 \Rightarrow A \circ B} \circ R \\
\\
\frac{\Delta \langle 1 \rangle \Rightarrow A: \phi}{\Delta \langle \vec{J}: x \rangle \Rightarrow A: \phi} JL \quad \frac{}{1 \Rightarrow J: 0} JR
\end{array}$$

Fig. 2. Semantically labelled multiplicative rules

3 Syntax and semantics of collocations

3.1 Expletives, discontinuous idioms and particle verbs

Expletives Consider the following lexical entry for the verb *rains* in (1a) which needs the semantically dummy expletive pronoun *it*:

(10) *rains*: $W(it) \text{--} \circ S$: **rain**

$$(11) \frac{W(it): x \Rightarrow W(it): x \quad S: z \Rightarrow S: z}{W(it): x, W(it) \text{--} \circ S: y \Rightarrow S: y} \text{--} \circ L$$

The proof (11) gives the derivational semantics y , and then with the lexical semantics in (10), we get **S: rain**.²

² Two referees object that this treatment requires two types for *seems* to generate both *It seems to rain* and *John seems to smile* whereas if we assume a special N category $N(it)$ we can get by with a single type $\forall x((N(x) \setminus S)/(N(x) \setminus Sinf))$. However, this single syntactic category fails to reflect that the semantics must be treated differently in the two cases.

$$\begin{array}{c}
\frac{\Gamma \Rightarrow A: \phi \quad \Delta \langle \vec{C}: z \rangle \Rightarrow D: \omega}{\Delta \langle \Gamma, \vec{A} \multimap \vec{C}: y \rangle \Rightarrow D: \omega\{y/z\}} \multimap\text{-L} \quad \frac{\vec{A}: x, \Gamma \Rightarrow C: \chi}{\Gamma \Rightarrow A \multimap C: \chi} \multimap\text{-R} \\
\frac{\Gamma \Rightarrow B: \psi \quad \Delta \langle \vec{C}: z \rangle \Rightarrow D: \omega}{\Delta \langle \vec{C} \bullet \vec{B}: x, \Gamma \rangle \Rightarrow D: \omega\{0/z\}} \bullet\text{-L} \quad \frac{\Gamma, \vec{B}: y \Rightarrow C: \chi}{\Gamma \Rightarrow C \bullet B: \lambda y 0} \bullet\text{-R} \\
\frac{\Delta \langle \vec{A}: x, \vec{B}: y \rangle \Rightarrow D: \omega}{\Delta \langle \vec{A} \bullet \vec{B}: z \rangle \Rightarrow D: \omega\{z/y\}} \bullet\text{L} \quad \frac{\Gamma_1 \Rightarrow A: \phi \quad \Gamma_2 \Rightarrow B: \psi}{\Gamma_1, \Gamma_2 \Rightarrow A \bullet B: \psi} \bullet\text{R} \\
\frac{\Gamma \Rightarrow A: \phi \quad \Delta \langle \vec{C}: z \rangle \Rightarrow D: \omega}{\Delta \langle \Gamma, \vec{A} \multimap \vec{C}: y \rangle \Rightarrow D: \omega\{0/z\}} \multimap\text{-L} \quad \frac{\vec{A}: x, \Gamma \Rightarrow C: \chi}{\Gamma \Rightarrow A \multimap C: \lambda x 0} \multimap\text{-R} \\
\frac{\Gamma \Rightarrow B: \psi \quad \Delta \langle \vec{C}: z \rangle \Rightarrow D: \omega}{\Delta \langle \vec{C} \circ \vec{B}: x, \Gamma \rangle \Rightarrow D: \omega\{x/z\}} \circ\text{-L} \quad \frac{\Gamma, \vec{B}: y \Rightarrow C: \chi}{\Gamma \Rightarrow C \circ B: \chi} \circ\text{-R} \\
\frac{\Delta \langle \vec{A}: x, \vec{B}: y \rangle \Rightarrow D: \omega}{\Delta \langle \vec{A} \bullet \vec{B}: z \rangle \Rightarrow D: \omega\{z/x\}} \bullet\text{L} \quad \frac{\Gamma_1 \Rightarrow A: \phi \quad \Gamma_2 \Rightarrow B: \psi}{\Gamma_1, \Gamma_2 \Rightarrow A \bullet B: \phi} \bullet\text{R}
\end{array}$$

Fig. 3. Semantically labelled semantically inactive conc. multiplicative rules

Discontinuous idioms The idiom *gives ... the cold shoulder* in (2) can be generated with the following lexical assignment:

(12) *gives*: $(N \setminus S) / (N \bullet W(\text{the} + \text{cold} + \text{shoulder}))$: **shun**

We are able to generate example (2) in labelled ND format as follows:

$$\frac{\frac{\frac{\frac{\frac{\text{cold}: W(\text{cold}): 0 \quad \text{shoulder}: W(\text{shoulder}): 0}{W(w)I}}{\text{the}: W(\text{the}): 0 \quad \text{cold} + \text{shoulder}: W(\text{cold} + \text{shoulder}): 0}{W(w)I}}{\text{peter}: N: \mathbf{p} \quad \text{the} + \text{cold} + \text{shoulder}: W(\text{the} + \text{cold} + \text{shoulder}): 0}{\bullet I}}{\text{gives}: (N \setminus S) / (N \bullet W(\text{the} + \text{cold} + \text{shoulder}))}: \mathbf{shun} \quad \text{peter} + \text{the} + \text{cold} + \text{shoulder}: N \bullet W(\text{the} + \text{cold} + \text{shoulder}): \mathbf{p}}{/E}}{\text{mary}: N: \mathbf{m} \quad \text{gives} + \text{peter} + \text{the} + \text{cold} + \text{shoulder}: N \setminus S: (\mathbf{shun} \mathbf{p})}{\setminus E}}{\text{mary} + \text{gives} + \text{peter} + \text{the} + \text{cold} + \text{shoulder}: S: (\mathbf{shun} \mathbf{p} \mathbf{m})}$$

Particle verbs Particle verbs like *calls ... up* can be accounted for with the following lexical assignment:

(13) *calls*: $(N \setminus S) / (N \bullet W(\text{up}))$: **phone**

The lexical assignment (13) generates (3a) as follows in labelled ND format:

$$(14) \frac{\frac{\frac{\text{peter}: N: \mathbf{p} \quad \text{up}: W(\text{up}): 0}{\bullet I}}{\text{calls}: (N \setminus S) / (N \bullet W(\text{up}))}: \mathbf{phone} \quad \text{peter} + \text{up}: N \bullet W(\text{up}): \mathbf{p}}{/E}}{\text{mary}: N: \mathbf{m} \quad \text{calls} + \text{peter} + \text{up}: N \setminus S: (\mathbf{phone} \mathbf{p})}{\setminus E}}{\text{mary} + \text{calls} + \text{peter} + \text{up}: S: (\mathbf{phone} \mathbf{p} \mathbf{m})}$$

If one wants to account for the non-determinism of examples (3a) and (3b), we need a more general assignment with a semantically inactive additive, for example:

(15) *calls*: $(N \setminus S) / ((N \bullet W(\text{up})) \sqcup (W(\text{up}) \bullet N))$: **phone**

$$\begin{array}{c}
\frac{\Gamma \Rightarrow A: \phi \quad \Delta \langle \vec{C}: z \rangle \Rightarrow D: \omega}{\Delta \langle \Gamma \downarrow A \downarrow C: y \rangle \Rightarrow D: \omega\{0/z\}} \downarrow_{\sigma} L \quad \frac{\vec{A}: x \downarrow \Gamma \Rightarrow C: \chi}{\Gamma \Rightarrow A \downarrow C: \lambda x 0} \downarrow_{\sigma} R \\
\frac{\Gamma \Rightarrow B: \psi \quad \Delta \langle \vec{C}: z \rangle \Rightarrow D: \omega}{\Delta \langle C \uparrow B: x \downarrow \Gamma \rangle \Rightarrow D: \omega\{x/z\}} \uparrow_{\sigma} L \quad \frac{\Gamma \downarrow \vec{B}: y \Rightarrow C: \chi}{\Gamma \Rightarrow C \uparrow B: \chi} \uparrow_{\sigma} R \\
\frac{\Delta \langle \vec{A}: x \downarrow \vec{B}: y \rangle \Rightarrow D: \omega}{\Delta \langle A \circ B: z \rangle \Rightarrow D: \omega\{z/y\}} \circ_L \quad \frac{\Gamma_1 \Rightarrow A: \phi \quad \Gamma_2 \Rightarrow B: \psi}{\Gamma_1 \downarrow \Gamma_2 \Rightarrow A \circ B: \psi} \circ_R \\
\frac{\Gamma \Rightarrow A: \phi \quad \Delta \langle \vec{C}: z \rangle \Rightarrow D: \omega}{\Delta \langle \Gamma \downarrow A \downarrow C: y \rangle \Rightarrow D: \omega\{y/z\}} \downarrow_{\sigma} L \quad \frac{\vec{A}: x \downarrow \Gamma \Rightarrow C: \chi}{\Gamma \Rightarrow A \downarrow C: \chi} \downarrow_{\sigma} R \\
\frac{\Gamma \Rightarrow B: \psi \quad \Delta \langle \vec{C}: z \rangle \Rightarrow D: \omega}{\Delta \langle C \uparrow B: x \downarrow \Gamma \rangle \Rightarrow D: \omega\{0/z\}} \uparrow_{\sigma} L \quad \frac{\Gamma \downarrow \vec{B}: y \Rightarrow C: \chi}{\Gamma \Rightarrow C \uparrow B: \lambda y 0} \uparrow_{\sigma} R \\
\frac{\Delta \langle \vec{A}: x \downarrow \vec{B}: y \rangle \Rightarrow D: \omega}{\Delta \langle A \circ B: z \rangle \Rightarrow D: \omega\{z/y\}} \circ_L \quad \frac{\Gamma_1 \Rightarrow A: \phi \quad \Gamma_2 \Rightarrow B: \psi}{\Gamma_1 \downarrow \Gamma_2 \Rightarrow A \circ B: \psi} \circ_R
\end{array}$$

Fig. 4. Semantically labelled semantically inactive interc. multiplicative rules

3.2 Interaction with Gapping

We turn now to gapping as like-category coordination. The standard type assignment for gapping (Morrill et al. 2011[16]) in the displacement calculus is the following:³

$$(16) \text{ and: } ((S \downarrow TV) \setminus (S \downarrow TV)) / (S \downarrow TV): \lambda P \lambda Q \lambda z [(Pz) \wedge (Qz)]$$

However, as noted by Kubota and Levine (2012[4]), the lexical assignment (16) cannot generate cases with *complex* transitive verbs like discontinuous idiomatic transitive verbs or particle verbs.

The HTLG account The set of types in HTLG is the smallest set which includes a denumerable set of atomic types closed by the standard Lambek divisions and the vertical slash |. This vertical slash allows HTLG to account for several complex phenomena of discontinuity. In particular, scoping interactions with gapping are elegantly and successfully explained.

HTLG's proof machinery is based on a labelled natural deduction system (LND) with the following constraints:

- Types the principal connective of which is Lambekian are inhabited by concatenations of syntactical (or prosodic) constants.
- Types the principal connective of which is the vertical slash | are inhabited by string lambda terms.

Despite the linguistic flexibility of the LND of HTLG, there are problems with the definition of types like $(CN \setminus CN) / (S \downarrow N)$ for a relative pronoun where the higher order argument subtype

³ We use the defined connective bridge, $\wedge A =_{df} A \circ I$, for which the derived Right rule is:

$$(i) \quad \frac{\Gamma \Rightarrow A: \phi}{\Gamma \downarrow \Lambda \Rightarrow \wedge A: \phi} \wedge R$$

would be a string lambda term and therefore its concatenation is not defined. It follows that the algebra of types of HTLG is not free (M. Moortgat, p.c.) and requires a sorting discipline such as that “|” always outscopes “\” and “/”. In HTLG a type for the relative pronoun would be then $(CN|CN)|(S|N)$.

The lexical entry proposed in HTLG for standard (simple transitive verb) gapping is the following:

$$(17) \lambda\Phi\lambda\Psi\lambda v[(\Psi v)+and+(\Phi \epsilon)]:((S|TV)|(S|TV))|(S|TV):\lambda P\lambda Q\lambda w.[(Q w) \wedge (P w)]$$

For a complex transitive verb like a particle verb, the type entry proposed is:

$$(18) ((S|(TV|N))|(S|(TV|N)))|(S|(TV|N))$$

The type assignment for standard gapping in the displacement calculus is:

$$(19) ((S^\dagger TV)\backslash(S^\dagger TV))/\wedge(S^\dagger TV)$$

If one assumes that in the displacement calculus the type of a complex transitive verb is $(N\backslash S)^\dagger N$ (cf. [16]), there is no possibility to make an extraction of this type as in the case of standard gapping:

$$(20) \frac{N, 1\{N\} \Rightarrow S^\dagger((N\backslash S)^\dagger N)}{N, ((N\backslash S)^\dagger N)\{N\} \Rightarrow S} *$$

Thus, provided it assumes two gapping lexical entries, HTLG is able to account for standard gapping and generalized gapping respectively whereas, apparently, displacement calculus can account for only the former. Here however, using the limited machinery which we have added to the displacement calculus, we will see that we are able to account for gapping with simple and complex transitive verbs, and furthermore do so with a single gapping lexical entry.

Generalized gapping in D with words as types and semantically inactive multiplicatives Here we propose a new gapping coordinator lexical assignment which can generate both gapping with *simple* transitive verbs and gapping with *complex* transitive verbs. Let there be the following abbreviations for types:

$$(21) \begin{aligned} GTV(w) &:= (N\backslash S)/(N\blacklozenge W(w)) \\ X(w) &:= (S^\dagger GTV(w))^\dagger W(w) \end{aligned}$$

We propose the following generalized gapping lexical entry:

$$(22) \text{and: } \forall u((X(u)\backslash X(u))/\sim X(u)): \lambda P\lambda Q\lambda z[(P z) \wedge (Q z)]$$

Notice the crucial use of both continuous and discontinuous semantically inactive connectives, as well as the use of the semantically inactive universal quantifier \forall . With this type for the coordinator we are able to generate the following instances of gapping:

- (23) a. Peter sees Mary and Robin, Clark.
 b. Peter calls up Mary and Robin, Clark.
 c. Peter calls Mary up and Robin, Clark.
 d. Peter gives Mary the cold shoulder and Robin, Clark.

Notice the following important equivalence:

$$(24) (N\backslash S)/(N\blacklozenge W(\epsilon)) \Leftrightarrow TV$$

The proof of $TV \Rightarrow GTV(\epsilon)$ uses the Left rule for $W(\epsilon)$, while the proof of $GTV(\epsilon) \Rightarrow TV$ uses the Right rule for $W(\epsilon)$:

$$(25) \text{ a. } \frac{\frac{\frac{N, TV, N \Rightarrow S}{N, TV, N, W(\epsilon) \Rightarrow S} W(\epsilon)L}{TV, N, W(\epsilon) \Rightarrow N\backslash S} \backslash R}{TV, N\blacklozenge W(\epsilon) \Rightarrow N\backslash S} \bullet L}{TV \Rightarrow GTV(\epsilon) = (N\backslash S)/(N\blacklozenge W(\epsilon))} /R$$

$$\text{ b. } \frac{\frac{\frac{N \Rightarrow N \quad \Lambda \Rightarrow W(\epsilon)}{N \Rightarrow N\blacklozenge W(\epsilon)} \bullet R \quad N\backslash S \Rightarrow N\backslash S}{GTV(\epsilon), N \Rightarrow N\backslash S} /L}{GTV(\epsilon) \Rightarrow (N\backslash S)/N} /R$$

Derivations of generalized gapping Using the lexical entry for generalized gapping (22), we are able to account for standard gapping examples (simple transitives) like (23a) as well as gapping instances with for example a particle verb like (23c). For (23c) we have:

$$(26) \frac{\frac{\frac{N \Rightarrow N \quad W(up) \Rightarrow W(up)}{N, W(up) \Rightarrow N \bullet W(up)} \bullet R \quad \frac{N \Rightarrow N \quad S \Rightarrow S}{N, N \setminus S \Rightarrow S} \setminus L}{\frac{N, GTV(up), N, W(up) \Rightarrow S}{N, 1, N, W(up) \Rightarrow S \downarrow GTV(up)} \downarrow R}{\frac{N, 1, N, 1 \Rightarrow X(up) = (S \downarrow GTV(up)) \downarrow W(up)}{N, N, 1 \Rightarrow \hat{X}(up)} \downarrow R \quad \uparrow R}{\frac{N, N \Rightarrow \sim X(up)}{N, N \Rightarrow \sim X(up)} \wedge R} \uparrow R$$

$$(27) \frac{\frac{\frac{\frac{N \Rightarrow N \quad W(up) \Rightarrow W(up)}{N, W(up) \Rightarrow N \bullet W(up)} \bullet R \quad \frac{N \Rightarrow N \quad S \Rightarrow S}{N, N \setminus S \Rightarrow S} \circ\text{-}L}{\frac{N, GTV(up), N, W(up) \Rightarrow S}{N, 1, N, W(up) \Rightarrow S \downarrow GTV(up)} \downarrow R}{\frac{N, 1, N, 1 \Rightarrow (S \downarrow GTV(up)) \downarrow W(up)}{N, GTV(up), N, X(up) \setminus X(up) \Rightarrow S} \downarrow R \quad \uparrow R}{\frac{GTV(up) \Rightarrow GTV(up) \quad S \Rightarrow S}{(S \downarrow GTV(up)) \downarrow GTV(up) \Rightarrow S} \downarrow L}{\frac{W(up) \Rightarrow W(up) \quad (S \downarrow GTV(up)) \downarrow GTV(up) \Rightarrow S}{(S \downarrow GTV(up)) \downarrow W(up) \downarrow GTV(up) : W(up) \Rightarrow S} \downarrow L}{\frac{N, GTV(up), N, X(up) \setminus X(up) \Rightarrow S}{N, GTV(up), N, X(up) \setminus X(up) \Rightarrow S} \setminus L} \downarrow L$$

$$(28) \frac{\frac{(26)}{\vdots} \quad \frac{(27)}{\vdots}}{\frac{N, N \Rightarrow \sim X(up) \quad N, GTV(up), N, X(up) \setminus X(up) \Rightarrow S}{N, GTV(up), N, W(up), (X(up) \setminus X(up)) / \sim X(up)), N, N \Rightarrow S} /L}{\frac{N, GTV(up), N, W(up), \forall u((X(u) \setminus X(u)) / \sim X(u)), N, N \Rightarrow S}{N, GTV(up), N, W(up), \forall u((X(u) \setminus X(u)) / \sim X(u)), N, N \Rightarrow S} \forall L, u := up}$$

Where lexical lookup gives $GTV(up)$ for *call*, (28) is a derivation of (23c). Using the lexical entry (22) for generalized gapping, one can account for gapping with simple transitive verbs like in sentence (23a), as derivation (29) shows:

$$(29) \frac{\frac{(30)}{\vdots} \quad \frac{(31)}{\vdots}}{\frac{N, N \Rightarrow \sim X(\epsilon) \quad N, TV, N, X(\epsilon) \setminus X(\epsilon) \Rightarrow S}{N, TV, N, (X(\epsilon) \setminus X(\epsilon)) / \sim X(\epsilon)), N, N \Rightarrow S} /L}{\frac{N, TV, N, \forall u((X(u) \setminus X(u)) / \sim X(u)), N, N \Rightarrow S}{N, TV, N, \forall u((X(u) \setminus X(u)) / \sim X(u)), N, N \Rightarrow S} \forall L, u := \epsilon}$$

We show below the derivations (30) and (31) used in derivation (29):

$$(30) \frac{\frac{\frac{N \Rightarrow N \quad W(\epsilon) \Rightarrow W(\epsilon)}{N, W(\epsilon) \Rightarrow N \bullet W(\epsilon)} \bullet R \quad \frac{N \Rightarrow N \quad S \Rightarrow S}{N, N \setminus S \Rightarrow S} \setminus L}{\frac{N, GTV(\epsilon), N, W(\epsilon) \Rightarrow S}{N, 1, N, W(\epsilon) \Rightarrow S \downarrow GTV(\epsilon)} \downarrow R}{\frac{N, 1, N, 1 \Rightarrow X(\epsilon) = (S \downarrow GTV(\epsilon)) \downarrow W(\epsilon)}{N, N, 1 \Rightarrow \hat{X}(\epsilon)} \downarrow R \quad \uparrow R}{\frac{N, N \Rightarrow \sim X(\epsilon)}{N, N \Rightarrow \sim X(\epsilon)} \wedge R} \uparrow R$$

Notice in (31) the sub-derivation (25a) which gives a proof of $TV \Rightarrow GTV(\epsilon)$. Finally, in the appendix we give the derivation for (23d).

$$\begin{array}{c}
 \frac{N \Rightarrow N \quad W(\epsilon) \Rightarrow W(\epsilon)}{N, W(\epsilon) \Rightarrow N \bullet W(\epsilon)} \bullet R \quad \frac{N \Rightarrow N \quad S \Rightarrow S}{N, N \setminus S \Rightarrow S} \setminus L \\
 \hline
 \frac{}{N, GTV(\epsilon), N, W(\epsilon) \Rightarrow S} /L \\
 \frac{}{N, 1, N, W(\epsilon) \Rightarrow S \dagger GTV(\epsilon)} \dagger R \\
 \frac{}{N, 1, N, 1 \Rightarrow (S \dagger GTV(\epsilon)) \dagger W(\epsilon)} \dagger R \\
 \hline
 \frac{}{N, TV, N, X(\epsilon) \setminus X(\epsilon) \Rightarrow S} \setminus L
 \end{array}
 \quad
 \begin{array}{c}
 \frac{}{TV \Rightarrow GTV(\epsilon) \quad S \Rightarrow S} (25a) \\
 \frac{}{S \dagger GTV(\epsilon) \{TV\} \Rightarrow S} \dagger L \\
 \frac{W(\epsilon)R}{\Lambda \Rightarrow W(\epsilon)} \\
 \frac{}{(S \dagger GTV(\epsilon)) \dagger W(\epsilon) \{TV : \Lambda\} \Rightarrow S} \dagger L \\
 \hline
 \frac{}{N, TV, N, X(\epsilon) \setminus X(\epsilon) \Rightarrow S} \setminus L
 \end{array}$$

4 Conclusion

In this paper we have extended (displacement) categorial logic with semantically inactive multiplicatives and a proposal for words as types, innovations which fill existing technical and conceptual gaps in the type logical categorial architecture, while remaining within that architecture, and we have illustrated application to the syntax and semantics of collocations.

The semantically inactive multiplicatives are not indispensable, in that they can be simulated, for example by the usual multiplicatives with vacuous abstraction, just as the other semantically inactive connectives can be simulated by their semantically active counterparts. But the inactive variants provide for more economical grammatical specification. These proposals are implementable in the parser/theorem-prover CatLog (Morrill 2012[13]): the new sequent rules are of the same kind as existing ones and we only require by means of further addition lexical entries $v: W(v): 0$ for each contentless word v of collocations.

We have also shown how the only published examples of HTLG (Kubota and Levine 2012[4]) beyond the scope of the displacement calculus can be *simulated* by this enriched apparatus. HTLG is one of a family of categorial formalisms invoking higher-order syntactic abstraction (Oerhle 1994[19], de Groot 2002[1], Muskens 2003[18]). We think that such abstraction is problematic because parsing a formalism which uses syntactic beta-reduction apparently requires computing higher-order matching, which rapidly becomes intractable. This may explain why no parser yet exists for such formalisms. The displacement calculus with this little extra formal machinery seems to approximate the required expressivity given by HTLG, while maintaining implementability.

References

1. Philippe de Groot. Towards Abstract Categorial Grammars. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL)*, Toulouse, 2001.
2. Mark Hepple. *The Grammar and Processing of Order and Dependency*. PhD thesis, University of Edinburgh, 1990.
3. Gerhard Jäger. *Anaphora and Type Logical Grammar*, volume 24 of *Trends in Logic – Studia Logica Library*. Springer, Dordrecht, 2005.
4. Yusuke Kubota and Robert Levine. Gapping as like-category coordination. In Denis Bchet and Alexander Dikovsky, editors, *Logical Aspects of Computational Linguistics*, volume 7351 of *Lecture Notes in Computer Science*, pages 135–150. Springer Berlin Heidelberg, 2012.
5. 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.
6. J. Lambek. Categorial and Categorical Grammars. In Richard T. Oerhle, Emmon Bach, and Deidre Wheeler, editors, *Categorial Grammars and Natural Language Structures*, volume 32 of *Studies in Linguistics and Philosophy*, pages 297–317. D. Reidel, Dordrecht, 1988.
7. Joachim Lambek. The mathematics of sentence structure. *American Mathematical Monthly*, 65:154–170, 1958. Reprinted in Buszkowski, Wojciech, Wojciech Marciszewski, and Johan van Benthem, editors, 1988, *Categorial Grammar, Linguistic & Literary Studies in Eastern Europe* volume 25, John Benjamins, Amsterdam, 153–172.

8. 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.
9. Michael Moortgat. Categorical 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.
10. G. Morrill. Grammar and Logical Types. In Martin Stockhof and Leen Torenvliet, editors, *Proceedings of the Seventh Amsterdam Colloquium*, pages 429–450, 1990. Also in G. Barry and G. Morrill, editors, *Studies in Categorical Grammar*, Edinburgh Working Papers in Cognitive Science, Volume 5, pages 127–148: 1990. Revised version published as Grammar and Logic, *Theoria*, LXII, 3:260–293, 1996.
11. Glyn Morrill. Intensionality and Boundedness. *Linguistics and Philosophy*, 13(6):699–726, 1990.
12. Glyn Morrill. Categorical 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.
13. Glyn Morrill. CatLog: A Categorical Parser/Theorem-Prover. In *LACL 2012 System Demonstrations*, Logical Aspects of Computational Linguistics 2012, pages 13–16, Nantes, 2012.
14. Glyn Morrill and Josep-Maria Merenciano. Generalising discontinuity. *Traitement automatique des langues*, 37(2):119–143, 1996.
15. 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.
16. 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.
17. Glyn V. Morrill. *Type Logical Grammar: Categorical Logic of Signs*. Kluwer Academic Publishers, Dordrecht, 1994.
18. Reinhard Muskens. Language, lambdas, and logic. In Geert-JanM. Kruijff and RichardT. Oehrle, editors, *Resource-Sensitivity, Binding and Anaphora*, volume 80 of *Studies in Linguistics and Philosophy*, pages 23–54. Springer Netherlands, 2003.
19. Richard T. Oehrle. Term-labeled categorical type systems. *Linguistics and Philosophy*, 17(6):633–678, 1994.
20. Thomas Wasow, Ivan A. Sag, and Geoffrey Nunberg. Idioms: an interim report. In Shiro Hattori and Kazuko Inove, editors, *Proceedings of the XIIIth International Congress of Linguists*, pages 102–15, Tokyo, 1983. CIPL.

Appendix: Derivation of *Peter gives Mary the cold shoulder and Robin, Clark*

$$\begin{array}{c}
\frac{N \Rightarrow N}{N, W(tc_s) \Rightarrow N \bullet W(tc_s)} \text{R} \quad \frac{N \Rightarrow N \quad S \Rightarrow S}{N, N \setminus S \Rightarrow S} \text{L} \\
\hline
\frac{N, GTV(tc_s), N, W(tc_s) \Rightarrow S}{N, 1, N, W(tc_s) \Rightarrow S \downarrow GTV(tc_s)} \text{L} \\
\frac{N, 1, N, 1 \Rightarrow (S \downarrow GTV(tc_s)) \downarrow W(tc_s)}{N, N, 1 \Rightarrow \sim X(tc_s)} \text{R} \\
\hline
\frac{N, 1, N, 1 \Rightarrow \sim X(tc_s)}{N, GTV(tc_s), N, W(tc_s), W(c), W(s), (X(tc_s) \setminus X(tc_s)) / \sim X(tc_s), N, N \Rightarrow S} \text{L} \\
\frac{N, GTV(tc_s), N, W(tc_s), W(c), W(s), \forall u ((X(u) \setminus X(u)) / \sim X(u)), N, N \Rightarrow S}{N, GTV(tc_s), N, W(\theta), W(c), W(s), X(tc_s) \setminus X(tc_s) \Rightarrow S} \text{L} \\
\hline
\frac{N, 1, N, 1 \Rightarrow (S \downarrow GTV(tc_s)) \downarrow W(tc_s)}{N, GTV(tc_s), N, W(tc_s) \Rightarrow S} \text{L} \\
\frac{N, 1, N, W(tc_s) \Rightarrow S \downarrow GTV(tc_s)}{N, 1, N, 1 \Rightarrow (S \downarrow GTV(tc_s)) \downarrow W(tc_s)} \text{R} \\
\frac{W(\theta) \Rightarrow W(\theta) \quad W(c), W(s) \Rightarrow W(tc_s)}{W(\theta), W(c), W(s) \Rightarrow W(tc_s)} \text{R} \\
\frac{W(c) \Rightarrow W(c) \quad W(s) \Rightarrow W(s)}{W(c), W(s) \Rightarrow W(s)} \text{R} \\
\frac{W(\theta) \Rightarrow W(\theta) \quad W(c), W(s) \Rightarrow W(tc_s)}{W(\theta), W(c), W(s) \Rightarrow W(tc_s)} \text{R} \\
\frac{W(\theta), W(c), W(s) \Rightarrow W(tc_s)}{(S \downarrow GTV(tc_s)) \downarrow W(tc_s) | GTV(tc_s) : W(\theta), W(c), W(s) \Rightarrow S} \text{L} \\
\frac{(S \downarrow GTV(tc_s)) \downarrow W(tc_s) | GTV(tc_s) : W(\theta), W(c), W(s) \Rightarrow S}{S \Rightarrow S} \text{L} \\
\frac{S \Rightarrow S}{GTV(tc_s) \Rightarrow GTV(tc_s)} \text{L} \\
\frac{GTV(tc_s) \Rightarrow GTV(tc_s)}{S \Rightarrow S} \text{L} \\
\frac{S \Rightarrow S}{S \Rightarrow S} \text{L}
\end{array}$$