

Towards a Montagovian Account of Dynamics

Philippe de Groote
Inria-Lorraine

1. Introduction

The thesis we support in this paper is that dynamic phenomena that are often considered to fall outside of the scope of Montague semantics may be handled by using the standard tools of mathematical logic. In particular, we do not want to appeal to any kind of dynamic notion such as assignment functions that are updated in an imperative way (which results in the so-called destructive assignment problem). We use instead the simply typed λ -calculus, and the only dynamic notion on which we rely is the usual relation of β -reduction. There are several advantages in doing so:

- We do not commit ourselves with any specific theory. Consequently, our approach is independent of the target logic that is used to express the meaning of the expressions. The user may choose her or his favorite framework, which possibly comes with its well developed proof and model theories. This allows one to reuse mathematical and logical knowledge that has been developed for other areas than computational linguistics.
- The variables we use are as usual in mathematics. In particular the notions of free and bound occurrences of a variable are the standard ones. As it is the case that we do not use any assignment function, the meaning we assign to expressions are closed λ -terms. Our approach is therefore akin to Jacobson's variable-free semantics (Jacobson 1999). In fact, the relation between the two approaches is deeper than a simple analogy. We will not have enough space, however, to develop this point in this paper.
- Finally, from an epistemological point of view, we find that explaining some phenomenon in term of an existing, well established, and well understood theory is more satisfactory than explaining it in term of a new (possibly made-up) theory. In the latter case, indeed, there is a risk that the phenomenon to be explained is simply reflected in the new theory by some ad hoc primitive.

As a canonical example of dynamics, we will address the problem of intra- and intersentential pronominal binding, including quantification scoping and donkey sentences. When trying to give a logical semantics to a discourse, one faces the problem that quantifiers should dynamically extend their scope from one sentence to another. This problem is solved in Kamp's Discourse Representation Theory (Kamp and Reyle 1993) by using sets of reference markers. From an intuitive point of view, these reference markers act as existential quantifiers. Nevertheless, from a

technical point of view, they must be considered as free variables, which explains that they can indeed extend their scope dynamically.

Because of this hybrid status of the variables, one has to choose carefully the names of the new reference markers when constructing a Discourse Representation Structure (DRS). This problem is particularly salient when trying to combine Discourse Representation Theory and Montague semantics (Groenendijk and Stokhof 1990, Muskens 1995) because the merging of two DRSs may result in variable clashes or, more precisely, in variable assignments that destroy the values previously assigned. As a consequence, merging two DRSs is either a partial operation or must include some kind of sophisticated variable renaming in order to avoid unwanted side effects.

In this paper, we tackle this problem by proposing a new way of combining Montague semantics and discourse representation. To this end, we provide Montague semantics with an appropriate notion of context that allow for some kind of dynamics without violating the usual quantifier scoping rules. In particular, the notions of free and bound variables are as usual, and the operation corresponding to the merging of two DRSs is performed by standard β -reduction.

2. Expressing Propositions in Context

Quoting van Eijck and Kamp (1997), *the key idea behind (...) Discourse Representation Theory is that each new sentence of a discourse is interpreted in the context provided by the sentences preceding it*. Here, we go two steps further:

1. We will interpret a sentence according to both its left and right contexts. In other words, the interpretation of a sentence will depend, on the one hand, on the context made of the sentences preceding it, and on the other hand, on the context made of the sentences following it.
2. These two kind of contexts will be abstracted over the meaning of the sentences. In other words, the contexts will be passed as parameters to the functions that assign a semantics to the sentences.

These two ideas will be made technically precise in a Montagovian framework.

Montague semantics (Montague 1973) is based on Church's simple type theory (Church 1940). This theory provides a full hierarchy of functional types built upon two atomic types: ι , the type of individuals, and o , the type of propositions.¹ In order to accommodate a notion of context, we add a third atomic type, namely, γ . This type stands for the type of the left contexts.

¹Montague uses different notations: e for the type of individuals (a.k.a. *entities*), and t for the type of propositions (a.k.a. *truth values*). We prefer to stick to Church's original notations in order to stress that the propositions do not need to be semantically interpreted as Boolean truth values. For instance, in order to take intensionality into account, the atomic type o may be interpreted as the set $\mathbf{w} \rightarrow \mathbf{t}$, where \mathbf{w} is the set of possible worlds (Carpenter 1996). In fact, our approach is compatible with non standard logics such as, for instance, intuitionistic logic.

The type of the right contexts is different. As usual, we only consider discourses made of declarative sentences. Ultimately, such discourses must be interpreted as propositions. Now, a right context is a piece of discourse that will be interpreted as a proposition provided it is given its left context. Consequently, the type of the right contexts must be $\gamma \rightarrow o$. Such a type may be seen as the type of a *continuation* (Strachey and Wadsworth 1974), which is indeed a standard way of modeling right contexts in denotational semantics.²

Let s be the syntactic category of sentences, and t be the syntactic category of texts (or discourses). As we said, we intend to abstract over the meaning of a sentence its left and right contexts. Consequently, we obtain the following semantic interpretation of s :³

$$(1) \quad \llbracket s \rrbracket = \gamma \rightarrow (\gamma \rightarrow o) \rightarrow o$$

and similarly for t :

$$(2) \quad \llbracket t \rrbracket = \gamma \rightarrow (\gamma \rightarrow o) \rightarrow o$$

We now face the following question: how does one compose meanings of sentences in order to get the meaning of a discourse? Let D be a piece of discourse, and S be a sentence. The semantics of the discourse made of D and S is given by the following semantic equation:

$$(3) \quad \llbracket D.S \rrbracket = \lambda e \phi. \llbracket D \rrbracket e (\lambda e'. \llbracket S \rrbracket e' \phi)$$

This equation may be explained as follows. The semantics of $(D.S)$ must be a term of type $\gamma \rightarrow (\gamma \rightarrow o) \rightarrow o$. It has therefore two parameters: e of type γ , which represents the left context of $(D.S)$, and ϕ of type $\gamma \rightarrow o$, which represents its right context. Then the body of the term, which must be of type o , is contributed by the semantic interpretation of the first component of $(D.S)$, namely, $\llbracket D \rrbracket$. Now, $\llbracket D \rrbracket$ is itself a term of type $\gamma \rightarrow (\gamma \rightarrow o) \rightarrow o$. Consequently, it must be passed its left and right contexts as arguments. The left context of D is nothing but the left context of $(D.S)$, i.e., e . As for the right context of D , it must be made of S and of the right context of $(D.S)$. This explains the form of the second argument, $\lambda e'. \llbracket S \rrbracket e' \phi$.

At this point, it would be interesting to see what is the possible relation between Kamp's DRSs and λ -terms of type $\gamma \rightarrow (\gamma \rightarrow o) \rightarrow o$. A DRS is made of a set of variables x_1, \dots, x_n , called reference markers, which are interpreted existentially, and of a set of logical constraints C_1, \dots, C_m , which are interpreted conjunctively:

$x_1 \dots x_n$
C_1
\vdots
C_m

²The use of continuations in natural language semantics has been recently advocated by Barker (2002, 2004), de Groote (2001b), and Shan (2004).

³In order to save parentheses when writing functional types, we apply the usual notational convention that $a \rightarrow b \rightarrow c$ stands for $a \rightarrow (b \rightarrow c)$.

To such a structure, the following λ -term corresponds :

$$\lambda e\phi. \exists x_1 \dots x_n. C_1 \wedge \dots \wedge C_m \wedge \phi e'$$

where e' is a context made of e and of the variables x_1, \dots, x_n . Therefore, these variables fall in the scope of the *continuation* of the discourse, i.e., ϕ . The precise way e' is constructed depends of the concrete structure given to the contexts. We will be more specific about this in the next section.

3. Updating and Accessing the Context

Data of type γ (i.e., left contexts) are mainly needed to solve pronominal anaphora. We will not discuss any real anaphoric resolution procedure in this paper, but focus on the simpler task of computing a discourse representation assuming that the anaphora are solved by some choice operators that act as oracles. We will use different oracles (sel_{he} , sel_{she} , sel_{it} , sel_{him} , sel_{her} , ...) for the different possible pronouns. These operators take a left context as their argument and yield back an individual. Hence, they obey the following general type assignment scheme:

$$(4) \quad \text{sel}_x : \gamma \rightarrow \iota$$

We also need some operators that allow the contexts to be updated. For the purpose of the present discussion, we may consider that the data of type γ are simply finite sets of individuals on which act the choice operators. If e is such a finite set and a, b are individuals, we write $a::e$ for $\{a\} \cup e$, and $a::b::e$ for $a::(b::e)$. The type of the operator “ $::$ ” is then as follows:

$$(5) \quad _::_ : \iota \rightarrow \gamma \rightarrow \gamma$$

Now, consider the two following sentences:

$$(6) \quad \text{John loves Mary. He smiles at her.}$$

Using the above primitives, these two sentences may be given the following representations, respectively:

$$(7) \quad \lambda e\phi. \text{love} \mathbf{j} \mathbf{m} \wedge \phi (\mathbf{m}::\mathbf{j}::e)$$

$$(8) \quad \lambda e\phi. \text{smile} (\text{sel}_{\text{he}} e) (\text{sel}_{\text{her}} e) \wedge \phi e$$

Then, composing (7) and (8) through Equation (3), we obtain a λ -term that β -reduces as follows:

$$\begin{aligned} & \lambda e\phi. (\lambda e\phi. \text{love} \mathbf{j} \mathbf{m} \wedge \phi (\mathbf{m}::\mathbf{j}::e)) e (\lambda e'. (\lambda e\phi. \text{smile} (\text{sel}_{\text{he}} e) (\text{sel}_{\text{her}} e) \wedge \phi e) e' \phi) \\ \rightarrow_{\beta} & \lambda e\phi. (\lambda \phi. \text{love} \mathbf{j} \mathbf{m} \wedge \phi (\mathbf{m}::\mathbf{j}::e)) (\lambda e'. (\lambda e\phi. \text{smile} (\text{sel}_{\text{he}} e) (\text{sel}_{\text{her}} e) \wedge \phi e) e' \phi) \\ \rightarrow_{\beta} & \lambda e\phi. (\lambda \phi. \text{love} \mathbf{j} \mathbf{m} \wedge \phi (\mathbf{m}::\mathbf{j}::e)) (\lambda e'. (\lambda \phi. \text{smile} (\text{sel}_{\text{he}} e') (\text{sel}_{\text{her}} e') \wedge \phi e') \phi) \\ \rightarrow_{\beta} & \lambda e\phi. (\lambda \phi. \text{love} \mathbf{j} \mathbf{m} \wedge \phi (\mathbf{m}::\mathbf{j}::e)) (\lambda e'. \text{smile} (\text{sel}_{\text{he}} e') (\text{sel}_{\text{her}} e') \wedge \phi e') \\ \rightarrow_{\beta} & \lambda e\phi. \text{love} \mathbf{j} \mathbf{m} \wedge (\lambda e'. \text{smile} (\text{sel}_{\text{he}} e') (\text{sel}_{\text{her}} e') \wedge \phi e') (\mathbf{m}::\mathbf{j}::e) \\ \rightarrow_{\beta} & \lambda e\phi. \text{love} \mathbf{j} \mathbf{m} \wedge \text{smile} (\text{sel}_{\text{he}} (\mathbf{m}::\mathbf{j}::e)) (\text{sel}_{\text{her}} (\mathbf{m}::\mathbf{j}::e)) \wedge \phi (\mathbf{m}::\mathbf{j}::e) \end{aligned}$$

Then, assuming that the choice operators make the correct predictions (which is only a matter of gender in the present case), we obtain the following semantic representation for (6):

$$\lambda e \phi. \mathbf{love} \mathbf{j} \mathbf{m} \wedge \mathbf{smile} \mathbf{j} \mathbf{m} \wedge \phi (\mathbf{m}::\mathbf{j}::e)$$

It remains to explain how terms such as (7) or (8) are constructed from the lexical components of (6). Besides s and t , we consider two other atomic syntactic categories: n for nouns, and np for noun phrases. Then, we assign the following syntactic categories to the several components of (6):⁴

$$\begin{aligned} \text{John, Mary, he, her} & : np \\ \text{loves, smiles_at} & : np \rightarrow np \rightarrow s \end{aligned}$$

The next step is to specify the semantic interpretation of np . In Montague semantics, noun phrases are interpreted as terms of type

$$(9) \quad (t \rightarrow o) \rightarrow o.$$

In the present approach, the interpretation of a proposition depends upon a context. Consequently, let us parameterize the two occurrences of o in (9) with γ . This leads to the following semantic interpretation:

$$(10) \quad \llbracket np \rrbracket = (t \rightarrow \gamma \rightarrow o) \rightarrow \gamma \rightarrow o$$

As we will see, the first occurrence of γ in (10) allows noun phrases such as *John* or *Mary* to update the current context, while the second occurrence of γ allows pronominal noun phrases such as *he* or *her* to access the current context. In fact, we will not discuss further Equation (10) because it is provisional.

We are now in a position of giving proper semantic interpretations to the different noun phrases occurring in (6):

$$(11) \quad \llbracket \text{John} \rrbracket = \lambda \psi e. \psi \mathbf{j} (\mathbf{j}::e)$$

$$(12) \quad \llbracket \text{Mary} \rrbracket = \lambda \psi e. \psi \mathbf{m} (\mathbf{m}::e)$$

$$(13) \quad \llbracket \text{he} \rrbracket = \lambda \psi e. \psi (\text{sel}_{\text{he}} e) e$$

$$(14) \quad \llbracket \text{her} \rrbracket = \lambda \psi e. \psi (\text{sel}_{\text{her}} e) e$$

It is not difficult to see that the right-hand sides of the above equations are λ -terms of the appropriate type, i.e., $(t \rightarrow \gamma \rightarrow o) \rightarrow \gamma \rightarrow o$.

The interpretations of the transitive verbs seem to be more involved because they must be of type $\llbracket np \rrbracket \rightarrow \llbracket np \rrbracket \rightarrow \llbracket s \rrbracket$. Nevertheless, mutatis mutandis, they fairly correspond to Montague usual interpretation. The values of $\llbracket \text{loves} \rrbracket$ and $\llbracket \text{smiles_at} \rrbracket$ are respectively as follows (assuming subject wide scope):

⁴In order not to obfuscate the discussion with syntactic details, we give the syntactic categories without any feature such as *gender*, *number*, *case*, ... As for the non atomic categories, we only specify their functionality. It should be clear, however, that our approach is completely consistent with more realistic grammatical formalisms such as, among others, type logical grammars (Morrill 1994), multimodal grammars (Moortgat 1997), abstract categorial grammars (de Groote 2001a), lambda-grammars (Muskens 2001), or high-order categorial grammars (Pollard 2004).

$$(15) \quad \llbracket \text{loves} \rrbracket = \lambda o s e \phi . s (\lambda x e . o (\lambda y e . \mathbf{love} x y \wedge \phi e) e) e$$

$$(16) \quad \llbracket \text{smiles_at} \rrbracket = \lambda o s e \phi . s (\lambda x e . o (\lambda y e . \mathbf{smile} x y \wedge \phi e) e) e$$

Besides the parameters of type γ , the most significant difference with respect to Montague interpretation is the presence of the conjunct (ϕe) , which models the right context in which the sentence produced by the transitive verb occurs.

In order to see contexts at work, it is interesting to see how the meaning of the first sentence of (6) may be computed from the above semantic lexical entries:

$$\begin{aligned}
\llbracket \text{loves} \rrbracket \llbracket \text{Mary} \rrbracket \llbracket \text{John} \rrbracket &= (\lambda o s e \phi . s (\lambda x e . o (\lambda y e . \mathbf{love} x y \wedge \phi e) e) e) \llbracket \text{Mary} \rrbracket \llbracket \text{John} \rrbracket \\
\rightarrow_{\beta} & (\lambda s e \phi . s (\lambda x e . \llbracket \text{Mary} \rrbracket (\lambda y e . \mathbf{love} x y \wedge \phi e) e) e) \llbracket \text{John} \rrbracket \\
\rightarrow_{\beta} & \lambda s e \phi . \llbracket \text{John} \rrbracket (\lambda x e . \llbracket \text{Mary} \rrbracket (\lambda y e . \mathbf{love} x y \wedge \phi e) e) e \\
&= \lambda s e \phi . (\lambda \psi e . \psi \mathbf{j} (\mathbf{j} :: e)) (\lambda x e . \llbracket \text{Mary} \rrbracket (\lambda y e . \mathbf{love} x y \wedge \phi e) e) e \\
\rightarrow_{\beta} & \lambda s e \phi . (\lambda e . (\lambda x e . \llbracket \text{Mary} \rrbracket (\lambda y e . \mathbf{love} x y \wedge \phi e) e) \mathbf{j} (\mathbf{j} :: e)) e \\
\rightarrow_{\beta} & \lambda s e \phi . (\lambda x e . \llbracket \text{Mary} \rrbracket (\lambda y e . \mathbf{love} x y \wedge \phi e) e) \mathbf{j} (\mathbf{j} :: e) \\
\rightarrow_{\beta} & \lambda s e \phi . (\lambda e . \llbracket \text{Mary} \rrbracket (\lambda y e . \mathbf{love} \mathbf{j} y \wedge \phi e) e) (\mathbf{j} :: e) \\
\rightarrow_{\beta} & \lambda s e \phi . \llbracket \text{Mary} \rrbracket (\lambda y e . \mathbf{love} \mathbf{j} y \wedge \phi e) (\mathbf{j} :: e) \\
&= \lambda s e \phi . (\lambda \psi e . \psi \mathbf{m} (\mathbf{m} :: e)) (\lambda y e . \mathbf{love} \mathbf{j} y \wedge \phi e) (\mathbf{j} :: e) \\
\rightarrow_{\beta} & \lambda s e \phi . (\lambda e . (\lambda y e . \mathbf{love} \mathbf{j} y \wedge \phi e) \mathbf{m} (\mathbf{m} :: e)) (\mathbf{j} :: e) \\
\rightarrow_{\beta} & \lambda s e \phi . (\lambda y e . \mathbf{love} \mathbf{j} y \wedge \phi e) \mathbf{m} (\mathbf{m} :: \mathbf{j} :: e) \\
\rightarrow_{\beta} & \lambda s e \phi . (\lambda e . \mathbf{love} \mathbf{j} \mathbf{m} \wedge \phi e) (\mathbf{m} :: \mathbf{j} :: e) \\
\rightarrow_{\beta} & \lambda s e \phi . \mathbf{love} \mathbf{j} \mathbf{m} \wedge \phi (\mathbf{m} :: \mathbf{j} :: e)
\end{aligned}$$

Other interesting examples are provided by quantified noun phrases as in the following example:

(17) Every man loves a woman.

The syntactic categories of the new lexical items are as follows:

$$\begin{aligned}
\text{man, woman} &: n \\
\text{every, a} &: n \rightarrow np
\end{aligned}$$

We keep Montague's interpretation of nouns as properties:

$$(18) \quad \llbracket n \rrbracket = \iota \rightarrow o,$$

which induces the following simple semantic translations:

$$(19) \quad \llbracket \text{man} \rrbracket = \lambda x . \mathbf{man} x$$

$$(20) \quad \llbracket \text{woman} \rrbracket = \lambda x . \mathbf{woman} x$$

The determiners must be interpreted as λ -terms of type $\llbracket n \rrbracket \rightarrow \llbracket np \rrbracket$, which is spelled out as follows:

$$(\iota \rightarrow o) \rightarrow (\iota \rightarrow \gamma \rightarrow o) \rightarrow \gamma \rightarrow o.$$

Quantified noun phrases should be allowed to update the current context. This explains the following interpretations of the quantifiers:

$$(21) \quad \llbracket \text{a} \rrbracket = \lambda n \psi e . \exists x . n x \wedge \psi x (x :: e)$$

$$(22) \quad \llbracket \text{every} \rrbracket = \lambda n \psi e . \forall x . n x \supset \psi x (x :: e)$$

Again, up to the management of contexts, the above interpretations correspond fairly to Montague's standard interpretations. Then, the semantic representation of (17) may be computed by reducing the λ -term

$$(23) \quad \llbracket \text{love} \rrbracket (\llbracket a^2 \rrbracket \llbracket \text{woman} \rrbracket) (\llbracket \text{every}^1 \rrbracket \llbracket \text{man} \rrbracket)$$

which yields the following result:

$$(24) \quad \lambda e\phi. \forall x. \mathbf{man}x \supset (\exists y. \mathbf{woman}y \wedge \mathbf{love}xy \wedge \phi(y::x::e))$$

4. Systematizing the Approach

The few examples given in the previous section demonstrate the way our approach, based on Equation (1), allows Montague semantics to be provided with a notion of context. Nevertheless, some of the semantic interpretations we propose presents several limitations.

A first defect, due to Equation (10), concerns the scope of the variables introduced by the quantified noun phrases. In Kamp's DRT, there are accessibility constraints that restrict the scope of the reference markers occurring in a DRS. This allows some problematic anaphoric links to be ruled out. For instance, it is not possible to process the following discourse:

$$(25) \quad \text{Every man loves a woman. ?He smiles at her.}$$

Indeed, the reference markers corresponding to the noun phrases *every man* and *a woman* are not available when processing the second sentence. Our approach should reflect properly such accessibility constraints. Unfortunately, this is not the case with the semantic interpretation given so far to the determiner *every*. This appears in (24) where the conjunct $\phi(y::x::e)$, which corresponds to the right context of the sentence, falls in the scope of both quantifiers. This would allow (25) to be processed in a completely unsatisfactory way.

Another defect concerns the interpretation of nouns as properties. This is certainly fine for simple nouns such as *man* or *woman*, but cannot work for more complex expressions such as *man who loves a woman*. Expressions of this kind, which belong to category n , possibly include noun phrases or sentences as sub-components. Therefore, processing a noun possibly appeals to the processing of noun phrases and sentences. Now, when processing noun phrases and sentences, one should be able to access contexts. Consequently, when processing a noun, one should also be able to access contexts. This implies that the interpretation of a noun should include somehow parameters of type γ .

In order to solve these problems, let us go back to Montague's original interpretations of the syntactic categories:

$$(26) \quad \llbracket s \rrbracket = o$$

$$(27) \quad \llbracket n \rrbracket = t \rightarrow o$$

$$(28) \quad \llbracket np \rrbracket = (t \rightarrow o) \rightarrow o$$

These equations may be rephrased as follows:

$$(29) \quad \llbracket s \rrbracket = o$$

$$(30) \quad \llbracket n \rrbracket = \iota \rightarrow \llbracket s \rrbracket$$

$$(31) \quad \llbracket np \rrbracket = (\iota \rightarrow \llbracket s \rrbracket) \rightarrow \llbracket s \rrbracket$$

Now, in order to be completely consistent with our analysis of Section 2, we must replace Equation (29) with Equation (1), and propagate this change in both Equations (30) and (31). This yields the following interpretations of n and np :

$$(32) \quad \llbracket n \rrbracket = \iota \rightarrow \gamma \rightarrow (\gamma \rightarrow o) \rightarrow o$$

$$(33) \quad \llbracket np \rrbracket = (\iota \rightarrow \gamma \rightarrow (\gamma \rightarrow o) \rightarrow o) \rightarrow \gamma \rightarrow (\gamma \rightarrow o) \rightarrow o$$

Although these two equations have been obtained in a systematic way, we must justify them on some intuitive ground because they appear to be rather complex. Nevertheless, we claim that they are not counterintuitive.

As we said, the interpretation of a noun possibly depends on the current context. A natural way of implementing this dependency is to abstract our notion of context (made of both a left and a right context) over the usual interpretation of a noun. This would yield $\gamma \rightarrow (\gamma \rightarrow o) \rightarrow \iota \rightarrow o$ as the interpretation of n , which is indeed isomorphic to (32).

As for the noun phrases, Equation (10) has been obtained by parameterizing with γ both occurrences of o in Montague's type $(\iota \rightarrow o) \rightarrow o$. By doing so, we went halfway to providing np with its proper interpretation. There is indeed no reason for giving a greater place to the left contexts than to the right ones. Consequently, we should also have parameterized the interpretation of np with $\gamma \rightarrow o$, which is the type of the right contexts. Doing so, we obtain (33) as an interpretation. This explanation is maybe too technical to be completely satisfactory. Therefore, let us see how Equation (31) sheds some light on Equation (33). By interpreting a noun phrase as a term of type $(\iota \rightarrow o) \rightarrow o$, Montague identifies an entity with the set of all its properties. This relies on Leibniz's principle which says that two entities are equal if there is no property that allows ones to discriminate one from the other. In Montague's case, a property (i.e., a term of type $\iota \rightarrow o$) may be seen as the interpretation of a sentence with a "hole" of type ι . In our case, a term of type $\iota \rightarrow \llbracket s \rrbracket$ may be seen as the interpretation of a complete discourse with a "hole" of type ι . Let us call such a term a semantic context. Then, what Equation (31) says is that we consider two noun phrases as semantically equivalent if there is no semantic context that allows ones to distinguish one from the other. Of course, as in Montague's case, this also allows for the interpretations of noun phrases that do not correspond to any entity, such as *nobody*.

Let us see now how Equations (32) and (33) affect the semantic interpretations given in Section 3.

4.1. Nouns

The interpretation of a noun such as *man* should be a λ -term of type (32), i.e.,

$$t \rightarrow \gamma \rightarrow (\gamma \rightarrow o) \rightarrow o$$

Consequently, it takes three parameters of types t , γ , and $\gamma \rightarrow o$, respectively. Then, assuming that the main mode of semantic composition is conjunctive, we obtain the following interpretation:

$$(34) \quad \llbracket \text{man} \rrbracket = \lambda x e \phi. \mathbf{man} x \wedge \phi e$$

4.2. Noun Phrases

The interpretation of a noun phrase such as *Mary*, which must be of type (33), is adapted as follows:

$$(35) \quad \llbracket \text{Mary} \rrbracket = \lambda \psi e \phi. \psi \mathbf{m} e (\lambda e. \phi (\mathbf{m} :: e))$$

The first parameter, ψ , is of type $t \rightarrow \llbracket s \rrbracket$. It needs three arguments of type t , γ , and $\gamma \rightarrow o$, respectively. The first argument must be an individual, in this case \mathbf{m} . The second argument corresponds to the current left context, namely, e . The third argument (i.e., $\lambda e. \phi (\mathbf{m} :: e)$) corresponds to a new right context made of the current right context (ϕ) together with an updating of the current left context.

As for the interpretation of a pronominal noun phrase such as *he*, it is straightforward:

$$(36) \quad \llbracket \text{he} \rrbracket = \lambda \psi e \phi. \psi (\text{sel}_{\text{he}} e) e \phi$$

4.3. Transitive Verbs

The interpretation of a transitive verb such as *love* must obey the following type:

$$(37) \quad \llbracket np \rrbracket \rightarrow \llbracket np \rrbracket \rightarrow \llbracket s \rrbracket$$

By unfolding completely the definitions of $\llbracket np \rrbracket$ and $\llbracket s \rrbracket$ in (37), one would obtain a quite complex type scheme. There is no reason, however, for doing so. In order to understand the interpretations of the transitive verbs, we first unfold partially the two occurrences of $\llbracket np \rrbracket$:

$$(38) \quad ((t \rightarrow \llbracket s \rrbracket) \rightarrow \llbracket s \rrbracket) \rightarrow ((t \rightarrow \llbracket s \rrbracket) \rightarrow \llbracket s \rrbracket) \rightarrow \llbracket s \rrbracket$$

then, we unfold the first occurrence of $\llbracket s \rrbracket$:

$$(39) \quad ((t \rightarrow \gamma \rightarrow (\gamma \rightarrow o) \rightarrow o) \rightarrow \llbracket s \rrbracket) \rightarrow ((t \rightarrow \llbracket s \rrbracket) \rightarrow \llbracket s \rrbracket) \rightarrow \llbracket s \rrbracket$$

This explains the following interpretation, which remains quite close to Montague's one:

$$(40) \quad \llbracket \text{love} \rrbracket = \lambda o s. s (\lambda x. o (\lambda y e \phi. \mathbf{love} x y \wedge \phi e))$$

4.4. Determiners

The interpretations of the determiners are more interesting. They must be of type

$$(41) \quad \llbracket n \rrbracket \rightarrow \llbracket np \rrbracket$$

which partially unfolds as:

$$(42) \quad \llbracket n \rrbracket \rightarrow (\iota \rightarrow \llbracket s \rrbracket) \rightarrow \gamma \rightarrow (\gamma \rightarrow o) \rightarrow o$$

Consequently, they have four parameters of types $\llbracket n \rrbracket$, $\iota \rightarrow \llbracket s \rrbracket$, γ , and $\gamma \rightarrow o$, respectively. The interpretation of the existential determiner is as follows:

$$(43) \quad \llbracket a \rrbracket = \lambda n \psi e \phi. \exists x. n x e (\lambda e. \psi x(x::e) \phi)$$

The first parameter, which is of type $\llbracket n \rrbracket$ is the head of the interpretation. It needs three arguments: an individual, which is the logical variable introduced by the existential quantifier, a left context, which is the current left context (e), and finally its right context, which is expressed as $(\lambda e. \psi x(x::e) \phi)$. Using (34) and (43), it is not difficult to see that the interpretation of a noun phrase such as *a man* reduces to:

$$(44) \quad \lambda \psi e \phi. \exists x. \mathbf{man} x \wedge \psi x(x::e) \phi$$

It is important to note that the conjunction occurring in this term does not come from the interpretation of the determiner but from the interpretation of the noun.

Finally, the interpretation of the universal quantifier *every* is as follows:

$$(45) \quad \llbracket \text{every} \rrbracket = \lambda n \psi e \phi. (\forall x. \neg(n x e (\lambda e. \neg(\psi x(x::e) (\lambda e. \top)))))) \wedge \phi e$$

Remark the use of negation in order to turn the conjunction that is implicitly contained in n into an implication.⁵ Remark also that the right context given to ψ is not the current right context but an empty right context $(\lambda e. \top)$. This allows DRT accessibility constraints to be satisfied by limiting the scope of the universal quantification.

4.5. Relative Pronouns

The case of the relative pronouns is also interesting. A relative pronoun such as *who* must be of type:

$$(46) \quad (\llbracket np \rrbracket \rightarrow \llbracket s \rrbracket) \rightarrow \llbracket n \rrbracket \rightarrow \llbracket n \rrbracket$$

This type may be partially unfolded as follows:

$$(47) \quad (\llbracket np \rrbracket \rightarrow \llbracket s \rrbracket) \rightarrow \llbracket n \rrbracket \rightarrow \iota \rightarrow \gamma \rightarrow (\gamma \rightarrow o) \rightarrow o$$

The first parameter, which is of type $(\llbracket np \rrbracket \rightarrow \llbracket s \rrbracket)$, corresponds to the relative clause, and the second parameter, of type $\llbracket n \rrbracket$, corresponds to the antecedent of the relative

⁵This use of negation together with de Morgan's laws seems to indicate that our approach is tied to classical logic. It is actually not the case because the way we use negation respects intuitionistic polarity constraints. Therefore, in a intuitionistic setting, \neg may be considered as a polarity switching operator.

pronoun. The other parameters come from the unfolding of the second occurrence of $\llbracket n \rrbracket$ in (46). They corresponds respectively to the individual to which the resulting semantic property applies, to the current left context, and to the continuation of the discourse. This leads to the following interpretation:

$$(48) \quad \llbracket \text{who} \rrbracket = \lambda r n x e \phi . n x e (\lambda e . r (\lambda \psi . \psi x) e \phi)$$

The head of this interpretation is n , the antecedent of the relative pronoun. It must be given three arguments: the individual x , its left context (i.e., e), and its right context. Now, the right context of n must be made of the relative clause, namely, r . In turn, r must be given three arguments: the individual x in its type-raised form (i.e., $\lambda \psi . \psi x$), its left context, and its right context. This explains the form of subterm $\lambda e . r (\lambda \psi . \psi x) e \phi$.

5. A toy Example

As a final example, we do not resist to consider Geach's famous donkey sentence (Geach 1962):

(49) Every farmer who owns a donkey beats it.

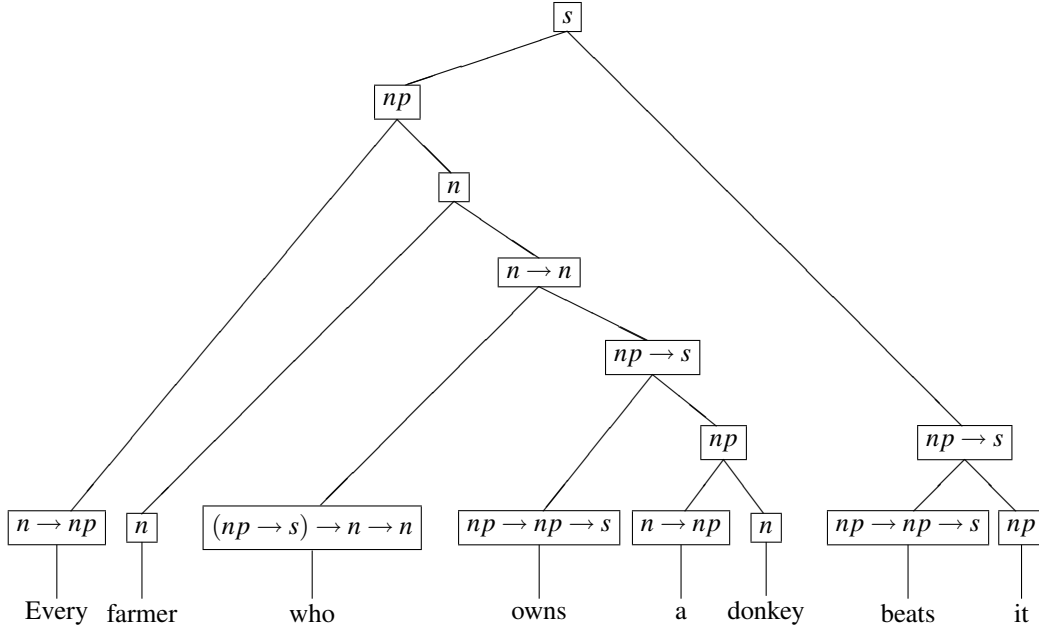
It is well known that such a sentence is quite a puzzle:

- There is a pronominal anaphora to be solved. This anaphora has the same complexity as an intersentential anaphora because the antecedent of the pronoun occurs in the relative clause.
- The existential quantification occurring in the relative clause must extend its scope over the complete sentence. Consequently, it must be semantically interpreted as a universal quantification.
- In order to respect DRT accessibility constraints, the reference markers corresponding to *every farmer* and to *a donkey* should not be available for further processing.

The following toy grammar, whose entries are consistent with Section 4, will allow (49) to be processed.

expression	category	interpretation
farmer	n	$\lambda x e \phi . \mathbf{farmer} x \wedge \phi e$
donkey	n	$\lambda x e \phi . \mathbf{donkey} x \wedge \phi e$
owns	$np \rightarrow np \rightarrow s$	$\lambda o s . s (\lambda x . o (\lambda y e \phi . \mathbf{own} x y \wedge \phi e))$
beats	$np \rightarrow np \rightarrow s$	$\lambda o s . s (\lambda x . o (\lambda y e \phi . \mathbf{beat} x y \wedge \phi e))$
who	$(np \rightarrow s) \rightarrow n \rightarrow n$	$\lambda r n x e \phi . n x e (\lambda e . r (\lambda \psi . \psi x) e \phi)$
a	$n \rightarrow np$	$\lambda n \psi e \phi . \exists x . n x e (\lambda e . \psi x (x :: e) \phi)$
every	$n \rightarrow np$	$\lambda n \psi e \phi . (\forall x . \neg (n x e (\lambda e . \neg (\psi x (x :: e) (\lambda e . \top)))))) \wedge \phi e$
it	np	$\lambda \psi e \phi . \psi (\mathbf{sel}_{it} e) e \phi$

Let us assume that some syntactic processing of (49) yields the following parse tree:



Then, the semantic representation of (49) may be computed by reducing the following λ -term:

$$(50) \quad \llbracket \text{beats} \rrbracket \llbracket \text{it} \rrbracket (\llbracket \text{every} \rrbracket (\llbracket \text{who} \rrbracket (\llbracket \text{owns} \rrbracket (\llbracket \text{a} \rrbracket \llbracket \text{donkey} \rrbracket))) \llbracket \text{farmer} \rrbracket))$$

Let us start by computing the meaning of the noun phrase *a donkey*.

$$\begin{aligned} & \llbracket \text{a} \rrbracket \llbracket \text{donkey} \rrbracket \\ &= (\lambda n \psi e \phi. \exists y. n y e (\lambda e. \psi y (y::e) \phi)) \llbracket \text{donkey} \rrbracket \\ &\rightarrow_{\beta} \lambda \psi e \phi. \exists y. \llbracket \text{donkey} \rrbracket y e (\lambda e. \psi y (y::e) \phi) \\ &= \lambda \psi e \phi. \exists y. (\lambda x e \phi. \mathbf{donkey} x \wedge \phi e) y e (\lambda e. \psi y (y::e) \phi) \\ &\rightarrow_{\beta} \lambda \psi e \phi. \exists y. (\lambda e \phi. \mathbf{donkey} y \wedge \phi e) e (\lambda e. \psi y (y::e) \phi) \\ &\rightarrow_{\beta} \lambda \psi e \phi. \exists y. (\lambda \phi. \mathbf{donkey} y \wedge \phi e) (\lambda e. \psi y (y::e) \phi) \\ &\rightarrow_{\beta} \lambda \psi e \phi. \exists y. \mathbf{donkey} y \wedge (\lambda e. \psi y (y::e) \phi) e \\ &\rightarrow_{\beta} \lambda \psi e \phi. \exists y. \mathbf{donkey} y \wedge \psi y (y::e) \phi \end{aligned}$$

At this point, it is already the case that the variable introduced by the existential quantifier has been made available in the context of the continuation. Remark, however, that the above λ -term is closed because the continuation falls into the scope of the existential quantifier.

We proceed with the verb phrase *owns a donkey*.

$$\begin{aligned} & \llbracket \text{owns} \rrbracket (\llbracket \text{a} \rrbracket \llbracket \text{donkey} \rrbracket) \\ &= (\lambda o s. s (\lambda x. o (\lambda y e \phi. \mathbf{own} x y \wedge \phi e))) (\llbracket \text{a} \rrbracket \llbracket \text{donkey} \rrbracket) \\ &\rightarrow_{\beta} \lambda s. s (\lambda x. \llbracket \text{a} \rrbracket \llbracket \text{donkey} \rrbracket) (\lambda y e \phi. \mathbf{own} x y \wedge \phi e) \\ &= \lambda s. s (\lambda x. (\lambda \psi e \phi. \exists y. \mathbf{donkey} y \wedge \psi y (y::e) \phi) (\lambda y e \phi. \mathbf{own} x y \wedge \phi e)) \\ &\rightarrow_{\beta} \lambda s. s (\lambda x e \phi. \exists y. \mathbf{donkey} y \wedge (\lambda y e \phi. \mathbf{own} x y \wedge \phi e) y (y::e) \phi) \\ &\rightarrow_{\beta} \lambda s. s (\lambda x e \phi. \exists y. \mathbf{donkey} y \wedge (\lambda e \phi. \mathbf{own} x y \wedge \phi e) (y::e) \phi) \\ &\rightarrow_{\beta} \lambda s. s (\lambda x e \phi. \exists y. \mathbf{donkey} y \wedge (\lambda \phi. \mathbf{own} x y \wedge \phi (y::e)) \phi) \\ &\rightarrow_{\beta} \lambda s. s (\lambda x e \phi. \exists y. \mathbf{donkey} y \wedge \mathbf{own} x y \wedge \phi (y::e)) \end{aligned}$$

The resulting λ -term expresses the meaning of the body of the relative clause *who owns a donkey*. It must be passed as an argument to the meaning of *who* in order to get the meaning of the relative clause.

$$\begin{aligned}
& \llbracket \text{who} \rrbracket (\llbracket \text{owns} \rrbracket (\llbracket \text{a} \rrbracket \llbracket \text{donkey} \rrbracket)) \\
&= (\lambda r n x e \phi. n x e (\lambda e. r (\lambda \psi. \psi x) e \phi)) (\llbracket \text{owns} \rrbracket (\llbracket \text{a} \rrbracket \llbracket \text{donkey} \rrbracket)) \\
&\rightarrow_{\beta} \lambda n x e \phi. n x e (\lambda e. \llbracket \text{owns} \rrbracket (\llbracket \text{a} \rrbracket \llbracket \text{donkey} \rrbracket)) (\lambda \psi. \psi x) e \phi \\
&= \lambda n x e \phi. n x e (\lambda e. (\lambda s. s (\lambda x e \phi. \exists y. \mathbf{donkey} y \wedge \mathbf{own} x y \wedge \phi (y::e))) (\lambda \psi. \psi x) e \phi) \\
&\rightarrow_{\beta} \lambda n x e \phi. n x e (\lambda e. (\lambda \psi. \psi x) (\lambda x e \phi. \exists y. \mathbf{donkey} y \wedge \mathbf{own} x y \wedge \phi (y::e)) e \phi) \\
&\rightarrow_{\beta} \lambda n x e \phi. n x e (\lambda e. (\lambda x e \phi. \exists y. \mathbf{donkey} y \wedge \mathbf{own} x y \wedge \phi (y::e)) x e \phi) \\
&\rightarrow_{\beta} \lambda n x e \phi. n x e (\lambda e. (\lambda e \phi. \exists y. \mathbf{donkey} y \wedge \mathbf{own} x y \wedge \phi (y::e)) e \phi) \\
&\rightarrow_{\beta} \lambda n x e \phi. n x e (\lambda e. (\lambda \phi. \exists y. \mathbf{donkey} y \wedge \mathbf{own} x y \wedge \phi (y::e)) \phi) \\
&\rightarrow_{\beta} \lambda n x e \phi. n x e (\lambda e. \exists y. \mathbf{donkey} y \wedge \mathbf{own} x y \wedge \phi (y::e))
\end{aligned}$$

We may now apply the above λ -term to $\llbracket \text{farmer} \rrbracket$ in order to get the meaning of *farmer who owns a donkey*.

$$\begin{aligned}
& \llbracket \text{who} \rrbracket (\llbracket \text{owns} \rrbracket (\llbracket \text{a} \rrbracket \llbracket \text{donkey} \rrbracket)) \llbracket \text{farmer} \rrbracket \\
&= (\lambda n x e \phi. n x e (\lambda e. \exists y. \mathbf{donkey} y \wedge \mathbf{own} x y \wedge \phi (y::e))) \llbracket \text{farmer} \rrbracket \\
&\rightarrow_{\beta} \lambda x e \phi. \llbracket \text{farmer} \rrbracket x e (\lambda e. \exists y. \mathbf{donkey} y \wedge \mathbf{own} x y \wedge \phi (y::e)) \\
&= \lambda x e \phi. (\lambda x e \phi. \mathbf{farmer} x \wedge \phi e) x e (\lambda e. \exists y. \mathbf{donkey} y \wedge \mathbf{own} x y \wedge \phi (y::e)) \\
&\rightarrow_{\beta} \lambda x e \phi. (\lambda e \phi. \mathbf{farmer} x \wedge \phi e) e (\lambda e. \exists y. \mathbf{donkey} y \wedge \mathbf{own} x y \wedge \phi (y::e)) \\
&\rightarrow_{\beta} \lambda x e \phi. (\lambda \phi. \mathbf{farmer} x \wedge \phi e) (\lambda e. \exists y. \mathbf{donkey} y \wedge \mathbf{own} x y \wedge \phi (y::e)) \\
&\rightarrow_{\beta} \lambda x e \phi. \mathbf{farmer} x \wedge (\lambda e. \exists y. \mathbf{donkey} y \wedge \mathbf{own} x y \wedge \phi (y::e)) e \\
&\rightarrow_{\beta} \lambda x e \phi. \mathbf{farmer} x \wedge \exists y. \mathbf{donkey} y \wedge \mathbf{own} x y \wedge \phi (y::e)
\end{aligned}$$

We are now in a position of computing the meaning of the noun phrase *every farmer who owns a donkey*.

$$\begin{aligned}
& \llbracket \text{every} \rrbracket (\llbracket \text{who} \rrbracket (\llbracket \text{owns} \rrbracket (\llbracket \text{a} \rrbracket \llbracket \text{donkey} \rrbracket)) \llbracket \text{farmer} \rrbracket) \\
&= (\lambda n \psi e \phi. (\forall x. \neg (n x e (\lambda e. \neg (\psi x (x::e) (\lambda e. \top)))))) \wedge \phi e \\
&\quad (\llbracket \text{who} \rrbracket (\llbracket \text{owns} \rrbracket (\llbracket \text{a} \rrbracket \llbracket \text{donkey} \rrbracket)) \llbracket \text{farmer} \rrbracket) \\
&\rightarrow_{\beta} \lambda \psi e \phi. (\forall x. \neg (\\
&\quad \llbracket \text{who} \rrbracket (\llbracket \text{owns} \rrbracket (\llbracket \text{a} \rrbracket \llbracket \text{donkey} \rrbracket)) \llbracket \text{farmer} \rrbracket x e (\lambda e. \neg (\psi x (x::e) (\lambda e. \top))) \\
&\quad)) \wedge \phi e \\
&= \lambda \psi e \phi. (\forall x. \neg (\\
&\quad (\lambda x e \phi. \mathbf{farmer} x \wedge \exists y. \mathbf{donkey} y \wedge \mathbf{own} x y \wedge \phi (y::e)) \\
&\quad x e (\lambda e. \neg (\psi x (x::e) (\lambda e. \top))) \\
&\quad)) \wedge \phi e
\end{aligned}$$

$$\begin{aligned}
&\rightarrow_{\beta} \lambda \psi e \phi. (\forall x. \neg (\\
&\quad (\lambda e \phi. \mathbf{farmer} x \wedge \exists y. \mathbf{donkey} y \wedge \mathbf{own} xy \wedge \phi (y::e)) \\
&\quad e (\lambda e. \neg (\psi x (x::e) (\lambda e. \top))) \\
&\quad)) \wedge \phi e \\
&\rightarrow_{\beta} \lambda \psi e \phi. (\forall x. \neg (\\
&\quad (\lambda \phi. \mathbf{farmer} x \wedge \exists y. \mathbf{donkey} y \wedge \mathbf{own} xy \wedge \phi (y::e)) \\
&\quad (\lambda e. \neg (\psi x (x::e) (\lambda e. \top))) \\
&\quad)) \wedge \phi e \\
&\rightarrow_{\beta} \lambda \psi e \phi. (\forall x. \neg (\mathbf{farmer} x \wedge \exists y. \mathbf{donkey} y \wedge \mathbf{own} xy \wedge \\
&\quad (\lambda e. \neg (\psi x (x::e) (\lambda e. \top))) (y::e) \\
&\quad)) \wedge \phi e \\
&\rightarrow_{\beta} \lambda \psi e \phi. (\forall x. \neg (\mathbf{farmer} x \wedge \exists y. \mathbf{donkey} y \wedge \mathbf{own} xy \wedge \neg (\psi x (x::y::e) (\lambda e. \top)))) \\
&\quad \wedge \phi e \\
&\equiv \lambda \psi e \phi. (\forall x. \mathbf{farmer} x \supset (\forall y. (\mathbf{donkey} y \wedge \mathbf{own} xy) \supset \psi x (x::y::e) (\lambda e. \top))) \wedge \phi e
\end{aligned}$$

There are two continuation parameters in the above λ -term, namely, ψ and ϕ . The first one, which comes from Montague type raising, is a local continuation that corresponds to the continuation of the sentence. The second one is the continuation of the discourse. Remark that the variables introduced by the quantifiers (x and y) are only available to the local continuation. This corresponds to Kamp's accessibility conditions.

We now compute the meaning of the verb phrase *beats it*.

$$\begin{aligned}
&\llbracket \text{beats} \rrbracket \llbracket \text{it} \rrbracket \\
&= (\lambda o s. s (\lambda x. o (\lambda y e \phi. \mathbf{beat} xy \wedge \phi e))) \llbracket \text{it} \rrbracket \\
&\rightarrow_{\beta} \lambda s. s (\lambda x. \llbracket \text{it} \rrbracket (\lambda y e \phi. \mathbf{beat} xy \wedge \phi e)) \\
&= \lambda s. s (\lambda x. (\lambda \psi e \phi. \psi (\mathbf{sel}_{\text{it}} e) e \phi) (\lambda y e \phi. \mathbf{beat} xy \wedge \phi e)) \\
&\rightarrow_{\beta} \lambda s. s (\lambda x e \phi. (\lambda y e \phi. \mathbf{beat} xy \wedge \phi e) (\mathbf{sel}_{\text{it}} e) e \phi) \\
&\rightarrow_{\beta} \lambda s. s (\lambda x e \phi. (\lambda e \phi. \mathbf{beat} x (\mathbf{sel}_{\text{it}} e) \wedge \phi e) e \phi) \\
&\rightarrow_{\beta} \lambda s. s (\lambda x e \phi. (\lambda \phi. \mathbf{beat} x (\mathbf{sel}_{\text{it}} e) \wedge \phi e) \phi) \\
&\rightarrow_{\beta} \lambda s. s (\lambda x e \phi. \mathbf{beat} x (\mathbf{sel}_{\text{it}} e) \wedge \phi e)
\end{aligned}$$

Finally, we may compute the meaning of the complete sentence.

$$\begin{aligned}
&\llbracket \text{beats} \rrbracket \llbracket \text{it} \rrbracket (\llbracket \text{every} \rrbracket (\llbracket \text{who} \rrbracket (\llbracket \text{owns} \rrbracket (\llbracket \text{a} \rrbracket \llbracket \text{donkey} \rrbracket)) \llbracket \text{farmer} \rrbracket)) \\
&= (\lambda s. s (\lambda x e \phi. \mathbf{beat} x (\mathbf{sel}_{\text{it}} e) \wedge \phi e) \\
&\quad (\llbracket \text{every} \rrbracket (\llbracket \text{who} \rrbracket (\llbracket \text{owns} \rrbracket (\llbracket \text{a} \rrbracket \llbracket \text{donkey} \rrbracket)) \llbracket \text{farmer} \rrbracket))) \\
&\rightarrow_{\beta} \llbracket \text{every} \rrbracket (\llbracket \text{who} \rrbracket (\llbracket \text{owns} \rrbracket (\llbracket \text{a} \rrbracket \llbracket \text{donkey} \rrbracket)) \llbracket \text{farmer} \rrbracket) (\lambda x e \phi. \mathbf{beat} x (\mathbf{sel}_{\text{it}} e) \wedge \phi e) \\
&= (\lambda \psi e \phi. (\forall x. \mathbf{farmer} x \supset (\forall y. (\mathbf{donkey} y \wedge \mathbf{own} xy) \supset \psi x (x::y::e) (\lambda e. \top))) \wedge \phi e) \\
&\quad (\lambda x e \phi. \mathbf{beat} x (\mathbf{sel}_{\text{it}} e) \wedge \phi e) \\
&\rightarrow_{\beta} \lambda e \phi. (\forall x. \mathbf{farmer} x \supset (\forall y. (\mathbf{donkey} y \wedge \mathbf{own} xy) \supset \\
&\quad (\lambda x e \phi. \mathbf{beat} x (\mathbf{sel}_{\text{it}} e) \wedge \phi e) x (x::y::e) (\lambda e. \top) \\
&\quad)) \wedge \phi e \\
&\rightarrow_{\beta} \lambda e \phi. (\forall x. \mathbf{farmer} x \supset (\forall y. (\mathbf{donkey} y \wedge \mathbf{own} xy) \supset \\
&\quad (\lambda e \phi. \mathbf{beat} x (\mathbf{sel}_{\text{it}} e) \wedge \phi e) (x::y::e) (\lambda e. \top) \\
&\quad)) \wedge \phi e \\
&\rightarrow_{\beta} \lambda e \phi. (\forall x. \mathbf{farmer} x \supset (\forall y. (\mathbf{donkey} y \wedge \mathbf{own} xy) \supset \\
&\quad (\lambda \phi. \mathbf{beat} x (\mathbf{sel}_{\text{it}} (x::y::e)) \wedge \phi (x::y::e)) (\lambda e. \top) \\
&\quad)) \wedge \phi e \\
&\rightarrow_{\beta} \lambda e \phi. (\forall x. \mathbf{farmer} x \supset (\forall y. (\mathbf{donkey} y \wedge \mathbf{own} xy) \supset \mathbf{beat} x (\mathbf{sel}_{\text{it}} (x::y::e)) \wedge \\
&\quad (\lambda e. \top) (x::y::e) \\
&\quad)) \wedge \phi e
\end{aligned}$$

$$\begin{aligned}
&\rightarrow_{\beta} \lambda e\phi. (\forall x. \mathbf{farmer} x \supset (\forall y. (\mathbf{donkey} y \wedge \mathbf{own} xy) \supset \mathbf{beat} x(\mathbf{sel}_{it}(x::y::e)) \wedge \top)) \\
&\quad \wedge \phi e \\
&\equiv \lambda e\phi. (\forall x. \mathbf{farmer} x \supset (\forall y. (\mathbf{donkey} y \wedge \mathbf{own} xy) \supset \mathbf{beat} x(\mathbf{sel}_{it}(x::y::e)))) \wedge \phi e
\end{aligned}$$

Assuming that the choice operator selects correctly y , we obtain the following representation:

$$(51) \quad \lambda e\phi. (\forall x. \mathbf{farmer} x \supset (\forall y. (\mathbf{donkey} y \wedge \mathbf{own} xy) \supset \mathbf{beat} xy)) \wedge \phi e$$

6. Conclusions

Our approach to handling contexts in a Montagovian framework solves several compositionality problems one encounters when trying to accommodate Discourse Representation Theory and Montague semantics.

In addition, it is rather general. As we suggested, it does not depend on the target logic. It does not depend either of the very structure given to the data of type γ . The $_::_$ operator could be used to record more relevant information such as morphosyntactic features (gender, number, case,...), actor focus, and discourse focus. Then, the choice operators could be turned in more realistic anaphora resolution algorithms.

In this paper, we restricted our analysis to the case of declarative discourses that may be interpreted as classical logical propositions, avoiding the several problems related to tense, aspect, intensionality, modality, ... We did not consider either definite descriptions. These different topics will be the subjects of future works.

References

- Barker, C.: 2002, ‘Continuations and the nature of quantification’, *Natural Language Semantics* **10**, 211–242.
- Barker, C.: 2004, ‘Continuations in Natural Language’, in H. Thielecke (ed.), *Proceedings of the Fourth ACM SIGPLAN Continuations Workshop*, 1–11.
- Carpenter, B.: 1996, *Type-Logical Semantics*. MIT Press, Cambridge, Massachusetts and London England.
- Church, A.: 1940, ‘A Formulation of the Simple Theory of Types’, *Journal of Symbolic Logic* **5**, 56–68.
- van Eijck, J. and H. Kamp: 1997, ‘Representing Discourse in Context’, in J. van Benthem and A. ter Meulen (eds.), *Handbook of Logic and Language*. Elsevier.
- Geach: 1962, *Reference and Generality*. Cornell University Press, Ithaca, NY.
- Groenendijk, J. and M. Stokhof: 1990, ‘Dynamic Montague Grammar’, Technical Report LP-90-02, Faculty of Mathematics and Computer Science, Amsterdam.
- de Groote, Ph.: 2001a, ‘Towards Abstract Categorical Grammars’, in *Association*

- for Computational Linguistics, 39th Annual Meeting and 10th Conference of the European Chapter, Proceedings of the Conference*, 148–155.
- de Groot, Ph.: 2001b, ‘Type raising, continuations, and classical logic’, in R. van Rooy and M. Stokhof (eds.), *Proceedings of the Thirteenth Amsterdam Colloquium*, 97–101.
- Jacobson, P.: 1999, ‘Towards a variable-free semantics’, *Linguistics and Philosophy* **22**, 117–184.
- Kamp, H. and U. Reyle: 1993, *From Discourse to Logic*. Kluwer Academic Publishers, Dordrecht.
- Montague, R.: 1973, ‘The proper treatment of quantification in ordinary English’, in J. Hintikka, J. Moravcsik, and P. Suppes (eds.), *Approaches to natural language: proceedings of the 1970 Stanford workshop on Grammar and Semantics*. Reidel, Dordrecht. Reprinted: (Montague 1974: pages 247–270).
- Montague, R.: 1974, *Formal Philosophy: selected papers of Richard Montague, edited and with an introduction by Richmond Thomason*. Yale University Press.
- Moortgat, M.: 1997, ‘Categorial Type Logics’, in J. van Benthem and A. ter Meulen (eds.), *Handbook of Logic and Language*. Elsevier.
- Morrill, G.: 1994, *Type Logical Grammar: Categorial Logic of Signs*. Kluwer Academic Publishers, Dordrecht.
- Muskens, R.: 1995, ‘Combining Montague semantics and discourse representation’, *Linguistics and Philosophy* **19**, 143–186.
- Muskens, R.: 2001, ‘Lambda Grammars and the Syntax-Semantics Interface’, in R. van Rooy and M. Stokhof (eds.), *Proceedings of the Thirteenth Amsterdam Colloquium*, 150–155.
- Pollard, C.: 2004, ‘High-Order Categorial Grammars’, in *Categorial Grammars, Montpellier, France*, 340–361.
- Shan, C.c.: 2004, ‘Delimited continuations in natural language: quantification and polarity sensitivity’, in H. Thielecke (ed.), *Proceedings of the Fourth ACM SIGPLAN Continuations Workshop*, 55–64.
- Strachey, C. and C. Wadsworth: 1974, ‘Continuations a mathematical semantics for handling full jumps’, Technical Report PRG-11, Oxford University, Computing Laboratory.