A Mechanism to Restrict the Scope of Clause-Bounded Quantifiers in 'Continuation' Semantics

Anca Dinu
Faculty of Foreign Languages and
Literatures, University of Bucharest
anca_d_dinu@yahoo.com

Abstract

This paper presents a formal mechanism to properly constrain the scope of negation and of certain quantificational determiners to their minimal clause in continuation semantics framework introduced in Barker and Shan (2008) and which was subsequently extended from sentential level to discourse level in Dinu (2011). In these works, type shifting is employed to account for side effects such as pronominal anaphora binding or quantifier scope. However, allowing arbitrary type shifting will result in overgenerating interpretations impossible in natural language. To filter out some of these impossible interpretations, once the negation or the quantifiers reach their maximal scope limits (that is their minimal clause), one should force their scope closing by applying a standard type shifter *Lower*. But the actual mechanism that forces the scope closing was left underspecified in previous work on continuation semantics. We propose here such a mechanism, designed to ensure that no lexical entries having the scope bounded to their minimal clause (such as *not, no, every, each, any*, etc) will ever take scope outside.

1 Introduction

The starting point of this paper is the continuation semantics introduced in Barker and Shan (2008) and extended from sentential level to discourse level in Dinu (2011). In this framework, type shifting is used to account for side effects such as pronominal anaphora binding or quantifier scope. However, allowing arbitrary type shifting will result in overgenerating interpretations impossible in natural language. To filter out these impossible interpretations, we first need to understand the scope behavior of each scope-taking lexical entry: its maximal scope limits and the scope precedence preferences w.r.t. other lexical entries. Second, we should force the scope closing of the quantifiers by applying a standard type shifter *Lower* (which is equivalent to identity function application), once their scope limits were reached. But the actual mechanism that ensures the scope closing was left underspecified in previous work on continuation semantics.

In what follows, we propose such a mechanism, designed to ensure that no lexical entry having the scope bounded to its minimal clause (such as *not, no, every, each, any*, etc) will ever take scope outside, thus getting right discourse truth conditions.

The programming language concept of continuations was successfully used by Barker and Shan in a series of articles (Barker 2002, Barker 2004, Shan 2005, Shan and Barker 2006, Barker and Shan 2008) to analyze intra-sentential linguistic phenomena (focus fronting, donkey anaphora, presupposition, crossover, superiority, etc). Moreover, (de Groote, 2006) proposed an elegant discourse semantics based on continuations. Continuations are a standard tool in computer science, used to control side effects of computation. They are a notoriously hard to understand notion. Actually, understanding what a continuation is per se is not so hard. What is more difficult is to understand how a grammar based on continuations (a 'continuized' grammar) works. The basic idea of continuizing a grammar is to provide subexpressions with direct access to their own continuations (future context), so subexpressions are modified to take a continuation as an argument. A continuized grammar is said to be written in *continuation*

Proceedings of Recent Advances in Natural Language Processing, pages 495–502,
Hissar, Bulgaria, 12-14 September 2011.

passing style. Continuation passing style is in fact a restricted (typed) form of λ-calculus.

Historically, the first continuation operators were undelimited (e.g., call/cc or J). An undelimited continuation of an expression represents "the entire (default) future for the computation" of that expression. Felleisen (1988) introduced delimited continuations (sometimes called 'composable' continuations) such as control ('C') and prompt ('%'). Delimited continuations represent the future of the computation of the expression up to a certain boundary. Interestingly, the natural-language phenomena discussed here make use only of delimited continuations.

For instance, if we take the local context to be restricted to the sentence, when computing the meaning of the sentence *John saw Mary.*, the default future of the value denoted by the subject is that it is destined to have the property of seeing Mary predicated of it. In symbols, the continuation of the subject denotation j is the function $\lambda x.\, saw\ \boldsymbol{m}\ x$. Similarly, the default future of the object denotation $\boldsymbol{m}$ is the property of being seen by John, the function $\lambda y.\, \boldsymbol{saw}\ y\ j$; the continuation of the transitive verb denotation *saw* is the function $\lambda R.\, R\ m\ j$; and the continuation of the verb phrase *saw Mary* is the function $\lambda P.\, P\ j$. This simple example illustrates two important aspects of continuations:

(1) every meaningful subexpression has a continuation;

(2) the continuation of an expression is always relative to some larger expression containing it.

Thus, when *John* occurs in the sentence *John left yesterday.*, its continuation is the property $\lambda x.\, \boldsymbol{yesterday\ left}\ x$; when it occurs in *Mary thought John left.*, its continuation is the property $\lambda x.\, \boldsymbol{thought}\ (\boldsymbol{left}\ x)\ m$ and when it occurs in the sentence *Mary or John left.*, its continuation is $\lambda x.\, (\boldsymbol{left}\ \boldsymbol{m}) \vee (\boldsymbol{left}\ x)$ and so on.

It is worth mentioning that some results of traditional semantic theories are particular cases of results in continuation semantics:

- The generalized quantifier type from Montague grammar (Montague, 1970) <<<e,t>,t>,t> is exactly the type of quantificational determiners in continuation-based semantics;
- The <<t,t>,t> type of sentences in dynamic semantics is exactly the type of sentences in continuation-based semantics. In fact, dynamic interpretation constitutes a partial continuization in which only the category S has been continuized.

This is by no means a coincidence, MG only continuizes the noun phrase meanings and dynamic semantics only continuizes the sentence meanings, rather than continuizing uniformly throughout the grammar as it is done in continuation semantics.

2 Preliminaries

We use Barker and Shan's (2008) tower notation for a given expression, which consists of three levels: the top level specifies the syntactic category of the expression couched in categorial grammar, the middle level is the expression itself and the bottom level is the semantic value:

$$\begin{array}{c} syntactic\ category \\ expression \\ semantic\ value \end{array}$$

The syntactic categories are written $\frac{C|B}{A}$, where *A*, *B* and *C* can be any categories. We read this counter clockwise as "the expression functions as a category *A* in local context, takes scope at an expression of category *B* to form an expression of category *C*."

The semantic value in linear notation $\lambda k.\, f[k(x)]$ is equivalently written vertically as $\frac{f[\,]}{x}$ omitting the future context (continuation) *k*. Here, *x* can be any expression, and *f[ ]* can be any expression with a gap *[ ]*. Free variables in *x* can be bound by binders in *f [ ]*. This vertical (layered) notational convention is meant to make the combination process of two expressions easier (more visual) then in linear notation. Here there are the two possible modes of combination (Barker and Shan 2008):

$$\begin{pmatrix} \frac{C|D}{A/B} & \frac{D|E}{B} \\ left-exp & right-exp \\ \frac{g[\,]}{f} & \frac{h[\,]}{x} \end{pmatrix} = \begin{array}{c} \frac{C|E}{A} \\ left-exp\ right-exp \\ \frac{g[h[\,]]}{f(x)} \end{array}$$

$$\begin{pmatrix} \frac{C|D}{B} & \frac{D|E}{B\backslash A} \\ left-exp & right-exp \\ \frac{g[\,]}{x} & \frac{h[\,]}{f} \end{pmatrix} = \begin{array}{c} \frac{C|E}{A} \\ left-exp\ right-exp \\ \frac{g[h[\,]]}{f(x)} \end{array}$$

Below the horizontal lines, combination proceeds simply as in combinatory categorial grammar: in the syntax, *B* combines with *A/B* or *B\A* to form *A*; in the semantics, *x* combines with *f* to form *f(x)*. Above the lines is where the

combination machinery for continuations kicks in. The syntax combines the two pairs of categories by a kind of cancellation: the *D* on the left cancels with the *D* on the right. The semantics combines the two expressions with gaps by a kind of composition: we plug *h[ ]* to the right into the gap of *g[ ]* to the left, to form *g[h[ ]]*. The expression with a gap on the left, g*[ ]*, always surrounds the expression with a gap on the right, *h[ ]*, no matter which side supplies the function and which side supplies the argument below the lines. This fact expresses the generalization that the default order of semantic evaluation is left-to-right.

When there is no quantification or anaphora involved, a simple sentence like *John came*. is derived as follows:

$$\begin{pmatrix} DP & DP\backslash S \\ John & came \\ j & came \end{pmatrix} = \begin{matrix} S \\ John\ came \\ came\ j \end{matrix}$$

In the syntactic layer, as usual in categorical grammar, the category under slash (here DP) cancels with the category of the argument expression; the semantics is function application.

Quantificational expressions have extra layers on top of their syntactic category and on top of their semantic value, making essential use of the powerful mechanism of continuations in ways proper names or definite descriptions do not. For instance, below is the derivation of *A man came.:*

$$\begin{pmatrix} \frac{S|S}{DP}/N & N & \frac{S|S}{DP\backslash S} \\ a & man & came \\ \lambda P.\frac{\exists x.P(x)\wedge[\,]}{x} & man & \frac{[\,]}{\boldsymbol{came}} \end{pmatrix} = \begin{matrix} \frac{S|S}{S} \\ a\ man\ came \\ \frac{\exists x.man(x)\wedge[\,]}{\boldsymbol{came}\ x} \end{matrix}$$

Comparing the analysis above of *John came* with that of *A man came* reveals that came has been given two distinct values. The first, simpler value is the basic lexical entry, the more complex value being derived through the standard type-shifter Lift, proposed by Partee and Rooth (1983), Jacobson (1999), Steedman (2000), and many others:

$$\begin{matrix} A \\ expression \\ x \end{matrix} \overset{Lift}{\Longrightarrow} \begin{matrix} \frac{B|B}{A} \\ expression \\ \frac{[\,]}{x} \end{matrix}$$

Syntactically, Lift adds a layer with arbitrary (but matching!) syntactic categories. Semantically, it adds a layer with empty brackets. In linear notation we have: $x \overset{Lift}{\Longrightarrow} \lambda k.k(x)$.

To derive the syntactic category and a semantic value with no horizontal line, Barker and Shan (2008) introduce the type-shifter Lower. In general, for any category *A*, any value *x*, and any semantic expression *f[ ]* with a gap, the following type-shifter is available:

$$\begin{matrix} \frac{A|S}{S} \\ expression \\ \frac{f[\,]}{x} \end{matrix} \overset{Lower}{\Longrightarrow} \begin{matrix} A \\ expression \\ f[x] \end{matrix}$$

Syntactically, Lower cancels an *S* above the line to the right with an *S* below the line. Semantically, Lower collapses a two-level meaning into a single level by plugging the value *x* below the line into the gap *[ ]* in the expression *f[ ]* above the line. Lower is equivalent to identity function application.

The third and the last type shifter we need is one that accounts for binding. We adopt the idea (in line with Barker and Shan (2008)) that the mechanism of binding is the same as the mechanism of scope taking. Binding is a term used both in logics and in linguistics with analog (but not identical) meaning. In logics, a variable is said to be bound by an operator (as the universal or existential operators) if the variable is inside the scope of the operator. If a variable is not in the scope of any operator, than the variable is said to be free. In linguistics, a binder may be a constituent such as a proper name (*John*), an indefinite common noun (*a book)*, an event or a situation, etc. Anaphoric expressions such as pronouns (*he, she, it, him, himself*, etc), definite common nouns (*the book, the book that John read*), demonstrative pronouns (like *this, that*), etc. act as variables that take the value of (are bind by) a previous binder.

In order to give a proper account of anaphoric relations in discourse, we need to formulate an explicit semantics for both the binder and the anaphoric expressions to be bound. Any determiner phrase (DP) may act as a binder, as the Bind rule from Barker and Shan (2008) explicitly states:

$$\begin{matrix} \frac{A|B}{DP} \\ expression \\ \frac{f[\,]}{x} \end{matrix} \overset{Bind}{\Longrightarrow} \begin{matrix} \frac{A|DP \rhd B}{DP} \\ expression \\ \frac{f([\,]x)}{x} \end{matrix}$$

At the syntactic level, the Bind rule says that an expression that functions in local context as a DP may offer to bind an anaphoric expression to

the right ((Barker and Shan 2008) encode that by the sign ⊳). At the semantic level, the expression transmits (copies) the value of the variable *x*. In linear notation, the semantic part of the Bind rule looks like this: $\lambda k.\, f[k(x)] \overset{Bind}{\Longrightarrow} \lambda k.\, f([k(x)]x)$

As for the elements that may be bound, (Barker and Shan 2008) give the following lexical entry for the singular pronoun *he*:

$$\frac{\begin{array}{c}DP \rhd S|S\\ \hline DP\\ he\\ \lambda y.[\,]\end{array}}{y}$$

To account for multiple anaphoric expressions (and their binders) or for inverse scope of multiple quantifiers, each binder can occupy a different scope-taking level in the compositional tower. With access to multiple levels, it is easy to handle multiple binders. Analyzing pronouns as two-level rules is the same thing as claiming that pronouns take scope (see Dowty (2007), who also advocates treating pronouns as scope-takers). Then, a pronoun or another anaphoric expression chooses its binder by choosing where to take scope. So, distinct scope-taking levels correspond to different binders, layers playing the role of indices: a binder and the pronoun it binds must take effect at the same layer in the compositional tower. A superior level takes scope at inferior levels and left expressions take scope at right expressions, to account for left-to-right natural language order of processing.

Dinu (2011) extends the formalism from sentence level to discourse level, giving the sentence connectors such as the dot the following semantics:

$$\begin{array}{c}S\backslash(S/S)\\ .\\ \lambda p\lambda q.\, p\wedge q\end{array}$$

that is, the dot is a function that takes two sentence denotations and returns a sentence denotation (the conjunction of original sentence denotation).

For two affirmative sentences with no anaphoric relations and no quantifiers, such as *John came. Mary left.*, the derivation trivially proceeds as follows:

$$\begin{array}{ccccc} S & S\backslash(S/S) & S & = & S\\ John\ came & . & Mary\ left & & John\ came.Mary\ left\\ \boldsymbol{came\ j} & \lambda p\lambda q.\, p\wedge q & \boldsymbol{left\ m} & & \boldsymbol{came\ j}\wedge\boldsymbol{left\ m}\end{array}$$

As one sees above, there is no need in this simple case to resort to type shifting at all. Nevertheless, type shifting and the powerful mechanism of continuations are employed when dealing with linguistic side effects such quantifier scope or binding. For instance, to derive the denotation of *A man came. He whistled*., type lifting, type lowering and Bound rule become necessary:

$$\lambda P.\frac{\begin{array}{c}\frac{S|S}{DP}/N\\ a\end{array}\quad \begin{array}{c}N\\ man\\ \boldsymbol{man}\end{array}}{\begin{array}{c}\exists x.P(x)\wedge[\,]\\ \hline x\end{array}} = \frac{\begin{array}{c}\frac{S|S}{DP}\\ a\ man\end{array}}{\begin{array}{c}\exists x.\boldsymbol{man}(x)\wedge[\,]\\ \hline x\end{array}}$$

$$\overset{Bind}{\Longrightarrow} \frac{\begin{array}{c}\frac{S|DP\rhd S}{DP}\\ a\ man\end{array}}{\begin{array}{c}\exists x.\boldsymbol{man}(x)\wedge([\,]x)\\ \hline x\end{array}}$$

$$\frac{\begin{array}{c}\frac{S|DP\rhd S}{DP}\\ a\ man\end{array}}{\begin{array}{c}\exists x.\boldsymbol{man}(x)\wedge([\,]x)\\ \hline x\end{array}}\quad \frac{\begin{array}{c}\frac{DP\rhd S|DP\rhd S}{DP\backslash S}\\ came\end{array}}{\begin{array}{c}[\,]\\ \hline \boldsymbol{came}\end{array}} =$$

$$\frac{\begin{array}{c}\frac{S|DP\rhd S}{S}\\ a\ man\ came\end{array}}{\begin{array}{c}\exists x.\boldsymbol{man}(x)\wedge([\,]x)\\ \hline \boldsymbol{came}\ x\end{array}}$$

$$\frac{\begin{array}{c}\frac{DP\rhd S|S}{DP}\\ he\\ \lambda y.[\,]\end{array}}{y}\quad \frac{\begin{array}{c}\frac{S|S}{DP\backslash S}\\ whistled\end{array}}{\begin{array}{c}[\,]\\ \hline \boldsymbol{whistled}\end{array}} = \frac{\begin{array}{c}\frac{DP\rhd S|S}{S}\\ he\ whistled\\ \lambda y.[\,]\end{array}}{\boldsymbol{whistled}\ y}$$

$$\frac{\begin{array}{c}\frac{S|DP\rhd S}{S}\\ a\ man\ came\end{array}}{\begin{array}{c}\exists x.\boldsymbol{man}(x)\wedge([\,]x)\\ \hline \boldsymbol{came}\ x\end{array}}\quad \frac{\begin{array}{c}\frac{DP\rhd S|DP\rhd S}{S\backslash(S/S)}\\ .\end{array}}{\begin{array}{c}[\,]\\ \hline \lambda p\lambda q.\, p\wedge q\end{array}}\quad \frac{\begin{array}{c}\frac{DP\rhd S|S}{S}\\ he\ whistled\\ \lambda y.[\,]\end{array}}{\boldsymbol{whistled}\ y}$$

$$\begin{array}{c}\frac{S|S}{S}\\ = \ a\ man\ came.he\ whistled \overset{Lower}{\Longrightarrow}\\ \exists x.\boldsymbol{man}(x)\wedge(\lambda y.[\,]x)\\ \hline \boldsymbol{came}\ x\wedge\boldsymbol{whistled}\ y\end{array}$$

$$\begin{array}{c}S\\ a\ man\ came.he\ whistled\\ \exists x.\boldsymbol{man}(x)\wedge(\lambda y.[\boldsymbol{came}\ x\wedge\boldsymbol{whistled}\ y\,]x)\end{array}$$

$$\begin{array}{c}S\\ = \quad a\ man\ came.he\ whistled\\ \exists x.\boldsymbol{man}(x)\wedge(\boldsymbol{came}\ x\wedge\boldsymbol{whistled}\ x)\end{array}$$

Note that the denotations of *came* and *whistled* were also lifted so as to match the ones of *a* and *he*, both being scope-takers. The last equality sign is due to routine lambda conversion.

3 Restricting the Scope of Clause-Bounded Lexical Entries

A first proposal for the lexical entry for the negation could look like this:

$$\frac{S|S}{(DP\backslash S)/(DP\backslash S)}$$
$$not$$
$$\frac{\neg[\,]}{[\,]}$$

meaning that negation functions in local context as a verb modifier and takes scope at a sentence to give a sentence.

Using this denotation for *not*, the piece of discourse *John does not own a car*. is interpreted as (ignoring the auxiliary *does* for simplicity):

$$\frac{\frac{S|S}{DP}}{\frac{John}{\frac{[\,]}{\boldsymbol{j}}}} \quad \frac{S|S}{\frac{(DP\backslash S)/(DP\backslash S)}{\frac{not}{\frac{\neg[\,]}{[\,]}}}} \quad \frac{S|S}{\frac{(DP\backslash S)/DP}{\frac{own}{\frac{[\,]}{\boldsymbol{own}}}}} \quad \frac{\frac{S|S}{DP}}{\frac{a\ car}{\frac{\exists x.\boldsymbol{car}(x)\wedge[\,]}{x}}}$$

$$= \frac{\frac{S|S}{S}}{\frac{John\ not\ own\ a\ car}{\frac{\neg(\exists x.\boldsymbol{car}(x)\wedge[\,])}{\boldsymbol{own}\ x\ \boldsymbol{j}}}} \overset{Lower}{\Longrightarrow} \frac{S}{John\ not\ own\ a\ car\ \neg(\exists x.\boldsymbol{car}(x)\wedge \boldsymbol{own}\ x\ \boldsymbol{j})}$$

meaning that there is no car that John owns, a fair approximation of the intended meaning.

It is generally accepted that negation cannot take scope outside its minimal clause. But, if we do not restrict the possible scope of negation, continuing the discourse with the sentence **It is red*., could result in the following derivation:

$$\frac{\frac{S|DP \rhd S}{S}}{\frac{John\ not\ own\ a\ car}{\frac{\neg(\exists x.\boldsymbol{car}(x)\wedge[\,]x)}{\boldsymbol{own}\ x\ \boldsymbol{j}}}} \quad \frac{\frac{DP \rhd S|DP \rhd S}{S\backslash(S/S)}}{\frac{\cdot}{\frac{[\,]}{\lambda p\lambda q.p\wedge q}}} \quad \frac{\frac{DP \rhd S|S}{S}}{\frac{It\ is\ red}{\frac{\lambda y.[\,]}{\boldsymbol{is\ red}\ y}}}$$

$$= \frac{\frac{S|S}{S}}{\frac{John\ not\ own\ a\ car.\ It\ is\ red}{\frac{\neg(\exists x.\boldsymbol{car}(x)\wedge\lambda y.[\,]x)}{\boldsymbol{own}\ x\ \boldsymbol{j}\ \wedge \boldsymbol{is\ red}\ y}}} \overset{Lower}{\Longrightarrow}$$

$$\frac{S}{John\ not\ own\ a\ car.\ It\ is\ red\ \neg(\exists x.\boldsymbol{car}(x)\wedge\lambda y.[\ \boldsymbol{own}\ x\ \boldsymbol{j}\wedge \boldsymbol{s\ red}\ y]x)}$$

$$= \frac{S}{John\ not\ own\ a\ car.\ It\ is\ red\ \neg(\exists x.\boldsymbol{car}(x)\wedge[\ \boldsymbol{own}\ x\ \boldsymbol{j}\wedge \boldsymbol{is\ red}\ x])}$$

which would incorrectly assert that there is no car which is owned by John and which is red. Moreover, *it* would wrongly refer back to *a car*. In fact, if we do not restrict the possible scope of negation, any following sentence may be wrongly interpreted inside the scope of negation.

In order to block such interpretations, we could adopt a similar strategy with the one proposed in Barker and Shan (2008): to force the scope closing of *not* immediately after the interpretation of its minimal clause, by applying Lower. This also closes the scope of any other *DP* inside the scope of negation, so it becomes impossible for it to bind subsequent anaphoric expressions. But this strategy leaves the actual mechanis that insures the scope closing unspecified. As Barker and Shan put it, when referring to the scope closing of *every*, "Like most leading accounts of donkey anaphora, we provide no formal mechanism here that bounds the scope-taking of universals".

In what follows, we propose such a mechanism within the continuation semantics framework. The mechanism is designed to ensure that no lexical entries having the scope bounded to their minimal clause (such as *not, no, every, each, any*, etc) will ever take scope outside.

We introduce a new category for clauses: *C*, of the same semantic type as the category *S*, namely *t*. *C* is the minimal discourse unit, whereas *S* is contains at least one such unit.

We constrain by definition the lexical entries with clause-bounded scope to take scope only at clauses. For instance, here there are the lexical entries for *not, no* and *every*:

$$\frac{C|C}{(DP\backslash C)/(DP\backslash C)}$$
$$not$$
$$\frac{\neg[\,]}{[\,]}$$

$$\frac{\frac{C|C}{DP}/N}{\lambda P.\frac{no}{\frac{\neg\exists x.(P(x)\ \wedge\ [\,])}{x}}}$$

$$\frac{\frac{C|C}{\frac{C|C}{DP}/N}}{\lambda P.\frac{every}{\frac{\forall x.[\,]}{\frac{P(x)\longrightarrow[\,]}{x}}}}$$

After the full interpretation of the minimal clause which they appear in, the category *C* has to be converted to category *S*. Specifically, one can use the following silent lexical entry:

$$\frac{S/C}{\Phi}$$
$$\lambda p.p([])$$

This step ensures that clauses (of category *C*) can be further processed as pieces of discourse (of category *S*), because all discourse connectors (such as the dot or *if*) are allowed to take only expressions of category *S* as arguments.

We modify the Lower rule such that category *C* may also be lowered similarly to category *S*:

$$\frac{\frac{A|C}{C}}{\frac{expression}{\frac{f[\,]}{x}}} \overset{Lower}{\Longrightarrow} \frac{A}{\frac{expression}{f[x]}}$$

With this clause-restricting mechanism, the derivation of *John does not own a car*. becomes:

$$\frac{\frac{C|C}{\frac{DP}{\frac{John}{[\,]}}}}{\frac{}{\boldsymbol{j}}} \quad \frac{C|C}{\frac{(DP\backslash C)/(DP\backslash C)}{\frac{not}{\frac{\neg[\,]}{[\,]}}}} \quad \frac{C|C}{\frac{(DP\backslash C)/DP}{\frac{own}{\frac{[\,]}{own}}}} \quad \frac{\frac{C|C}{\frac{DP}{a\ car}}}{\frac{\exists x.\boldsymbol{car}(x)\wedge[\,]}{x}}$$

$$= \frac{\frac{C|C}{C}}{\frac{John\ not\ own\ a\ car}{\frac{\neg(\exists x.\boldsymbol{car}(x)\wedge[\,])}{\boldsymbol{own}\ x\ \boldsymbol{j}}}} \overset{Lower}{\Longrightarrow} \frac{C}{\frac{John\ not\ own\ a\ car}{\neg(\exists x.\boldsymbol{car}(x)\wedge\ \boldsymbol{own}\ x\ \boldsymbol{j})}}$$

Now that the scope of negation is closed, it is obviously impossible for it to stretch over the following discourse. We only have to change the category *C* into *S* in order to connect it to the discourse:

$$\frac{S/C}{\frac{\Phi}{\lambda p.p([])}} \quad \frac{C}{\frac{John\ not\ own\ a\ car}{\neg(\exists x.\boldsymbol{car}(x)\wedge\ \boldsymbol{own}\ x\ \boldsymbol{j})}}$$

$$= \frac{S}{\frac{John\ not\ own\ a\ car}{\lambda p.p([\neg(\exists x.\boldsymbol{car}(x)\wedge\ \boldsymbol{own}\ x\ \boldsymbol{j})])}}$$

$$= \frac{S}{\frac{John\ not\ own\ a\ car}{\neg(\exists x.\boldsymbol{car}(x)\wedge\ \boldsymbol{own}\ x\ \boldsymbol{j})]}}$$

What about the binding capabilities of the expressions in a clause whose scope has been closed? The subject, for instance, should be able to bind subsequent anaphora. It can do so by lifting over the negation and being available to bind from that position:

$$\frac{\frac{S|DP\rhd S}{\frac{C|C}{\frac{DP}{\frac{John}{\frac{[\,]\boldsymbol{j}}{[\,]}}}}}}{\boldsymbol{j}} \quad \frac{\frac{DP\rhd S|DP\rhd S}{\frac{C|C}{(DP\backslash C)/(DP\backslash C)}}}{\frac{not}{\frac{[\,]}{\frac{\neg[\,]}{[\,]}}}} \quad \frac{\frac{DP\rhd S|DP\rhd S}{\frac{C|C}{(DP\backslash C)/DP}}}{\frac{own}{\frac{[\,]}{\frac{[\,]}{\boldsymbol{own}}}}} \quad \frac{\frac{DP\rhd S|DP\rhd S}{\frac{C|C}{\frac{DP}{a\ car}}}}{\frac{[\,]}{\frac{\exists x.\boldsymbol{car}(x)\wedge[\,]}{x}}}$$

$$= \frac{\frac{S|DP\rhd S}{\frac{C|C}{C}}}{\frac{John\ not\ own\ a\ car}{\frac{[\,]\boldsymbol{j}}{\frac{\neg\exists x.\boldsymbol{car}(x)\wedge[\,]}{\boldsymbol{own}\ x}}}} \overset{Lower}{\Longrightarrow} \frac{\frac{S|DP\rhd S}{C}}{\frac{John\ not\ own\ a\ car}{\frac{[\,]\boldsymbol{j}}{\neg\exists x.\boldsymbol{car}(x)\wedge\boldsymbol{own}\ x}}}$$

$$\frac{\frac{S|S}{S/C}}{\frac{\Phi}{\frac{[\,]}{\lambda p.p([])}}} \quad \frac{\frac{S|DP\rhd S}{C}}{\frac{John\ not\ own\ a\ car}{\frac{[\,]\boldsymbol{j}}{\neg\exists x.\boldsymbol{car}(x)\wedge\boldsymbol{own}\ x}}} = \frac{\frac{S|DP\rhd S}{S}}{\frac{John\ not\ own\ a\ car}{\frac{[\,]\boldsymbol{j}}{\neg\exists x.\boldsymbol{car}(x)\wedge\boldsymbol{own}\ x}}}$$

$$\frac{\frac{S|DP\rhd S}{S}}{\frac{John\ not\ own\ a\ car}{\frac{[\,]\boldsymbol{j}}{\neg\exists x.\boldsymbol{car}(x)\wedge\boldsymbol{own}\ x}}} \quad \frac{\frac{DP\rhd S|DP\rhd S}{S\backslash(S/S)}}{\frac{\cdot}{\frac{[\,]}{\lambda p\lambda q.p\wedge q}}} \quad \frac{\frac{DP\rhd S|S}{S}}{\frac{He\ came\ by\ foot}{\frac{\lambda y.[\,]}{\boldsymbol{came\ by\ foot}\ y}}}$$

$$\frac{S|S}{S}$$

$$= \frac{John\ not\ own\ a\ car.He\ came\ by\ foot}{\frac{\lambda y.[\,]\boldsymbol{j}}{\neg\exists x.\boldsymbol{car}(x)\wedge\boldsymbol{own}\ x\wedge\boldsymbol{came\ by\ foot}\ y}} \overset{Lower}{\Longrightarrow}$$

$$\frac{S}{\frac{John\ not\ own\ a\ car.He\ came\ by\ foot}{\neg\exists x.\boldsymbol{car}(x)\wedge\boldsymbol{own}\ x\wedge\boldsymbol{came\ by\ foot}\ \boldsymbol{j}}}$$

It is conceivable that an indefinite in direct object position may also rise from its minimal negated clause to give the inverse scope interpretation. This interpretation may sometimes be ruled out on pragmatic grounds as being too uninformative (for instance, there is a car that John does not own is not a valid interpretation for *John does not own a car*.) or may be the preferred interpretation (there is a certain colleague Mary does not like is the preferred interpretation *of Mary does not like a colleague.*). Also, there are lexical entries such as negative polarity items (for instance, *any*) or definite descriptions (such as *John, the man, the man who entered)* that, when in direct object position of a negated verb phrase, take wide scope over negation and thus bind subsequent anaphora. For instance, here it is the derivation of *Mary does not like John. He is rude.*:

$$\frac{\frac{S|S}{\frac{C|C}{\frac{DP}{\frac{Mary}{\frac{[\,]}{\frac{[\,]}{\boldsymbol{m}}}}}}}} \quad \frac{S|S}{\frac{C|C}{(DP\backslash C)/(DP\backslash C)}} \quad \frac{\frac{S|S}{\frac{C|C}{DP\backslash C}}}{\frac{like}{\frac{[\,]}{\frac{[\,]}{\boldsymbol{like}}}}} \quad \frac{S|DP\rhd S}{\frac{\frac{C|C}{\frac{DP}{John}}}{\frac{[\,]\boldsymbol{j}}{\frac{[\,]}{\boldsymbol{j}}}}} \quad =$$

$$\dfrac{\dfrac{S|DP \rhd S}{\dfrac{C|C}{C}}}{\begin{array}{c}Mary\ not\ like\ John\\ \dfrac{[\,]\boldsymbol{j}}{\neg[\,]}\\ \boldsymbol{like\ j\ m}\end{array}} \overset{Lower}{\Longrightarrow} \dfrac{\dfrac{S|DP \rhd S}{C}}{\begin{array}{c}Mary\ not\ like\ John\\ \dfrac{[\,]\boldsymbol{j}}{\neg[\ \boldsymbol{like\ j\ m}]}\end{array}}$$

$$\dfrac{\dfrac{S|S}{S/C}}{\begin{array}{c}\Phi\\ \dfrac{[\,]}{\lambda p.\,p([])}\end{array}} \quad \dfrac{\dfrac{S|DP \rhd S}{C}}{\begin{array}{c}Mary\ not\ like\ John\\ \dfrac{[\,]\boldsymbol{j}}{\neg[\ \boldsymbol{like\ j\ m}]}\end{array}} = \dfrac{\dfrac{S|DP \rhd S}{S}}{\begin{array}{c}Mary\ not\ like\ John\\ \dfrac{[\,]\boldsymbol{j}}{\neg[\ \boldsymbol{like\ j\ m}]}\end{array}}$$

$$\dfrac{\dfrac{S|DP \rhd S}{S}}{\begin{array}{c}Mary\ not\ like\ John\\ \dfrac{[\,]\boldsymbol{j}}{\neg[\ \boldsymbol{like\ j\ m}]}\end{array}} \quad \dfrac{\dfrac{S|S}{S\backslash(S/S)}}{\begin{array}{c}\cdot\\ \dfrac{[\,]}{\lambda p\lambda q.\,p\wedge q}\end{array}} \quad \dfrac{\dfrac{DP \rhd S|S}{S}}{\begin{array}{c}He\ is\ rude\\ \dfrac{\lambda y.[\,]}{\boldsymbol{is\ rude}\ y}\end{array}}$$

$$\dfrac{S|S}{S}$$
$$= Mary\ not\ like\ John.\ He\ is\ rude \overset{Lower}{\Longrightarrow}$$
$$\dfrac{\lambda y.[\,]\boldsymbol{j}}{\neg[\ \boldsymbol{like\ j\ m}]\wedge \boldsymbol{is\ rude}\ y}$$

$$S$$
$$Mary\ not\ like\ John.\ He\ is\ rude$$
$$\lambda y.[\neg[\ \boldsymbol{like\ j\ m}]\wedge \boldsymbol{is\ rude}\ y\ ]\boldsymbol{j}$$

$$S$$
$$= Mary\ not\ like\ John.\ He\ is\ rude$$
$$\neg[\ \boldsymbol{like\ j\ m}]\wedge \boldsymbol{is\ rude\ j}$$

The scope behavior of the quantificational determiners *every* and *any* may be accounted for in a similar manner. Consider for instance the following examples:

*John does not know every poem. *It is nice.*

*John does not know any poem. *It is nice.*

The interpretative difference between *every* and *any* is made (in line with Quine and Geach among others) by the scope behavior of the two quantificational determiners. *Any* prefers to take wide scope, whereas *every* rather takes narrow scope:

$$\dfrac{C|C}{\dfrac{C|C}{DP}}\ \begin{array}{c}John\\ [\,]\\ \hline [\,]\\ \boldsymbol{j}\end{array} \qquad \dfrac{C|C}{\dfrac{C|C}{(DP\backslash C)/(DP\backslash C)}}\ \begin{array}{c}not\\ \neg[\,]\\ \hline [\,]\\ [\,]\end{array} \qquad \left(\dfrac{\dfrac{C|C}{\dfrac{C|C}{(DP\backslash C)/DP}}\ \begin{array}{c}know\\ [\,]\\ \hline [\,]\\ \boldsymbol{know}\end{array} \qquad \dfrac{\dfrac{C|C}{\dfrac{C|C}{DP/N}}\ \begin{array}{c}every\\ \neg\exists x.[\,]\\ \hline P(x)\wedge\neg[\,]\\ x\end{array}\ \lambda P. \qquad \dfrac{C|C}{N}\ \begin{array}{c}poem\\ [\,]\\ \hline \boldsymbol{poem}\end{array}}\right)$$

$$\dfrac{\dfrac{C|C}{C|C}}{C}$$
$$= John\ not\ know\ every\ poem \overset{Lower\ \ two\ \ times}{\Longrightarrow}$$
$$\dfrac{\neg[\neg\exists x.[\,]]}{\boldsymbol{poem}(x)\wedge\neg[\,]}$$
$$\boldsymbol{know\ x\ j}$$

$$C$$
$$John\ does\ not\ know\ every\ poem$$
$$\neg[\neg\exists x.[\boldsymbol{poem}(x)\wedge\neg[\boldsymbol{know}\ x\ \boldsymbol{j}\ ]\ ]]$$

$$\begin{array}{ccc}S/C & & C\\ \Phi & & John\ does\ not\ know\ every\ poem\\ \lambda p.\,p([]) & & \neg[\neg\exists x.[\mathbf{poem}(x)\wedge\neg[\mathrm{know}\ x\ \mathbf{j}\ ]\ ]]\end{array} =$$

$$S$$
$$John\ does\ not\ know\ every\ poem$$
$$\neg[\neg\exists x.[\boldsymbol{poem}(x)\wedge\neg[\boldsymbol{know}\ x\ \boldsymbol{j}\ ]\ ]]$$

which means that there is (at least) one poem that John does not know, a fare approximation of the intended meaning. In this context, the interpretation of *It is nice.* crashes, because *it* cannot find a suitable antecedent into the preceding discourse. It would have been useless for *poem* to offer to bind in the first place, because *not* takes scope over it and negation has to close its scope before its minimal clause is interpreted in discourse.

The interpretation of the quantificational determiner *any* in discourse proceeds similarly:

$$\dfrac{\dfrac{C|C}{\dfrac{C|C}{DP}/N}}{\begin{array}{c}any\\ \neg\exists x.[\,]\\ \hline \lambda P.\dfrac{P(x)\wedge\neg[\,]}{x}\end{array}} \quad \dfrac{\dfrac{C|C}{N}}{\begin{array}{c}poem\\ [\,]\\ \hline \boldsymbol{poem}\end{array}} = \dfrac{\dfrac{C|C}{\dfrac{C|C}{DP}}}{\begin{array}{c}any\ poem\\ \neg\exists x.[\,]\\ \hline \boldsymbol{poem}(x)\wedge\neg[\,]\\ x\end{array}}$$

$$\overset{Lift}{\Longrightarrow} \dfrac{\dfrac{C|C}{\dfrac{C|C}{\dfrac{C|C}{DP}}}}{\begin{array}{c}any\ poem\\ \neg\exists x.[\,]\\ \hline \boldsymbol{poem}(x)\wedge\neg[\,]\\ \hline [\,]\\ x\end{array}}$$

$$\dfrac{C|C}{\dfrac{C|C}{\dfrac{C|C}{DP}}}\ \begin{array}{c}John\\ [\,]\\ \hline [\,]\\ [\,]\\ \hline \boldsymbol{j}\end{array} \qquad \dfrac{C|C}{\dfrac{C|C}{(DP\backslash C)/(DP\backslash C)}}\ \begin{array}{c}not\\ [\,]\\ \hline [\,]\\ \neg[\,]\\ \hline [\,]\end{array} \qquad \left(\dfrac{\dfrac{C|C}{\dfrac{C|C}{(DP\backslash C)/DP}}\ \begin{array}{c}know\\ [\,]\\ \hline [\,]\\ [\,]\\ \hline \boldsymbol{know}\end{array} \qquad \dfrac{\dfrac{C|C}{\dfrac{C|C}{\dfrac{C|C}{DP}}}}{\begin{array}{c}any\ poem\\ \neg\exists x.[\,]\\ \hline \boldsymbol{poem}(x)\wedge\neg[\,]\\ \hline [\,]\\ x\end{array}}\right)$$

$$\dfrac{\dfrac{C|C}{\dfrac{C|C}{\dfrac{C|C}{DP}}}}{\begin{array}{c}= John\ not\ know\ any\ poem \overset{Lower\ \ three\ \ times}{\Longrightarrow}\\ \neg\exists x.[\,]\\ \hline \boldsymbol{poem}(x)\wedge\neg[\,]\\ \hline \neg[\,]\\ \boldsymbol{know\ x\ j}\end{array}}$$

$$\frac{C}{\textit{John does not know any poem}}$$
$$\neg\exists x.\,\textbf{\textit{poem}}(x) \wedge [\textbf{\textit{know}}\ x\ \textbf{\textit{j}}\ ]$$

which means that there is no poem that John knows, a fare approximation of the intended meaning. It cannot be argued that it is the negation which prevents further referring to *any poem*, because *any* takes wide scope over negation. Obviously, the same mechanism prevents *poem* to bind subsequent anaphora both in the case of *every* and of *any*.

Notice that there is a third intermediate possibility of scope taking, with negation taking scope at the second level of the compositional tower:

$$\frac{\frac{C|C}{\frac{C|C}{DP}}\ \textit{John}\ \frac{[\,]}{\frac{[\,]}{\textbf{\textit{j}}}}}{\left(\frac{C|C}{\frac{C|C}{(DP\backslash C)/(DP\backslash C)}\ \textit{not}\ \frac{[\,]}{\frac{\neg[\,]}{[\,]}}}\left(\frac{C|C}{\frac{C|C}{(DP\backslash C)/DP}\ \textit{know}\ \frac{[\,]}{\frac{[\,]}{\textbf{\textit{know}}}}}\left(\frac{\frac{C|C}{\frac{C|C}{DP/N}\ \textit{any}}\quad\frac{C|C}{\frac{N}{\textit{poem}}\ \frac{[\,]}{\textbf{\textit{poem}}}}}{\lambda P.\frac{P(x)\wedge\neg[\,]}{x}\ \frac{\neg\exists x.[\,]}{\neg(\textbf{poem}(x)\wedge\neg[\,])}\ \textbf{know}\ x\ \textbf{j}}\right)\right)\right)}$$

$$\frac{\frac{C|C}{C|C}}{C}$$
$$= \text{John not know any poem}$$

$$\frac{\neg\exists x.[\,]}{\neg(\textbf{poem}(x)\wedge\neg[\,])}\ \textbf{know}\ x\ \textbf{j}$$

$$\xRightarrow{\text{Lower two times}}\quad \frac{C}{\text{John does not know any poem}}\ \neg\exists x.\neg[\textbf{poem}(x)\wedge\neg[\textbf{know}\ x\ \textbf{j}\ ]]$$

$$\frac{S}{= \text{John does not know any poem}}\ \neg\exists x.\neg\textbf{poem}(x)\vee\textbf{know}\ x\ \textbf{j}$$

This interpretation is impossible in natural language. Thus, it may be said that *any* obligatory takes wide scope over negation not only with its general (first level) scope, but also with its nuclear scope.

4 Conclusions

To conclude, allowing arbitrary type shifting overgenerates interpretations impossible in natural language. In order to filter some of them out, we proposed a mechanism that forbids clause bounded lexical entries to take scope outside their minimal clause. For this natural language fragment, the mechanism and the scope precedence preference of the lexical entries (for instance, *not* > indefinites, *not* > *every*, *not* < *any*) ensures the right discourse truth conditions.

References

Barker, Chris. 2002. Continuations and the nature of quantification. *Natural Language Semantics* 10(3). 211-242.

Barker, Chris. 2004. Continuations in natural language. In Hayo Thielecke, editor, Proceedings of *the fourth ACM SIGPLAN workshop on continuations*, pages 55-64, 2004.

Barker, C and Shan Chung-chieh. 2008. Donkey anaphora is in-scope binding. In *Semantics and Pragmatics* Volume 1, pages 1-46.

Dowty, David. 2007. Compositionality as an empirical problem. In Chris Barker & Pauline Jacobson (eds.), *Direct compositionality*. Oxford University Press.

Dinu, Anca. 2011. Versatility of 'continuations' in discourse semantics. Fundamenta Informaticae (to appear).

de Groote, Philippe. 2006. Towards a montagovian account of dynamics. In *Semantics and Linguistic Theory* XVI.

Jacobson, Pauline. 1999. Towards a variable-free semantics. *Linguistics and Philosophy* 22(2). 117-185.

M. Felleisen. 1988. The theory and practice of first-class prompts. In J. Ferrante and P. Mager, editors, Proceedings of the *Fifteenth Annual ACM Symposium on Principles of Programming Languages*, pages 180-190, San Diego, California, Jan. 1988. ACM Press.

Montague, Richard. 1970. The Proper Treatment of Quantification in English. In R. Thomason (ed). *Formal Philosophy: Selected Papers of Richard Montague*,247-270. New Haven:Yale.

Partee, Barbara H. & Mats Rooth. 1983. Generalized conjunction and type ambiguity. In Rainer Buerle, Christoph Schwarze & Arnim von Stechow. (eds.), *Meaning, use, and interpretation of language*, 361-383.Walter de Gruyter and Co.

Shan, Chung-chieh and Chris Barker. 2006. Explaining crossover and superiority as left-to-right evaluation. *Linguistics and Philosophy* 29.1:91-134.

Shan, Chung-chieh. 2005. *Linguistic side effects*. Ph.D. thesis, Harvard University.

Steedman, Mark. 2000. *The syntactic process*. MIT Press.