Is Coordination Quantification?

Kevin Lerman⋆ **and Owen Rambow**†
⋆ Department of Computer Science, † Center for Computational Learning Systems
Columbia University
New York, NY, USA
klerman@cs.columbia.edu, rambow@ccls.columbia.edu

Abstract

We explore the semantics of conjunction using a neo-Davidsonian semantics expressed in a synchronous grammar. We propose to model conjunction as quantification over the set of conjoined entities, and discuss problems that arise in this approach when we have conjoined quantified noun phrases.

1 Introduction

The semantics of conjunction in natural language has proven particularly difficult for formal linguistic theories to model. We present some preliminary work on this problem using synchronous grammars, specifically the SynchUVGDL formalism. We observe similarities between quantification and coordination, and therefore attempt to model the latter as the former. While this reduces the complexity of many simple NP-coordinated sentences, those with quantified NPs prove difficult to model due to the multi-component nature of quantifiers in semantics. We describe our attempts at adopting an underspecified, single-component quantifier to get around this problem, and present the implications of such a representation.

The paper is structured as follows. We first review work in semantics and Tree Adjoining Grammar (TAG), and then briefly present synchronous UVGDL (Section 3). In Section 4, we summarize our approach to modeling semantics using SynchUVGDL, and then discuss the similarities between quantification and conjunction in Section 5 and propose a simple approach. Section 6 presents a serious problem for the proposed approach (the conjunction of quantified NPs), and Section 7 presents our solution.

2 TAG and Semantics

Shieber and Schabes (1990) were the first to propose a syntactic-semantic grammar in the TAG framework, by using synchronous TAG (SynchTAG). In SynchTAG, two TAGs are linked in such a way that trees in one grammar correspond to trees in the other grammar, and the nodes in corresponding trees are linked. When we substitute or adjoin a tree in a node, then we must substitute or adjoin a corresponding tree in the linked node in the tree in the other grammar. Several different definitions of SynchTAG are possible (Shieber, 1994), and the most interesting definition has the property that the derivation trees for the two derivations in the two synchronized grammars are isomorphic, so that we can talk of a single, TAG-style derivation tree for a SynchTAG. Subsequently, a series of research was published which did not use SynchTAG for semantics, but instead generated a semantic representation during the syntactic derivation, often using feature structures (Kallmeyer, 2002; Kallmeyer and Joshi, 2003; Gardent and Kallmeyer, 2003; M. Romero, 2004). The principal difference is that in this line of work, the semantics is not itself modeled in a TAG.

Recently, Nesson and Shieber (2006; 2007) have revived the approach using SynchTAG. They have shown that a large number of different constructions can be given elegant analyses using a SynchTAG-based analysis. Our work is in this tradition.

3 UVGDL and SynchUVGDL

If TAG can be seen as a partial derivation in a CFG, pre-assembled for convenience, in a UVGDL (Rambow, 1994) the partial derivation is collected into one kit, but not actually fully assembled. More technically, in UVGDL, the elementary structures of the grammar are sets of context-free rules which can be augmented with **dominance links**. A dominance link stipulates a relation of (immediate or non-immediate) dominance which must hold in the derived tree structure. Note that despite the formal differences between a TAG and a UVGDL, they share exactly the same notion of extended domain of locality, and both formalisms can be lexicalized; linguistically, an elementary structure in a UVGDL can be used to represent, as in TAG, a lexical head, its (extended) projection, and positions for its arguments. Rambow (1994) shows that the parsing problem (with a fixed grammar) is polynomial in the length of the input sentence, if the UVGDL is lexicalized (as we assume all our grammars are). This formalism can also be seen as a tree description language, with the context-free rules in a set as statements of immediate dominance between one node and one or more daughters (along with constraints on linear precedence among the daughters), the dominance links as statements of dominance (Vijay-Shanker, 1992; Rambow et al., 2001). We choose the rewriting formulation because a synchronous version (SynchUVGDL) was defined by Rambow and Satta (1996) and some initial results on computation were proposed. Specifically, they claim that the parse-to-forest translation problem for a lexicalized SynchUVGDL can be computed in polynomial time.

4 Overview of Semantics with SynchUVGDL

We have been developing a semantic formalism that can be easily modeled using SynchUVGDL (see (Lerman and Rambow, 2008) for details). We adopt the notation of Montague semantics, wherein e is the type of entities, t' is the type of truth values, and for any two types x and y, $< x, y >$ is the type of functions from x's to y's. Our formalism relies heavily on set theory, for example explicitly interpreting $< e, t >$'s as sets rather than as propositions. For instance, Montague semantics might represent *boy* as "boy(x)", a function whose value is true or false depending on whether or not x is a boy. We instead represent *boy* as simply the set of all boys. While this does not alter any truth conditions, the representation is more convenient for this sort of tree-based formalism. This is because at times a traditional Montague $< e, t >$ such as *boy* must be treated as a function (taking an entity as an argument and returning true iff it is a boy), and at other times it must be treated as a first-order object (when being modified by an adjective such as *tall* to produce a new $< e, t >$ representing "tall-boy"). In certain other formalisms it is easy to underspecify these two cases, but in ours the two views will be directly manifested in the structure of the productions for a word such as *boy* – it either will have a substitution node for an entity and itself be an argument of type t, or it will have no substitution nodes and itself be an argument of type $< e, t >$. Because these two views have different structural representations, we are forced to choose which to use, and generally choose the set representation as being easier to work with under the UVGDL formalism.

Adjectives are then analyzed as functions from sets of entities to "filtered" sets of entities. By way of example, *tall boy* would be represented as "tall(boy)" – "tall()" takes the set of all boys, filters it for those who are tall, and returns a new set consisting of all boys who pass the filter.[1] Note that while we often call these "filters", they are opaque enough to support the semantics of nonintersective adjectives such as *fake*, which would perhaps take a set of entities and yield a new set that contains all entities that appear similar to elements of the argument set, but that nonetheless are not members of the argument set.

The piecewise nature of UVGDL lends itself to neo-Davidsonian semantics, wherein the events denoted by verbs (or possibly by nouns) are treated as first-order objects, akin to entities (we assign them the type of v).[2] Thus, our analysis of verbs and adverbs is nearly identical to that of nouns and adjectives – a verb like *visit* returns the set of all "visit" events, and an adverb like *quickly* would take a set of events (a $< v, t >$) and filter the set for those which were done happily. This is logically equivalent to something like $visit(x) \wedge quickly(x)$, but has more flexibility in representing the meaning of manner adverbs (what is quick

[1]Technically, *boy* should be analyzed this way too, as "boy(U)" – a function that takes a set of entities (here, the [U]niverse of all entities) and filters the set for those entities which are boys. In this paper, we will represent this as simply 'boy' for easier reading.

[2]For a detailed discussion of neo-Davidsonian semantics, see (Schein, 2002). The idea is not new; our goal has been to integrate it into a robust syntactic formalism.

depends on the event – Mary quickly batting an eye is different from humanoids quickly spreading through Asia).

Slightly more complicated filters exist to handle verb arguments. For instance, subjecthood is represented as a filter that selects only events which have a certain entity as their agent (or whatever the semantic role of the subject is for the verb). The filter actually comes in two parts: one part that filters a set of events for those that make a certain condition true, and another that specifies that the condition consists of having a certain entity as an agent (see the top left structure in Figure 1). This distinction will become important in the next section. Finally, to preserve the notion that statements are formulas that ultimately resolve to a truth type, an existential quantifier dominates all logic relating to the verb. Thus, the semantics for a statement like *John visited Mary* reads as "There exists an event in the set of (visit events whose subject is John and whose object is Mary)." This matches intuition, as the utterance does not imply any additional detail about the nature of the event being described – just that some event matching this description happened.[3]

Using this formalism, we can with a simple toy grammar (Figure 1) obtain the typical two readings for a sentence like sentence 1 – the one where all the boys visited the same store, and the one where they may have all visited distinct stores (see Figure 2).[4]

(1) Every boy visited a store

Upon inspection, it should become clear that we actually license several additional readings for (1). Because of the argument positions introduced by the semantics for subjecthood and objecthood, the two quantifiers associated with *every boy* and *a store* can actually scope underneath the existential quantifier for the event – in the spots occupied above by "has-agent" and "has-patient". This gives rise to an additional three readings, whose meanings may not be immediately obvious. To make the matter clearer, consider (2).

(2) Every boy lifted-the-piano

[3]We do not consider the information contained in the tense of *visited* in this paper

[4]Note that, unlike (Nesson and Shieber, 2006), our semantic trees do not derive a string which represents the semantics; rather, our derived tree itself represents the semantics. Our trees could be easily modified (with additional terminal symbols) to allow for the semantics to be read off as a string. We see no urgent theoretical of practical need for this, however.

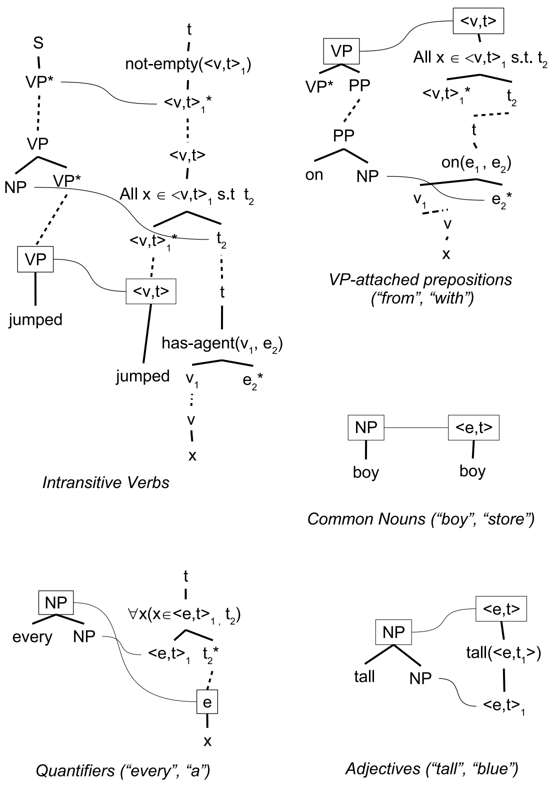

Figure 1: A toy grammar from the Synch-UVGDL framework.

We will treat 'lifted-the-piano' as a simple intransitive verb to simplify our analysis.[5] As is shown in Figure 3, the neo-Davidsonian representation of verbs licenses two readings:

- All the boys gathered around the piano, counted to three, and lifted it together. This corresponds to the reading where $\forall$ appears directly over 'has-agent'. More technically, "There is a single piano-lifting event, of which all the boys are agents."
- Taking turns, perhaps in a piano-lifting competition, each of the boys lifted the piano. This corresponds to the reading where $\forall$ is the root of the sentence's semantics. More technically, "For every boy, there exists some piano-lifting event of which he is the subject".

There is actually some debate as to whether the first reading can be obtained with the word "every". See (Winter, 2002) for a detailed discussion.

[5]Alternatively, substitute your favorite intransitive verb. We use "lifted-the-piano" because it makes the newly available readings easy to visualize.

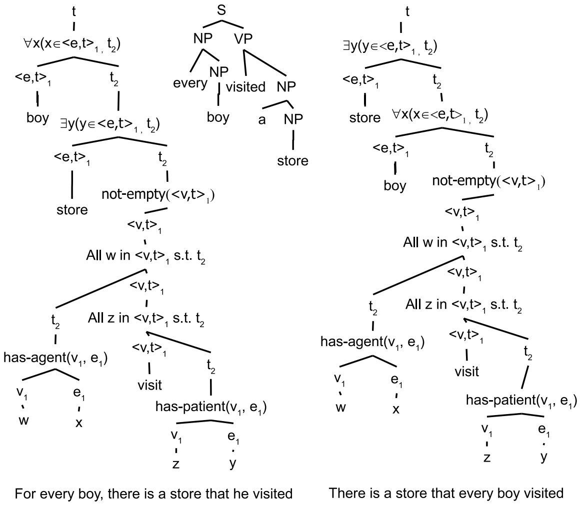

Figure 2: Two readings for *Every boy visited a store*.

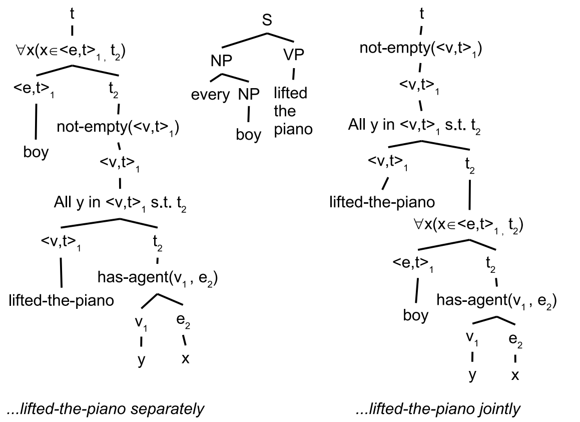

Figure 3: Two meanings for *Every boy lifted a piano*

The precise treatment is beyond the scope of this paper, but if you prefer, replace *every boy* with *all-the boys*, and the readings become readily available.

5 Conjunction as Quantification

The ambiguity in sentence (2) is very similar to a general ambiguity observed when studying conjunction, namely the issue of entity coordination versus sentence coordination. Roughly speaking, it is unclear whether in (2) we are constructing a compound subject out of *every boy*, and applying that subject (containing the set of all boys) to a single event, or if we are constructing many events, each of which has a simple subject. The same ambiguity arises if we substitute a conjunction for the quantifier:

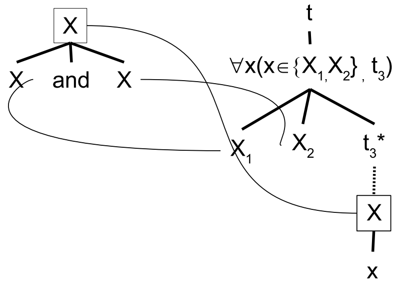

Figure 4: Conjunction modeled as quantification in UVGDL

(3) John and Mary lifted-the-piano

Here it is unclear if we are conjoining at the entity level (John and Mary lifted the piano together) or the sentence level (John lifted the piano and separately, Mary lifted the piano). Devising an analysis of coordination that accounts for both boolean coordination (as in the second case) and entity coordination (as in the first) has been a challenge for many researchers. Sometimes two fundamentally different meanings of words like *and* are proposed to account for this phenomenon: one that conjoins multiple entities or sets of entities into a larger set, and another that conjoins multiple propositions into a single proposition (Partee and Rooth, 1983).

We attempt to construct a single semantic treatment for conjunction by modeling it as quantification over an explicitly defined set (the conjuncts). Following a rough syntactic treatment more or less as suggested by (Sarkar and Joshi, 1996), the semantics for words duplicated across all conjuncts form the "assertion" of this quantifier, and the words specific to the individual conjuncts form the elements of the set to be quantified over (figure 4). Besides eliminating the need to potentially duplicate semantic formulas, this allows the conjuncts to be of any one semantic type, while preserving conjunction as an operation over t-type values. With this analysis, we neatly obtain two readings for *John and Mary lifted-the-piano*, with no duplication of semantic rules and a single analysis of *and* (see Figure 5).

It should be noted that we will require the *entire* semantic components of the words forming the conjuncts – not just their synchronous productions – to scope underneath the conjunction. Conjunction will need to be a sort of island in this sense. Otherwise, we run the risk of having elements of

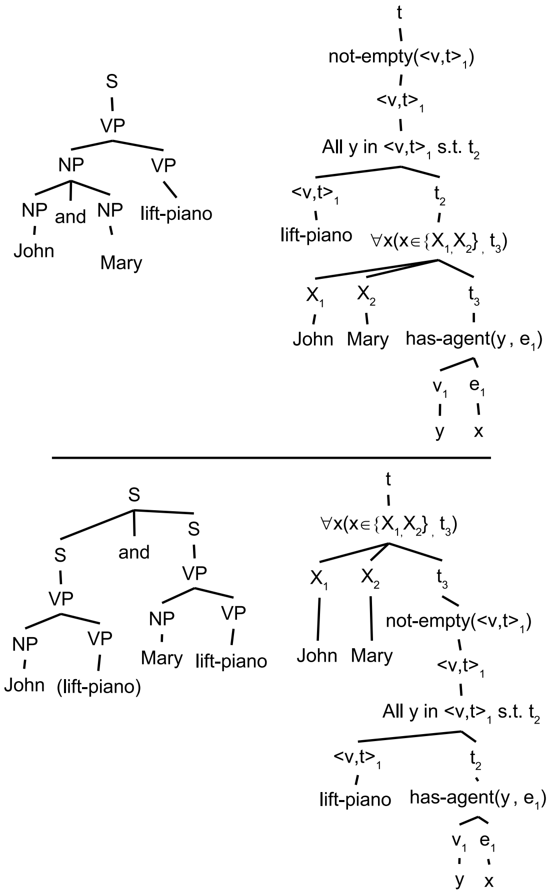

Figure 5: Two analyses for *John and Mary lifted a piano*

one conjunct scope over elements of another.

6 Conjunction Of Quantifiers

Simple sentences such as (3) can be handled very easily with this approach, but we encounter difficulties when we replace *John* and *Mary* with quantified NPs, as in (4).

(4) Every boy and most girls lifted-the-piano

The difficulty comes in the non-contiguous nature of the quantifiers: the individual conjuncts for *every boy* and *most girls* must contain the quantifier, the quantifier restrictor, and the variable the quantifier introduces into its assertion. However, the semantics of *lifted-the-piano* needs to show up in the assertion of the conjunction. This is not possible under the formalism as presented. Approaches that do not have this problem, such as the generalized conjunction of (Partee and Rooth, 1983), instead require conjuncts to be "cast up" and their representations changed depending on the other conjuncts in question – not easily represented within the SynchUVGDL formalism. To handle these cases, we reformulate our quantifiers in such a way that their scope is underspecified, but their form is contiguous.

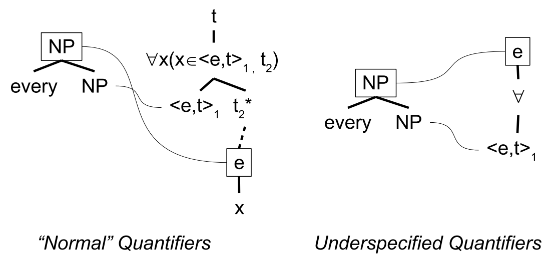

Figure 6: Underspecified quantifiers.

A quantifier can be viewed as "iterating" through all possible values as given by its restrictor, and seeing what happens when they are substituted in place of the bound variable it introduces in its assertion. The value of the quantifier expression as a whole (true or false) is dependent upon the values observed from its assertion when different possible values are substituted in. Under the SynchUVGDL formalism, formulae are not duplicated, and so quantifiers are able to project only a single copy of their bound variable into their assertion. Furthermore, as per the semantic treatment developed in (Lerman and Rambow, 2008), a quantifier's restrictor is simply a set – it does not contain any instances of the quantifier's bound variable. Thus, we are able to construct an underspecified quantifier as shown in figure 6 – quantifiers with a specified restrictor set, but with no specific assertion (yet).[6]

Intuitively, these underspecified quantifiers may be thought of as "choice functions," selecting an arbitrary element of their restrictor set. In the case of universal quantifiers, the choice function would be something like "pick any" (implying, as per the normal universal quantifier, that the same truth conditions hold for any member of the set). For an existential quantifier, the choice function would be something like "nondeterministically pick a privileged member" (implying that there exists at least one privileged member of whom something is true). These quantifiers are now contiguous, and may be used with our conjunction framework trivially, as seen in figure 7.

[6]The notion of underspecifying a quantifier in some manner is not new; as will be shown shortly, this representation is similar to one used in (Hobbs and Shieber, 1987).

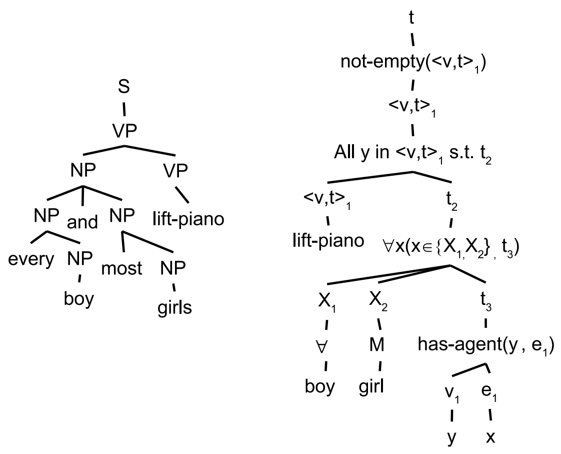

Figure 7: "Every boy and most girls lifted-the-piano [together]" with underspecified quantifiers.

The only challenge arising from this representation is that we have destroyed all notion of quantifier scope. Because quantifiers are now local to their restrictor sets, we are no longer able to distinguish the two common readings of sentences such as "Every boy visited a store." At some level, this is a good thing, as sentences such as these are in fact ambiguous between their two readings. A semantic formula for the meaning of a sentence that doesn't commit to a particular reading is desirable in many cases. However, it is important to be able to produce from this representation the complete set of valid readings for a sentence. As Hobbs and Shieber (1987) and others have pointed out, finding valid scopings is not a trivial task.

7 Restoring Quantifier Scope

Hobbs and Shieber (1987) present an algorithm for finding the valid quantifier scopings in a sentence. Roughly speaking, they iterate through the (currently underspecified) quantifiers of a sentence, in such an order that no quantifier is visited before any other quantifier that dominates it. Each quantifier visited is "moved" to some opaque argument position dominating it, or to the root of the sentence, such that bound variables don't lose their binding. The quantifier and its restrictor are moved, the quantifier "leaves behind" a copy of its bound variable, and whatever logic used to fill that opaque argument (including the left-behind bound variable) become the quantifier's assertion. By choosing a different iteration order, or by choosing different "landing sites" for each quantifier, all possible scope relations can be generated.

If we ignore conjunction for a moment, this algorithm can be applied to the SynchUVGDL framework with almost no modification. After having constructed the semantic tree for a sentence (with underspecified quantifiers), one can iterate through the quantifiers in any order permitted by the Shieber algorithm, and move each to any t-type argument position that dominates it (so long as no variables lose their binding). As before, the quantifier and its restrictor move, the quantifier's bound variable is left behind as a trace, and the logic that used to fill that t position becomes the quantifier assertion. This is facilitated by the structure of fully-specified quantifiers – they take a t-type argument, and are themselves t-type. Thus, the movement operation is essentially adjunction, plus leaving a trace. Because the semantic framework we are working with is neo-Davidsonian, this adaptation of the scope-restoration algorithm enables us to generate the additional single- and multiple-event readings as well, as the introduction of subjects and objects create more arguments of type t in the semantic tree.

Thus, so far the standard SynchUVGDL approach and the approach using underspecified quantifiers (along with the Hobbs-Shieber algorithm for disambiguating scope) are equivalent: in the standard SynchUVGDL approach, the dominance links in the semantics for quantifiers express the potential for extended scope, while the combination of the definition of the formalism (the meaning of dominance links) and the way the semantic side of the SynchUVGDL is constructed determine the actual scope readings. The standard SynchUVGDL approach is declarative, and scope is actually computed using general algorithms for processing SynchUVGDLs, not *ad-hoc* algorithms for scope.

When we introduce conjunction back in, the algorithm must be extended somewhat. First, in the case of coordinating quantified NPs, the quantifier would need to be able to expand to some node within the assertion of the conjunction. That is to say, the conjunction would need to be viewed as iterating through its conjuncts, substituting each into its assertion, then letting any quantifiers expand from its temporary position in the assertion, and then repeating the process with the next one. Crucially, the expansion happens "after" the conjunct has been substituted into the assertion of the conjunction.

- All the boys and most of the girls gathered together around the piano and together lifted it.
- Each of the children individually lifted it.
- All of the boys lifted the piano together, then they left and most of the girls lifted it together.
- All of the boys lifted the piano together, then most of the girls lifted it individually.
- Most of the girls lifted the piano together, then all of the boys lifted it individually.

Figure 8: Readings for *Every boy and most girls lifted-the-piano*

This is necessary because, although we are analyzing conjunctions as having the same structure as quantifiers, they nonetheless quantify over much richer objects. No matter how complex a (normal) quantifier's restrictor may be, it will ultimately yield a set of simple, atomic objects such as entities. Because in the case of conjunction we are quantifying over arbitrary expressions (so arbitrary in fact, that each element of the restrictor set must be vocalized individually, rather than in some compact expression as with a phrase like "every boy"), additional processing may be needed after the elements are substituted into the assertion. While this requires significantly more computation, note that there are tremendously fewer objects to iterate through: whereas a phrase like "every boy" may refer to hundreds or thousands of boys, each element in a conjunction must be vocalized individually, and so we rarely see more than three or four of these in a single sentence.

Additionally, recall that conjunctions must be treated as islands for the semantics of their conjuncts. This property must be retained in the context of quantifier expansion – otherwise we might license readings for (4) wherein the girls lift the piano once for each boy present. So, we prohibit quantifiers from expanding over any conjunctions they may be under. Note that this will never create a problem wherein quantifiers have no place to expand: they expand once they're substituted into the assertion of the conjunction, and the root of the assertion is always of type t.

We see now that the new approach for conjoined quantifiers has no clear equivalent representation in the standard SynchUVGDL approach: this is because the quantifiers are "temporarily" moved into the assertion for expansion, which cannot be replicated in a declarative approach. Thus, these kinds of semantic derivation pose a problem for semantic theories relying entirely on synchronous formalisms.

To this point we have experimented with treating conjunction as quantification, and with an underspecified model of quantification. The next logical step would be to examine the possibility of using the underspecified quantifier model with the quantifier we have introduced for conjunction – in short, underspecified conjunction. If conjunction is made underspecified in the same way as quantification (see figure 9), the semantic trees for sentences with conjunction become much more intuitive. Scope disambiguation would then proceed in the same manner as before – for instance, any quantifiers embedded in a conjunct of a conjunction could only be raised after the conjunction itself did so. The only difference is that conjunction must still be an island for quantifier raising: embedded quantifiers still may not ever scope above the conjunction.

Whereas previously sentences such as *Every boy and most girls lifted-the-piano* had different possible readings depending on the scope selected for *and*, the representation in figure 10 encompasses all 5 possible scope orderings which are summarized in Figure 8. It is, however, unclear whether all five readings really exist. We believe the first two readings are clearly licensed by the sentence, but the last three are somewhat dubious. Intuitively, it seems that the quantifiers *every* and *most* ought to move in parallel, but this behavior is hard to enforce in a way that still makes sense in sentences without such similar NPs (*Every boy and Susan lifted-the-piano*). In our example sentence, the two desired readings could be obtained neatly by declaring the assertion of a conjunction to be "opaque" with respect to quantifiers – they must raise above or below the entire thing (but obviously, stay under the conjunction itself). This would create exactly one reading for each possible position for the conjunction to raise to. It is not clear whether this approach would work in more complex cases.

8 Conclusion

In conclusion, we have explored how we can express the semantics of coordination in a syn-

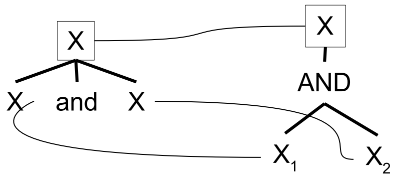

Figure 9: Underspecified conjunction

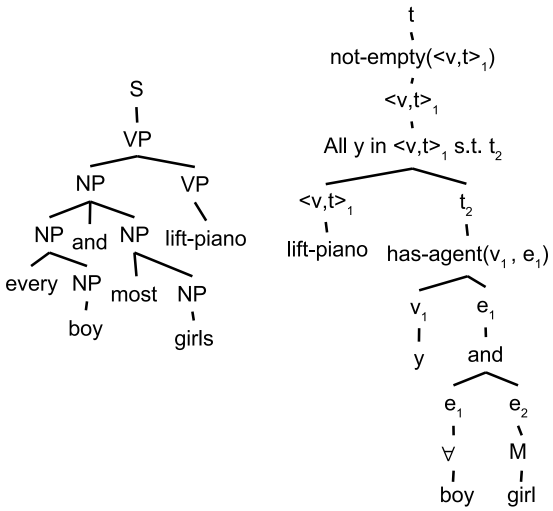

Figure 10: Example derivation with underspecified conjunction

chronous formalism. By modeling conjunction as quantification, we can easily derive the scope ambiguities with respect to the event variable which we see in coordination (do John and Mary lift the piano together, or individually?). We have seen that the conjunction of quantified NPs poses problems that apparently prevent us from expressing scope within the synchronous framework.

References

Gardent, Claire and Laura Kallmeyer. 2003. Semantic construction in F-TAG. In *Tenth Conference of the European Chapter of the Association for Computational Linguistics (EACL'03)*, Budapest, Hungary.

Hobbs, J. R. and S. M. Shieber. 1987. An algorithm for generating quantifier scopings. *Computational Linguistics*, 13:47–63.

Kallmeyer, L. and A. Joshi. 2003. Factoring predicate argument and scope semantics: Underspecified semantics with ltag. *Research on Language and Computation*, 1:3–58.

Kallmeyer, L. 2002. Using an enriched tag derivation structure as basis for semantics. In *Proceedings of the Sixth International Workshop on Tree Adjoining Grammar and Related Frameworks (TAG+6)*, pages 127–136.

Lerman, Kevin and Owen Rambow. 2008. Event semantics in synchronous formalisms. Technical report, Center for Computational Learning Systems, Columbia University.

M. Romero, L. Kallmeyer, O. Babko-Malaya. 2004. Ltag semantics for questions. In *Proceedings of the TAG+7*, pages 186–193, Vancouver.

Nesson, R. and S. Shieber. 2006. Simpler tag semantics through synchronization. In *Proceedings of the 11th Conference on Formal Grammar*, Malaga, Spain.

Nesson, R. and S. Shieber. 2007. Extraction phenomena in synchronous tag syntax and semantics. In *Proceedings of the Workshop on Syntax and Structure in Statistical Translation*, Rochester, New York.

Partee, B. and M. Rooth. 1983. Generalized conjunction and type ambiguity. In Bauerle, R., C. Schwarze, and A von Stechow, editors, *Meaning, use, and the interpretation of language*. Walter de Gruyter, Berlin.

Rambow, Owen and Giorgio Satta. 1996. Synchronous models of language. In *34th Meeting of the Association for Computational Linguistics (ACL'96)*, pages 116–123. ACL.

Rambow, Owen, K. Vijay-Shanker, and David Weir. 2001. D-Tree Substitution Grammars. *Computational Linguistics*, 27(1).

Rambow, Owen. 1994. Multiset-valued linear index grammars. In *32nd Meeting of the Association for Computational Linguistics (ACL'94)*, pages 263–270. ACL.

Sarkar, A. and A. Joshi. 1996. Handling coordination in a tree adjoining grammar. Technical report, Department of Computer and Information Science, University of Pennsylvania, Philadelphia.

Schein, B. 2002. Events and the semantic content of thematic relations. *Logical Form and Language*, page 263344.

Shieber, Stuart and Yves Schabes. 1990. Synchronous tree adjoining grammars. In *Proceedings of the 13th International Conference on Computational Linguistics*, Helsinki.

Shieber, Stuart B. 1994. Restricting the weak generative capacity of Synchronous Tree Adjoining Grammar. *Computational Intelligence*, 10(4):371–385.

Vijay-Shanker, K. 1992. Using descriptions of trees in a Tree Adjoining Grammar. *Computational Linguistics*, 18(4):481–518.

Winter, Y. 2002. Atoms and sets: A characterization of semantic number. *Linguistic Inquiry*, pages 493–505.