Noname manuscript No.
(will be inserted by the editor)

A Reply to Kubota and Levine on Gapping

Glyn Morrill and Oriol Valentín

Received: date / Accepted: date

Abstract In a series of papers Kubota and Levine give an account of gapping and determiner gapping in terms of hybrid type logical grammar, including anomalous scopal interactions with auxiliaries and negative quantifiers. We make three observations: i) under the counterpart assumptions that Kubota and Levine make, the existent displacement type logical grammar account of gapping already accounts for the scopal interactions, ii) Kubota and Levine overgenerate determiner-verb order permutations in determiner gapping conjuncts whereas the immediate adaptation of their proposal to displacement type logical grammar does not do so, and iii) Kubota and Levine do not capture simplex gapping as a special case of complex gapping, but require distinct lexical entries for the two cases; we show how a generalisation of displacement type logical grammar allows both simplex and discontinuous gapping under a single type assignment.

Keywords gapping, determiner gapping, discontinuous gapping, hybrid type logical grammar, displacement type logical grammar

This research was partially financed by an ICREA Acadèmia to Morrill, MINECO project APCOM (TIN2014-57226-P), and Generalitat de Catalunya project SGR2014-890 (MACDA). We thank two anonymous NLLT referees for comments and suggestions, and Louise McNally for editorial guidance. All errors are our own.

G. Morrill & O. Valentín
Department of Computer Science
Universitat Politècnica de Catalunya
Barcelona 08034
Tel.: +34 93 413 7843
Fax: +34 413 7833
E-mail: morrill@cs.upc.edu & oriol.valentin@gmail.com

1 Introduction

Consider the following four types of gapping: simple gapping, discontinuous gapping, determiner gapping, and discontinuous determiner gapping.

(1) a. Leslie met Sandy and Robin (met) Bill.
b. John wants Watford to win and Daniel (wants) Chelsea (to win).

(2) a. Some dogs like Whiskas and (some) cats (like) Alpo.
b. Every cook wants Barça to win and (every) waiter (wants) Madrid (to win).

Kubota and Levine (2012[2]; 2013[3]; 2015[4]; henceforth K&L) develop an account of gapping in hybrid type logical grammar (HTLG), an extension of Lambek calculus admitting functional expressions in the phonological component.[1] K&L (2015[4]) provide a review of the literature and argue in broad terms the advantages of an analysis of gapping as hypothetical reasoning, to which we have nothing to add; we in turn review their type logical proposal. The Lambek rules of HTLG are as follows:[2]

$$(3)\quad \frac{\begin{array}{cc}\vdots & \vdots\\ \alpha\colon C/B\colon \phi & \beta\colon B\colon \psi\end{array}}{\alpha{+}\beta\colon C\colon (\phi\ \psi)}\,/E \qquad \frac{\begin{array}{cc}\vdots & \vdots\\ \alpha\colon A\colon \phi & \beta\colon A\backslash C\colon \psi\end{array}}{\alpha{+}\beta\colon C\colon (\psi\ \phi)}\,\backslash E$$

$$\frac{\begin{array}{c}\overline{b\colon B\colon y}\ n\\ \vdots\\ \alpha{+}b\colon C\colon \chi\end{array}}{\alpha\colon C/B\colon \lambda y\chi}\,/I^n \qquad \frac{\begin{array}{c}\overline{a\colon A\colon x}\ n\\ \vdots\\ a{+}\beta\colon C\colon \chi\end{array}}{\beta\colon A\backslash C\colon \lambda x\chi}\,\backslash I^n$$

In these rules $\alpha\colon A\colon \phi$ signifies a sign with phonology/prosodics α, syntactic type A, and semantics ϕ. The elimination (E) rules combine signs by concatenation prosodically, *modus ponens* type-logically, and functional application semantically; the introduction (I) rules of hypothetical reasoning conditionalise assumptions: when passing from premise to conclusion the assumption coindexed with the rule name n becomes closed. Every rule instance has a fresh index. The introduction rules are semantically interpreted by functional

[1] An anonymous referee questions whether gapping is after all a purely combinatoric phenomenon citing split antecedent gapping (i), and non-ATB gapping (ii):

i) Sue goes running 6 times a week, and Alex lifts weights 3 times a week, but neither every day.
ii) Either Pat came with Chris and Sandy came with Kim, or Pat with Kim and the others were alone.

We cannot enter fully into this question here except to note that such examples do not show that there is *no* combinatoric component to gapping, but rather that it is a more generalised phenomenon in the case of iterated coordination, which we do not address here.

[2] We make some notational adjustments in order to smoothen comparison of hybrid TLG and displacement TLG.

abstraction. To these directional rules HTLG adds inference rules for a nondirectional type constructor | interpreted by phonological functional application and phonological functional abstraction:

$$(4)\quad \frac{\begin{array}{cc}\vdots & \vdots\\ \alpha\colon C|B\colon \phi & \beta\colon B\colon \psi\end{array}}{\alpha(\beta)\colon C\colon (\phi\ \psi)}\,|E \qquad \frac{\begin{array}{c}\overline{v\colon B\colon y}\,n\\ \vdots\\ \alpha\colon C\colon \chi\end{array}}{\lambda v.\alpha\colon C|B\colon \lambda y\chi}\,|I^n$$

These are the characteristic rules of HTLG (for more details we refer the reader to the papers of Kubota and Levine).

In relation to gapping, K&L (2015[4]) (53) present a type assignment which is in essence:[3]

(5) $\lambda\sigma_2\lambda\sigma_1\lambda\phi.\sigma_1(\phi)$+**and**+$\sigma_2(0)$: $(X|X)|X$: $\lambda x\lambda y\lambda z[(y\ z)\wedge(x\ z)]$
where $X = S|(VP/N)$

And they derive from this simple gapping such as (1a); see *ibid* (52) and (55). In our notation, their (52) is:

$$(6)\quad \frac{\dfrac{\dfrac{}{\phi\colon VP/N\colon x}\,i \quad \textbf{bill}\colon N\colon b}{\phi\text{+}\textbf{bill}\colon VP\colon (x\ b)}\,E/ \quad \textbf{robin}\colon N\colon r}{\textbf{robin}\text{+}\phi\text{+}\textbf{bill}\colon S\colon ((x\ b)\ r)}\,E\backslash}{\lambda\phi.\textbf{robin}\text{+}\phi\text{+}\textbf{bill}\colon S|(VP/N)\colon \lambda x((x\ b)\ r)}\,I|^i$$

Continuing the derivation as in their (55) yields:

(7) **leslie** + **met** + **sandy** + **and** + **robin** + **bill**: S: $((met\ s)\ l)\wedge((met\ b)\ r)$

K&L also assume in their (56) a raised type for an auxiliary:

(8) $\lambda\sigma.\sigma$(**must**): $S|(S|(VP/VP))$: $\lambda x(Nec\ (x\ \lambda yy))$

They show that such assignments to auxiliaries license the auxiliary wide-scope reading of, say, *John must eat steak and Mary pizza*.

Likewise, K&L (2015[4]) (66) present a type assignment for determiner gapping which is:

(9) $\lambda\rho_2\lambda\rho_1\lambda\phi\lambda\sigma.\rho_1(\phi)(\sigma)$+**and**+$\rho_2(0)(\lambda\chi\lambda\psi.\psi(\chi))$: $(X|X)|X$:
$\lambda x\lambda y\lambda z\lambda w[((y\ z)\ w)\wedge((x\ z)\ w)]$
where $X = ((S|((S|(S|N))|CN)|(VP/N)$

And they assume, K&L (2015[4]) (65), a raised type for a negative determiner of the form:

(10) $\lambda\rho(\lambda\phi\lambda\sigma.\sigma(\textbf{no}\text{+}\phi))$: $S|(S|((S|(S|N))|CN))$: $\lambda x(\neg(x\ \lambda y\lambda z\exists w[(y\ w)\wedge(z\ w)]))$

These enable derivation of the split scope reading of:

(11) No dog eats whiskas or cat alpo.

[3] Throughout, *VP* abbreviates $N\backslash S$. We limit attention to gapping of the transitive verb category; so far as we are aware gapping in other categories raises no new issues differentiating between hybrid TLG and displacement TLG.

2 The scopal anomalies are already available in displacement TLG

K&L (2015[4]) state that 'our own analysis resembles most closely Morrill et al.'s (2011) (which is a refinement of Hendriks 1995)'. The displacement type logical grammar (DTLG) gapping assignment is as follows, where $\odot$ is discontinuous product, which is semantically interpreted by ordered pairing, and '$\pi_1 x$' selects the first component of x:[4]

(12) **and**: $(X\backslash X)/(X\odot I)$: $\lambda x\lambda y\lambda z[(y\ z) \wedge (\pi_1 x\ z)]$
where $X = S{\uparrow}(VP/N)$

Here, the symbols such as I and $\uparrow$ are connectives of the displacement calculus, which we define later. The corresponding determiner gapping assignment would be:

(13) **and**: $(X\backslash X)/((X\odot I)\odot I)$: $\lambda x\lambda y\lambda z\lambda w[((y\ z)\ w) \wedge ((\pi_1\pi_1 x\ z)\ w)]$
where $X = (S{\uparrow}(VP/N)){\uparrow}(((S{\uparrow}N){\downarrow}S)/CN)$

The auxiliary and negative determiner assignments would be:

(14) **must**: $(S{\uparrow}(VP/VP)){\downarrow}S$: $\lambda x(Nec\ (x\ \lambda yy))$

(15) **no**: $(S{\uparrow}(((S{\uparrow}N){\downarrow}S)/CN)){\downarrow}S$: $\lambda x(\neg(x\ \lambda y\lambda z\exists w[(y\ w) \wedge (z\ w)]))$

Thus the analyses of gapping scope anomalies are already available in DTLG. Indeed K&L (2015[4]) state, 'To be fair, the core of our empirical results, so far as we can tell, seems to straightforwardly carry over to Morrill et al.'s (2011) system.' So K&L's type logical contribution is the determiner gapping, and the observation that raised auxiliary and negative determiner assignments capture the scopal anomalies in HTLG, but also in DTLG. Thus, although K&L couch their solution in terms of HTLG, HTLG and DTLG are on a par in respect of gapping and the scopal anomalies. Since this point appears to have been granted we do not elaborate on it further. But we will see a respect in which DTLG improves on HTLG, and another respect in which DTLG can be made to improve further on HTLG.

3 The HTLG determiner gapping assignment overgenerates

Kubota (p.c.) points out that the HTLG analysis of determiner gapping incorrectly predicts examples such as the following, where the determiner and the transitive verb orders are not consistent in the conjuncts:

(16) a. ∗Some dogs like Whiskas and I (like) (some) cats.
b. ∗I like some cats and (some) dogs (like) Whiskas.

This permutation overgeneration arises because the $|(((S|(S|N))|CN)$ and $|(VP/N)$ arguments in the types for determiner gapping do not distinguish the linear

[4] The notation π_1 (and π_2) represents first (and second) projection of an ordered pair, so that e.g. $\pi_1(\phi, \psi) = \phi$.

order of the phonological variables they bind. Thus, for example, in relation to (16a), both $\lambda v \lambda q.q$+**dogs**+v+**Whiskas** and $\lambda v \lambda q$.**I**+v+q+**cats** have type $((S|((S|(S|N))|CN)|(VP/N)$. Note that the issue has nothing to do with the order in which the quantifier and transitive verb arguments are abstracted, but rather with the left to right position of the variables they bind in the body of the phonological term, to which the type constructor | is insensitive. DTLG does not overgenerate in this way because the positions of discontinuity are indexed for left to right order.

4 HTLG requires distinct type assignments for simplex and discontinuous gapping

Our analysis is inspired by that of Kubota and Levine, which assigns coordinator conjunct types and phonology as follows:

conjunt type & coordinator phonology	gapping	determiner gapping
simplex	$S\|(VP/N)$ $\lambda\sigma_2\lambda\sigma_1\lambda\phi.\sigma_1(\phi)$+**and**+$\sigma_2(0)$ K&L (2012[2]) (5) K&L (2013[3]) (13) K&L (2015[4]) (53)	$(S\|(VP/N))\|((S\|(S\|N))\|CN)$ $\lambda\rho_2\lambda\rho_1\lambda\phi\lambda\sigma.\rho_1(\phi)(\sigma)$+**and**+$\rho_2(\lambda\chi\lambda\psi.\psi(\chi))(0)$ K&L (2013[3]) (24) K&L (2015[4]) (66) (83)
discont.	$S\|(VP\|N)$ $\lambda\rho_2\lambda\rho_1\lambda\phi.\rho_1(\phi)$+**and**+$\rho_2(\lambda\pi.\pi)$ K&L (2012[2]) (20)	$(S\|(VP\|N))\|((S\|(S\|N))\|CN)$ $\lambda\rho_2\lambda\rho_1\lambda\phi\lambda\sigma.\rho_1(\phi)(\sigma)$+**and**+$\rho_2(\lambda\chi\lambda\psi.\psi(\chi))(\lambda\omega.\omega)$ (by extrapolation)

However, our analysis of gapping represents an improvement on the HTLG analysis in that we require only a single type for simple and discontinuous gapping. HTLG requires two types because simplex gapping conditionalises VP/N of sort *string* whereas discontinuous gapping conditionalises $VP|N$ of sort *function*, and these require two distinct phonological operations: application to the empty string 0 (of sort *string*) and to the identity function $\lambda\pi.\pi$ (of sort *function*) respectively.

5 Our account

We give an account of simple and discontinuous gapping as in (1) and simplex and discontinuous determiner gapping as in (2) which is optimal in that a single coordinator type generates discontinuous gapping with simple gapping as a special case and a single coordinator type generates discontinuous determiner gapping with simplex determiner gapping as a special case. The account is expressed in an extension of displacement calculus (**D**; Morrill et al. 2011[5]) which we call second-order displacement calculus $\mathbf{D}^2$.

5.1 Second-order displacement calculus

Displacement calculus is a logic of discontinuous strings. By discontinuous strings we mean strings punctuated by a distinguished vocabulary item '1' called the *separator*. In contrast to HTLG, the phonological terms of displacement calculus have no lambda abstraction and, instead of a set of variable placeholders, DTLG has a single placeholder, the separator. The sort of a discontinuous string is the number of separators it contains. We notate by $L_i, i \geq 0$, the set of all strings of sort i with respect to some alphabet. We consider the operations concatenation, intercalation, and adjunction on discontinuous strings. Concatenation is represented in (17).

(17) $\boxed{\alpha} + \boxed{\beta}$
$=$
$\boxed{\alpha}\boxed{\beta}$

concatenation $+ : L_i, L_j \to L_{i+j}$

For example, the concatenation of **Leslie**+1+**Sandy** and **and**+**Robin**+**Bill** is:

(18) **Leslie**+1+**Sandy** + **and**+**Robin**+**Bill**
$=$
Leslie+1+**Sandy**+**and**+**Robin**+**Bill**

Intercalation is represented in (19):

(19) $\boxed{\alpha}\boxed{1}\boxed{\gamma} \times_k \boxed{\beta}$
$=$
$\boxed{\alpha}\boxed{\beta}\boxed{\gamma}$

intercalation $\times_k : L_{i+1}, L_j \to L_{i+j}$

For example, the intercalation at the second separator of 1+**dogs**+1+**Whiskas**+**and**+**cats**+**Alpo** and **like** is:

(20) 1+**dogs**+1+**Whiskas**+**and**+**cats**+**Alpo** $\times_2$ **like**
$=$
1+**dogs**+**like**+**Whiskas**+**and**+**cats**+**Alpo**

Finally, adjunction is represented in (21):

(21) $\boxed{\alpha}\boxed{1}\boxed{\gamma}\boxed{1}\boxed{\epsilon} \wedge_{k,l} \boxed{\beta}\boxed{1}\boxed{\delta}$
$=$
$\boxed{\alpha}\boxed{\beta}\boxed{\gamma}\boxed{\delta}\boxed{\epsilon}$

adjunction $\wedge_{k,l} : L_{i+2}, L_{j+1} \to L_{i+j}$

For example, the adjunction at the first separators of **John**+1+**Watford**+1+**and**+**Daniel**+**Chelsea** and **wants**+1+**to**+**win** is:

(22) **John**+1+**Watford**+1+**and**+**Daniel**+**Chelsea** $^{\wedge_{1,1}}$ **wants**+1+**to**+**win**
=
John+**wants**+**Watford**+**to**+**win**+**and**+**Daniel**+**Chelsea**

We will have three families of type-constructors defined in relation to the three prosodic operations of concatenation, intercalation, and adjunction. The syntactic types are sorted $Tp_0, Tp_1, Tp_2, \ldots$ according to the number of points of discontinuity $0, 1, 2, \ldots$ their expressions contain. The sets Tp_i of types of sort i are defined by mutual recursion in terms of sets $\mathcal{P}_i$ of primitive types of sort i as follows:

$$Tp_i ::= \mathcal{P}_i$$

Tp_i	$::= Tp_{i+j}/Tp_j$	$T(C/B)$	$= T(B)\rightarrow T(C)$	over
Tp_j	$::= Tp_i\backslash Tp_{i+j}$	$T(A\backslash C)$	$= T(A)\rightarrow T(C)$	under
Tp_{i+j}	$::= Tp_i \bullet Tp_j$	$T(A\bullet B)$	$= T(A)\&T(B)$	continuous product
Tp_0	$::= I$	$T(I)$	$= \top$	cont. unit
Tp_{i+1}	$::= Tp_{i+j}\uparrow_k Tp_j, 1 \le k \le i{+}1$	$T(C\uparrow_k B)$	$= T(B)\rightarrow T(C)$	extract
Tp_j	$::= Tp_{i+1}\downarrow_k Tp_{i+j}, 1 \le k \le i{+}1$	$T(A\downarrow_k C)$	$= T(A)\rightarrow T(C)$	infix
Tp_{i+j}	$::= Tp_{i+1}\odot_k Tp_j, 1 \le k \le i{+}1$	$T(A\odot_k B)$	$= T(A)\&T(B)$	disc. product
Tp_1	$::= J$	$T(J)$	$= \top$	disc. unit
Tp_{i+2}	$::= Tp_{i+j}\Uparrow_{k,l} Tp_{j+1}, 1 \le k \le i{+}1, 1 \le l \le j{+}1$	$T(C\Uparrow_{k,l} B)$	$= T(B)\rightarrow T(C)$	2nd order extract
Tp_{j+1}	$::= Tp_{i+2}\Downarrow_{k,l} Tp_{i+j}, 1 \le k \le i{+}1, 1 \le l \le j{+}1$	$T(A\downarrow_{k,l} C)$	$= T(A)\rightarrow T(C)$	2nd order infix
Tp_{i+j}	$::= Tp_{i+2}\circledcirc_{k,l} Tp_{j+1}, 1 \le k \le i{+}1, 1 \le l \le j{+}1$	$T(A\circledcirc_{k,l} B)$	$= T(A)\&T(B)$	2nd order disc. product
Tp_2	$::= K$	$T(K)$	$= \top$	2nd order disc. unit

The second column of this table shows the standard categorial semantic type map for the connectives.[5] Each type of sort i is interpreted as a set of (discontinuous) strings of sort i. The prosodic interpretation is as follows:

$$\begin{aligned}
[[C/B]] &= \{s_1 |\ \forall s_2 \in [[B]], s_1{+}s_2 \in [[C]]\}\\
[[A\backslash C]] &= \{s_2 |\ \forall s_1 \in [[A]], s_1{+}s_2 \in [[C]]\}\\
[[A\bullet B]] &= \{s_1{+}s_2 |\ s_1 \in [[A]] \ \&\ s_2 \in [[B]]\}\\
[[I]] &= \{0\}\\
\\
[[C\uparrow_k B]] &= \{s_1 |\ \forall s_2 \in [[B]], s_1 \times_k s_2 \in [[C]]\}\\
[[A\downarrow_k C]] &= \{s_2 |\ \forall s_1 \in [[A]], s_1 \times_k s_2 \in [[C]]\}\\
[[A\odot_k B]] &= \{s_1 \times_k s_2 |\ s_1 \in [[A]] \ \&\ s_2 \in [[B]]\}\\
[[J]] &= \{1\}\\
\\
[[C\Uparrow_{k,l} B]] &= \{s_1 |\ \forall s_2 \in [[B]], s_1 {}^{\wedge_{k,l}} s_2 \in [[C]]\}\\
[[A\Downarrow_{k,l} C]] &= \{s_2 |\ \forall s_1 \in [[A]], s_1 {}^{\wedge_{k,l}} s_2 \in [[C]]\}\\
[[A\circledcirc_{k,l} B]] &= \{s_1 {}^{\wedge_{k,l}} s_2 |\ s_1 \in [[A]] \ \&\ s_2 \in [[B]]\}\\
[[K]] &= \{1{+}1{+}1\}
\end{aligned}$$

Although linguistically only some of the power, and hence only some of the rules, are necessary here, the framework of DTLG complies with the modern

[5] For example, if $T(N) = e$ where e is the semantic type for individuals, and $T(S) = t$ where t is the semantic type corresponding to truth-values, we have then that $T(N\backslash S) = e \rightarrow t$.

logical paradigm of logic as an interpreted formal language, and aspiration to *soundness* (that everything said is true) and *completeness* (that everything true is said). This, to us, is the rock on which type *logical* grammar is founded. Thus, we present here all the rules so that the reader has the complete picture of which the gapping analysis uses just a part.

The rules for second-order displacement calculus fall into three groups for the concatenative, intercalative, and adjunctive connective families. Each family contains four connectives: the two implicational residuals, the conjunctive product, and product unit. Each connective has two rules, namely a rule of elimination (E), eliminating the connective reading from premise to conclusion, and a rule of introduction (I) introducing the connective reading from premise to conclusion.[6] Although there are many rules, the reader should be aware of the high degree of symmetry between them, and that the rules simply formalise the necessary and sufficient conditions for membership of syntactic types: the rules are essentially the result of restating the interpretation clauses given above.

The labelled natural deduction for second-order displacement calculus is as follows, where we use three conventions. Firstly, where $\mathbf{a}$ is a syntactical constant of sort i, the *vector* $\overrightarrow{\mathbf{a}}$ is $\mathbf{a}_0{+}1{+}\mathbf{a}_1{+}1{+}\cdots{+}1{+}\mathbf{a}_i$; for example, if $\mathbf{a}$ is of sort 1, $\overrightarrow{\mathbf{a}} = \mathbf{a}_0{+}1{+}\mathbf{a}_1$. Secondly, where α is a discontinuous string of sort $i > 0$, $\alpha\,|_k\,\beta$, $1 \leq k \leq i$, is the result of replacing the kth separator in α by β (counting from the left); for example **John**+1+**Watford**+1+**and**+**Daniel**+**Chelsea** $|_2$ **to**+**win** = **John**+1+**Watford**+**to**+**win**+**and**+**Daniel**+**Chelsea**. Thirdly, $\alpha\,\|_k\,\beta$ abbreviates $\alpha\,|_k\,1{+}\beta{+}1$; for example **John**+1+**and**+**Daniel**+**Chelsea** $\|_1$ **Watford** = **John**+1+**Watford**+1+**and**+**Daniel**+**Chelsea**.

Continuous family:

– Elimination rules for implications

$$\frac{\begin{array}{cc}\vdots & \vdots\\ \alpha\colon C/B\colon \phi & \beta\colon B\colon \psi\end{array}}{\alpha{+}\beta\colon C\colon (\phi\ \psi)}\,/E \qquad \frac{\begin{array}{cc}\vdots & \vdots\\ \alpha\colon A\colon \phi & \beta\colon A\backslash C\colon \psi\end{array}}{\alpha{+}\beta\colon C\colon (\psi\ \phi)}\,\backslash E$$

– Elimination rule for product

$$\frac{\begin{array}{cc} & \overline{\overrightarrow{a}\colon A\colon x}^{\,n} \quad \overline{\overrightarrow{b}\colon B\colon y}^{\,n}\\ & \vdots\\ \gamma\colon A\bullet B\colon \chi & \delta(\overrightarrow{a}{+}\overrightarrow{b})\colon D\colon \omega(x,y)\end{array}}{\delta(\gamma)\colon D\colon \omega(\pi_1\chi,\pi_2\chi)}\,\bullet E^n$$

[6] Except we omit the elimination rules for product units which, as well as being unmotivated linguistically, are awkward to formulate in the format used here.

– Introduction rules for implications

$$\dfrac{\begin{array}{c}\overrightarrow{b}:B:y\ ^{n}\\ \vdots\\ \alpha+\overrightarrow{b}:C:\chi\end{array}}{\alpha:C/B:\lambda y\chi}\ /I^{n} \qquad \dfrac{\begin{array}{c}\overrightarrow{a}:A:x\ ^{n}\\ \vdots\\ \overrightarrow{a}+\beta:C:\chi\end{array}}{\beta:A\backslash C:\lambda x\chi}\ \backslash I^{n}$$

– Introduction rules for product and product unit

$$\dfrac{\begin{array}{cc}\vdots & \vdots\\ \alpha:A:\phi & \beta:B:\psi\end{array}}{\alpha+\beta:A\bullet B:(\phi,\psi)}\ \bullet I \qquad \dfrac{}{0:I:0}\ II$$

Discontinuous family:

– Elimination rules for implications

$$\dfrac{\begin{array}{cc}\vdots & \vdots\\ \alpha:C\uparrow_k B:\phi & \beta:B:\psi\end{array}}{\alpha\mid_k\beta:C:(\phi\ \psi)}\ \uparrow E \qquad \dfrac{\begin{array}{cc}\vdots & \vdots\\ \alpha:A:\phi & \beta:A\downarrow_k C:\psi\end{array}}{\alpha\mid_k\beta:C:(\psi\ \phi)}\ \downarrow E$$

– Elimination rule for product

$$\dfrac{\begin{array}{ccc} & \overrightarrow{a}:A:x\ ^{n} & \overrightarrow{b}:B:y\ ^{n}\\ \vdots & \vdots & \\ \gamma:A\odot B:\chi & \delta(\overrightarrow{a}\mid_k\overrightarrow{b}):D:\omega(x,y) & \end{array}}{\delta(\gamma):D:\omega(\pi_1\chi,\pi_2\chi)}\ \odot E^{n}$$

– Introduction rules for implications

$$\dfrac{\begin{array}{c}\overrightarrow{b}:B:y\ ^{n}\\ \vdots\\ \alpha\mid_k\overrightarrow{b}:C:\chi\end{array}}{\alpha:C\uparrow_k B:\lambda y\chi}\ \uparrow I^{n} \qquad \dfrac{\begin{array}{c}\overrightarrow{a}:A:x\ ^{n}\\ \vdots\\ \overrightarrow{a}\mid_k\beta:C:\chi\end{array}}{\beta:A\downarrow_k C:\lambda x\chi}\ \downarrow I^{n}$$

– Introduction rules for product and product unit

$$\frac{\begin{array}{cc}\vdots & \vdots\\ \alpha\colon A\colon \phi & \beta\colon B\colon \psi\end{array}}{\alpha\mid_k\beta\colon A\odot_k B\colon(\phi,\psi)}\bullet I \qquad \frac{}{1\colon J\colon 0}\,IJ$$

Second-order discontinuous family:

– Elimination rules for implications

$$\frac{\begin{array}{cc}\vdots & \vdots\\ \alpha\parallel_k\gamma\colon C\Uparrow_{k,l}B\colon\phi & \beta\colon B\colon\psi\end{array}}{\alpha\mid_k(\beta\mid_l\gamma)\colon C\colon(\phi\ \psi)}\Uparrow E \qquad \frac{\begin{array}{cc}\vdots & \vdots\\ \alpha\parallel_k\gamma\colon A\colon\phi & \beta\colon A\Downarrow_{k,l}C\colon\psi\end{array}}{\alpha\mid_k(\beta\mid_l\gamma)\beta\colon C\colon(\psi\ \phi)}\Downarrow E$$

– Elimination rule for product

$$\frac{\begin{array}{cc}\vdots & \overline{\overrightarrow{a}\parallel_k\overrightarrow{c}\colon A\colon x}\,n \quad \overline{\overrightarrow{b}\colon B\colon y}\,n\\ \gamma\colon A\circledcirc_{k,l}B\colon\chi & \delta(\overrightarrow{a}\mid_k(\overrightarrow{b}\mid_l\overrightarrow{c}))\colon D\colon\omega(x,y)\end{array}}{\delta(\gamma)\colon D\colon\omega(\pi_1\chi,\pi_2\chi)}\circledcirc E^n$$

– Introduction rules for implications

$$\frac{\begin{array}{c}\overline{\overrightarrow{b}\colon B\colon y}\,n\\ \vdots\\ \alpha\mid_k(\overrightarrow{b}\mid_l\gamma)\colon C\colon\chi\end{array}}{\alpha\parallel_k\gamma\colon C\Uparrow_{k,l}B\colon\lambda y\chi}\Uparrow I^n \qquad \frac{\begin{array}{c}\overline{\overrightarrow{a}\mid_l\overrightarrow{c}\colon A\colon x}\,n\\ \vdots\\ \overrightarrow{a}\mid_k(\beta\mid_l\overrightarrow{c})\colon C\colon\chi\end{array}}{\beta\colon A\Downarrow_{k,l}C\colon\lambda x\chi}\Downarrow I^n$$

– Introduction rules for product and product unit

$$\frac{\begin{array}{cc}\vdots & \vdots\\ \alpha\parallel_k\gamma\colon A\colon\phi & \beta\colon B\colon\psi\end{array}}{\alpha\mid_k(\beta\mid_l\gamma)\colon A\circledcirc_{k,l}B\colon(\phi,\psi)}\bullet E \qquad \frac{}{1{+}1{+}1\colon K\colon 0}\,IK$$

We adopt the convention that when subscripts k and l are omitted they are 1, i.e. they default to 1.

By way of example, the following auxiliary derivation shows that a subject followed by an object has type $S\Uparrow(VP\uparrow N)$:

$$(23)\quad \dfrac{\dfrac{\mathbf{sbj}: N: sbj \qquad \dfrac{\overline{}^{\,i}}{a_0{+}1{+}a_1: VP{\uparrow}N: x}\ \mathbf{obj}: N: obj}{a_0{+}\mathbf{obj}{+}a_1: VP: (x\ obj)}\uparrow E}{\mathbf{sbj}{+}a_0{+}\mathbf{obj}{+}a_1: S: ((x\ obj)\ sbj)}\backslash E}{\mathbf{sbj}{+}1{+}\mathbf{obj}{+}1: S{\Uparrow}(VP{\uparrow}N): \lambda x((x\ obj)\ sbj)}\Uparrow I^{i}$$

5.2 Analyses

The account of gapping consists in an assignment to the coordinator of the following type:

(24) **and**: $(X\backslash X)/(X{\circledcirc}J)$: $\lambda x\lambda y\lambda z[(y\ z) \wedge (\pi_1 x\ z)]$
where $X = S{\Uparrow}(VP{\uparrow}N)$

Consider example (1a) of simple gapping: *Leslie met Sandy and Robin Bill*. Then there is the following derivation of (1a) using (23):

$$\dfrac{\dfrac{\text{Leslie Sandy}}{\begin{array}{c}\mathbf{L}{+}1{+}\mathbf{S}{+}1:\\ S{\Uparrow}(VP{\uparrow}N):\\ \lambda x((x\ s)\ l)\end{array}} \qquad \dfrac{\begin{array}{c}\mathbf{and}:\\ ((S{\Uparrow}(VP{\uparrow}N))\backslash(S{\Uparrow}(VP{\uparrow}N)))/\\ /((S{\Uparrow}(VP{\uparrow}N)){\circledcirc}J):\\ \lambda x\lambda y\lambda z[(y\ z) \wedge (\pi_1 x\ z)]\end{array} \qquad \dfrac{\dfrac{\text{Robin Bill}}{\begin{array}{c}\mathbf{R}{+}1{+}\mathbf{B}{+}1:\\ S{\Uparrow}(VP{\uparrow}N):\\ \lambda x((x\ b)\ r)\end{array}} \quad \begin{array}{c}JR\\ 1{:}\,J{:}\,0\end{array}}{\mathbf{R}{+}\mathbf{B}: (S{\uparrow}(VP/N)){\circledcirc}J: (\lambda x((x\ b)\ r), 0)}{\circledcirc}I}{\mathbf{and}{+}\mathbf{R}{+}\mathbf{B}: (S{\Uparrow}(VP{\uparrow}N))\backslash(S{\Uparrow}(VP{\uparrow}N)): \lambda y\lambda z[(y\ z) \wedge ((z\ b)\ r)]}/E}{\mathbf{L}{+}1{+}\mathbf{S}{+}1{+}\mathbf{and}{+}\mathbf{R}{+}\mathbf{B}: S{\Uparrow}(VP{\uparrow}N): \lambda z[((z\ s)\ l) \wedge ((z\ b)\ r)]}\backslash E \qquad \dfrac{\dfrac{\dfrac{\begin{array}{c}\mathbf{met}: VP/N: meet\\ \end{array} \quad \begin{array}{c}\overline{}^{\,i}\\ a: N: x\end{array}}{\mathbf{met}{+}a: VP: (meet\ x)}/E}{\mathbf{met}{+}1: VP{\uparrow}N: \lambda x(meet\ x)}\uparrow I^{i}}{\mathbf{L}{+}\mathbf{met}{+}\mathbf{S}{+}\mathbf{and}{+}\mathbf{R}{+}\mathbf{B}: S: [((meet\ s)\ l) \wedge ((meet\ b)\ r)]}\Uparrow E$$

And from the same coordinator type assignment (24) there is the following derivation of the discontinuous gapping (1b) *John wants Watford to win and Daniel Chelsea*:

$$\dfrac{\dfrac{\text{John Watford}}{\begin{array}{c}\mathbf{J}{+}1{+}\mathbf{W}{+}1:\\ S{\Uparrow}(VP{\uparrow}N):\\ \lambda x((x\ w)\ j)\end{array}} \qquad \dfrac{\begin{array}{c}\mathbf{and}:\\ ((S{\Uparrow}(VP{\uparrow}N))\backslash(S{\Uparrow}(VP{\uparrow}N)))/\\ /((S{\Uparrow}(VP{\uparrow}N)){\circledcirc}J):\\ \lambda x\lambda y\lambda z[(y\ z) \wedge (\pi_1 x\ z)]\end{array} \qquad \dfrac{\dfrac{\text{Daniel Chelsea}}{\begin{array}{c}\mathbf{D}{+}1{+}\mathbf{S}{+}1:\\ S{\Uparrow}(VP{\uparrow}N):\\ \lambda x((x\ s)\ d)\end{array}} \quad \begin{array}{c}JR\\ 1{:}\\ J{:}\\ 0\end{array}}{\begin{array}{c}\mathbf{D}{+}\mathbf{C}:\\ (S{\uparrow}(VP/N)){\circledcirc}J:\\ (\lambda x((x\ c)\ d), 0)\end{array}}{\circledcirc}I}{\mathbf{and}{+}\mathbf{D}{+}\mathbf{C}:\ (S{\Uparrow}(VP{\uparrow}N))\backslash(S{\Uparrow}(VP{\uparrow}N)):\ \lambda y\lambda z[(y\ z) \wedge ((z\ s)\ d)]}/E}{\begin{array}{c}\mathbf{J}{+}1{+}\mathbf{W}{+}1{+}\mathbf{and}{+}\mathbf{D}{+}\mathbf{C}:\\ S{\Uparrow}(VP{\uparrow}N):\\ \lambda z[((z\ w)\ j) \wedge ((z\ c)\ d)]\end{array}}\backslash E \qquad \dfrac{\dfrac{\dfrac{\begin{array}{c}\mathbf{wants}:\\ (VP/VP)/N:\\ want\end{array} \quad \begin{array}{c}\overline{}^{\,i}\\ a:\\ N:\\ x\end{array}}{\begin{array}{c}\mathbf{wants}{+}a:\\ VP/VP:\\ (want\ x)\end{array}}/E \qquad \begin{array}{c}\text{to win}\\ \mathbf{to}{+}\mathbf{win}:\\ VP:\\ win\end{array}}{\begin{array}{c}\mathbf{wants}{+}a{+}\mathbf{to}{+}\mathbf{win}:\\ VP\\ :((want\ x)\ win)\end{array}}/E}{\begin{array}{c}\mathbf{wants}{+}1{+}\mathbf{to}{+}\mathbf{win}:\\ VP\\ \uparrow N: \lambda x((want\ x)\ win)\end{array}}\uparrow I^{i}}{\begin{array}{c}\mathbf{J}{+}\mathbf{wants}{+}\mathbf{W}{+}\mathbf{to}{+}\mathbf{win}{+}\mathbf{and}{+}\mathbf{D}{+}\mathbf{C}: S: [(((want\ w)\ win)\ j) \wedge (((want\ c)\ win)\ d)]\end{array}}\Uparrow E$$

Observe how in the last step of both of the above derivations adjunction combines a string of sort 2 and a string of sort 1. But, in the first, simplex, case the separator of the second operand is right peripheral, whereas in the second, complex, case the separator of the second operand is medial. This is how the

account unifies simplex and complex gapping under a single coordinator type.

The account of determiner gapping consists in an assignment to the coordinator of the following type (Q is $((S{\uparrow}N){\downarrow}S)/CN$):

(25) **and**: $(X\backslash X)/((X{\circledcirc}J){\odot}I)$: $\lambda x\lambda y\lambda z\lambda w[((y\ z)\ w) \wedge ((\pi_1\pi_1 x\ z)\ w)]$
where $X = (S{\Uparrow}(VP{\uparrow}N)){\uparrow}Q$

Consider example (2a) of simplex determiner gapping: *Some dogs like Whiskas and cats Alpo*. We use the following auxiliary derivation showing that a common noun followed by an object has type $(S{\Uparrow}(VP{\uparrow}N)){\uparrow}Q$:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{a\colon N\colon x}\,i \quad \dfrac{\overline{}^{\,j}}{b_0{+}1{+}b_1\colon VP{\uparrow}N\colon y} \quad \mathbf{obj}\colon N\colon obj}{b_0{+}\mathbf{obj}{+}b_1\colon VP\colon (y\ obj)}\,{\uparrow}E}{a{+}b_0{+}\mathbf{obj}{+}b_1\colon S\colon ((y\ obj)\ x)}\,{\backslash}E}{1{+}b_0{+}\mathbf{obj}{+}b_1\colon S{\uparrow}N\colon \lambda x((y\ obj)\ x)}\,{\uparrow}I^i \qquad \dfrac{\dfrac{}{c\colon Q\colon z}\,k \quad \mathbf{cn}\colon CN\colon cn}{c{+}\mathbf{cn}\colon (S{\uparrow}N){\downarrow}S\colon (z\ cn)}\,{/}E}{c{+}\mathbf{cn}{+}b_0{+}\mathbf{obj}{+}b_1\colon S\colon ((z\ cn)\ \lambda x((y\ obj)\ x))}\,{\downarrow}E$$
$$\dfrac{c{+}\mathbf{cn}{+}b_0{+}\mathbf{obj}{+}b_1\colon S\colon ((z\ cn)\ \lambda x((y\ obj)\ x))}{c{+}\mathbf{cn}{+}1{+}\mathbf{obj}{+}1\colon S{\Uparrow}(VP{\uparrow}N)\colon \lambda y((z\ cn)\ \lambda x((y\ obj)\ x))}\,{\Uparrow}I^j$$
$$\dfrac{c{+}\mathbf{cn}{+}1{+}\mathbf{obj}{+}1\colon S{\Uparrow}(VP{\uparrow}N)\colon \lambda y((z\ cn)\ \lambda x((y\ obj)\ x))}{1{+}\mathbf{cn}{+}1{+}\mathbf{obj}{+}1\colon (S{\Uparrow}(VP{\uparrow}N)){\uparrow}Q\colon \lambda z\lambda y((z\ cn)\ \lambda x((y\ obj)\ x))}\,{\uparrow}I^k \tag{26}$$

The derivation of (2a) is given in Figure 1. Finally, from the same coordinator type assignment (25) we can derive the case of discontinuous determiner gapping (2b) *Every cook wants Barça to win and waiter Madrid* in Figure 2.

6 Conclusion

The categorial analysis of gapping as like-type coordination was established in Steedman (1990[6]) and Hendriks (1995[1]). In the framework of HTLG Kubota and Levine (2012[2]) go further in that they provide like-type coordination for discontinuous gapping. Our analysis is inspired by that of Kubota and Levine (2012[2]; 2013[3]). However, our analysis of gapping represents an improvement on the HTLG analysis because we do not require two types for simple and discontinuous gapping: a single type suffices.

Finally, we have noted that the HTLG account of gapping suffers from determiner-transitive verb permutation overgeneration; the same problem would arise for HTLG in relation to discontinuous determiner gapping:

(27) a. ∗Some boy wants Everton to win and Mary (wants) (some) London club (to win)
b. ∗Mary wants some London club to win and (some) boy (wants) Everton (to win)

In addition to capturing simplex gapping (and determiner gapping) as a special case of complex gapping (and determiner gapping), our DTLG account does not have the permutation overgeneration problem of determiner gapping and discontinuous determiner gapping.

Fig. 1 Determiner gapping

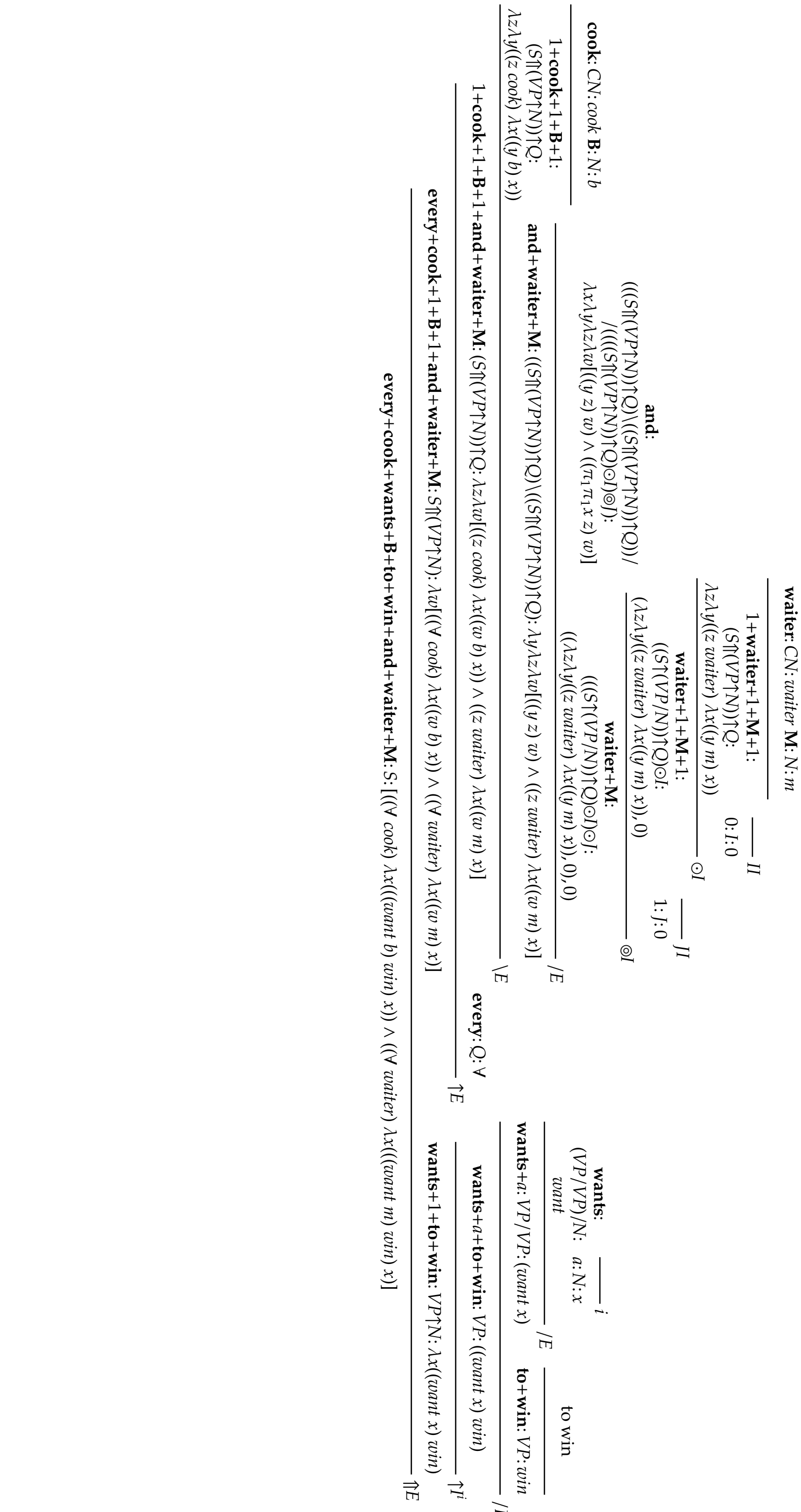

Fig. 2 Discontinuous determiner gapping

References

1. Petra Hendriks. *Comparatives and Categorial Grammar*. PhD thesis, Rijksuniversiteit Groningen, Groningen, 1995.
2. Yusuke Kubota and Robert Levine. Gapping as like-category coordination. In Denis Bechet and Alexander Dikovsky, editors, *Logical Aspects of Computational Linguistics*, volume 7351 of *Lecture Notes in Computer Science*, pages 135–150. Springer Berlin Heidelberg, 2012.
3. Yusuke Kubota and Robert Levine. Determiner Gapping as Higher-Order Discontinuous Constituency. 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 225–241, Berlin, 2013. Springer.
4. Yusuke Kubota and Robert Levine. Gapping as hypothetical reasoning. *Natural Language and Linguistic Theory*, 34(1):107–156, February 2016.
5. 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.
6. Mark Steedman. Gapping as Constituent Coordination. *Linguistics and Philosophy*, 13(2):207–263, 1990.