Quantification Over Situation Variables in LTAG: Some Constraints

Maribel Romero
University of Pennsylvania

1. Introduction

Some natural language expressions –namely, determiners like *every, some, most*, etc.— introduce quantification over individuals (or, in other words, they express relations between sets of individuals). For example, the truth conditions of a sentence like (1a) are represented in Predicate Logic (PrL) by binding the occurrences of the individual variable *x* with the quantifier ∀, as in (1b).[1]

(1) a. Every professor run the marathon.
b. ∀x [professor(x) → run-the-marathon(x)]

In a similar way, it has been argued that certain expressions introduce quantification over possible worlds or possible situations (Lewis 1973, among many others). The set of possible worlds includes the actual world --where the individuals are the way they actually are-- and any other logically possible world --where individuals may have different properties from the ones they have in actuality. In this paper, I assume a Situation Semantic framework (Barwise-Perry 1983, Kratzer 1989, von Fintel 1994) and do not quantify over entire possible worlds, but over *parts* of possible worlds, i.e., over possible situations.[2] In (2a), e.g., the speaker predicates the property of being in the Bahamas by 5pm of Jorge in all the (actual or non-actual relevant) situations *s* where all my actual obligations are fulfilled. This is informally represented in (2b). The variable s_0 stands for the situation at which the sentence as a whole is evaluated (the actual situation), and the variable *s* ranges over the possible (actual or non-actual) situations considered.

(2) a. Jorge has to be in the Bahamas by 5pm.
b. [[(2a)]](s_0) = 1 iff ∀s [all my actual obligations in s_0 are fulfilled in s → in(jorge, bahamas, 5pm, s)]

The present paper has two goals.

The first one is to implement in LTAG the semantics of some natural language expressions that quantify over possible situations. In particular, I will propose lexical entries for the modal auxiliary *must*, for the intensional adverb *probably* and for the adverb of quantification *sometimes*.[3] The proposal will model insights from the philosophical and linguistic literature (Stalnaker 1968, Lewis 1973, Cresswell 1990, Kratzer 1979) into the LTAG quantificational schemata developed in Kallmeyer-Joshi (2001).

The second, more important aim is to account, within LTAG, for a certain constraint on binding of situation variables discussed in Percus (2000) (see also Musan 1995 for related observations on time variables). In an nutshell, this binding constraint requires the following. Whereas predicates within (simple) Noun Phrases (NPs) can be evaluated with respect to a non-local situation binder, the situation variable of the main predicate in the Verb Phrase (VP) has to be bound by the closest c-commanding situation operator. It will be argued that this constraint follows automatically if, using the LTAG denotations proposed in this paper, the compositional semantics is computed on the derivation tree rather than on the derived tree.

The paper is organized as follows. In section 2, I will briefly present Kallmeyer-Joshi's (2001) proposal for quantifiers over individuals and extend it to quantifiers over situations. Semantic denotations for *must*, *probably* and *sometimes* will be spelled out in LTAG. Section 3 introduces the core data on the behavior of NPs versus

1. (1b) is a simplified version. Once we add situation variables, the denotation of (1a) for a given evaluation situation s_0 is represented as in (i). [[.]] is the interpretation function from linguistic expressions to their intensions.
(i) [[(1a)]](s_0) = 1 iff ∀x [professor(x,s_0) → run-the-marathon(x,s_0)]
2. The choice of situations instead of worlds is orthogonal to the arguments in this paper. Situations are best fitted to account for adverbs of quantification (*always, sometimes, often*, etc.) involving indefinites and donkey-anaphora (see von Fintel 1994).
3. The denotation of other members of each category can, of course, be easily modeled after the proposed entries (e.g., for modals like *can*, *might*, *would, should*; intensional adverbs like *necessarily*, *possibly*, *perhaps*, *unlikely*; and adverbs of quantification like *always, usually*, *often, rarely*.)

 Proceedings of the Sixth International Workshop on Tree Adjoining Grammar and Related Frameworks (TAG+6), pp. 108-117. Universitá di Venezia.

VPs with respect to situation binding. Section 4 derives the asymmetric behavior of NPs and VPs with respect to situation binding. We will consider a simple case with one situation adverb. It will be shown that the freedom of the NP situation variable and the locality of the VP situation variable follow in a straightforward way if we apply the compositional semantics to the derivation tree instead of the derived tree. A more complex case will be briefly considered, where two situation adverbs are at issue. Section 5 concludes.

2. Quantification over possible worlds in LTAG

A sentence α denotes a proposition, $[[\alpha]]$, i.e., a function from possible worlds to the truth values in $\{0,1\}$. A way to encode the denoted proposition $[[\alpha]]$ is to equate with a Ty2 formula the conditions under which $[[\alpha]]$ applied to the actual situation s_0 will yield 1. This is done for a simple example in (3), and its LTAG correspondent is given next to it. Note that, in a Ty2 formula (Gallin 1975), all logical predicates (formal translations of nouns, adjectives and verbs) include a situation argument, represented using the variables s_0 (by convention, the actual situation, the evaluation situation of the sentence as a whole), and s, s', s'', etc.

(3) $[[$*Pat visits Kate*$]](s_0) = 1$ iff visit(p,k,s_0)

l_0: visit(x,y,s_0) pat(x) kate(y)
arg: -

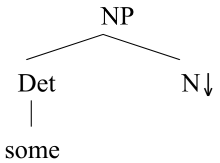
Let us introduce quantificational determiners into this Ty2 intensional framework. Kallmeyer-Joshi propose that the contribution of a quantificational determiner consists of two parts: on the one hand, the quantificational NP adds an argument to the predicate-argument structure of the sentence; on the other, it introduces a scopal element, a logical operator which takes scope over (at least) that predicate-argument structure. For each quantificational determiner, these two components are separated into two trees, as exemplified in (4)-(5): the predicate-argument component substitutes into the appropriate argument slot in the verb's tree, and the scopal component adjoins to the root node of the verb tree.

(4) Some / A:

β_1 S*

l_1: some(x,h_1,h_2) $r_1 \leq h_2$
arg: r_1

α_1



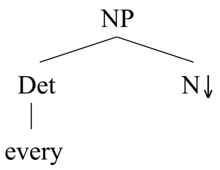
l_2: $p_2(x,s')$ $l_2 \leq h_1$
arg: $\langle p_2, 01\rangle$

(5) Every:

β_2 S*

l_3: every(y,h_3,h_4) $r_2 \leq h_4$
arg: r_2

α_2



l_4: $p_4(y,s'')$ $l_4 \leq h_3$
arg: $\langle p_4, 01\rangle$

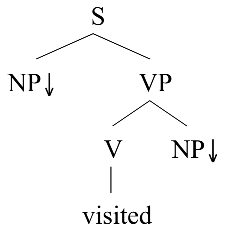
An example sentence is given in (6). The remaining basic trees and their LTAG denotations are spelled out in (7)-(9). Following the derivation tree in (10), these denotations compose to yield the final denotation in (11). Note that the relative scope of two quantifiers *every* and *some* remains underspecified in this output denotation.

The two possible scopal readings are determined by the disambiguation functions δ_1 and δ_2 in (12).[4]

(6) A student visited every club.

(7) Visited:

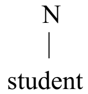
α_0



l_o: visit(x,y,s''')
arg: <x, 00>, <y,011>

(8) Student:

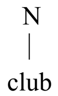
α_3



q_1: student
arg: -

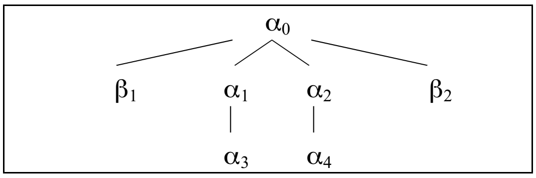
(9) Club:

α_4



q_2: club
arg: -

(10) Derivation tree:



(11) Output denotation:

l_1: some(x,h_1,h_2), l_3: every(y,h_3,h_4), l_o: visit(x,y,s'''), l_2: (q_1: student)(x,s'), l_4: (q_2: club)(y,s'') $l_o \leq h_2$, $l_2 \leq h_1$, $l_0 \leq h_4$, $l_4 \leq h_3$
arg: -

(12) Scope disambiguation functions:

a. δ_1 for the scope *some* >> *every* = { <h_1,l_2>, <h_2,l_3>,<h_3,l_4>,<h_4,l_0>}

b. δ_2 for the scope *every* >> *some* = { <h_1,l_2>, <h_2,l_0>,<h_3,l_4>,<h_4,l_1>}

The semantic representations above are faithful to Kallmeyer-Joshi's proposal expect for the fact that we have introduced a situation variable for each predicate, namely *s'*, *s''*, *s'''*. In this case, since there is no situation binder in the sentence, all three situations variables are identified with the evaluation situation s_0. The result of this variable identification is spelled out in (13).

(13) [[*A student visited every club*]](s_0) = 1 iff $\exists$x [student(x,s_0) $\wedge$ $\forall$y [club(y,s_0) $\rightarrow$ visit(x,y,s_0)]] or $\forall$y [club(y,s_0) $\rightarrow$ $\exists$x [student(x,s_0) $\wedge$ visit(x,y,s_0)]]

l_1: some(x,h_1,h_2), l_3: every(y,h_3,h_4), l_o: visit(x,y,s_0), l_2: (q_1: student)(x,s_0), l_4: (q_2: club)(y,s_0) $l_o \leq h_2$, $l_2 \leq h_1$, $l_0 \leq h_4$, $l_4 \leq h_3$
arg: -

4. I will ignore the semantic contribution of tense throughout the paper.

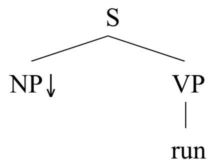
Let us now extend Kallmeyer-Joshi's quantification procedure, already intensionalized, to quantification over situations. In the same way that quantificational NPs introduce quantification over individuals, it has been argued that modal auxiliaries, intensional adverbs and adverbs of quantification quantify over possible situations (Kratzer 1979, 1989, von Fintel 1994). A small difference between the two quantification procedures concerns the so called "restrictor". The restrictor subformula for NPs –e.g., club(y,s_0) in (13)-- originates from the noun inside it, whereas the restrictor for modals is (mostly) contextually given. For example, (14) can be understood as quantifying over deontic situations (roughly, situations where all our actual obligations are fulfilled) or over epistemic situations (roughly, situations such that, as far as the speaker knows, could be the actual situation s_0). These two readings are encoded in the Ty2 formulae (14a)-(14b) by placing the 2-place predicates *Deo* and *Epi* --defined in (15)-- in the restrictor of the situation quantifier.

(14) John must run.
a. $\forall$s' [Deo(s',s_0) $\rightarrow$ run(j, s')]
b. $\forall$s' [Epi(s',s_0) $\rightarrow$ run(j, s')]

(15) a. Deo(s",s') =1 iff s" is a situation accessible from s' such that all the obligations in s' are fulfilled in s".
b. Epi(s",s') =1 iff s" is a situation accessible from s' such that, for all the speaker knows, s' could be s".

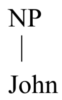
Following Kallmeyer-Joshi's quantification schemata, I propose to implement quantification over situations in LTAG as follows. The semantic contribution of the tree for *run* and the tree for *John* is given in (16)-(17). The double semantic contribution of *must* under its deontic reading is provided in (18). Note that *must* carries universal quantificational force, that is, the scopal part of *must* includes the quantifier every, this time applied to a situation variable *s'*. Other items like *may* or *perhaps*, expressing existential force, would yield a comparable double semantic value. Furthermore, note that, whereas *every* in (5) needs to look for its restrictor in its 01 address, I have spelled out the restrictor of *must* –contextually provided—directly in the predicate-argument part of *must* itself, for simplicity reasons.

(16) Run:
α_1



l_0: run(x,s')
arg: <x, 00>

(17) John:
α_2

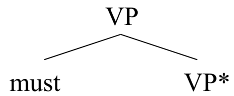


john(x)
arg: -

(18) Must:
β_1 S*

l_1: every(s''',h_1,h_2)
$r_1 \leq h_2$
arg: r_1

β_2

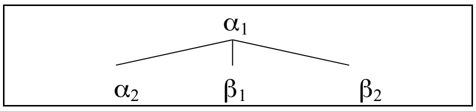


l_2: Deo(s''',s'')
$l_2 \leq h_1$
arg: <s"', 01>

The reader may have noticed a further difference between the predicate-argument value of *must* in (18) and the predicate-argument value of *every* in (5): *must*'s β_2 has an address for one of its internal variables (namely *s"'*, marked in boldface), but *every*'s α_2 does not. This difference stems from the fact that the predicate-argument part of *every* will be substituted into the verb's tree, whereas the predicate-argument part of *must* will be adjoined to the verb's tree. Now, recall that the difference between a substituted element and an adjoined element yields the following effect in (L)TAG. The to-be-adjoined element can identify one or more of the variables within its semantic value with the variables (of the appropriate type) provided at a given address in its tree. But the to-be-substituted element does not have any address in its tree for the tree it is substituting into. Hence, there is no way it can force any such variable identification. For the case at issue, this means that the predicate-argument part of *every* in (5) cannot force identification of *y* or *s"* with any variable in the tree for *visit*. The predicate-argument part of *must* in (18), instead, is allowed to identify *s"'* with a variable provided at the 01 address, that is, with the situation variable provided by the tree of *run*. This difference, as we will see, will play a crucial role to capture the data forthcoming in section 3.

The denotations in (16)-(18) are combined following the derivation tree in (19). The result is (20).

(19) Derivation tree for (14):



(20) [[*John must run*]](s_0) = 1 iff
$\forall$s' [Deo(s',s_0) → run(j, s')]

l_1: every(s''', h_1, h_2) l_2: Deo(s''', s_0), l_0: run(x, s'''), john(x) $l_2 \leq h_1$, $l_0 \leq h_2$
arg: -

Finally, before turning to the data in section 3, let me introduce the double semantic value of two situation adverbs that will be needed later in the paper: *probably* and *sometimes*. The restrictor predicate for *sometimes* is described in (21).

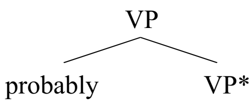
(21) Part(s",s') =1 iff the situation s" is part of the situation s'.

(22) Probably:

β_1 S*

l_1: most(s''', h_1, h_2) $r_1 \leq h_2$
arg: r_1

β_2



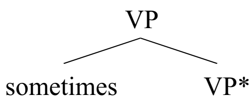
l_2: Epi(s''', s'') $l_2 \leq h_1$
arg: <s"', 01>

(23) Sometimes:

β_3 S*

l_1: every(s''', h_1, h_2) $r_1 \leq h_2$
arg: r_1

β_4



l_2: Part(s''', s'') $l_2 \leq h_1$
arg: <s"', 01>

3. An asymmetry with situation variables: NPs can be transparent or opaque, but VPs must be opaque

Take the sentence in (24) under the reading where *every* has scope inside the *if*-clause. Farkas (1997) –among others-- notes that there is still an ambiguity, rooted on the situation variable that we assign to the complex predicate *poor child*: we may be talking about the set of poor children in the hypothetical situations *s'* introduced by the conditional (opaque reading), or, interestingly, we may interpret it as the set of poor children in the actual situation s_0 (transparent reading). The transparent reading is particularly salient in (25), since (25)'s opaque reading yields a contradiction in the hypothetical situations and, hence, is pragmatically odd.

(24) If you fed every poor child, I would be happy.

a. Opaque NP: In every situation s' accessible to s_0: if you fed in s' all the **poor children in s'**, I am happy in s'.

b. Transparent NP: In every situation s' accessible to s_0: if you fed in s' all the people who are **poor children in the actual situation s_0**, I am happy in s'.

(25) If every poor child was very rich instead, I would be happy
a. # Opaque NP: In every situation s' accessible to s_0: if all the **poor children in s'** are very rich in s', I am happy in s'.
b. Transparent NP: In every situation s' accessible to s_0: if all the people who are **poor children in the actual situation s_0** are very rich in s', I am happy in s'.

The same ambiguity obtains in simpler sentences with modals and intensional adverbs. Take (26)-(27) under the reading where the indefinite determiner scopes under the (deontic) modal *must* (or *should*). Still, the complex predicate *poor child* can be interpreted with respect to the deontic situations *s'* (opaque reading) or with respect to the actual situation s_0 (transparent reading). Again, the transparent reading is particularly clear in (27), since the opaque reading is pragmatically out.

(26) A poor person must / should be fed.
a. Opaque NP: In every situation s' accessible to s_0 where all our obligations in s_0 are fulfilled: a **poor person in s'** in fed in s'.
b. Transparent NP: In every situation s' accessible to s_0 where all our obligations in s_0 are fulfilled: a person who is a **poor person in the actual situation s_0** is fed in s'.

(27) A poor person must / should be rich.
a. # Opaque NP: In every situation s' accessible to s_0 where all our obligations in s_0 are fulfilled: a **poor person in s'** is rich in s'.
b. Transparent NP: In every situation s' accessible to s_0 where all our obligations in s_0 are fulfilled: a person who is a **poor person in the actual situation s_0** is rich in s'.

Percus (2000) adds the interesting observation that a transparent reading of the main predicate in the VP is impossible. That is, even if we give the subject NP scope under the relevant intensional operator and we interpret it opaquely (so that the intensional operator binds at least one situation variable and it does not yield vacuous quantification), the VP predicate cannot be interpreted as transparent:

(24) If every poor child was fed, I would be happy.
c. * Transparent VP (and opaque NP): In every situation s' accessible to s_0: if all the poor children in s' **are fed in s_0**, I am happy in s'.

(25) If every poor child was very rich instead, I would be happy
c. * Transparent VP (and opaque NP): In every situation s' accessible to s_0: if all the poor children in s' **are rich in s_0**, I am happy in s'.

(26) Some poor person must / should be fed.
c. * Transparent VP (and opaque NP): In every situation s' accessible to s_0 where all our obligations in s_0 are fulfilled: a poor person in s' in **fed in s_0**.

(27) Some poor person must / should be rich.
c. * Transparent VP (and opaque NP): In every situation s' accessible to s_0 where all our obligations in s_0 are fulfilled: a poor person in s' in **rich in s_0**.

The readings (24c-27c) are simply impossible. Let us take sentence (26) and judge it in the scenario Σ_{26}. The sentence is judged false. But the reading (26c) is true in this scenario. Since a sentence with reading (26c) should be judged true in the all scenarios that make the reading (26c) is true, and since (26) is not judged true in one such scenario, (26) lacks the reading (26c).

(28) Scenario Σ_{26} for (26) :
In the actual situation s_0, Pat, Lucy, Miguel and nobody else are fed. Our obligation (as evil witches and wizards) is to make at least one of them (any of them) poor. There are no further obligations in s_0. In particular, there is no obligation to feed anybody.

The same reasoning applies to (27) and the scenario Σ_{27}. Sentence (27) is false in Σ_{27}, whereas the reading (27c) is true in Σ_{27}. Hence, sentence (27) lacks reading (27c).

(29) Scenario Σ_{27} for (27) :
In the actual situation s_0, Pat, Lucy, Miguel and nobody else are rich. Our obligation (as evil witches and wizards) is to make at least one of them (any of them) poor. There are no further obligations in s_0. In particular, there is no obligation to make anybody rich.

I leave examples (24)-(25) and the construction of the relevant scenarios as an exercise for the reader. However, before concluding this section, let me illustrate the asymmetry between the situation variables in NPs and VPs with adverbs of quantification as well. The following example, from Percus (2000), has a transparent NP reading ((30a)), but it lacks a transparent VP reading ((30b)). The sentence is judged true in scenario Σ_{30a} –a scenario that makes the transparent NP, opaque VP reading true-- but false in scenario Σ_{30b} –a scenario where the opaque NP, transparent VP reading is true.

(30) The winner sometimes lost.
a. Transparent NP, opaque VP reading: In some (relevant) situations s' that are part of s_0: the **winner in s_0** lost in s'.
b. * Opaque NP, transparent VP reading: In some (relevant) situations s' that are part of s_0: the winner in s' **lost in s_0**.

(31) Scenario Σ_{30} for (30):
We are in a situation s_0 that contains a game among five participants. The game is such that there is exactly one winner of the game and exactly one loser of the game. The other three participants neither win nor lose (e.g., if the winner receives money and the loser pays, the other three participants neither receive nor pay money). The game consists of fifteen rounds (each of which can be considered a natural sub-situation s' of s_0). Each round has exactly one winner and exactly one looser, and, as before, the other three participants of each round neither win nor lose. The winner of the game is the person that wins more rounds, and the loser of the game is the person who loses more rounds. (In case of tie, the relevant participants play until there is no tie.).
a. Σ_{30a}: This time, in situation s_0, Sue, the winner of the game, lost rounds 2 and 3, whereas Mario, the loser of the game, won no round at all.
b. Σ_{30b}: This time, in situation s_0, Sue, the winner of the game, lost no round at all, whereas Mario, the loser of the game, won rounds 6 and 9.

In sum, the question we need to answer is the following: Why is the main predicate in a VP necessarily opaque with respect to the immediate situation operator, whereas predicates embedded in an NP can be interpreted as opaque or transparent?

4. Capturing the asymmetry in LTAG semantics

To rephrase the question in LTAG terms, take the semantic representation in (32). Why is there a choice between l_4: poor-person(x, $\boldsymbol{s_0}$) and l_4: poor-person(x, $\boldsymbol{s'}$), whereas only the opaque situation option l_2: rich(x, $\boldsymbol{s'}$) is available?.

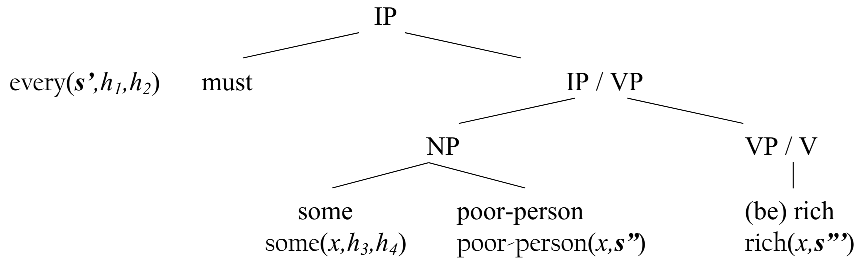
(32) [[*Some poor person must be rich*]](s_0) = 1
iff $\forall$s' [Deo(s',s_0) $\rightarrow$
$\exists$x [poor-person(x, $\mathbf{s_0}$/**s'**) $\wedge$ rich(x,**s'**)]]

δ (for *must*>>*every)* =
{<h_4,h_2>, <h_2,l_3>, <h_1,l_1>, <h_3,l_4>}

l_0: every(s',h_1,h_2) l_3: some(x,h_3,h_4) l_1: Deo(s',s_0), l_2: rich(x, $\boldsymbol{s'}$), l_4: poor-child(x, $\boldsymbol{s_0}$ / $\boldsymbol{s'}$) $l_1 \leq h_1$, $l_2 \leq h_2$, $l_4 \leq h_3$, $l_2 \leq h_4$
arg: -

The question is particularly puzzling for grammars where the compositional semantics is performed on the derived tree. Take a GB Logical Form tree or an LTAG derived tree where the modal *must* takes scope over the determiner *some*. We have assumed, as proven in Gallin (1975), that the expressive power needed to generate intensional readings in natural language amounts to a Ty2 formal language where we have direct quantification over situation (or world) variables. Furthermore, following Percus (2000), every predicate in a sentence is in principle allotted its own situation variable. That yields, roughly, the syntactico-semantic representation in (33). Note that, in the derived tree, the situation operator [[*must*]] combines with the denotation of its sister as a whole. Why should [[*must*]], then, make a distinction between NP situation variables and verbal situation variables if they are all equally available within its sister's denotation? How could we possibly account for the fact that [[*must*]] will necessarily bind *s'''* in rich(*x,s'''*) and will only optionally bind *s''* in poor-person(*x,s''*)?[5]

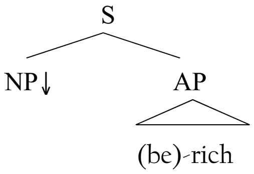
(33) Some poor person must be rich.



If, instead, we perform the semantic computation on the LTAG derivation tree using the proposed denotations, the asymmetry between NPs and main predicates follows straightforwardly from the way the derivation proceeds. Take the denotations below and the derivation tree in (38):

(34) (Be) rich:

α_1



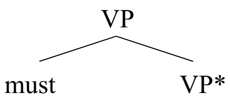
l_0: rich(x,s'')
arg: <x, 00>

(35) Must

β_1 S*

l_1: every(s',h_1,h_2) $r_1 \leq h_2$
arg: r_1

β_2



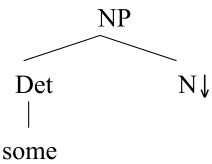
l_2: Deo(s',s) $l_2 \leq h_1$
arg: <s', 01>

(36) Some / A:

β_3 S*

l_3: some($x,h_3,,h_4$) $r_2 \leq h_4$
arg: r_2

α_2

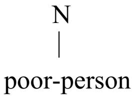


l_4: $p_2(x,s''')$ $l_4 \leq h_3$
arg: <p_2, 01>

5. A reviewer suggested the possibility that situation variables are not indices generated as sister of predicates, but indices introduced by the determiner in the NP. This way, NPs would have a free situation variable that may be optionally bound higher up, whereas the main predicate in a VP would not have a free situation variable at any point (e.g. [[*(be) rich*]] would be λxλs.rich(x,s)). However, being introduced by a determiner is neither a necessary nor a sufficient condition for an NP to be optionally transparent. First, bare plurals can have a transparent reading, as Kratzer's (i) illustratres. Second, NPs with a determiner acting as main predicates in copular sentences cannot be transparent: (ii) lacks the VP transparent reading as much as (25) and (27).
(i) Sue wanted to put belladonna berries in the salad because she mistook them for raspberries.
(ii) If some poor child was the richest child instead, I would be happy.
In fact, Percus (2000), who assumes a GB Logical Form derived tree as the input to the semantics, does not capture the binding asymmetry in the semantics. He proposes, instead, a syntactic constraint on LF, basically a Binding Theory for situation variables.

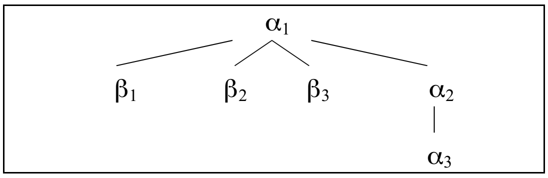
(37) Poor-person:

α_3



q_1: poor-person
arg: -

(38) Derivation tree for *Some poor people must be rich*:



In this derivation tree, β_2 (the semantic part of *must* in charge of identifying the variable *s'* with another situation variable at the address 01) applies to α_1, the main predicate's denotation, and it ensures that the two situation variables are identified. β_2 never applies to the denotation α_2 of the NP *some poor people*, thus it cannot enforce variable identification with it. In fact, given that we follow the derivation tree and not the derived tree, β_2 does not even apply to a semantic object that includes the contribution of α_2. That is, β_2 only finds α_1 at the 01 address, and not α_1 composed with α_2. Hence, the obligatory situation variable identification encoded in the denotation β_2 cannot choose a variable from α_2, but only from the main predicate's denotation α_1 found at the 01 address.

This way, the main predicate in *Some poor people must be rich*, namely *rich*, is necessarily opaque with respect to *must*, whereas no such constraint can be imposed for the predicate *poor-people* buried in the NP. The NP is, hence, free to be interpreted as transparent or as opaque with respect to the modal. If it is transparent, we take the option of identifying its situation variable with the actual situation s_0; if it is opaque, we identify it with the situation variable *s'* introduced by the modal. This derives the existing readings of the sentence represented in (32), and it successfully excludes the non-existing ones.

To conclude this section, I will briefly consider a more complicated case involving two situation quantifiers, about which I will tentatively make some speculations. Recall example (30), repeated below with the added intensional adverb *probably*. Besides the readings discussed in section 3, Percus (2000:214ff) notes that, in examples with two situation operators in a c-command relation, the c-commanding one necessarily binds the second situation variable of the immediately c-commanded one. In our example (39), this means that the second situation variable introduced by *sometimes* cannot be identified with the actual situation s_0, but it has to be identified with the situation variable that *probably* quantifies over, namely *s'*. This is shown in (40): the second situation variable in l_3:Part(*s''*,***s'***) (in boldface) has to be locally bound by the quantifier most introduced by *sometimes*.[6]

(39) The winner probably sometimes lost.

(40) [[*The winner probably sometimes lost*]](s_0) = 1
iff
MOST s' [Epi(s',s_0)]
[∃s'' [Part(s'',**s'**) ∧ ιy.winner(y, s_0/s'/#s'')=x ∧ lost(x,s'')]]

l_0: most(s',h_1,h_2) l_2: some(s'',h_3,h_4) l_1: Epi(s',s_0), l_3: Part(s'',$\boldsymbol{s'}$), l_4: lost(x, s'') l_5: ιy.winner(y, $s_0/s'/\#s''$)=x $l_1 \leq h_1$, $l_4 \leq h_2$, $l_3 \leq h_3$, $l_4 \leq h_4$
arg: -

Using the denotations of *probably* and *sometimes* provided in (22) and (23), this result can be easily achieved if the β_2 denotation of *probably* in (22) adjoins to the β_4 denotation of *sometimes* in (23), and β_4 adjoins to the tree for *lost*. This type of dependent adjunction is defended in Vijay-Shanker 1987. The

6. The NP *the winner* can now be evaluated with respect to any of the tree situations s_0, *s'* and *s''*, though the last one yields a pragmatically odd reading.

mandatory identification obtains straightforwardly. I leave to the reader the compositional semantic computation of this example.

A second possibility, presented in Schabes-Shieber (1991), consists of performing multi-adjunction of both adverbs at the same node of the main predicate's tree. If this syntactic approach is pursued, both the β_2 denotation of *probably* and the β_4 denotation of *sometimes* apply to the meaning of *lost*. The mandatory identification of variables encoded in β_2 and β_4 would then yield the wrong result in (41):

(41) MOST s" [Epi(s",s_0)]
[∃s" [Part(s",s_0/s") ∧ ιy.winner(y, s_0/#s")=x ∧ lost(x,s")]]

l_0: most(*s",h₁,h₂*)
l_2: some(*s",h₃,h₄*)
l_1: Epi(*s",s₀*), l_3: Part(*s", s₀/s'*), l_4: lost(*x, s"*)
l_5: ιy.winner(*y, s₀/#s"*)=x
$l_1 \leq h_1$, $l_4 \leq h_2$, $l_3 \leq h_3$, $l_4 \leq h_4$
arg: -

This is what the wrong result consists of: the variable identification instructions in β_2 and β_4 force both most and some to try to bind the same variable occurrence *s"*. This is not just an empirical wrong result, but an impossible task in Predicate Logic (PrL): one variable occurrence can only be bound by one quantifier. Hence, if this type of multi-adjunction is pursued, perhaps it is possible to ban this result on principled logical grounds, by appealing to a secondary variable identification procedure when the default one cannot be successfully implemented in PrL. I leave the issue open at this point.

5. Conclusions

We have seen that situation variables in NPs and in main predicates behave asymmetrically: NPs can be transparent and opaque with respect to the immediately c-commanding situation operator, whereas main predicates can only be opaque. Following Kallmeyer-Joshi's (2001) quantification procedure, I have proposed a double semantic value for the modal *must*, for the intensional adverb *probably* and for the adverb of quantification *sometimes*. The asymmetry between NPs and main predicates has been shown to follow if we apply the proposed denotations to the derivation tree.

References

Barwise, Jon and John Perry. 1983. *Situations and Attitudes*, Cambridge, Mass.: MIT Press.

Cresswell, M. 1990. *Entities and indices*, Dordrect: Kluwer.

von Fintel, Kai. 1994. *Restrictions on Quantifier Domains*, Amherst, Mass: GLSA.

Farkas, Donca. 1997. Evaluation indices and scope. In: A. Szabolcsi (ed.), *Ways of scope taking*, Drodrecht: Kluwer.

Gallin, Daniel. 1975. *Intensional and Higher-Order Modal Logic: with Application to Montague Semantics*, Oxford: North-Holland.

Kallmeyer, Laura and Aravind Joshi. 2001. Factoring Predicate Argument and Scope Semantics: Underspecified Semantics with LTAG. Penn- Univ. Paris 7 manuscript.

Joshi, Aravind. and K. Vijay-Shanker. 1999. Compositional semantics with LTAG: How much underspecification is necessary? In: Blunt H. C. and E.G.C. Thijsse, eds., *Proceedings of the third international workshop on computational semantics*, Tilburg.

Kratzer, Angelika. 1979. Conditional necessity and possibility. In: Baeuerle, Egli and von Stechow, eds., *Semantics from different points of view*, Berlin: Springer.

Kratzer, Angelika. 1989. An Investigation of the Lumps of Thought. Linguistics and Philosophy 12: 607-653.

Lewis, D. 1973. *Counterfactuals*, Oxford: Blackwell.

Musan, R. 1995. *On the temporal interpretation on noun phrases*, MIT Ph. D. diss.

Percus, O. 2000. Constraints on some other variables in syntax. Natural Language Semantics 8:173-229.

Schabes, Yves and Stuart M. Shieber. 1991. An Alternative Conception of Tree-Adjoining Derivation. Computational Linguistics 20.1: 91-124.

Stalnaker, R. 1968. A theory of conditionals. In: Rescher, ed., *Studies in Logical Theory*, Oxford: Blackwell.

Vijay-Shanker, K. 1987. *A Study of Tree Adjoining Grammars*, University of Pennsylvania Ph. D. dissertation.