Representing the effort in resolving ambiguous scope

Asad Sayeed — Saarland University

Abstract. This work proposes a way to formally model online scope interpretation in terms of recent experimental results. Specifically, it attempts to reconcile underspecified representations of semantic processing with results that show that there are higher-order dependencies between relative quantifier scope orderings that the processor may assert. It proposes a constrained data structure and movement operator that provides just enough specification to allow these higher-order dependencies to be represented. The operation reflects regression probabilities in one of the cited experiments.

Keywords: ambiguous scope, neo-Davidsonian semantics, online sentence processing

1. Introduction

In this paper, I reconcile conflicting factors in the representation of quantifiers and their scopes, particularly in the context of incremental parsing. Recent results in experimental psycholinguistics appear to suggest that higher-order constraints over scope ambiguity resolution seem to operate in the actual behaviour of language users (see section 2.2); these higher-order constraints have often been encoded in the theoretical linguistic literature as restrictions on covert movements, e.g., Quantifier Raising (QR; May, 1985). However, there are clearly pragmatic factors at play in how listeners choose scope order. The apparent complex interaction of these factors calls for a formal approach that accommodates three factors: (1) the incremental construction of the semantic representation, (2) the online pragmatic decision-making capacity of the processor, and (3) the formal/algorithmic constraints on ambiguity resolution. I use the aforementioned recent results to illustrate the challenge, and I present the outlines of a formal approach that hinges around the neo-Davidsonian event variable as the anchor that unifies both the pragmatic and algorithmic components of reanalysis and ambiguity resolution.

1.1. Scope and incrementality

Scope is a property of human language that connects syntax, semantics, and pragmatics, exposing aspects of the interfaces between each of them. The basic phenomenon of scope presents itself as follows: a logical operator that binds some variable within an area of syntactic or semantic structure. Insofar as there are multiple overlapping operators and scopes, there is potential for ambiguity. For example,

(1) Every child climbed a tree.

could mean either that there is a single tree that all the children climbed (inverse scope) or that for each child, there is a tree which that child climbed (linear scope). In this case, it may be that lexical-pragmatic bias about children and tree-climbing prompts a multiple-tree interpretation. However, sometimes these interpretations are constrained by grammatical factors. The sentence

(2) A tree that every child climbed was damaged.

is considerably more constrained to be a single tree, but not because of a "child-climb-tree" lexical-pragmatic relationship.

Ruys and Winter (2011) provide a thorough recent survey of approaches to the question of scope, but most of their examples come from traditional theoretical approaches that deal with scope ambiguity "offline". Offline scope ambiguity retains its ambiguity at the end of the sentence and is ideally tested in the absence of pragmatic bias.

Consider the sentence:

(3) Some woman admires every man.

Both readings (that there is a single woman who admires all the men or that each man has a woman who admires him) are difficult for many English-speakers to disambiguate without more context. In actual interaction, however, ambiguity comes and goes throughout the process.

(4) Some woman $\|_1$ admires every man. $\|_2$ These women $\|_3$ ...

At $\|_1$, the possibility that there could be a set of women involved is established. At $\|_2$, the ambiguity may be fully established, given no other context. At $\|_3$, the inverse interpretation is established. This involves costs that may accrue to cognitive or formal limits on the representation of meaning or to the cost of registering pragmatic or contextual information.

1.2. Movement, compositionality, and incrementality

If we make the assumption that some form of computational tractability must play a role in representing the operation of the human parser, then we would prefer as much as possible to eliminate the role of movement-style operations from the grammar (Kroch and Joshi, 1985). Thus, parsing formalisms rarely include space for QR-style operations. Formalisms that rely on a highly compositional semantics, such as categorial grammar, either simply omit covert operations from their representation or are required to posit highly divergent parallel representations; if there are a large number of scopal items, there may be a proliferation of parallel representations during the parse, to an extent implausible even under parallel architectures of processing, since most parallel parsing approaches use a search space of limited breadth (Staub, 2015)[1].

Formalisms that rely on less aggressively compositional mechanisms, such as neo-Davidsonian semantic formalisms (Parsons, 1990), avoid some of the problems of parallel computation and backtracking, but nevertheless confront the same problem in the incremental context: how to represent the possibilities of scope ambiguity resolution in the semantic output representation. But from where does this conflict ultimately stem?

At root, the problem is that it is challenging to represent the possibility of scope ambiguity in an incremental context, because incrementality by definition forces representations of the input to be only partially available, and yet linguistic constraints on interpretation are sometimes created by objects that are late in appearing. To resolve this conflict, predictive frameworks for syntactic parsing (e.g., Roark et al., 2009) attempt to match human behaviour in experimental settings by employing a form of *underspecification* (e.g., Ebert, 2005). At each step in the parse, the parser posits a structure with constraint-laden placeholders for future structure that may come with words not yet seen in the parse. The quality of the predictions and their match to human behaviour is controlled by fine-tuning the appearance of these placeholders and the costs of satisfying their constraints.

In the remainder of this work, we proceed through some arguments for and against underspecification approaches to scope representation. We describe a couple of recent experimental results that show that there are higher-order constraints on the interpretation of scopal ambiguities that "pure" underspecification grammars cannot represent. Keeping in mind the benefits of underspecification, we then describe an approach to accommodating higher-order scope interpretation effects by admitting a very limited form of movement that pertains to scope relationships – effectively, a type of stripped-down QR. We then return to the psycholinguistic results and describe how our system accommodates those facts.

2. Underspecification: for and against

2.1. For underspecification in scope

Underspecification approaches avoid generating the full listing of possible scope order interpretations until it is actually necessary, instead producing a compact description of the possibilities.

[1]This is not *entirely* salvaged by the idea that most of these readings may be pragmatically excluded. Consider a quantifier arriving late in an pragmatically implausible position. If the parser has already made commitments to a particular derivation, it would require considerable backtracking to return to a more plausible derivation, restating the problem in terms of a covert operation of backtracking.

Given the sentence,

(5) Every child climbed a tree.

we find that there are two ways to interpret the relationship between the existential and universal quantifiers, $\forall > \exists$ and $\exists > \forall$. Without further evidence from the context, we can instead use a placeholder operator to say that there *should* be a dominance relationship between the two, without having to enumerate them all: $\exists \approx \forall$. Koller et al. (2010) show that it is possible to identify a vast number of readings from simple narrative sentences, even if some of these can be identified as semantically equivalent *post hoc* – which they do through the use of an underspecification formalism. They also find that potential ambiguities are relatively common; their annotation effort on the German-language NEGRA corpus finds that 121 of 322 annotated sentences potentially contain a scope relationship.

Underspecification theories of scope treat readings as equivalent unless otherwise required: they contain no default hierarchy. Although (5) seems to suggest that there was a different tree for every child, Dwivedi (2013) suggests that this effect may stem merely from the pragmatics of the situation, because sentences like

(6) Every jeweller appraised a diamond.

do not, experimentally, have so strong a bias, as language users are more willing to believe that there is a single diamond appraised by all the jewellers. Underspecification theories allow us to abstract away from pragmatically-driven aspects of interpretation; all that matters is achieving a compact representation of what is and is not allowed.

Dwivedi (2013) used reading time experiments involving sentences with two quantifiers preceding continuation sentences. She compared conditions under which the first sentence describes a scenario with a strong pragmatic bias vs. when they do not, as in:

(7) a. Every child climbed a/that/those tree(s). The tree(s) was/were in the park.
b. Every jeweler appraised a/that/those diamond(s). The diamond(s) was/were clear and flawless.

Dwivedi found that sentence pairs such as that in (7-a) take less time to read when left scopally ambiguous (with *a*), while scope ambiguity has no effect on the reading time of the second sentence. On the other hand, in a question-answer task about the number of trees, she found that the singular variant of the second sentence with a scopally ambiguous first sentence produces chance accuracy rates (subjects disprefer the inverse interpretation that there is a single tree that all children climbed).

In sentence pairs such as (7-b), she finds the same effect for the first sentence, with unambiguous scope taking longer to read. On the other hand, the second sentence takes longer to read when the first sentence is scopally ambiguous. That is, the reduced lexical-pragmatic bias of the verb requires the processor to acknowledge the specification of number when the continuation sentence is given (subjects update their expectations of number). However, many subjects once again have difficulty when a question-answer task is used to force an inverse reading; it is possible, but dispreferred.

On the face of it, we can interpret this as strong evidence for a split system in scope processing: one in which there is a conceptual level that is specified only insofar as there is previous lexical bias on the verb; otherwise, underspecification applies until further information updates the scope expectations at this conceptual level. Then there is an algorithmic level which remains underspecified until a reading is forced, and this again costs some effort.

2.2. Against underspecification in scope

But while it appears that underspecification is present in the grammar, the extent to which underspecification applies is a matter of debate. Dwivedi's experiment involved sentences with two quantifiers. Dotlačil and Brasoveanu (2015) point out that it is difficult to draw distinctions between theories when the evidence involves only two quantifiers. We can use the influence of lexical-pragmatic bias in Dwivedi's result as an example. There are only three possible combinations, $\forall > \exists$, $\exists > \forall$, and $\exists \approx \forall$, the latter being the "default" under an underspecification story. Lexical-pragmatic information can force a specification, one that is complete for the entire sentence. But given a third quantifier, there remain unspecified possibilities. Under a "pure" underspecification story, the relationship of the third quantifier can remain unresolved indefinitely. But does it? Dotlačil and Brasoveanu (2015) experimented with adding a third quantifier, and their reading-time results show that there are higher-order relationships between quantifier specifications that cannot really be accommodated in a pure underspecification framework.

Dotlačil and Brasoveanu (2015) tested sentences like these on adult speakers:

(8) A caregiver (x) comforted a child (y) every night (n).
  a. The caregivers wanted the children to get some rest. ($\forall n > \exists x \exists y$)
  b. The caregivers wanted the child to get some rest. ($\forall n > \exists x$)
  c. The caregiver wanted the children to get some rest. ($\forall n > \exists y$)
  d. The caregiver wanted the child to get some rest. ($\exists x \exists y > \forall n$)

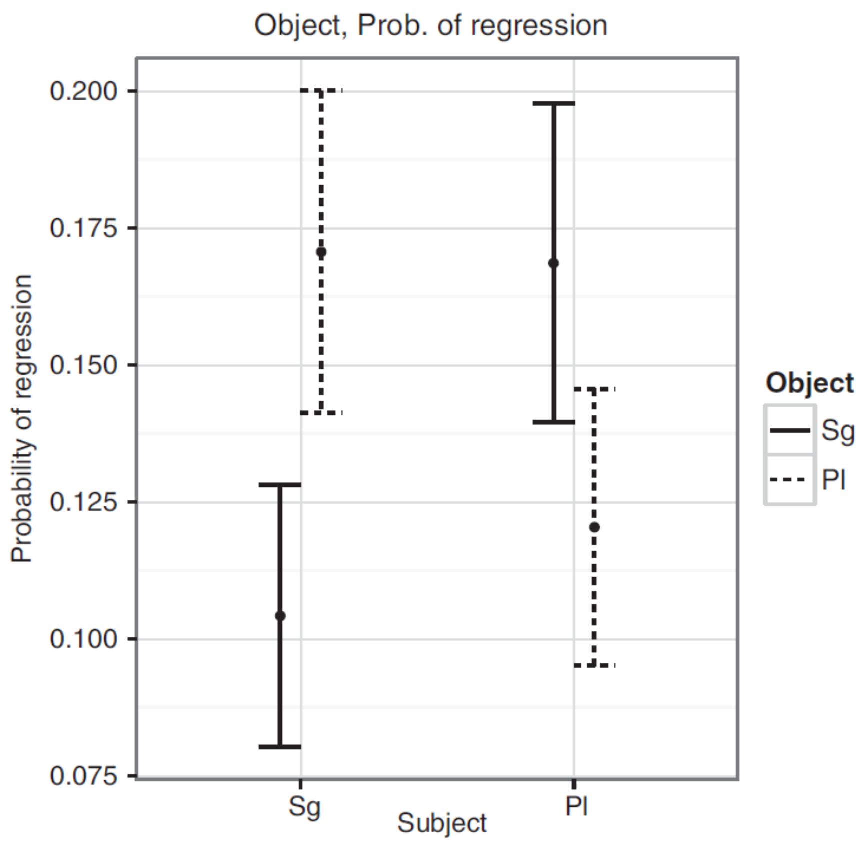


Figure 1: Probability of regression at the object in the continuation sentence. The result for re-reading probability is similar. Result from Dotlačil and Brasoveanu (2015).

In an eye-tracking setting, Dotlačil and Brasoveanu presented a sentence like (8) to a given subject, and then one continuation from (8-a)-(8-d). The crucial details of their result are in Figure 1, which shows the probability of regression for the object in the continuation sentence.

In summary, they found that in (8-a), there is a facilitating effect of the plural reading of "caregiver" on the plural reading of "child". The presence of a singular reading of "child" after a plural reading of "caregiver" (8-b), on the other hand, forces regressions and re-readings. In a purely underspecified framework, the readings of "caregiver" and "child" should be independent of one another; that they are dependent implies that there is a default structure already posited by the parser, that is defeated by the forced raising of "every night" on encountering the plural. On the other hand (8-d) is a kind of "baseline" scenario, in which "shallow" processing creates the linear order, and no covert operations are required. Finally, (8-c) is the complete leftward raising of "every night", leading to plural readings for both "caregiver" and "children"; the difference between it and the both-singular construction is not large.

Here, apparently *contra* Dwivedi, we have a result that does require a “higher-order” dependency between scope constraints, one revealed by the presence of the third quantifier. In both cases, however, the distinction is largely detected between the first and second sentences. Thus, it is not completely correct to say that the results contradict one another; without the third quantifier, the result from Dwivedi can be subsumed by that from Dotlačil and Brasoveanu, as there is no possibility of a higher-order dependency.

How do we draw a line in order to define exactly *how much* underspecification we need? An experiment by Radó and Bott (2011) may be useful in this case. They tested German sentences of the form :

(9) Genau ein Affe ist auf allen/jeder Karte(n) zu finden.
Exactly one monkey is on all/each card(s) to find.

They used self-paced reading followed by a display of cards with sets of images that may or may not contain a progression of monkeys; each card was revealed one-by-one in the same manner as the self-paced reading, and subjects were solicited to respond whether the statement has been proven true or false by the cards displayed so far, or whether they need more information (by revealing more cards). Compared with a control sentence with a single quantifier, subjects tended to have higher response times on the very first picture card, suggesting that the scopes were already fully specified by the time the card was read; the doubly-quantified sentence required the subject to examine the entire card, in order to confirm the truth of *exactly one*. On the other hand, Radó and Bott tested inverse linking versions of (9) (*Exactly one monkey on all/each card(s)...*) and found that there was no reading slowdown at *all/each*, while there was when the verb stood between the quantifiers, suggesting that the verb creates a minimal domain in which scope is computed when the second quantifier is seen.

3. Scope trees

Putting Dwivedi (2013), Dotlačil and Brasoveanu (2015), and Radó and Bott (2011) together, we see evidence for a model that does deep, pragmatically-influenced processing of scopes, but only at the completion of some form of “minimal scope domain”. Thereafter the processing is potentially subject to higher-order algorithmic constraints that prevent an analysis that is fully underspecified, only positing constraints whenever there is direct evidence in the string.

Other evidence for the importance of processing domain in scope interpretation includes Syrett and Lidz (2011) who find that children and some adults do not respect a tensed clause barrier in antecedent-contained deletion (ACD) interpretation; they suggest that online processing capacity affects QR-constraining ability. Specifically, they test ACD sentences of the form:

(10) Miss Piggy wanted to drive every car that Kermit did.

Most adults take this sentence to imply that for every car that Kermit must have driven, Miss Piggy must have wanted to drive that car. That is to say, the quantifier in the deleted portion of the sentence remains within the scope of the "drive"-clause, and the deleted portion is the infinite regression "…drive every car that Kermit did drive every car that Kermit did …". However, many more children than adults take the quantifier to raise to the matrix portion, implying that for every car Miss Piggy wanted to drive, Kermit wanted to drive that car too: "…want to drive every car that Miss Piggy did want to drive every car…".

Syrett and Lidz suggest that this may have to do with a reduced ability in children to distinguish between the matrix and embedded VPs, so that children more often resolve the ambiguity by raising the quantifier to the "wrong" VP in ellipsis resolution.

Sayeed and Demberg (2013b) propose an approach to the joint incremental representation of syntactic and semantic processing that allows for maximum underspecification at the level of predicate calculus. This TAG-based syntactic formalism makes use of the neo-Davidsonian event variable as a formal device that provides a great deal of representational flexibility. It allows the output semantic expression to grow mostly rightwards:

(11) a. A caregiver comforted …
b. $\exists x \mathsf{caregiver}(x) \wedge \exists e \mathsf{comfort}(e) \wedge \mathsf{agent}(x, e)$
c. A caregiver comforted a child.
d. $\exists x \mathsf{caregiver}(x) \wedge \exists e \mathsf{comfort}(e) \wedge \mathsf{agent}(x, e) \wedge \exists y \mathsf{child}(y) \wedge \mathsf{patient}(y, e)$

Sayeed and Demberg (2013a) then propose a system that represents ambiguous variable scopes without having to resort to inference rules that require the direct editing of the semantic representation. They do this by proposing a parallel structure called a "variable scope tree" (VST), in which strictly the participants in covert operations (event and entity variables) are contained in relations analogous to a syntactic tree. Then QR-style restrictions can be imposed over an operation called VST-move, which uses the event variable as a ceiling over QR.

3.1. Defining the VST system

The variable scope trees (VSTs) contain three types of nodes, event nodes, entity nodes, and traces. The event node simply contains the event variable. The entity node contains an entity variable along with a quantification[2]. Event and entity nodes can have child nodes. Event nodes can have entity nodes or other events nodes as children. When an entity node is a child of an event node, it normally represents that the entity variable is fulfilling a semantic role in the event. Some events assign roles to other events, so an event node can be a child of an event node[3].

Trace nodes are coindexed with entity (or event) nodes in a manner familiar to movement theories of syntax. Trace nodes are generated only at the application of the “VST-move” operation. In the VST formalism, traces are currently used in the representation of the history of the derivation.

C-command is the principle means by which a VST is interpreted. The order of sisters under a parent node does not matter. When a node bearing a scope operator c-commands another node, it takes scope over it. Traces, as above, are currently only formal entities and are not subject to scope.

VST-move is also relatively familiar to recent movement theories of syntax. VST-move targets a node other than a root node, detaches it, replaces it with a coindexed trace, and makes the node a sister of an ancestor node. The ancestor node is copied to become its own parent as well as the parent of the reattached node. VSTs are not necessarily binary-branching, but the result of VST-move is a binary branching node.

VST-move with events and entities is limited by a ceiling. Specifically, nodes can only move to the most immediately containing event. This can be voided if there is some kind of semantic identity or overlap between two events. However, nodes can also only move to the root node. Together, these constraints have an effect of defining an equivalent to the Phase Impenetrability Condition. In other words, in the VST system, the event variable functions as a kind of minimal domain. Results such as Syrett and Lidz can be explained by memory constraints “blurring” event variables together, creating escape-hatches for otherwise illicit raising.

3.2. Online VST construction

How do we construct a VST? We describe this in terms of improvements we now propose to Sayeed and Demberg’s system that enhance the generally rightward expansion of semantic expressions under parsing while allowing us to account for observations we have so far described.

In keeping with the incremental aims of this formalism, construction of a VST happens in parallel to the syntactic parsing procedure. A compatible incremental syntactic parser should generate one

[2]For now, I am restricting this to quantificational noun phrases; other kinds of scope-bearing elements may introduce other types of variables, such as, for example, discourses and situations.

[3]An event node can be the child of an entity node in the case of a relative clause, a condition we leave for future work.

or more neo-Davidsonian terms with every word processed. The terms are processed as soon as they arrive and are used to expand the VST. These terms are usually connected by conjunctions or implications, depending on the introduction of universal or existential quantifiers (and nuclear and restriction scope). Because we use the VST to handle scope relations, we replace all logical operators between terms with a generic connective operator $\bullet$.

An initial "root" event is assumed. Whenever a term representing a predicate contains a binary relation that mentions a variable ready in the VST, the lowest node representing the variable is expanded with a copy of that variable as the first child and the unmentioned variable as its sister. In other words, if event e is already in the VST, the term $\mathsf{Role}(e, x)$ is sent by the parser, and x is already bound by the universal quantifier, then the lowest node mentioning e is expanded to have children e and $\forall x$. Event variable expansion pushes the existential event quantifier to the lower variable in compliance with Champollion (2011), in which the event variable's quantifier normally takes the lowest scope in the event.

I now provide an example of the incremental construction of a VST using the sentence in (8). At the beginning of the parse, we have:

(12) a. $\|$ *A caregiver comforted a child every night.*
b. **Semantic expression**: $\emptyset$ (empty expression)
c. **VST**: $\exists\mathbf{e}$

We mark the variables introduced via semantic output expression terms with in italics and the variable to be expanded in the next step with bold.

Now we process the first word, which gets us only one term.

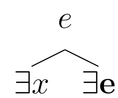
(13) a. A $\|$ *caregiver comforted a child every night.*
b. **Semantic expression**: $\mathsf{agent}(e, x)$
c. **VST**:

$$\begin{array}{c} e \\ \exists x \quad \exists\mathbf{e} \end{array}$$

The semantic expression does not contain the quantifiers. Instead, these are mentioned strictly in the tree, allowing the VST to be the sole representation of scope. As described above, the existential quantifier on e is lowered.

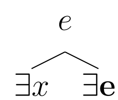
(14) a. A caregiver $\|$ *comforted a child every night.*
b. **Semantic expression**: $\mathsf{agent}(e, x) \bullet \mathsf{caregiver}(x)$

c. **VST**:



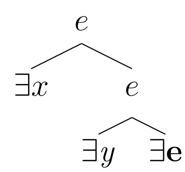
The introduction of “caregiver” gives us no additional information as to the variables, so it only introduces a term.

(15) a. A caregiver comforted || *a child every night.*
b. **Semantic expression**: $\mathsf{agent}(e, x) \bullet \mathsf{caregiver}(x) \bullet \mathsf{comforted}(e)$
c. **VST**:



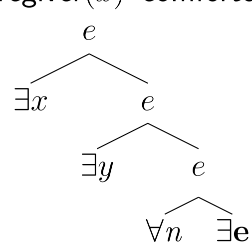
“Comforted” also produces nothing new in the VST, as no new variables are introduced.

(16) a. A caregiver comforted a || *child every night.*
b. **Semantic expression**: $\mathsf{agent}(e, x) \bullet \mathsf{caregiver}(x) \bullet \mathsf{comforted}(e) \bullet \mathsf{patient}(e, y)$
c. **VST**:



Once we have the second determiner, we obtain a new variable and expand the event node once again. “Child” will include no new variable information, so I will skip over that step for the purposes of explanation. Instead, the arrival of “every” introduces a new role-filler, but without telling us the role.

(17) a. A caregiver comforted a child every || *night.*
b. **Semantic expression**: $\mathsf{agent}(e, x) \bullet \mathsf{caregiver}(x) \bullet \mathsf{comforted}(e) \bullet \mathsf{patient}(e, y) \bullet \mathsf{child}(y) \bullet \_\_(e, n)$
c. **VST**:



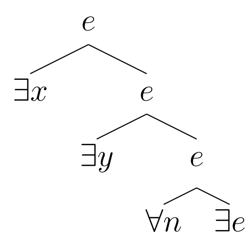
A properly incremental semantic parser would be aware that a role is upcoming without actually knowing what role is heralded by the appearance of “every”. Consequently, we use __ as a placeholder.

Finally, the expression is completed with the arrival of “night”. The VST is already complete, but “night” specifies the role of time/occurrence.

(18) a. A caregiver comforted a child every night. ||

b. **Semantic expression**: $\mathsf{agent}(e, x) \bullet \mathsf{caregiver}(x) \bullet \mathsf{comforted}(e) \bullet \mathsf{patient}(e, y) \bullet \mathsf{child}(y) \bullet \mathsf{OCCUR}(e, n) \bullet \mathsf{night}(n)$

c. **VST**:



At each point in the parse, it is possible to apply VST-move to obtain alternative scope interpretations based on the demands of pragmatics, when enough variables are available. These are not obligatory; these are optional and are thus adaptable to experimental results in incremental scope resolution. Nevertheless, these are highly constrained, as not all possible movements are available (permitting, among other things, the development of tractable probabilistic models of scope resolution).

4. Accounting for reanalysis under specification

Now I will accommodate the result of Dotlačil and Brasoveanu (2015). We thus need to include “every night” as in (8) in the expression in (11-d). Without the VST system, we could insert the universal quantifier above the event, meaning that the nights scope over the event, as expected. However, our VST-less incremental parse, having joined all the other terms in the order in which they appeared, has “child” entity variable y scoping under the event.

(19) $\exists x\, \mathsf{caregiver}(x) \wedge \forall n\, \mathsf{night}(n) \rightarrow \exists e\, \mathsf{comfort}(e) \wedge \mathsf{OCCUR}(n, e) \wedge \mathsf{agent}(x, e) \wedge \exists y\, \mathsf{child}(y) \wedge \mathsf{patient}(y, e)$

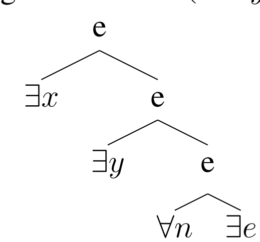
This would be acceptable when there are only existential quantifiers, as they are all logically interchangeable in scope. However, this late insertion of “night” also forces an incorrect default scope order, as well as requiring complex inference rules; the baseline order should not have a distributive meaning of “comforted” over “child”. Instead, we take seriously the idea that the scopes are only computed when the event domain is complete. Then we no longer need the quantifiers to be mixed in among the predicates and can hold these bindings entirely in the VST. This has the side-benefit of eliminating late leftward insertion:

(20) $\mathsf{caregiver}(x^{\exists}) \bullet \mathsf{comfort}(e^{\exists}) \bullet \mathsf{agent}(x,e) \bullet \mathsf{child}(y^{\exists}) \bullet \mathsf{patient}(y,e) \bullet \mathsf{night}(n^{\forall}) \bullet \mathsf{OCCUR}(n,e)$

where $\bullet \in \{\wedge, \rightarrow\}$, to be left underspecified until a final interpretation is selected based on the quantifier order.

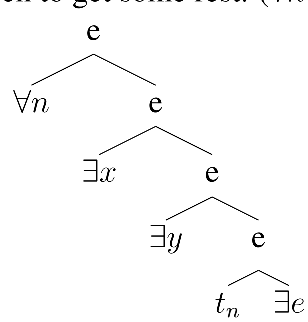
The initial state of the VST at the end of the sentence is in (18-c). This corresponds to the order in (8-d). I repeat these here:

(21) a. The caregiver wanted the child to get some rest. ($\exists x \exists y > \forall n$)
b.



This is the baseline in Dotlačil and Brasoveanu (2015). Reaching the other interpretations provoked by the continuation sentences requires the operation *VST-move*. Application of VST-move to the *every night* variable corresponds to the following:

(22) a. The caregivers wanted the children to get some rest. ($\forall n > \exists x \exists y$)
b.

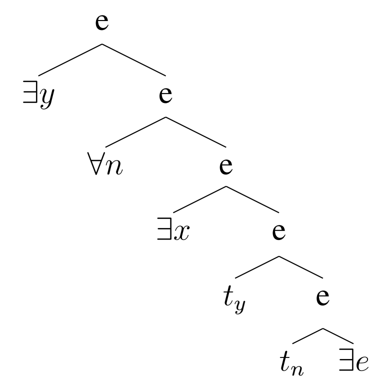


This is the dual plural reading, which has only a slightly increased probability of regression, due to the facilitation effect found by Dotlačil and Brasoveanu.

The processor VST-moves n to the highest position, so it already knows that not only one but both must have a plural reading. The more difficult readings are the plural-single readings, which are reached *after* the first VST-move by a single additional VST-move:

(23) a. The caregivers wanted the child to get some rest. ($\forall n > \exists x$)

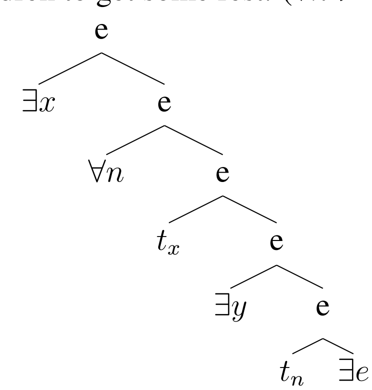
b.



We can accommodate the plural-single reading using a similar mechanism (it has the same regression probability):

(24) a. The caregiver wanted the children to get some rest. ($\forall n > \exists y$)

b.



Both of these last cases are derived from the plural-plural reading – and they are both more difficult.

Repopulating the expression in (20) with quantifiers and logical operators is straightforward and can be done as necessary. This structure represents a limited degree of underspecification, without requiring the semantics to parallel a full bottom-up syntax, while leaving a structure in which the experimentally-observed reanalysis takes place.

4.1. Discouraging infinite movement

I introduce one additional behaviour of VST-move in order to discourage infinite movement, since our current definition of VST-move currently has no restriction other than the "ceiling" of an event domain; variables can be disconnected and reattached at will. I thus add the constraint that the processor disprefers achieving the same scope configuration twice.

Radó and Bott (2011) find that constructions as in (9) are strongly biased, in a judgement study, to an inverse scope reading, even though the linear scope reading remains possible for German speakers. This is reflected in their online disambiguation study, wherein subjects usually rejected the sentence early during the card sequence, if the sequence guided them to a linear scope reading. Given subjects' tendency to compute the plausible scope after the minimal domain is reached, this is consistent with a story in which the inverse scope is computed with an immediate VST-move, but then subjects resist being guided back to the original relative scope configuration. Repeating (9) here with a semantic expression:

(25) a. Genau ein Affe ist auf allen/jeder Karte(n) zu finden.
Exactly one monkey is on all/each card(s) to find.
b. $\mathsf{monkey}(m) \bullet \mathsf{subject}(e, m) \bullet \mathsf{location}(e, c) \bullet \mathsf{card}(c) \bullet \mathsf{find}(e)$

Which yields the following initial VST, which is essentially complete when "cards/Karten" is reached, since the rest of the sentence yields no additional variables (the root event e has already been inferred):

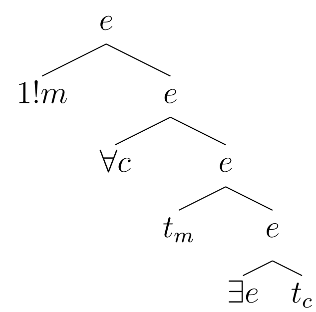
(26)
$$[_{e}\ 1!m\ [_{e}\ \exists e\ \ \forall c\ ]]$$

But the pragmatics seem to demand that the cards scope over the monkey. So the processor raises the scope of c:

(27)
$$[_{e}\ \forall c\ [_{e}\ 1!m\ [_{e}\ \exists e\ \ t_c\ ]]]$$

When the evidence actually forces the linear reading, the processor must raise the "exactly one monkey".

(28)



The bottom of the tree now only contains the existentially quantified event "head" variable $\exists e$ and traces. When a configuration like this exists, we see that the remainder of the tree looks like an image of (26) – that is, it has been returned to an initial state, given that sister nodes in the VST formalism are unordered. The processor highly disfavors creating a structure that contains an image of the initial VST, and so subjects tend to reject the linear interpretation. Another way of looking at this is to use the traces: at least one entity variable should not have produced a trace of VST-move for the scope structure under a given event to remain acceptable.

We would then predict that if we were to do an experiment similar to Dwivedi (2013), but with sentences that were highly lexically biased towards inverse scope, we would also see that continuation sentences that forced a linear reading would produce a significant slowdown. In other words, backing out of an already-inverse reading would be costly.

5. Conclusions and future work

In this work, I described some underlying challenges in accommodating ambiguous scope resolution in a formal incremental framework. I then combined some recent results in scope processing in order to define a model in which there are default scopes, but they are constrained by higher-order dependencies which show up as priming behaviour experimentally. Variable scope trees and VST-move allow for the highly constrained representation of possible scope configurations in a manner that replicates observations about the effort in updating scope representations; however, they are flexible enough to accommodate some variation in the underlying theory of scope processing.

There are many avenues for future work, both experimental, formal, and computational. For example, it would be possible test this system against observations about ACD and other long-distance scopal phenomena as well as to test it against scope interactions at levels other than quantifiers (e.g., negation). The latter requires a more fine-grained formal treatment of events and, potentially, discourses. This system also makes experimental predictions about the effort in scope processing, such as in reversing an already inverted scope.

One major advantage of a system like this is that the constraints imply a small derivational "horizon" at each step. That is, the number of possible VST-moves is limited both at each step in processing (since the system accommodates the possibility of pragmatically-driven ambiguity resolution before the end of the parse, if necessary) and at each step of pragmatic interpretation. Keeping a partially-underspecified representation of scope relations separate from the predicate logic is thus a further step towards constructing tractable probabilistic representations of scope ambiguity resolution and brings linguistic theory and formalism closer to computational applications.

References

Champollion, L. (2011). Quantification and negation in event semantics. *Baltic International Yearbook of Cognition, Logic, and Communication*, 6(1):1–23.

Dotlačil, J. and Brasoveanu, A. (2015). The manner and time course of updating quantifier scope representations in discourse. *Language, Cognition and Neuroscience*, 30(3):305–323.

Dwivedi, V. D. (2013). Interpreting quantifier scope ambiguity: Evidence of heuristic first, algorithmic second processing. *PloS one*, 8(11):e81461.

Ebert, C. (2005). *Formal investigations of underspecified representations*. PhD thesis, King's College London.

Koller, A., Thater, S., and Pinkal, M. (2010). Scope underspecification with tree descriptions: Theory and practice. In Crocker, M. W. and Siekmann, J., editors, *Resource-Adaptive Cognitive Processes*, Cognitive Technologies, pages 337–364. Springer.

Kroch, A. S. and Joshi, A. K. (1985). The linguistic relevance of tree adjoining grammar. Technical report, University of Pennsylvania.

May, R. (1985). *Logical Form: Its Structure and Derivation*. MIT Press.

Parsons, T. (1990). *Events in the semantics of English*. MIT Press, Cambridge, MA, USA.

Radó, J. and Bott, O. (2011). Underspecified representations of scope ambiguity? In *Logic, Language, and Meaning: 18th Amsterdam Colloquium*, pages 180–189, Amsterdam. Springer.

Roark, B., Bachrach, A., Cardenas, C., and Pallier, C. (2009). Deriving lexical and syntactic expectation-based measures for psycholinguistic modeling via incremental top-down parsing. In *Proc. of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 324–333, Singapore. Association for Computational Linguistics.

Ruys, E. and Winter, Y. (2011). Quantifier scope in formal linguistics. In *Handbook of philosophical logic*, pages 159–225. Springer.

Sayeed, A. and Demberg, V. (2013a). Covert without overt: QR for movementless parsing frameworks. In *Generative Linguistics in the Old World 36*.

Sayeed, A. and Demberg, V. (2013b). The semantic augmentation of a psycholinguistically-motivated syntactic formalism. In *Proceedings of the Fourth Annual Workshop on Cognitive Modeling and Computational Linguistics (CMCL). Association for Computational Linguistics, Sofia, Bulgaria*, pages 57–65.

Staub, A. (2015). Reading sentences: Syntactic parsing and semantic interpretation. In Pollatsek, A. and Treiman, R., editors, *The Oxford Handbook of Reading*, pages 202–216. Oxford University Press.

Syrett, K. and Lidz, J. (2011). Competence, performance, and the locality of quantifier raising: Evidence from 4-year-old children. *Linguistic Inquiry*, 42(2):305–337.