Quantifiers in Dependency Tree Semantics

Leonardo Lesmo, Livio Robaldo, Jelle Gerbrandy

Dipartimento di Informatica - Universitá di Torino
{lesmo,robaldo,gerbrand}@di.unito.it

Abstract

Dependency Tree Semantics (DTS) is an underspecified formalism for representing quantifier scope ambiguities in natural language. DTS features a direct interface with a Dependency grammar and an incremental, constraint-based disambiguation mechanism. In this paper, we discuss the meaning of quantifier dependency in DTS by translating its well formed structures into formulae of a Second Order Logic augmented with Mostowskian generalized quantifiers.

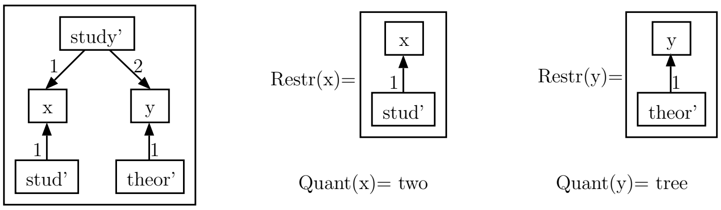
1 Introduction

Dependency Tree Semantics (DTS) is an underspecified formalism for dealing with quantifier scope ambiguity. DTS tries to keep the advantages of most common underspecification techniques: it has a straightforward syntax-semantics interface with a Dependency Grammar, just as QLF has [1], and it allows for monotonically adding constraints to take partial disambiguations into account, just as in UDRT [12], MRS [3] or CLLS [4]. These features have been presented in [7] and [8], whereas in [9] DTS is proposed as a possible underspecified semantic structure of Meaning⇔Text Theory [10]. This paper discusses a third property of DTS in further depth: the possibility to represent branching quantifier (BQ) readings. Branching quantification in DTS has partially been discussed in [7] and [8], in which we compared DTS with First Order Logic (FOL). However, FOL is limited in that it allows to represent only standard quantifiers ($\exists$ and $\forall$); in this paper we compare DTS with the logic developed in [13] and [14], which is a fragment of Second Order Logic which allows for a representation of branching quantification with Generalized Quantifiers.

1.1 Intuitions behind Dependency Tree Semantics

The key idea of DTS is to specify quantifier scope by explicitly showing the dependencies between involved (quantified) groups of entities, i.e. by implementing a sort of "Skolemization" in the underspecified representation. Well-formed structures in DTS are based on a simple graph G that represents the predicate-arguments relations, without any quantification. The nodes of G are either predicates or discourse referents; each arc connects a predicate with a discourse referent and is labelled with the number of the predicate argument position. With each discourse referent we associate a *quantifier* (given by a function $QUANT$ from discourse referents to quantifiers) and its *restriction*, which is given by a function $RESTR$ that associates a subgraph of G to each discourse referent. In (1), we show a first simple example

(1) Two students study three theorems



The representation in (1) is still ambiguous; to disambiguate, we need to specify how the quantifiers depend on each other. This is done by inserting dotted arcs between discourse referents, named *semdep* arc. In figure 1.a and fig 1.b two fully-specified representations of sentence (1) are given. Fig.1.a shows the reading in which the quantifier 'three' depends on (has scope inside) the quantifier 'two'. In figure 1.b, the arc linking x to y specifies that the two students depend on the theorems. In both interpretations, the wide-scope quantifier is linked to a new node called *Ctx* – the context.

But DTS allows for very natural representation of a third reading of sentence (1): in figure 1.c, both discourse referents are linked to the context. This is the branching quantifier (BQ) reading. As we will see, the BQ reading is true only in those models in which we can find a set of two students and a set of three theorems, for which it holds that each student in the first set studies each theorem in the second one. In NL, there are many cases in which the correct

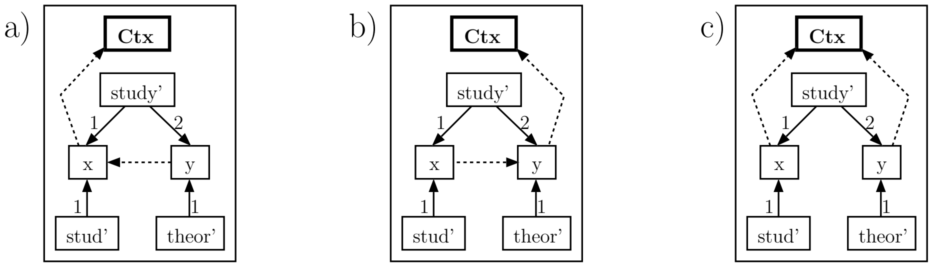


Fig. 1. The three readings of sentence (1)

truth conditions can be captured only via a BQ reading; in fact, it is easy to add some context elements in the sentence in order to force the two involved sets to be constant; for instance, in (2.i), the involved students and theorems are explicitly mentioned in two appositions, while in (2.ii) the prepositional modifier *with my sister* favours an interpretation in which three persons, two friends of mine and my sister, went together to three same concerts.

Finally, even if there are not explicit syntactic elements forcing a BQ reading, in many cases this is done by world knowledge; for example, in (2.iii), world knowledge seems to render the reading in which two students have seen the same three drug dealers the most salient; in fact, the presence of drug-dealers in front of a school is (fortunately) a rare event and this induces to prefer the reading minimizing the number of involved drug dealers.

(2) (i) Two students, *John and Jack,* study three theorems*: the first three of the book.*

(ii) Two friends of mine went to three concerts *with my sister.*

(iii) Two students of mine have seen three drug dealers in front of the school.

Not all possible configurations of *semdep* arcs are allowed. For instance, a well-formed DTS cannot contain cycling paths, which would correspond to a reading in which two sets of entities depend on each other, which is clearly absurd. Furthermore, there are constraints to reduce the available readings to those admitted in NL. In this paper, we will focus on the expressivity of the general formalism, and provide a precise definition of the meaning of all configurations that respect a minimal set of syntactic constraints, and abstract from the question whether they correspond to an actual reading in NL. In other words, in DTS the set of logical admitted readings is kept separate from the subset of readings admitted in NL, and this paper focus on the former.

1.2 Formalisation: Syntax of DTS

A well-formed structure (wfs) in DTS is a Scoped Dependency Graph (SDG) as defined below. We take as given a set of predicates pred and a set of discourse referents D.

Definition 1.1 [Flat Dependency Graphs (FDG)]
A Flat Dependency Graph is a tuple $\langle N, L, A, Dom, f\rangle$ s.t.:

- N is a set of nodes $\{n_1, n_2, \ldots, n_k\}$.
- L is a set of labels $\{l_1,\, l_2,\, \ldots,\, l_m\}$; in fig.1, $L{\equiv}\{1, 2\}$.
- $Dom \equiv \mathsf{pred} \cup \mathsf{D}$ is a set *domain objects*: predicates and discourse referents
- f is a function $f : N \mapsto Dom$, specifying the node referent, i.e. the domain object with which the node is associated. In the following, whenever $f(n) \in X$, we will say that node n is of type X.
- A is a set of *arcs*. An arc is a triple $(n_s,\, n_d,\, l)$, where $n_s, n_d \in N$, n_s is of type pred, n_d is of type D and $l \in L$.

Without going into further details, we stipulate that G_f is a connected acyclic graph such that each node of type pred has one node of type D for each of its places. Note that there can be two different nodes u and v s.t. $f(u){=}f(v)$, i.e. the nodes in N can be seen as *occurrences* of symbols from Dom.

Definition 1.2 [Scoped Dependency Graph (SDG)]
A Scoped Dependency Graph is a tuple $\langle G_f, \mathsf{ctx}, \mathsf{Q}, \mathsf{quant}, \mathsf{restr}, \mathsf{SemDep}\rangle$ s.t.:

- $G_f = \langle N, L, A, Dom, f \rangle$ is an FDG.
- ctx is a special element called the context.
- Q is a set of 2-place Mostowskian quantifiers $\{$*every*, *most*, *two*, $\ldots\}$ [1]
- quant is a total function $N_D \mapsto Q$, where $N_D \subseteq N$ are the nodes of type D
- restr is a function assigning to each $d \in N_D$ its *restriction*, which is a subgraph of G_f.
- SemDep is a relation $N_D \times (N_D \cup \{\{\mathsf{ctx}\}\})$.

When $\mathsf{SemDep}(\mathsf{d}, \mathsf{d}')$, we say that d *depends on* d'. Note that a discourse referent can depend on more than one other discourse referent. The dependence relation needs to satisfy the following constraints:

- The transitive closure of SemDep is a partial order on all discourse referents and ctx, with ctx as its maximal element.
- Let d be a discourse referent, and let $R(d)$ be the smallest set that contains d, and for which it holds that if d' is in $R(d)$ and d'' occurs in the restriction of d', then also $d'' \in D$. It must hold that:
  · If $d_1 \in R(d)$, $d_2 \notin R(d)$, and d_1 depends on d_2, then also d depends on d_2
  · If $d_1 \in R(d)$, $d_2 \notin R(d)$, and d_2 depends on d_1, then also d depends on d_1

These last two constraints serve to exclude certain dependency relations that are 'logically impossible', and make sure that, for example, a sentence like "Most representatives of a company took every sample" does not get a reading in which 'a' depends on (only) 'every' and 'every' depends (only) on 'most'.

2 Branching quantification

Branching quantification was introduced by Henkin [5] in the context of FOL; Hintikka [6] showed that it can occur also in NL. A great step toward the definition of a model-theoretic schema for BQ was made by Barwise [2] who merged Hintikka's BQ account with the theory of Generalized Quantifiers. Barwise's idea was that the truth-conditions of BQ readings are connected with the monotonicity of the involved quantifiers. He claimed that there is no uniform schema for BQ: the formulae associated to sentences featuring all monotone increasing (M$\uparrow$) quantifiers are different from those associated to sentences featuring all monotone decreasing (M$\downarrow$) quantifiers. According to Barwise, sentences with mixed quantifiers (some M$\uparrow$ and some M$\downarrow$) make no

[1] A 2-place Mostowskian Quantifier [11] (see also [13]) is a symbol Q such that, if x is an individual variable and Ψ, Φ are formulae then $Q_x(\Psi, \Phi)$ is also a formula. Semantically, Q denotes, in every model M with universe A, a function q which takes in input two subsets B and C of A and returns a truth-value. Mostowskian Quantifiers are cardinality quantifiers, in the sense that $q(B, C)$ depends only on the cardinalities of the sets $(B \cap C)$, $(B \setminus C)$, $(C \setminus B)$ and $(A \setminus (B \cup C))$. Some examples are
· $\|All_x(P_1(x), P_2(x))\|^M = true$ iff $|(\|P_1(x) \wedge \neg P_2(x)\|^M)| = 0$
· $\|Few_x(P_1(x), P_2(x))\|^M = true$ iff $|(\|P_1(x) \wedge P_2(x)\|^M)| > \eta$

sense from a linguistic point of view.
On the other hand, Sher [13], [14] observed that since the semantics of linearly ordered quantification is provided regardless to monotonicity, there seems to be no methodological reason for imposing further constraints in case of partially ordered quantification. In other words, even if readings from NL are not available, this should not exclude their logical interpretation.
Sher specified the semantics of BQ on the basis of a precise definition of the involved groups, according to so-called *maximality conditions*; roughly, her claim is that the interpretation of a BQ reading with quantifiers of any type corresponds to the one of Barwise for M↑ quantifiers augmented with a maximality condition requiring that the involved sets are maximal with respect to the body of the formula. Consider the two following sentences:

(3) (i) Most of the dots and most of the stars are all connected by lines.
(ii) Few of the dots and few of the stars are all connected by lines.

In Sher's logic (let us name it L_0) sentences in (3) are associated with formulas of the following form:

$$(4) \quad \exists P_1, P_2[\ \begin{array}{l} C1 : Q1_x(dot(x), P_1(x))\wedge \\ C2 : Q2_y(star(y), P_2(y))\wedge \\ IN : \forall_{xy}[(P_1(x) \wedge P_2(y)) \rightarrow conn(x, y)]\wedge \\ \mathrm{Max}(\langle P_1, P_2\rangle, IN)\ ] \end{array}$$

where $Q1$ and $Q2$ are the Mostowskian quantifiers corresponding to the determiners in our example: $Q1{=}Q2{=}Most$ for (3.i); and $Q1{=}Q2{=}Few$ for (3.ii). The symbols $C1$, $C2$, IN are labels on the subformulae and $\mathrm{Max}(\langle P_1, P_2\rangle, IN)$ is an abbreviation for a maximality condition that states that two sets P_1 and P_2 are maximal with respect to the formula with label IN, in the sense that there are no strict supersets of P_1 and P_2 that satisfy IN. Formally, the maximality condition in (4) is the following formula:

$$\begin{array}{l} \mathrm{Max}(\langle P_1, P_2\rangle, IN) \Leftrightarrow \\ \quad \forall P'_1, P'_2[\ \forall_{xy}[\ (P_1(x) \wedge P_2(y)) \rightarrow (P'_1(x) \wedge P'_2(y))\wedge \\ \qquad\qquad (P'_1(x) \wedge P'_2(y)) \rightarrow conn(x, y)\ ] \rightarrow \\ \qquad \forall_{xy}[\ (P'_1(x) \wedge P'_2(y)) \rightarrow (P_1(x) \wedge P_2(y))\ ]] \end{array}$$

Sher generalizes the schema of (4), so that it applies to any partially ordered set of arbitrary quantifiers. To achieve this, it is necessary to existentially quantify n-ary *generalized Skolem functions* H_i rather than simple sets P_i, and to assert maximality conditions also on the subformulae with label C_i. Here, an n-ary Skolem function is just an $n+1$-ary relation H – we will write $H(x_1, \ldots x_{n+1})$ if $x_1 \ldots x_{n+1}$ stand in the relation H, but also write $H(x_1 \ldots x_n)$ for the set of objects x_{n+1} s.t. $H(x_1, \ldots x_{n+1})$. Consider now a branching reading such as in the following sentence:

(5) Few men inserted a coin in three coffee machines.

$$\begin{array}{lll} Few_x(\text{man'}(x)) & & \\ & \rangle & A_z(\text{Coin'}(z)) \text{ — } \text{Inserted'}(x,\ z,\ y) \\ Three_y(\text{CoffeeMach'}(y)) & & \end{array}$$

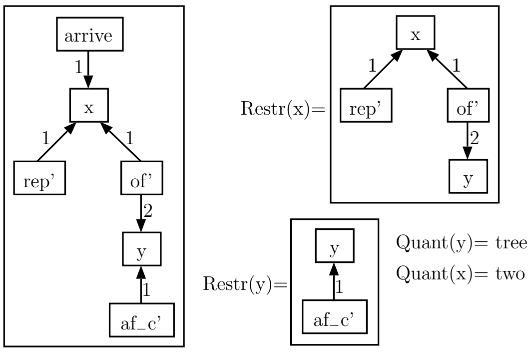
$$\begin{array}{l} =_{df}\ \exists H_x, H_y, H_z[\ \text{C}_x\text{: } Few_x(\text{man'}(x),\ H_x(x))\ \& \\ \quad \text{C}_y\text{: } Three_y(\text{CoffeeMach'}(y),\ H_y(y))\ \& \\ \quad \text{C}_z\text{: } \forall_{xy}[(H_x(x) \wedge H_y(y)) \rightarrow A_z(\text{coin'}(z),\ H_z(x,\ y))]\ \& \\ \quad \text{IN: } \forall_{xyz}[H_z(x,\ y,\ z) \rightarrow \text{inserted'}(x,\ y,\ z)]\ \& \\ \quad \text{Max}(\langle H_x,\ H_y\rangle,\ \text{C}_z)\ \&\ \text{Max}(\langle H_z\rangle,\ \text{IN})\ ] \end{array}$$

In this reading, the quantifier A depends on both *Three* and *Few*: there can be a different coin for every pair of a man and a coffee machine. This is reflected by the fact that H_z, the Skolem function associated with the quantifier A, is a 2-ary function, while H_x, H_y are 0-ary Skolem functions (that is, predicates). The formula states that we have to find witnesses H_x, H_y and H_z such that H_z corresponds to the extension of *inserted'*, and H_x and H_y are maximal sets of individuals x and y such that the set of objects z inserted by x in y, $H_z(x, y, z)$, includes at least one coin; H_x is a set of a "few men" and H_y contains "three coffee machines". See [14] for the formal details.

3 Nested Quantification

A limitation of Sher's logic is that it does not handle the case in which one quantifier occurs in the syntactical restriction of another quantifier. Consider:

(6) Two representatives of three African countries arrive.



In this example, the quantifier *Three* occurs in the syntactic restriction of *Two*. This corresponds to the fact that the discourse referent y occurs in the graph $RESTR(x)$. This type of reading cannot be directly represented in Sher's logic. Therefore, we propose to extend her definitions to accommodate for these cases as well. Lack of space does not permit us to state the precise definitions; we will give two examples instead which should illustrate how the definitions work. Before discussing the three possible disambiguations of (6),

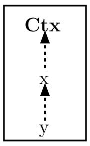
we introduce a new abbreviation to increase readability.
If Φ is a well formed formula, $x_1 \ldots x_n$ a sequence of discourse referents, and $S_1, \ldots, S_n$ a sequence of predicates, we define:

$$\begin{aligned}&\langle S_1, \ldots, S_n\rangle \underset{\max}{\subseteq} \Phi[x_1 \ldots x_n] \Leftrightarrow \\ &\qquad \mathrm{Max}(\langle S_1, \ldots, S_n\rangle, \forall x_1 \ldots x_n[(S_1(x_1) \wedge \ldots \wedge S_n(x_n)) \rightarrow \Phi])\end{aligned}$$

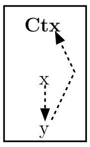
We will omit the reference to the variables $x_1 \ldots x_n$ in the notation when this does not lead to confusion. By using $\underset{\max}{\subseteq}$, the formula in (5) can be replaced by the following equivalent

$$\begin{aligned}&\exists H_x, H_y, H_z[\ \mathit{Few}_x(\text{man'}(x),\ H_x(x))\ \&\ \mathit{Every}_y(\text{CoffeeMach'}(y),\ H_y(y))\ \& \\ &\qquad \langle H_x, H_y\rangle \underset{\max}{\subseteq} [\ A_z(\text{coin'}(z),\ H_z(x, y, z))\& \\ &\qquad \langle H_z(x, y)\rangle \underset{\max}{\subseteq} \text{inserted'}(x, y, z)\ ]\ ]\end{aligned}$$

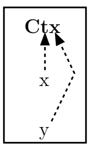
For representing the restriction of quantifiers in the logic, in addition to the Skolem functions H_x that represent the body of the quantifiers, we introduce *restriction sets* Ψ_x. The three readings of (6) can now be represented as:



$$\begin{aligned}&\exists H_x, H_y, \Psi_x, \Psi_y[\ \mathit{Two}_x(\Psi_x(x), H_x(x))\&\ \langle H_x\rangle \underset{\max}{\subseteq} (\text{arrive'}(x))\ \& \\ &\quad \langle \Psi_x\rangle \underset{\max}{\subseteq} [\mathit{Three}_y(\Psi_y(x, y),\ H_y(x,\ y))\ \& \\ &\qquad\qquad \langle \Psi_y(x)\rangle \underset{\max}{\subseteq} (\text{af\_c'}(y))\ \& \\ &\qquad\qquad \langle H_y(x)\rangle \underset{\max}{\subseteq} (\text{repr\_of'}(x,y))\ ]]\end{aligned}$$



$$\begin{aligned}&\exists H_x, H_y, \Psi_x, \Psi_y[\ \mathit{Three}_y(\Psi_y(y),\ H_y(y))\ \&\ \langle \Psi_y\rangle \underset{\max}{\subseteq} (\text{af\_c'}(y))\ \& \\ &\quad \langle H_y\rangle \underset{\max}{\subseteq} [\mathit{Two}_x(\Psi_x(y, x),\ H_x(y, x))\ \& \\ &\qquad\qquad \langle \Psi_x(y)\rangle \underset{\max}{\subseteq} (\text{repr\_of'}(x,y))\ \& \\ &\qquad\qquad \langle H_x(y)\rangle \underset{\max}{\subseteq} (\text{arrive'}(x))\ ]]\end{aligned}$$



$$\begin{aligned}&\exists H_x, H_y, \Psi_x, \Psi_y[\ \mathit{Two}_x(\Psi_x(x), H_x(x))\ \&\ \mathit{Three}_y(\Psi_y(y),\ H_y(y))\ \& \\ &\qquad \langle \Psi_x, H_y\rangle \underset{\max}{\subseteq} (\text{repr\_of'}(x,y))\ \&\ \langle \Psi_y\rangle \underset{\max}{\subseteq} (\text{af\_c'}(y))\ \& \\ &\qquad \langle H_x\rangle \underset{\max}{\subseteq} (\text{arrive'}(x))\ ]\end{aligned}$$

Let us shortly discuss each of these readings.
In the first reading, y depends on x, which is reflected in the fact that Ψ_y and

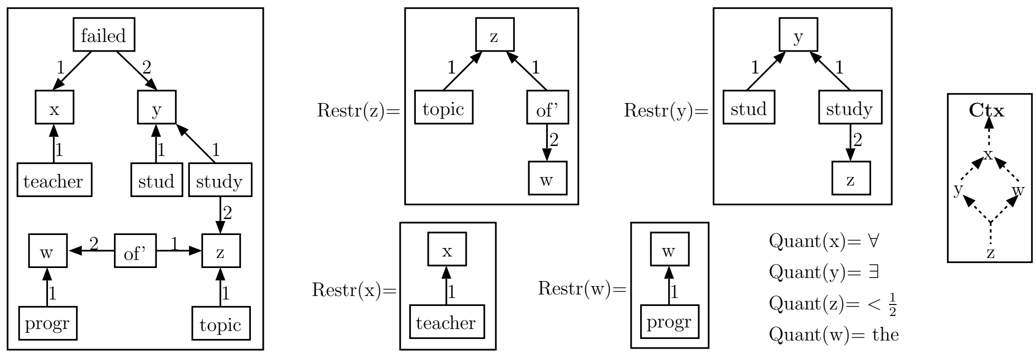
H_y are unary Skolem functions whose values depend on the value for x. The restriction set of 'three', $\Psi_y(x)$, is (for each x) the set of all African countries, while $H_y(x)$ is the set of objects represented by x. Therefore, the restriction set of 'two', Ψ_x, is a maximal set of individuals x that represent three African countries. Two of these individuals must be in H_x – the set of those that arrive.

In the second reading, x depends on y. The set Ψ_y consists of all African countries. The set H_y must contain three of these, and it is required that for each element y in H_y there are two individuals in the set of all its representatives $\Psi_x(y)$ that are in $H_x(y)$, which consists of all individuals that arrive.

The third formula represents the branching reading of the sentence, in which the two discourse referents do not depend on each other. This formula states that there are sets Ψ_x and H_y such that each individual in Ψ_x represents all elements from H_y (this is expressed by the maximality condition on the pair (Ψ_x, H_y)), and for which it holds that H_y contains three African countries, and that two of the representatives from Ψ_x must arrive. In the following, we report a last complex example:

(7) Every$_x$ teacher failed two$_y$ students that studied less than half$_z$ of the topics in the$_w$ program.

The following DTS represents a reading of (7) in which the discourse referent w depends on both y and z, and y and z depend on x.



This DTS gets the translation reported below; in this interpretation, the two students and the program depend on a teacher, while the set of topics depends both on a program and on a student. In the formula, the pair of students associated to a teacher $x \in H_x$ has to belong to the set Ψ_y, i.e. the set of students y such that the set of things studied by y, i.e. $H_z(x, y, w)$, contains less than half elements of Ψ_z, i.e. the set of topic in $H_w(x)$, i.e. the program of x.

$$
\begin{array}{l}
\exists H_x, H_y, H_z, H_w, \Psi_x, \Psi_y, \Psi_z, \Psi_w[\\
\quad Every_x(\Psi_x(x),\ H_x(x))\ \&\ \{\Psi_x\} \underset{\max}{\subseteq} (\text{teacher'}(x))\ \& \\
\quad \{H_x\} \underset{\max}{\subseteq} [\ The_w(\Psi_w(x,w),\ H_w(x,w))\ \&\ \{\Psi_w(x)\} \underset{\max}{\subseteq} (\text{progr'}(w))\ \& \\
\qquad\qquad Two_y(\Psi_y(x,y),\ H_y(x,y))]\ \&\ \{H_y(x)\} \underset{\max}{\subseteq} (\text{failed'}(x,\ y))\ \& \\
\qquad\qquad \{\Psi_y(x), H_w(x)\} \underset{\max}{\subseteq} [\ Lth_z(\Psi_z(x,y,w,z),\ H_z(x,y,w,z))\ \& \\
\qquad\qquad\qquad\qquad \{\Psi_z(x,y,w)\} \underset{\max}{\subseteq} (\text{topic'}(z) \wedge \text{of'}(z,\ w))\ \& \\
\qquad\qquad\qquad\qquad \{H_z(x,y,w)\} \underset{\max}{\subseteq} (\text{stud'}(y) \wedge \text{study'}(y,\ z))]]]
\end{array}
$$

4 Conclusions and further works

In this paper, a comparison between Dependency Tree Semantics and Sher's work on Branching Quantification and Generalized Quantifiers has been presented. In particular, we have shown how disambiguated DTS structures can be related to formulae of an extension of the formalism from [14] to represent branching quantification. This provides a way to model-theoretically interpret disambiguated DTS structures. Concerning further work, one of the next steps in research on DTS will be extending its expressivity in order to deal with cumulativity, which is a topic that has received very little attention in recent studies on underspecification. Cumulative readings arise from a different kind of branching quantification, as argued in [13], so the step for including them is more natural in DTS than in other underspecified logics that do not take BQ into account.

References

[1] Alshawi, H., editor, "The Core Language Engine," Mit Press, Cambridge, MA, 1992.

[2] Barwise, J., *On branching quantifiers in english*, The Journal of Philosophical Logic (1979), pp. 47–80.

[3] Copestake, A., D. Flickinger and I. Sag, *Minimal recursion semantics. an introduction*, Technical report, Manuscript, Stanford University (1999).

[4] Egg, M., A. Koller and J. Niehren, *The constraint language for lambda structures*, J. of Logic, Language and Information **10** (2001), pp. 457–485.

[5] Henkin, L., *Some remarks on infinitely long formulas*, in: *Finitistic methods, Proc. Symphosium of Foundations Math*, Warsaw, 1961, pp. 167–183.

[6] Hintikka, J., *Quantifiers vs quantification theory*, Dialectica (1973), pp. 329–358.

[7] Lesmo, L. and L. Robaldo, *Dependency tree semantics and underspecification*, in: *Proc. Int. Conf. On Natural language processing (ICON2004)*, Hyderabad, India, 2004.

[8] Lesmo, L. and L. Robaldo, *From dependency tree semantics to fol*, in: *Proc. 6th Workshop on Computational Semantics (IWCS-6)*, Tilburg, 2005, pp. 384–386.

[9] Lesmo, L. and L. Robaldo, *Underspecification of quantifier scope in mtt*, in: *Proc. 2th Int.Conf. on Meaning Text Theory*, Moscow, 2005.

[10] Melcuk, I., *Semantics and the lexicon in modern linguistics.*, in: A. Gelbukh, editor, *In Proc. of the 1st International Conference on Intelligent Text Processing and Computational Linguistics (CICLing)*, 2000, pp. 6–18.
URL `www.CICLing.com`

[11] Mostowski, A., *On a generalization of quantifiers.*, Fundamenta Mathematicae **44** (1957), pp. 12–36.

[12] Reyle, U., *Dealing with ambiguities by underspecification: Construction, representation and deduction*, Journal of Semantics (1993), pp. 123–179.

[13] Sher, G., *Ways of branching quantifiers*, Linguistics and Philosophy (1990), pp. 393–422.

[14] Sher, G., *Partially-ordered (branching) generalized quantifiers: a general definition*, The Journal of Philosophical Logic (1997), pp. 1–43.