Proof-Theoretic Semantics for Subsentential Phrases

Nissim Francez[†] and Roy Dyckhoff[‡] and Gilad Ben-Avi[§]

November 24, 2009

1. Introduction

In [3], a *proof-theoretic semantics (PTS)* for a fragment E_0^+ of English, the core of which containing (common, count) nouns, determiners, (extensional) verbs and the copula is a (abbreviated isa), was presented, providing *proof-theoretic sentential meaning*: The proof-theoretic meaning of a sentence $S \in E_0^+$ is a function from contexts (finite collections of sentences) Γ, returning the collection of all *canonical* (see below) derivations of S from Γ, in a "dedicated" proof-system N_0^+ (see Figure 1). This PTS is intended to replace the more traditional *model-theoretic semantics (MTS)*, that views meanings as *truth-conditions* (in arbitrary models), that has been criticized by many philosophers of language as an adequate theory of meaning. See (the full version of) [3] for a more detailed discussion and the appropriate references. The current paper extends the natural language PTS-programme by defining proof-theoretic meanings for *subsentential phrases*, down to lexical elements (single words), within the framework of type-logical grammar (TLG) [6], based on Frege's *Context Principle (CP)*. By this principle, the meaning (i.e., semantic value) of a word is its *contribution* to the sentential meanings of the sentences in which it occurs. Such an extension allows the incorporation of proof-theoretic meanings into actual (type-logical) grammars, for use in computational linguistics. A similar extension for the propositional calculus and for the 1st-order predicate calculus is presented in [2].

[*] Work started while the first author was on leave at St Andrews university.

[†] Computer Science dept., the Technion-IIT, Haifa, Israel (francez@cs.technion.ac.il)

[‡] School of Computer Science, University of St Andrews, Scotland, UK (rd@st-andrews.ac.uk)

[§] Computer Science dept., the Technion-IIT, Haifa, Israel (bagilad@cs.technion.ac.il)

Ever since Gentzen's casual remark, *Proof-Theoretical Semantics (PTS)* is an approach that regards meaning of logical constant as *determined* by rules of use, as embodied in a natural-deduction (ND) proof-system, instead of being determined model-theoretically by truth-conditions (in *arbitrary* models).

In spite of the vast literature on the proof-theoretic semantics for logical constants, there is nowhere an explicit definition for a *semantic value* as determined by the I-rules of the ND-system; something of the form

$$[\![*]\!] = \cdots$$

where '$*$' is a logical constant. Such a definition is needed, for example, to facilitate a discussion of the compositionality of the meaning of the constant as determined by the rules at hand. It is also needed when one wants to consider a grammar for a logical language of which the constant is a part, for computational purposes. In [2], such a definition is provided for logical constants, also based on Frege's *context principle* (CF) and a *type-logical grammar*.

A central claim advocated here, originating in [2], is that it is *sentential* meaning of compound sentences that is determined directly by the ND-rules; *contributions* to sentential meanings are determined by the function-argument relationships induced by the grammar, determined *indirectly* only by the rules of the logic. We adhere to a common view in PTS, by taking sentential meanings (semantic value) of compound sentences as constructed by means of *canonical derivations* in the ND-system. Specifically, we take the meaning of a sentence S to be a function from contexts (sets of (open) assumptions) Γ to canonical derivations of S from Γ. This definition corroborates, and makes precise, a common observation about PTS (cf. for example, [8, 9] for recent discussions): sentential meanings are not *directly compositional*. The reason *here* is, that a canonical derivation has as its *immediate* sub-derivations *arbitrary* derivations of the premises, not just canonical derivations. However, this does not mean that sentential meanings do not depend on their component phrases: only the dependence is somewhat indirect, "almost" compositional. This is reflected here by having the ND-rules determining *two* semantic values (both for sentences and for sub-sentential components). For an expression ξ (either a whole sentence or a sub-sentential component), the one semantic value is its *contributed* value $[\![\xi]\!]$, serving also as its meaning; the other, auxiliary, semantic value is its *contributing* value $[\![\xi]\!]^*$, used when ξ is part of some larger expression ξ'. Thus, the function-argument relationships, induced by the given grammar of the language, plays a central role in obtaining semantic values.

The paper is structured as follows. Section 2 briefly reviews the main points of the PTS in [3] for sentential meanings. Section 3 discusses the use of Frege's context principle, and introduces a new, proof-theoretic, interpretation of the type system needed for its use in a TLG-framework. Section 4 shows how to extract subsentential meanings from sentential meanings, here for noun-phrases and determiners, and exhibits the resulting grammar.

2. Review of the Proof-Theoretic Semantics for sentences

The core fragment E_0^+ of English consists of sentences headed by (extensional) intransitive and transitive verbs, and determiner phrases (occasionally called also noun-phrase, *np*) with a (count) noun* and a determiner. In addition, there is the copula. This is a typical fragment of many NLs, syntactically focusing on *subcategorization*, and semantically focusing on *predication* and *quantification*. Some typical sentences are listed below.

(1) every/some girl smiles

(2) every/some girl is a student

(3) every/some girl loves every/some boy

We omit here *proper names* that do appear in the detailed presentation of sentential meanings (see [3]). Note the absence of *negative determiners* like no (hence the superscript '+'). We refer to expressions such as every girl, some boy as *determiner phrases* (*dp*s).

The *PTS* is based on a dedicated natural deduction proof system N_0^+ with introduction and elimination rules (I-rules, E-rules) (see Figure 1). The structural rule of *contraction* (C) is assumed primitive (see [3] for a justification of its inclusion). The proof-system is formulated over the language L_0^+, slightly extending E_0^+, *disambiguating* ambiguous E_0^+ sentences. Meta-variables X schematize nouns, P over intransitive verbs and R - transitive verbs. Meta-variable S ranges over sentences, and boldface lower-case $\mathbf{j}$, $\mathbf{k}$, etc., range over $\mathcal{P}$, a denumerable set of *(individual) parameters*, artifacts of the proof-system (not used to make assertions). Syntactically, a parameter in L_0^+ is also regarded as a *dp*. If a parameter occurs in S in some position, we refer to S as a *pseudo-sentence*, and if *all dp*s in S are parameters, the pseudo-sentence S is *ground*. The ground pseudo-sentences play the role of atomic sentences, and their meaning is assumed *given*, externally to the ND proof-system. The latter defines sentential meanings of non-ground pseudo-sentenses (and, in particular, E_0^+-sentences), *relative* to the given meanings of ground pseudo-sentences.

*Currently, only singular (and not plural) nouns are considered.

In contrast to logic, where the introduced operator by an I-rule is always the (unique) *main operator*, in E_0^+ sentences there is no such main operator: every position that can be filled with a *dp* is a *locus of introduction* (of the quantifier corresponding to the determiner of the introduced *dp*). This is a major source of *ambiguity* in E_0^+, known as quantifier-scope ambiguity. See [3] for the way ambiguity is treated, recapitulated briefly below. For any *dp*-expression D having a quantifier, we use the notation $S[(D)_n]$ to refer to a sentence S having a designated position filled by D, where n is the *scope level (sl)* of the quantifier in D. In case D has no quantifier (i.e., it is a parameter), $sl = 0$. The higher the sl, the higher the scope. For example, $S[(\mathsf{every}\ X)_1]$ refers to a sentence S with a designated occurrence of $\mathsf{every}\ X$ of the lowest scope. An example of a higher scope is $S[(\mathsf{some}\ X)_2]$, having $\mathsf{some}\ X$ in the higher scope, like in the object wide-scope reading of $(\mathsf{every}\ X)_1\ \mathsf{loves}\ (\mathsf{some}\ Y)_2$. We use the conventions that within a rule, both $S[D_1], S[D_2]$ refer to the *same* designated position in S, and when the sl can be unambiguously determined it is omitted. We use $r(S)$ to indicate the *rank* of S, the highest sl on a *dp* within S. Note that for a ground S, $r(S) = 0$. The notation is extended to more than one *dp*, where, say, $S[(D_1)_i, (D_2)_j]$ indicates a sentence S with two designated positions filled, respectively, by D_1, D_2, each with its respective sl.

In a rule, the notation $[\cdots]_i$ indicates an assumption *discharged* by an application of that rule. The indices of the assumptions discharged by a rule appear as superscripts on the rule name. The usual notion of (tree-shaped) derivation is assumed. We use $\mathcal{D}$ for derivations, where $\mathcal{D}^{\Gamma\vdash S}$ is a derivation of sentence S from context Γ. We use Γ, S for the context extending Γ with sentence S. $\mathcal{F}(\Gamma; \mathbf{j})$ means $\mathbf{j}$ is *fresh* for Γ.

The following is a convenient *derived* E-rule, that will be used to shorten derivations.

$$\frac{\Gamma\vdash S[(\mathsf{every}\ X)_{r(S[\mathbf{j}])+1}] \quad \Gamma\vdash \mathbf{j}\ \mathsf{isa}\ X}{\Gamma\vdash S[\mathbf{j}]}\ (e\hat{E})$$

(see [3] for its easy proof of being derived). Below is an example derivation establishing

$$\mathsf{some}\ U\ \mathsf{isa}\ X,\ (\mathsf{every}\ X)_2\ R\ (\mathsf{some}\ Y)_1,\ \mathsf{every}\ Y\ \mathsf{isa}\ Z \vdash (\mathsf{some}\ U)_1\ R\ (\mathsf{some}\ Z)_2$$

Let $(\mathsf{some}\ U)_2\ \overset{\mathcal{D}_1}{R}\ (\mathsf{some}\ Z)_1$ and $(\mathsf{some}\ U)_1\ \overset{\mathcal{D}_2}{R}\ (\mathsf{some}\ Y)_2$ be the following

$$\frac{}{\Gamma, S \vdash S} \quad (Ax)$$

$$\frac{\Gamma, [\mathbf{j} \text{ isa } X]_i \vdash S[\mathbf{j}]}{\Gamma \vdash S[(\text{every } X)_{r(S[\mathbf{j}])+1}]} \ (eI^i)$$

$$\frac{\Gamma \vdash \mathbf{j} \text{ isa } X \quad \Gamma \vdash S[\mathbf{j}]}{\Gamma \vdash S[(\text{some } X)_{r(S[\mathbf{j}])+1}]} \ (sI)$$

$$\frac{\Gamma \vdash S[(\text{every } X)_{r(S[\mathbf{j}])+1}] \quad \Gamma \vdash \mathbf{j} \text{ isa } X \quad \Gamma, [S[\mathbf{j}]]_i \vdash S'}{\Gamma \vdash S'} \ (eE^i)$$

$$\frac{\Gamma \vdash S[(\text{some } X)_{r(S[\mathbf{j}])+1}] \quad \Gamma, [\mathbf{j} \text{ isa } X]_j, [S[\mathbf{j}]]_i \vdash S'}{\Gamma \vdash S'} \ (sE^{i,j})$$

where $\mathcal{F}(\Gamma, S[\text{every } X]; \mathbf{j})$ in (eI), and $\mathcal{F}(\Gamma, S[\text{some } X], S';\ \mathbf{j})$ for (sE).

Figure 1. The meta-rules for N_0^+

two sub-derivations.

$$\mathcal{D}_1 : \dfrac{\text{some } U \text{ isa } X \qquad \dfrac{[\mathbf{r} \text{ isa } U]_1 \qquad \dfrac{(\text{every } X)_2 \ R \ (\text{some } Y)_1 \quad [\mathbf{r} \text{ isa } X]_2}{\mathbf{r} \ R \text{ some } Y} \ (e\hat{E})}{(\text{some } U)_2 \ R \ (\text{some } Y)_1} \ (sI)}{(\text{some } U)_2 \ R \ (\text{some } Y)_1} \ (sE^{1,2})$$

$$\mathcal{D}_2 : \dfrac{[\text{some } U \ R \ \mathbf{j}]_3 \qquad \dfrac{\text{every } Y \text{ isa } Z \quad [\mathbf{j} \text{ isa } Y]_4}{\mathbf{j} \text{ isa } Z} \ (e\hat{E})}{(\text{some } U)_1 \ R \ (\text{some } Z)_2} \ (sI)$$

The whole derivation combines the two sub-derivations by

$$\dfrac{\overset{\mathcal{D}_1}{(\text{some } U)_2 \ R \ (\text{some } Y)_1} \quad \overset{\mathcal{D}_2}{(\text{some } U)_1 \ R \ (\text{some } Z)_2}}{(\text{some } U)_1 \ R \ (\text{some } Z)_2} \ (sE^{3,4})$$

For a derivation $\mathcal{D}$ of S, we define its *root* by $\rho(\mathcal{D}) = S$. This function is extended to collections of derivations Δ by $\rho(\Delta) = \{\rho(\mathcal{D}) \mid \mathcal{D} \in \Delta\}$, and further extended to contextualized functions $\mathcal{F}$ by $\rho(\mathcal{F}) = \cup_\Gamma \rho(\mathcal{F}(\Gamma))$.

In order to understand better the PTS of E_0^+, consider one of its well-known features: *quantifier scope ambiguity*. The following E_0^+ sentences are

usually attributed to two readings each, with the following FOL expressions of their respective truth-conditions in model-theoretic semantics.

(4) Every girl loves some boy

(5) Some girl loves every boy

Consider sentence (4).

$$\textbf{\textit{Subject wide-scope (sws):}}\ \forall x.\mathbf{girl}(x)\rightarrow\exists y.\mathbf{boy}(y)\wedge\mathbf{love}(x,y)$$

$$\textbf{\textit{Subject narrow-scope (sns):}}\ \exists y.\mathbf{boy}(y)\wedge\forall x.\mathbf{girl}(x)\rightarrow\mathbf{love}(x,y)$$

In our PTS, the difference in meanings reflects itself by the two readings having *different grounds of assertion*. This is manifested in derivations by different *order of introduction* of the subject and object *dp*s. Following [7], we disambiguate ambiguous sentences taking part in derivations.

Subject wide-scope (sws):

$$\dfrac{\dfrac{\dfrac{[\mathbf{r}\ \textsf{isa girl}]_i}{\mathcal{D}_1}}{\mathbf{r}\ \textsf{loves}\ \mathbf{j}}\quad\dfrac{\mathcal{D}_2}{\mathbf{j}\ \textsf{isa boy}}\ (sI)}{(\textsf{every girl})_2\ \textsf{loves}\ (\textsf{some boy})_1}\ (eI^i)$$

Subject narrow-scope (sns):

$$\dfrac{\dfrac{\dfrac{[\mathbf{r}\ \textsf{isa girl}]_i}{\mathcal{D}_1}}{\mathbf{r}\ \textsf{loves}\ \mathbf{j}}\ (eI^i)\quad\dfrac{\mathcal{D}_2}{\mathbf{j}\ \textsf{isa boy}}}{(\textsf{every girl})_1\ \textsf{loves}\ (\textsf{some boy})_2}\ (sI)$$

Note that there is no way to introduce a *dp* with a narrow-scope where the *dp* with the wider-scope has already been introduced. In the N_0^+ calculus, only disambiguated sentences participate.

A derivation is *canonical* if it ends with an application of an I-rule; we use $\vdash^c$ for canonical derivability. Denote by $[\![S]\!]^c_\Gamma$ the collection of canonical derivations of S from Γ, and by $[\![S]\!]^*_\Gamma$ the collection of all derivations of S from Γ.

Definition (PTS-meaning, semantic values):

1. For a non-ground $S{\in}L_0^+$, its *meaning* (referred to also as its *contributed* semantic value) is given by $[\![S]\!] =^{df\cdot} \lambda\Gamma.[\![S]\!]^c_\Gamma\ \ [=\lambda\Gamma.\{\mathcal{D}^{\Gamma\vdash^c S}\}]$.
   Recall that for a *ground* S, $[\![S]\!]$ is assumed *given*. The meaning of non-ground pseudo-sentences (and E_0^+-sentences in particular) is defined *relative* to the given meanings of ground pseudo-sentences.

2. For an arbirary $S{\in}L_0^+$, its *contributing semantic value* is given by $[\![S]\!]^* =^{\text{df.}} \lambda\Gamma.[\![S]\!]^*_\Gamma$.

By this way of defining sentential meanings, we do not allude to any "logical form" of the sentence, differing from its surface form. In accordance with many views in philosophy of language, every derivation in the meaning of a sentence S can be viewed as providing $G[\![S]\!]$, *grounds of asserting* S. We thus define, for $S{\in}E_0^+$, $G[\![S]\!] =^{df.} \{\Gamma \mid \Gamma{\vdash^c}S\}$, where Γ consists of E_0^+-sentences only. Parameters are not "observable" in grounds of assertion. A more comprehensive discussion of extensions of the fragment and some technicalities accompany the original presentation of the PTS in [3].

3. Frege's context principle and proof-theoretic types

3.1. The Context Principle and its use for meaning extraction

The original formulation of *CP* ([4], p. 116) is: "We must never try to define the meaning of a word in isolation, but only as it is used in the context of a proposition." For a recent philosophical discussion (and scrutiny) of the *CP*, see [10]. The *CP* can be generalized from words to arbitrary subsentential phrases, yielding[†] ([5]):

Principle F: The meaning of a phrase w is the *contribution* that w makes to the meanings of phrases u containing w.

Contributions: Frege did not give any explicit definition of 'contribution'. We propose one here. Our general method of identifying contributions is to decompose (in accordance to the TLG-categorization) the sentential meanings into function-argument structures: In the base case, the subphrase contributing the argument obtains its meaning from a ground sentence; in general, the subphrase contributing the function obtains its meaning by abstraction over the argument's contributing semantic value.

Suppose that by the TLG-categorization, a phrase w_1 of functional category is adjoined to a phrase w_2 of the suitable argument category (w.l.o.g, we consider the case where the function precedes the argument). We have the following decomposition principle:

$$\delta: \quad [\![w_1 w_2]\!] = [\![w_1]\!]([\![w_2]\!]) \tag{1}$$

Now, the subphrase w_1 contributing the function can obtain its contributed

[†]We consider 'subsentential phrase' where Hodges considers 'term' in his more algebraic setting.

semantic values by abstraction over the argument's[‡] contributing semantic value. Thus, transitively, subsentential meanings are characterized by their contribution to (the independently defined) sentential meanings, where the extraction of contributions is driven by the TLG, which assigns syntactic categories to phrases. We can observe here an important difference in the role TLG plays in our PTS, as opposed to its traditional role in MTS. In the latter, (1) is viewed from right to left, determining the $[\![w_1 w_2]\!]$ based on $[\![w_1]\!]$ and $[\![w_2]\!]$. Here, (1) is viewed from left to right, using $[\![w_1 w_2]\!]$ (given by the proof-system for full sentences) to yield $[\![w_1]\!]$ and $[\![w_2]\!]$. Recall that the recursive decomposition terminates when a "pure argument" is reached, as meanings of ground pseudo-sentences are given externally.

The whole approach is in concert also with Frege's *compositionality principle*: word-meaning extraction occurs *once only*; but when the extracted word-meanings are combined – per (successfully) parsed sentence – according to the grammar, they reconstruct compositionally its "original" sentential meaning. This is shown in more detail below. This view of reconciling the two of Frege's principles, Context and compositionality, shows that difficulties like those pointed out in [8] are only apparent.

3.2. Proof-Theoretic Type-Interpretation

In order to express the above mentioned abstractions, we propose a *proof-theoretic interpretation* of types, replacing the usual Montagovian model-theoretic interpretation of those types ((full) Henkin models), using *proof-theoretic domains*, comprised of functions from contexts to collections of N_0^+-derivations or functions therein. The issue of a proof-term language, the terms of which are typed using the proposed proof-theoretically interpreted types, will be treated elsewhere. Here, those interpretations of types are used only for forming proof-theoretic meanings for sub-sentential phrases.

Definition: (types, proof-theoretic type-interpretation) Let T_0^+ be the following system of types, ranged over by α, β:

- t is a (basic) type, with $D_t = \{[\![S]\!] \mid S \in L_0^+\}$.
- p is a (basic) type of (individual) parameters, with $D_p = P$.
- If α, β are types, then (α, β) is a functional type, with $D_{(\alpha,\beta)} = D_\beta^{D_\alpha}$, the collection of all functions from D_α to D_β.

[‡]Note that there is a strong dependency of the sub-sentential meanings obtained on the categorization used by the grammar; changing it may lead to a change in the sub-sentential meanings obtained from the *same* sentential meanings.

Note that for $z{\in}D_t$, $\rho(z)$ is a singleton. Specifically, if $z = \lambda\Gamma.[\![S]\!]^c_\Gamma$, then $\rho(z) = \{S\}$. We also assume that $\rho([\![S]\!]) = S$ for a *ground* S, to simplify the notation. This facilitates the following definition, of a function mapping the meaning of S to its contributing semantic value:

$$\mathbf{ex} =^{\text{df.}} \lambda z^t \lambda\Gamma.[\![\rho(z)]\!]^*_\Gamma$$

We refer to $\mathbf{ex}([\![S]\!])$ as the *expansion* of (the meaning of) S. Thus, the role of the expansion is to take a sentential meaning (contributed semantic value), comprised of canonical derivations only, and convert it to the collection of all derivations of the sentence (the root of the meaning), constituting the contributing semantic value. See its use below in the I-functions. This function facilitates "burying" the difference between the two semantic values, the source of non-direct-compositionality. Thus, in the TLG-lexicon, only meanings need being provided.

We introduce a means for forming certain *subtypes*, for some of the more frequently used functional types, where the argument parameter has to occupy some position in a pseudo-sentence type (i.e., preventing constant functions).

- t_p is a subtype of (p, t) (nouns), s.t. $D_{t_p} = \{\lambda\mathbf{j}.[\![S[\mathbf{j}]]\!] \mid S[\mathbf{j}]{\in}L^+_0\}$.
- $t_{p,p}$ is a subtype of $(p,(p,t))$, s.t. $D_{t_{p,p}} = \{\lambda\mathbf{k}\lambda\mathbf{j}.[\![S[\mathbf{j},\mathbf{k}]]\!] \mid S[\mathbf{j},\mathbf{k}]{\in}L^+_0\}$.
- n is a subtype of (p,t), s.t. $D_n = \{\lambda\mathbf{j}.[\![\mathbf{j}\ \mathsf{isa}\ X]\!] \mid X \text{ a noun}\}$. We let $\nu(\lambda\mathbf{j}.[\![\mathbf{j}\ \mathsf{isa}\ X]\!]) =^{\text{df.}} X$, recovering the noun from an element od D_n.

For expressing meanings we use PT^+_0-typed variables z_1, z_2.

3.2.1. I-functions

It is useful to view the I-rules as inducing *I-functions*, functions from (sets of) derivations (or pairs thereof) to (sets of) derivations; the result derivations are obtained by applying the respective I-rule to the argument derivation(s). There are two such functions for the core of N^+_0, one for each determiner. The functions are presented in Figure 2. Note that when I_e is applied to a non-fresh $\mathbf{j}$, the resulting set of derivations is empty. Note also the use of expansions within the I-functions, the latter forming the interface between contributed and contributing semantic values.

4. Extracting PTS subsentential meanings

In this section, we demonstrate the extraction of PT-meanings for the main E^+_0 subsentential phrases: determiner-phrases (*dp*s) and determiners. The

$$I_e = \lambda z_1^n \lambda z_2^{t_p} \lambda \mathbf{j} \lambda \Gamma . \{ \frac{\mathcal{D}}{\Gamma \vdash \rho(\mathcal{D})[(\mathsf{every}\ \nu(z_1))_{r(\rho(\mathcal{D}))+1}/\mathbf{j}]} (eI^i) : \\ \mathcal{D} \in \mathbf{ex}(z_2(\mathbf{j}))(\Gamma, [\rho(z_1(\mathbf{j}))]_i \& \mathcal{F}(\Gamma, \mathbf{j}) \} \quad (2)$$

Note that the *value* of z_1 does not contribute to the result, only its *noun* X is used (retrieved via ν), by augmenting Γ with $\mathbf{j}\ \mathsf{isa}\ X$. This reflects the role of $\mathbf{j}\ \mathsf{isa}\ X$ as a discharged assumption in the N_0^+-derivation of $S[\mathsf{every}\ X]$ from Γ.

$$I_s = \lambda z_1^n \lambda z_2^{t_p} \lambda \mathbf{j} \lambda \Gamma . \{ \frac{\mathcal{D}_1 \quad \mathcal{D}_2}{\Gamma \vdash \rho(\mathcal{D}_2)[(\mathsf{some}\ \nu(z_1))_{r(\rho(\mathcal{D}_2))+1}/\mathbf{j}]} (sI) : \\ \mathcal{D}_1 \in \mathbf{ex}(z_1(\mathbf{j}))(\Gamma) \mathcal{D}_2 \in \mathbf{ex}(z_2(\mathbf{j}))(\Gamma) \} \quad (3)$$

Note that here the value of the argument z_2 *is* used.

Figure 2. I-functions for the core of N_0^+

syntactic lexicon driving the process is based on the (associative) Lambek calculus **L** (see Appendix). The basic categories are n (noun[§]), dp (determiner-phrase) and s (sentence). *Directed arrows* are used for forming functional categories. Abbreviate the *raised* categories[¶] of dp, namely $(s \leftarrow (dp \rightarrow s))$ and $((s \leftarrow dp) \rightarrow s)$ as $dp_n \uparrow$ and $dp_a \uparrow$, respectively. Following is a typical syntactic lexicon for (the core of) E_0^+.

nouns	n
determiners	$(dp_n \uparrow \leftarrow n)$
	$(dp_a \uparrow \leftarrow n)$
intransitive verbs	$(dp \rightarrow s)$
transitive verbs	$((dp \rightarrow s) \leftarrow dp)$

4.1. Determiner-phrases and determiners

We start by noting that by applying decompositions δ as described above, we extract semantic values for sub-sentential phrases that serve as functions,

[§]Note that we use n both as a category and as a type; as categories and types appear in different contexts, no confusion should occur.

[¶]This distinction between *nominative* and *accusative* categories is morphologically realized in the form of the determiner itself in many languages.

by abstracting over the argument (shown below in detail). However, some expressions are "pure arguments" in their contribution to sentential meaning. For such expression, their meaning is obtained via (the given) meanings of ground pseudo-sentences. We have three such cases.

Nouns: Nouns are of a basic category, and can only contribute arguments. We take their meaning (of type n, a subtype of (p,t)) to originate from the corresponding (*given!*) meaning of the ground pseudo-sentence: $[\![X]\!] = \lambda\mathbf{j}.[\![\mathbf{j}\ \mathsf{isa}\ X]\!]$.

Intransitive verbs: If P is an intransitive verb, its meaning (of type t_p) originates from the (*given*) meaning of a ground pseudo-sentence headed by it: $[\![P]\!] = \lambda\mathbf{j}.[\![\mathbf{j}\ P]\!]$.

Transitive verbs: If R is a transitive verb, its meaning (of type $t_{p,p}$) originates from the (given) meaning of a ground pseudo-sentence headed by it: $[\![R]\!] = \lambda\mathbf{k}\lambda\mathbf{j}.[\![\mathbf{j}\ R\ \mathbf{k}]\!]$.

Our point of departure will be a schematic sentence, with a quantified dp in its subject position, say of the form

$$S = \mathsf{every}\ X\ VP$$

Here VP, a verb-phrase, can be either an intransitive verb, or a transitive verb with its object dp with a quantifier, already incorporated, or with an object being a parameter. Consider first the first two possibilities, behaving identically. The (sentential) meaning of S is given by

$$[\![S]\!] = \lambda\Gamma.\cup_{\mathbf{j}} I_e([\![X]\!])([\![VP]\!])(\mathbf{j})(\Gamma) \tag{4}$$

where the type of $[\![VP]\!]$ is t_p. By the syntactic categorization, S has the following derivation in the **L**-based TLG.

$$\frac{\dfrac{\mathsf{every}\ X}{dp_n \uparrow: Q_s} \quad \dfrac{VP}{(dp{\rightarrow}s) : V}}{s : (Q_s V)}\ (\leftarrow E) \tag{5}$$

Here Q_s is a variable of type (t_p, t), and V of type t_p. From (5) we get

$$[\![S]\!] = [\![\mathsf{every}\ X]\!]([\![VP]\!]) \tag{6}$$

By combining (4) and (6), introducing a variable z_2 of type t_p, and abstracting over it, we obtain the following meaning (semantic value) for the subject dp.

$$[\![\mathsf{every}\ X]\!] = \lambda z_2^{t_p} \lambda\Gamma.\cup_{\mathbf{j}} I_e([\![X]\!])(z_2)(\mathbf{j})(\Gamma) \tag{7}$$

Note that in the second case, I_e introduces every X with $sl = 2$, since the value of z_2 has already an np with a quantifier incorporated.

Next, we have from the categorization and the inner derivation of every X in the grammar that

$$[\![\text{every } X]\!] = [\![\text{every}]\!]([\![X]\!]) \tag{8}$$

Combining (8) with (7), we can introduce another variable z_1, of type n, and abstract over it, to obtain a meaning (semantic value) for the determiner itself.

$$[\![\text{every}]\!] = \lambda z_1^n \lambda z_2^{t_p} \lambda\Gamma.\cup_{\mathbf{j}} I_e(z_1)(z_2)(\mathbf{j})(\Gamma) \tag{9}$$

By a similar analysis of, say, $S =$ some X VP, we obtain

$$[\![\text{some } X]\!] = \lambda z_2^{t_p} \lambda\Gamma.\cup_{\mathbf{j}} I_s([\![X]\!])(z_2)(\mathbf{j})(\Gamma) \tag{10}$$

and

$$[\![\text{some}]\!] = \lambda z_1^n \lambda z_2^{t_p} \lambda\Gamma.\cup_{\mathbf{j}} I_s(z_1)(z_2)(\mathbf{j})(\Gamma) \tag{11}$$

Next, we consider the remaining case of the VP. If its object is a parameter, say $\mathbf{k}$, like in every X R $\mathbf{k}$, then $[\![R]\!]$ is of type $t_{p,p}$. The function I_e will cause the introduction of every X with index $sl = 1$, bearing a low scope to the object quantifier, once introduced too. Examples of determiners in object position are shown below, in the discussion of reconstructing quantifier scope ambiguity.

Below is a sample derivation (after β-reductions), with actual noun and (intransitive) verb. This derivation exemplifies how, during parsing, the lexical proof-theoretic meanings are combined, to reconstruct the pre-existing sentential meaning. To save space, we ignore the unions $\cup_{\mathbf{j}}$, assuming $\mathbf{j}$ is fresh.

$$\dfrac{\dfrac{\dfrac{\text{every}}{\begin{array}{c}(dp_n \uparrow\leftarrow n) : \\ \lambda z_1^n \lambda z_2^{t_p} \lambda\Gamma.I_e(z_1)(z_2)(\mathbf{j})(\Gamma)\end{array}} \quad \dfrac{\text{girl}}{\begin{array}{c}n : \\ [\![\text{girl}]\!]\end{array}}}{\begin{array}{c}dp_n \uparrow : \\ \lambda z_2^{t_p} \lambda\Gamma.I_e([\![\text{girl}]\!])(z_2)(\mathbf{j})(\Gamma)\end{array}} (\leftarrow E) \quad \dfrac{\text{smiles}}{\begin{array}{c}(dp\rightarrow s) : \\ [\![\text{smiles}]\!]\end{array}}}{\begin{array}{c}s : \\ \lambda\Gamma.I_e([\![\text{girl}]\!])([\![\text{smiles}]\!])(\mathbf{j})(\Gamma) = [\![\text{every girl smiles}]\!]\end{array}} (\leftarrow E)$$

We end the discussion of dp meanings by showing how the quantifier scope ambiguity is generated. Note that in the following derivation, the power of **L** to discharge assumptions is relied upon.

Our point of departure will be a schematic sentence, headed by a transitive verb, with two determiners: $S = \mathsf{every}\ X\ R\ \mathsf{some}\ Y$. Recall that such an S has quantifier scope ambiguity, reflected in N_0^+ by the order of introduction into its two *dp*s. By the syntactic categorization, S has two derivations in **L**, one for each scopal relation, as shown below. We note here that in displaying the derivation, we deviate somewhat from the standard TLG presentation (as explained in the appendix). The reason is, that in standard TLG, meanings are themselves represented as λ-terms, that can be substituted for the respective free variables of the proof-term of the derivations, bound variables left intact. Here, meanings are proof-theoretic objects, written in a pseudo λ-notation in the meta-language. Hence, we use meanings in the derivation "on-the-fly". In particular, bound variables are those of L_0^+, over which our proof-theoretic objects are defined. Still, to keep readability, we do use free meta-variables for the proof-theoretic objects, substituting for them at the end of the derivation, and keep only bound variables as bound parameters.

$$\dfrac{\dfrac{\mathsf{every\ X}}{\begin{array}{c}dp_n\uparrow:\\ Q_s\end{array}} \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\mathsf{R}}{\begin{array}{c}((dp{\rightarrow}s){\leftarrow}dp):\\ R\end{array}} \qquad [dp:\mathbf{k}]_1}{\begin{array}{c}(dp{\rightarrow}s):\\ R(\mathbf{k})\end{array}}(\leftarrow E) \qquad [dp:\mathbf{j}]_2}{\begin{array}{c}s:\\ R(\mathbf{k})(\mathbf{j})\end{array}}(\rightarrow E)}{\begin{array}{c}(s\leftarrow dp):\\ \lambda\mathbf{k}.R(\mathbf{k})(\mathbf{j})\end{array}}(\leftarrow I_1) \qquad \dfrac{\mathsf{some\ Y}}{\begin{array}{c}dp_a\uparrow:\\ Q_o\end{array}}}{\begin{array}{c}s:\\ Q_o(\lambda\mathbf{k}.R(\mathbf{k})(\mathbf{j}))\end{array}}(\rightarrow E)}{\begin{array}{c}(dp{\rightarrow}s):\\ \lambda\mathbf{j}.Q_o(\lambda\mathbf{k}.R(\mathbf{k})(\mathbf{j}))\end{array}}(\rightarrow I_2)}{\begin{array}{c}s:\\ Q_s(\lambda\mathbf{j}.Q_o(\lambda\mathbf{k}.R(\mathbf{k})(\mathbf{j})))\end{array}}(\leftarrow E) \tag{12}$$

$$\dfrac{\mathsf{R}}{((dp{\to}s) \leftarrow dp) : R}$$

$$\dfrac{((dp{\to}s) \leftarrow dp) : R \qquad [dp : \mathbf{k}]_1}{(dp{\to}s) : R(y)} \; (\leftarrow E)$$

$$\dfrac{[dp : \mathbf{j}]_2 \qquad (dp{\to}s) : R(y)}{s : R(\mathbf{k})(\mathbf{j})} \; ({\to}E)$$

$$\dfrac{s : R(\mathbf{k})(\mathbf{j})}{(dp{\to}s) : \lambda\mathbf{j}.R(\mathbf{k})(\mathbf{j})} \; ({\to}I_2)$$

$$\dfrac{\dfrac{\mathsf{every\ X}}{dp_n \uparrow : Q_s} \qquad (dp{\to}s) : \lambda\mathbf{j}.R(\mathbf{k})(\mathbf{j})}{s : Q_s(\lambda\mathbf{j}.Q_o(\lambda\mathbf{k}.R(\mathbf{k})(\mathbf{j})))} \; (\leftarrow E)$$

$$\dfrac{s : Q_s(\lambda\mathbf{j}.Q_o(\lambda\mathbf{k}.R(\mathbf{k})(\mathbf{j})))}{(s \leftarrow dp) : \lambda\mathbf{k}.Q_s(\lambda\mathbf{j}.R(\mathbf{k})(\mathbf{j}))} \; (\leftarrow I_1)$$

$$\dfrac{(s \leftarrow dp) : \lambda\mathbf{k}.Q_s(\lambda\mathbf{j}.R(\mathbf{k})(\mathbf{j})) \qquad \dfrac{\mathsf{some\ Y}}{dp_a \uparrow : Q_o}}{s : Q_o(\lambda\mathbf{k}.Q_s(\lambda\mathbf{j}.R(\mathbf{k})(\mathbf{j})))} \; ({\to}E) \tag{13}$$

By substituting the dp meanings for Q_s and Q_o, namely

$$Q_s = \lambda z_2^{t_p} \lambda\Gamma.\cup_{\mathbf{j}} I_e([\![X]\!])(z_2)(\mathbf{j})(\Gamma)$$

$$Q_o = \lambda z_2^{t_p} \lambda\Gamma.\cup_{\mathbf{k}} I_s([\![Y]\!])(z_2)(\mathbf{k})(\Gamma)$$

we obtain the following two readings for the S above.

$$\begin{array}{l}[\![(\mathsf{every}\ X)_2\ R\ (\mathsf{some}\ Y)_1]\!] = \\ \lambda\Gamma.\cup_{\mathbf{j}} I_e([\![X]\!])(\lambda\mathbf{j}'\lambda\Gamma'.\cup_{\mathbf{k}}.I_s([\![Y]\!])(\lambda\mathbf{k}'.[\![\mathbf{j}'\ R\ \mathbf{k}']\!])]\!])(\mathbf{k})(\Gamma'))(\mathbf{j})(\Gamma)\end{array} \tag{14}$$

$$\begin{array}{l}[\![(\mathsf{every}\ X)_1\ R\ (\mathsf{some}\ Y)_2]\!] = \\ \lambda\Gamma.\cup_{\mathbf{k}} I_s([\![Y]\!])(\lambda\mathbf{j}'\lambda\Gamma'.\cup_{\mathbf{j}} I_e([\![X]\!])([\![\mathbf{j}'\ R\ \mathbf{k}']\!])(\mathbf{j})(\Gamma'))(\mathbf{k})(\Gamma)\end{array} \tag{15}$$

We show the first derivation. Substituting the value of Q_s into (12), renaming $\mathbf{j}$ and omitting the type of z_2, we obtain

$$\lambda z_2 \lambda\Gamma.\cup_{\mathbf{j}} I_e([\![X]\!](z_2)(\mathbf{j})(\Gamma)(\lambda\mathbf{j}'.Q_o(\lambda\mathbf{k}.R(\mathbf{k})(\mathbf{j}')))$$

which reduces to

$$\lambda\Gamma.\cup_{\mathbf{j}} I_e([\![X]\!])(\lambda\mathbf{j}'.Q_o(\lambda\mathbf{k}.R(\mathbf{k})(\mathbf{j}'))(\mathbf{j})(\Gamma)$$

Substituting the value of Q_o, renaming $\mathbf{k}$ and Γ, yields

$$\lambda\Gamma.\cup_{\mathbf{j}} I_e([\![X]\!])(\lambda\mathbf{j}'\lambda z_2 \lambda\Gamma'.\cup_{\mathbf{k}} I_s([\![Y]\!])(z_2)(\mathbf{k})(\Gamma'))(\lambda\mathbf{k}'.R(\mathbf{k}')(\mathbf{j}'))(\mathbf{j})(\Gamma)$$

which reduces to

$$\lambda\Gamma.\cup_{\mathbf{j}} I_e([\![X]\!])(\lambda\mathbf{j}'\lambda\Gamma'\cup_{\mathbf{k}} I_s([\![Y]\!])(\lambda\mathbf{k}'.R(\mathbf{k}')(\mathbf{j}'))(\mathbf{k})(\Gamma'))(\mathbf{j})(\Gamma)$$

and after substituting for R its lexical meaning $[\![\mathbf{j}'\ R\ \mathbf{k}']\!]$ the result follows.

4.2. The resulting E_0^+-TLG

We summarize the full grammar of E_0^+ (i.e., including PT-meanings) in the following lexicon. To fit it into the figure, we again omit the (standard) categories.

	type	**meaning**
nouns X	n	$\lambda\mathbf{j}.[\![\mathbf{j}\ isa\ X]\!]$
every	$(n,(t_p,t))$	$\lambda z_1^n \lambda z_2^{t_p} \lambda\Gamma.$ $\cup_{\mathbf{j}} I_e(z_1)(z_2)(\mathbf{j})(\Gamma)$
some	$(n,(t_p,t))$	$\lambda z_1^n \lambda z_2^{t_p} \lambda\Gamma.$ $\cup_{\mathbf{j}} I_s(z_1)(z_2)(\mathbf{j})(\Gamma)$
intransitive verbs V	t_p	$\lambda\mathbf{j}.[\![\mathbf{j}\ V]\!]$
transitive verbs R	$t_{p,p}$	$\lambda\mathbf{k}\lambda\mathbf{j}.[\![\mathbf{j}\ R\ \mathbf{k}]\!]$
copula	(n,n)	$\lambda z^n.z$

5. E_1^+: Adding Relative Clauses

5.1. Extending the proof system

Following [3], we next add to the fragment *relative clauses*. This fragment transcends the locality of subcategorization in E_0^+, in having *long-distance dependencies*. We refer to this (still positive) fragment as E_1^+. Typical sentences include the following.

(6) Jacob/every boy/some boy loves every/some girl who(m) smiles/loves every flower/Rachel loves

(7) Rachel/every girl/some girl is a girl who loves Jacob/every boy

(8) Jacob loves every girl who loves every boy who smiles (nested relative clause)

So, girl who smiles and girl who loves every boy are *compound nouns*. We treat somewhat loosely the issue of the case of the relative pronoun, in the form of who(m), abbreviating either *who* or *whom*, as the case requires. We extend our notation with $S[-]$, that denotes, for S including a parameter in some distinguished position, the result of removing that parameter, leaving that position unoccupied. Examples are loves every girl (a parameter removed from subject position in **j** loves every girl), and every girl loves (a parameter removed from object position in every girl loves **k**).

The syntactic categories of the relative pronouns are

$$\left|\begin{array}{c|c} \mathsf{who} & ((n \leftarrow n) \leftarrow (dp \rightarrow s)) \\ \mathsf{whom} & ((n \rightarrow n) \leftarrow (s \leftarrow dp)) \end{array}\right|$$

forming a compound noun from a simpler noun and a sentence missing its object. The **L**-derivations of compound nouns are given below.

$$\dfrac{\overset{X}{n} \quad \dfrac{\overset{\mathsf{who}}{((n\rightarrow n)\leftarrow(dp\rightarrow s))} \quad \overset{S[-]}{(dp\rightarrow s)}}{(n\rightarrow n)}\,(\leftarrow E)}{n}\,(\rightarrow E) \tag{16}$$

$$\dfrac{\overset{X}{n} \quad \dfrac{\overset{\mathsf{whom}}{((n\rightarrow n)\leftarrow(s\leftarrow dp))} \quad \overset{S[-]}{(s\leftarrow dp)}}{(n\rightarrow n)}\,(\leftarrow E)}{n}\,(\rightarrow E) \tag{17}$$

The corresponding ND-system N_1^+ extends N_0^+ by adding the following main I-rules and E-rules.

$$\frac{\Gamma\vdash \mathbf{j}\ \mathsf{isa}\ X \quad \Gamma\vdash S[\mathbf{j}]}{\Gamma\vdash \mathbf{j}\ \mathsf{isa}\ X\ \mathsf{who}\ S[-]}\ (relI)$$

$$\frac{\Gamma\vdash \mathbf{j}\ \mathsf{isa}\ X\ \mathsf{who}\ S[-] \quad \Gamma, [\mathbf{j}\ \mathsf{isa}\ X]_i, [S[\mathbf{j}]]_j\vdash S'}{\Gamma\vdash S'}\ (relE^{i,j})$$

The simplified elimination-rules are:

$$\frac{\Gamma\vdash \mathbf{j}\ \mathsf{isa}\ X\ \mathsf{who}\ S[-]}{\Gamma\vdash \mathbf{j}\ \mathsf{isa}\ X}\ (rel\hat{E})_1 \quad \frac{\Gamma\vdash \mathbf{j}\ \mathsf{isa}\ X\ \mathsf{who}\ S[-]}{\Gamma\vdash S[\mathbf{j}]}\ (rel\hat{E})_2$$

The familiar conjunctive behavior of relative clauses is exhibited here by its rules, resembling the rules for logical conjunction.

As an example of a derivation in this fragment, consider

$$\textsf{some girl who smiles sings} \vdash_{N_1^+} \textsf{some girl sings}$$

exhibiting the *upward monotonicity*[‖] of some in its first argument.

$$\dfrac{\textsf{some } X \textsf{ who } P_1 \ P_2 \qquad \dfrac{\dfrac{[\mathbf{r} \textsf{ isa } X \textsf{ who } P_1]_1}{\mathbf{r} \textsf{ isa } X}\ (rel\hat{E})_1 \quad [\mathbf{r} \ P_2]_2}{\textsf{some } X \ P_2}\ (sE^{1,2})}{\textsf{some } X \ P_2}\ (sI)$$

The decidability of N_1^+-derivability is shown in [3].

5.2. Extracting meanings for relative clauses and relative pronouns

As before, we introduce an I-function for (rel).

$$I_r = \lambda z_1^n \lambda z_2^{t_p} \lambda \mathbf{j} \lambda \Gamma . \{ \frac{\mathcal{D}_1 \quad \mathcal{D}_2}{\Gamma \vdash \mathbf{j} \textsf{ isa } \nu(z_1) \textsf{ who } \rho(\mathcal{D}_2)[-]}\ (relI) : \begin{array}{c} \mathcal{D}_1 \in \mathbf{ex}(z_1)(\mathbf{j})(\Gamma) \\ \& \mathcal{D}_2 \in \mathbf{ex}(z_2)(\mathbf{j})(\Gamma) \end{array} \} \tag{18}$$

The meaning of compound nouns resembles that of lexical nouns.

$$\begin{array}{l} [\![X \textsf{ who(m) } S[-]]\!] = \lambda \mathbf{j} . [\![\mathbf{j} \textsf{ isa } X \textsf{ who(m) } S[-]]\!] = \\ \lambda \mathbf{j} \lambda \Gamma . I_r([\![X]\!])([\![S[\mathbf{j}]]\!])(\mathbf{j})(\Gamma) \end{array} \tag{19}$$

By (16,17),

$$[\![X \textsf{ who(m) } S[-]]\!] = [\![\textsf{who(m)}]\!]([\![S[-]]\!]([\![X]\!])) \tag{20}$$

Introducing a variable z_1^n and abstracting over it in (19), by $\mathbf{F}$, yields

$$[\![\textsf{who(m) } S[-]]\!] = \lambda z_1^n \lambda \mathbf{j} \lambda \Gamma . I_r(z_1)([\![S[\mathbf{j}]]\!])(\mathbf{j})(\Gamma) \tag{21}$$

and by another variable introduction and abstraction over it,

$$[\![\textsf{who(m)}]\!] = \lambda z_2^{t_p} \lambda z_1^n \lambda \mathbf{j} \lambda \Gamma . I_r(z_1)(z_2)(\mathbf{j})(\Gamma) \tag{22}$$

[‖] A proof-theoretical definition of monotonicity, as well as other properties of *dp*s, like conservativity, are presented and discussed in more detail in [1].

6. Conclusions

Our point of departure is a dedicated natural-deduction proof-system, by which a *sentential proof-theoretic semantics* is defined for a fragment of natural language, providing an *effective* semantics. The original presentation of proof-theoretical sentential meanings in (the full version of) [3] contains a comparison to related approaches, which we do not repeat here. We show, alluding to Frege's *context principle*, how to extend the semantics, obtaining the novel concept of proof-theoretic meanings for subsentential phrases, down to single words. This is done by obtaining the *contribution* of word meanings to sentential meanings, driven by the categories of a type-logical grammar for the language fragment. The resulting semantics is "almost compositional": when sentencehood of a sequence of single words is derived in the grammar, the meaning obtained by combining the word-meanings is the original sentential meaning. The types of meanings used during this process are *proof-theoreticaly interpreted types*, replacing Montague's denotational model-theoretic interpretation of types, the former interpreted as derivations in the dedicated sentential system (and functions therein), while the latter are interpreted in Henkin frames.

7. Appendix

The (associative, product-free) Lambek-calculus **L**, on which the TLG considered here is based, is presented (in its natural-deduction presentation) in Figure 3 (cf. [6], p.120). Categories $\mathcal{C}$ are formed from a given finite set $\mathcal{B}$ of *basic categories* by** *directed implications*. Note that contexts Γ here are *sequences* (not sets or multisets) of *signs*, pairs $\mathbf{c} : x$ ($\mathbf{c}$ – a category, x – a variable), where $subjects(\Gamma)$, the variables in Γ, are pairwise distinct.

A *type-logical grammar (TLG)* G over a set Σ of *terminal symbols* (*words* in case of natural language), is given by a *lexicon* α_G, assigning to each $\sigma{\in}\Sigma$ a finite set of pairs of categories from $\mathcal{C}$, and meanings (proof-theoretic meanings here), and a designated category $\mathbf{c}_0$. The lexical assignment is naturally lifted by concatenation to $\alpha[\![w]\!]$, $w{\in}\Sigma^+$ by setting $\alpha_G[\![\sigma w]\!] = \alpha_G[\![\sigma]\!]\alpha_G[\![w]\!]$. The *interpreted* language $L[\![G]\!]$ defined by G is:

$$L[\![G]\!] =^{df.} \{(w, M){\in}\Sigma^+ \times Term \mid \exists\Gamma{\in}\alpha_G[\![w]\!].\ \Gamma{\vdash_{\mathbf{L}}}(\mathbf{c}_0, M)\}$$

where M is a linear λ-term over $subjects(\Gamma)$. The actual meaning of w is obtained by substituting M_i, the meaning term of $\alpha_G[\![w_i]\!]$ for x_i in M, where

**In Lambek's original notation, \ and / are used instead of directed arrows.

$$\frac{}{\mathbf{c} : x \rhd \mathbf{c} : x}\ (Ax)$$

$$\frac{\Gamma_1 \rhd \mathbf{c}_1 : N \quad \Gamma_2 \rhd (\mathbf{c}_1 \rightarrow \mathbf{c}_2) : M}{\Gamma_1\Gamma_2 \rhd \mathbf{c}_2 : M(N)}\ (\rightarrow E)$$

$$\frac{\Gamma_2 \rhd (\mathbf{c}_2 \leftarrow \mathbf{c}_1) : M \quad \Gamma_1 \rhd \mathbf{c}_1 : N}{\Gamma_2\Gamma_1 \rhd \mathbf{c}_2 : M(N)}\ (\leftarrow E)$$

$$\frac{[\mathbf{c}_1]_i : x,\ \Gamma \rhd \mathbf{c}_2 : M}{\Gamma \rhd (\mathbf{c}_1 \rightarrow \mathbf{c}_2) : \lambda x.M}\ (\rightarrow I_i)$$

$$\frac{\Gamma,\ [\mathbf{c}_1]_i : x \rhd \mathbf{c}_2 : M}{\Gamma \rhd (\mathbf{c}_2 \leftarrow \mathbf{c}_1) : \lambda x.M}\ (\leftarrow I_i),$$

where in the I-rules Γ is not empty and x is fresh.

Figure 3. The **L**-calculus

$w = w_1, \cdots, w_n$. See [6] for a detailed exposition of TLG.

8. Acknowledgements

The work reported in this paper was supported by EPSRC grant number EP/D064015/1, and grant number 2006938 by the Israeli Academy for Sciences (ISF), both gratefully acknowledged.

References

[1] Gilad Ben-Avi and Nissim Francez. A proof-theoretic reconstruction of generalized quantifiers. *Submitted for publication*, 2009.

[2] Nissim Francez and Gilad Ben-Avi. Proof-theoretic semantic values for logical operators. *Synthese*, 2009. Under refereeing.

[3] Nissim Francez and Roy Dyckhoff. Proof-theoretic semantics for a natural language fragment. In *Proceedings of the 10th and 11th meetings of the Association for Mathematics of Language (MOL)*, July 2007, to appear. Full version under refereeing for *Linguistics and Philosophy*.

[4] Gottlob Frege. *Grundlagen der Aritmetik, 1884*. translated as The Foundations of Arithmetic, J.L. Austin (trans.), (2nd edition) Basil Blackwell, Oxford, 1953.

[5] Wilfrid Hodges. Formal features of compositionality. *Journal of Logic, Language, and Information*, 10:7–28, 2001.

[6] Michael Moortgat. Categorial type logics. In Johan van Benthem and Alice ter Meulen, editors, *Handbook of Logic and Language*, pages 93–178. North Holland, 1997.

[7] Lawrence Moss. Logics with verbs. 2007. In preparation.

[8] Peter Pagin. Compositionality, understanding, and proofs. *Mind*, 2009, to appear.

[9] Jaroslav Peregrin. Inferentialism and the compositionality of meaning. *International Review of Pragmatics*, 2009, to appear.

[10] Richard J. Stainton. Context principle. In Keith Brown (editor-in chief), editor, *Encyclopedia of Language and Linguistics*, volume 3, pages 108–115. Elsevier, 2006. Second edition.