Parsing with Dynamic Continuized CCG

Michael White and **Jordan Needle**
Department of Linguistics
The Ohio State University
Columbus, OH 43210
mwhite@ling.osu.edu
needle.6@osu.edu

Simon Charlow
Department of Linguistics
Rutgers University
New Brunswick, NJ 08901
simon.charlow@rutgers.edu

Dylan Bumford
Department of Linguistics
New York University
New York, NY 10003
dbumford@nyu.edu

Abstract

We present an implemented method of parsing with Combinatory Categorial Grammar (CCG) that for the first time derives the exceptional scope behavior of indefinites in a principled and plausibly practical way. The account implements Charlow's (2014) monadic approach to dynamic semantics, in which indefinites' exceptional scope taking follows from the way the *side effect* of introducing a discourse referent survives the process of delimiting the scope of true quantifiers in a continuized grammar. To efficiently parse with this system, we extend Barker and Shan's (2014) method of parsing with continuized grammars to only invoke monadic lifting and lowering where necessary, and define novel normal form constraints on lifting and lowering to avoid spurious ambiguities. We also integrate Steedman's (2000) CCG for deriving basic predicate-argument structure and enrich it with a method of lexicalizing scope island constraints. We argue that the resulting system improves upon Steedman's CCG in terms of theoretical perspicuity and empirical coverage while retaining many of its attractive computational properties.

1 Introduction

A long-standing puzzle in natural language semantics has been how to explain the exceptional scope behavior of indefinites. For example, (1a) has a reading where there's a specific relative (a steel magnate, say) such that if she dies, the speaker will be rich. By contrast, (1b) has no analogous reading where the universal takes wide scope: this sentence cannot mean that every relative is such that if that particular relative dies, I'll be rich.[1] If one takes the antecedent of a conditionals to be a *scope island* (as suggested by the $< \ldots >$ bracketing), then it's not surprising that the universal in (1b) is blocked from taking wide scope; what instead requires explanation is how the indefinite in (1a) can exceptionally take scope out of this island.

(1) a. If <$\underline{\text{a}}$ relative of mine dies>, I'll inherit a fortune. ($\exists >$ if)

b. If <$\underline{\text{every}}$ relative of mine dies>, I'll inherit a fortune. (* $\forall >$ if)

Charlow (2014) has recently shown that the exceptional scope behavior of indefinites can be derived from their role of introducing discourse referents in a dynamic semantics. To do so, he showed that (1) a monadic approach to dynamic semantics can be seamlessly integrated with Barker and Shan's (2014) approach to scope taking in continuized grammars, and (2) once one does so, the exceptional scope of indefinites follows from the way the *side effect* of introducing a discourse referent survives the process of delimiting the scope of true quantifiers, such as those expressed with *each* and *every*.

To date, computationally implemented approaches to scope taking[2] have not distinguished indefinites from true quantifiers in a way that accounts for their exceptional scope taking. In Bos's (2003) implementation of Discourse Representation Theory (Kamp and Reyle, 1993, DRT), for example, scope taking is independent of how

[1] That is, (1b) cannot mean the same thing as *If any relative of mine dies, I'll inherit a fortune*; see Barker and Shan (2014) for a compatible treatment of negative polarity items such as *any*.

[2] See e.g. (Copestake et al., 2005; Koller et al., 2003; Gardent and Kallmeyer, 2003; Nesson and Shieber, 2006; Pogodalla and Pompigne, 2012), inter alia.

indefinites are treated. Although Steedman (2012) has developed an account of indefinites' exceptional scope taking in a non-standard static semantics for Combinatory Categorial Grammar (Steedman, 2000, CCG), this treatment has not been fully implemented (to our knowledge); moreover, as Barker and Shan point out, Steedman's theory appears to undergenerate by not allowing true quantifiers to take scope from medial positions.

Barker and Shan offer a brief sketch of how a parser for their continuized grammars can be implemented, including how lifting can be invoked lazily to ensure parsing terminates. In this paper, we show how their approach can be seamlessly combined with Steedman's CCG and extended to include Charlow's monadic dynamic semantics, thereby providing the **first computational implementation** of a system that accounts for the exceptional scope behavior of indefinites in a **principled** and **plausibly practical** way. To efficiently parse with this system, we devise rules to **only invoke monadic lifting and lowering where necessary**, and define novel **normal form constraints on lifting and lowering** to avoid spurious ambiguities. We also integrate a method of lexicalizing scope island constraints (Barker and Shan, 2006), as Charlow's account does not provide a practical and empirically satisfactory means of enforcing such constraints. We argue that the resulting system improves upon Steedman's CCG in terms of **theoretical perspicuity**—insofar as it builds upon an account of dynamic semantics that is independently necessary—and **empirical coverage**, in that it allows quantifiers to take scope from medial positions and from some subordinate clauses. At the same time, it also retains many of CCG's attractive computational properties; in particular, it respects Steedman's Principle of Adjacency, only combining overtly realized adjacent constituents, thereby making it easy to use with well-studied parsing algorithms. An open source prototype implementation, suitable for testing out grammatical analyses, is available online.[3]

2 Are Scope Islands Real?

Steedman (2012) observes that although the empirical status of scope islands remains unsettled in the linguistics literature (Farkas and Giannakidou, 1996; Reinhart, 1997; Ruys and Winter, 2011; Syrett and Lidz, 2011; Syrett, 2015), the possible scopings of true quantifiers appear to be much more limited than commonly assumed in computational approaches to scope taking, arguing therefore in favor of a surface-compositional approach that aims to capture all and only the attested readings; in particular, Steedman takes as his working hypothesis that scope inversion should be subject to syntactic island constraints. While we are sympathetic to Steedman's point of view, we are skeptical of his working hypothesis, as it appears to incorrectly predict that quantifiers should never be able to take scope from subjects of finite complement clauses. Acknowledging that universals sometime appear to do so, Steedman appeals to Fox & Sauerland's (1996) illusory scope analyis, where the quantificational force is argued to stem from a main clause generic. However, Farkas and Giannakidou (1996) provide numerous examples in English and Greek of episodic sentences such as (2) where the universal takes extra-wide scope.

(2) Yesterday, a guide made sure that <every tour to the Louvre was fun>. $(\forall > \exists)$

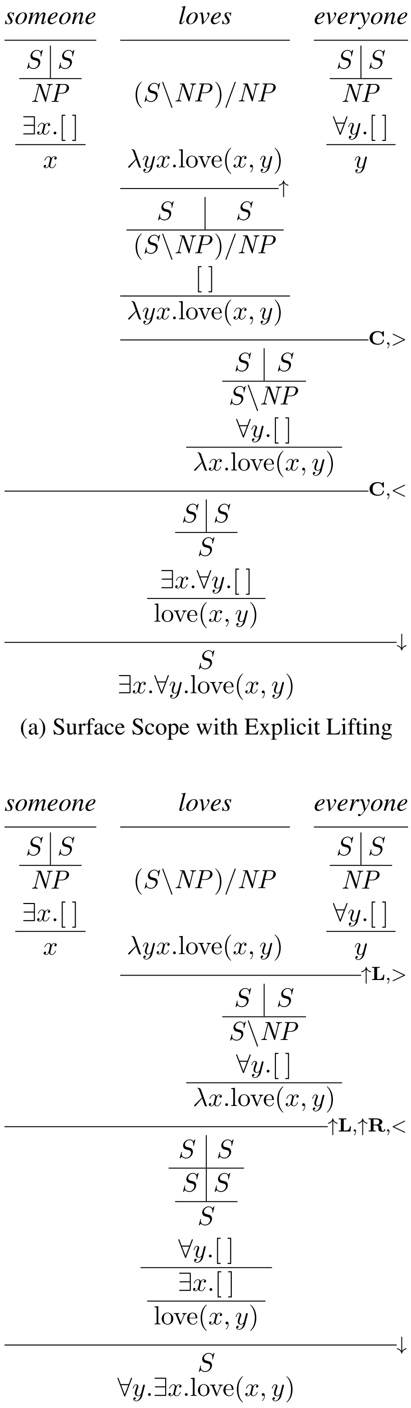
By contrast, a corpus analysis given in the appendix suggests that conditionals and relative clauses plausibly represent cases where scope islands should be treated as hard constraints. As such, in this paper we adopt the working hypothesis that scope island constraints can be given an accurate lexicalized treatment. Alternatively, one could pursue an approach based solely on soft constraints, where a probabilistic model simply makes scope taking beyond finite clause boundaries very unlikely. Even in this scenario, we contend that the approach to exceptionally scoping indefinites implemented here will greatly simplify the learning task, since the ability of indefinites to take exceptional scope would not need to be learned.

3 Continuized CCG

A continuized grammar is one where the meaning of expressions can be defined as a function on a portion of its surrounding context, or *continuation* (Barker, 2002; Shan and Barker, 2006; Barker and Shan, 2014). To make it easier to reason about continuized grammars, Barker & Shan devised the "tower" notation illustrated in Figure 1.[4] For example, *everyone* has a tower category with *NP* on

[3] `https://github.com/mwhite14850/dyc3g`

[4] Semantic types are suppressed in this and subsequent figures, except where essential for understanding.



(a) Surface Scope with Explicit Lifting

(b) Inverse Scope with Integrated Lifting

Figure 1: Continuized CCG Derivations

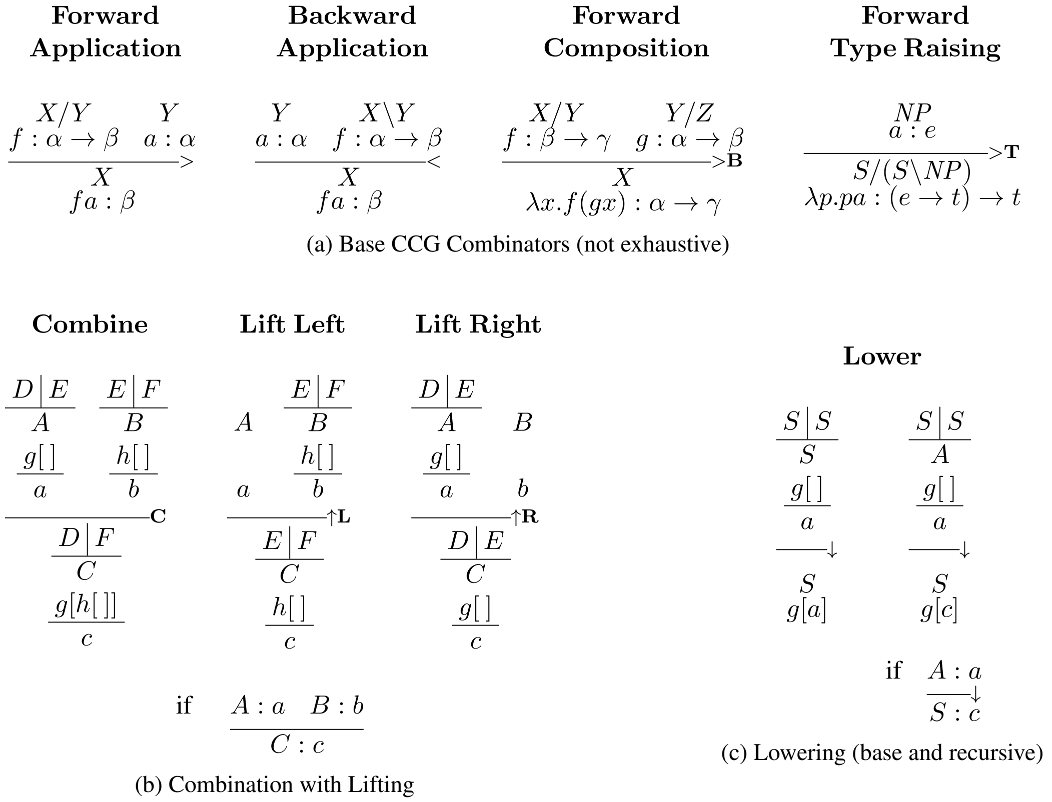
the bottom and two Ss on top; reading counterclockwise from the bottom, this category represents a constituent that acts locally as an NP, takes scope over an S, and returns an S. The semantics is $\lambda k.\forall y.ky$, a function from a continuation k of type $e \to t$ to a universally quantified expression of type t. A continuized meaning of this form can be abbreviated by representing the location where the continuation argument applies with [ ] and putting the argument to the continuation on the bottom of the tower, as shown.

In a continuized grammar, all expressions can potentially be given continuized meanings via the Lift (↑) operation. This is illustrated in Figure 1a where the category for *loves* is lifted, taking on the semantics $\lambda k.k(\lambda yx.\text{love}(x, y))$. Lifting the category for *loves* allows it to combine with that of *everyone* using scopal combination. The way in which scopal combination works in the tower notation is shown in Figure 2b (left): on the tower top, the continuized functions $g[\,]$ and $h[\,]$ compose in surface order, yielding $g[h[\,]]$; meanwhile, recursing on the tower bottom, the expressions a and b combine as they normally would in CCG (using the combinators in Figure 2a), yielding c.[5] In the example, [ ] and $\forall y.[\,]$ compose to again yield $\forall y.[\,]$ on the tower top, with $\lambda yx.\text{love}(x, y)$ applying to y and yielding $\lambda x.\text{love}(x, y)$ on the bottom.

As Barker & Shan observe, the explicit lifting step seen in Figure 1a can be integrated with the scopal combination step, as shown in the other recursively defined rules in Figure 2b, thereby avoiding an infinite regress when applying the lifting rule. Figure 1b shows how Lift Left (↑**L**) and Lift Right (↑**R**) can be applied in sequence—as part of a single parsing step combining adjacent signs—to create a three-level tower where *everyone* ends up taking inverse scope over the subject:[6] first, in applying Lift Left, the entire tower for *someone* is matched as A, while the bottom of the tower for *loves everyone*, $S\backslash NP$, is matched as B, and then the rules are reapplied with A and B as inputs; next, Lift Right is applied, with the bottom of the tower for *someone*, NP, matched as A, and $S\backslash NP$ again matched as B, and the rules are reapplied once more; this time, the categories can combine directly using Backward Application (<), ending the recursion; as the rules unwind, the three-level tower for *someone loves everyone* is constructed, with inverse scope semantics, as shown. The final representations are derived by collapsing the towers using the recursively defined Lower (↓) operation in Figure 2c, which repeat-

[5] The combinator for combining two scopal terms m and n is $\lambda mnk.m(\lambda x.n(\lambda y.k(xy)))$, assuming forward application on the tower bottom. Formulating the rules recursively allows the base combinator to be factored out while also generalizing to multi-level towers.

[6] Though *everyone* is right peripheral in the example, nothing would prevent it from taking inverse scope from medial position, in contrast to Steedman's (2012) approach.



(a) Base CCG Combinators (not exhaustive)

(b) Combination with Lifting

(c) Lowering (base and recursive)

Figure 2: Continuized CCG

edly applies the continuized semantics to the identity continuation $\lambda k.k$.

4 Monadic Dynamic Semantics

Charlow's (2014) dynamic semantics makes use of the State.Set monad (Hutton and Meijer, 1996), which combines the State monad for handling side effects with the Set monad for non-determinism. The State monad pairs ordinary semantic values with a state, which is threaded through computations. The Set monad models non-deterministic choices as sets, facilitating a non-deterministic treatment of indefinites. For example, the dynamic meaning of *a linguist swims* appears in (3): here, the proposition that x swims, where x is some linguist, is paired with a state that augments the input state s with the discourse referent x.

(3) $\lambda s.\{\langle \text{swim}(x), \widehat{sx}\rangle \mid \text{linguist}(x)\}$

More formally, the State.Set monad is defined as in (4). For each type α, the corresponding monadic type $M\alpha$ is a function from states of type s to sets pairing items of type α with such states. The η function injects values into the monad, simply yielding a singleton set consisting of the input item paired with the input state. The *bind* operation $\multimap$ sequences two monadic computations by sequencing the two computations pointwise, feeding each result of m applied to the input state s into π and unioning the results.[7] Less formally, the $\multimap$ operation can be thought of as "run m to determine v in π."

(4) State.Set Monad

$$\begin{array}{rcl} M\alpha & = & s \to \alpha \times s \to t \\ a^{\eta} & = & \lambda s.\{\langle a, s\rangle\} \\ m_v \multimap \pi & = & \lambda s.\bigcup_{\langle a,s'\rangle \in ms} \pi[a/v]s' \end{array}$$

Since the only operation on states that we will be concerned with in this paper is adding discourse referents, it suffices to leave the states implicit in the implementation, only explicitly representing the new discourse referents—much as in computational implementations of Discourse Representation Theory (Bos, 2003), where assignments are not explicitly represented in Discourse Representation Structures. Consequently, we will represent

[7] Note that the notation $m_v \multimap \pi$ is just syntactic sugar for $m \multimap \lambda v.\pi$, which may be more familiar.

(3) as (5), which can be translated to first-order logic in much the same way as with DRT.[8]

(5) $\{\langle \mathrm{swim}(x), x\rangle \mid \mathrm{linguist}(x)\}$
$\rightsquigarrow \exists x.\mathrm{linguist}(x) \wedge \mathrm{swim}(x)$

The definition of State.Set sequencing allows us to define a sequence reduction operation where the value of m is substituted into π for v and the discourse referents and conditions are combined. For example, the representation of *a linguist* can be sequenced with that of *swim* and simplified as in (6).

(6) $\{\langle x, x\rangle \mid \mathrm{linguist}(x)\}_y \multimap \{\langle \mathrm{swim}(y), \epsilon\rangle\}$
$\rightsquigarrow \{\langle \mathrm{swim}(x), x\rangle \mid \mathrm{linguist}(x)\}$

As in DRT, negation in Charlow's dynamic semantics is defined in a way that captures discourse referents, making them inaccessible for subsequent reference. Conditionals and universals are defined in terms of negation, thereby explaining their effects on discourse referent accessiblity; for representational simplicity, we will instead assume directly defined meanings for conditionals and universals, as in DRT.

5 Dynamic Combinatory Rules

The rules for combining signs in Dynamic Continuized CCG appear in Figures 3 and 4, augmenting those in Figure 2. We first give an overview of these rules and then illustrate with examples.[9]

As Charlow (2014) explains in detail, continuized grammars can be reconceptualized as operating over an underlying monad, where monadic lifting is identified with applying the underlying monad's sequencing operator ($\multimap$) and monadic lowering with applying the injection function (η). Accordingly, we include the rules for monadic lifting and lowering in Figure 3a-b. The lifting rule takes a category A with monadic value a of type $M\alpha$ and sequences it with a new continuation, yielding a function $\lambda k.a_v \multimap kv$ of type $(\alpha \rightarrow M\beta) \rightarrow M\beta$ for a tower category with A on the bottom. Monadic lowering is defined recursively, with the two base cases on the left and the two recursive cases on the right. The base cases apply η to the value a on the tower bottom before filling it in for the continuation; the second base case enables lowering to apply to the CCG categories S/NP and $S\backslash NP$ used in relative clauses. The recursive cases on the right again enable multi-level towers to be lowered in one fell swoop, with the second rule enabling towers with tower-result categories on the bottom to be fully lowered.

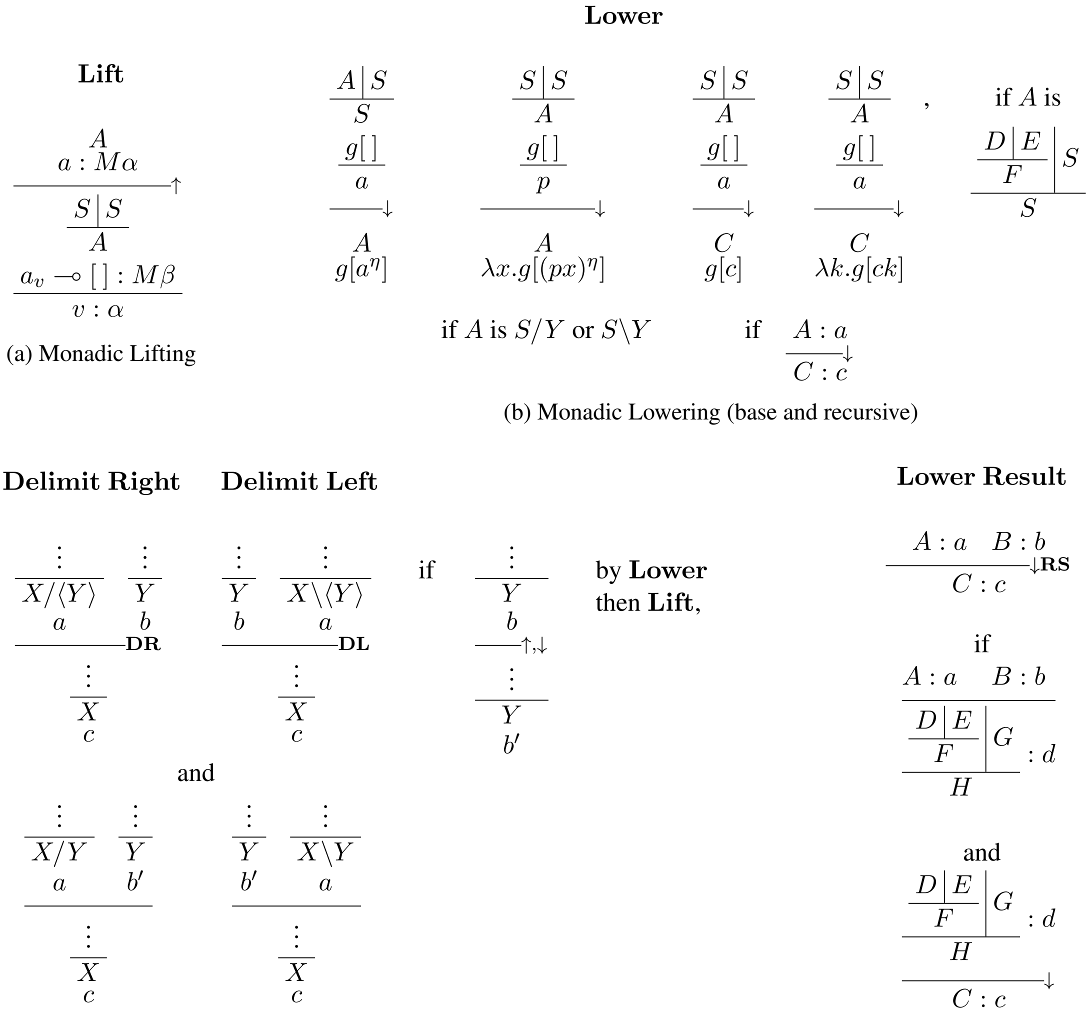
To implement scope islands, the rules in Figure 3c together with the unary type constructor $\langle\cdot\rangle$ enable categories to specify that their arguments must be scope delimited by undergoing a reset (i.e. lower then re-lift) before combination is permitted.[10] Figure 3d enables a double-continuation analysis of determiners by triggering a lowering when two categories combine to yield a category with a lowerable tower result. Finally, the rules in Figure 4 apply when the functor category expects a monadic value on the tower bottom; the rules in Figure 4a use η to coerce the input to the right type, while the ones in Figure 4b invoke lowering to do so.

An example illustrating exceptional scope for an indefinite appears in Figure 5a. Even though the category for *if* requires the category for its antecedent *someone complains* to be reset prior to combination, the side effect of discourse referent introduction survives the reset operation—enabling a wide-scope reading of the indefinite—irrespective of whether sequence reduction is carried out immediately, as in Figure 5c. (Figure 8 in the appendix shows how side effects are unaffected by reset in the general case, using the monadic identity and associativity laws.) Figure 5b shows how the narrow scope reading for the indefinite can be derived instead using monadic type–driven lowering. By contrast, Figure 6 shows why the narrow scope reading is the only one available for the universal in *if everyone complains*, since the reset operation closes off the scope of the universal, as illustrated in detail in Figure 6b. The appendix gives two further examples: Figure 10a illustrates result tower lowering in an inverse linking example—including the possibility of medial scoping, which is not possible with Steedman's CCG—while Figure 9 shows

[8] Explicitly representing the states could simplify the treatment of discourse referent accessibility; we leave investigating this alternative for future work.

[9] The side conditions for these rules (preceded by 'if') sometimes serve to define the rules recursively, as in the earlier Figure 2, and sometimes serve to specify sub-cases of interest. Rules for anaphora resolution are left for future work.

[10] The Delimit rules must apply first to ensure that entire towers are reset. This is accomplished using a cut in the Prolog implementation; alternatively, these rules could be defined at the level of signs rather than categories.



(a) Monadic Lifting

(b) Monadic Lowering (base and recursive)

(c) Delimiting Scope with Reset

(d) Result Lowering

Figure 3: Dynamic Continuized CCG: Monadic Lifting and Lowering

by contrast how universals are trapped in relative clause scope islands.

6 Prototype Implementation

Barker & Shan suggest that the rules in Figure 2 can form the basis of a practical parser. While the worst-case complexity of parsing with such rules has yet to be investigated, the way in which towers can grow to arbitrary heights is apt to at least limit the utility of dynamic programming in practice, potentially posing efficiency problems even when using aggressive statistical pruning (Clark and Curran, 2007). However, recent work on parsing with global neural network models has moved away from dynamic programming solutions, as the global models are incompatible with dynamic programming locality requirements. In particular, Lee et al. (2016) have shown that global neural models can be used with A* search to obtain a new state-of-the-art in CCG in parsing accuracy while maintaining impressive speed, even though the search space is exponential. As such, given that our approach respects Steedman's Principle of Adjacency, we suggest that it may be possible to extend CCG statistical parsing methods to the current setting, thereby resolving scope ambiguities the same way as other derivational ambiguities, rather than in a post-process as in earlier computational work on scope taking. While we are aware of no large-scale scope-annotated corpora at present, small-scale corpora do exist that would enable this conjecture to be tested in future work, such as the corpora used in work on CCG semantic parsing (Artzi and Zettlemoyer, 2013, inter alia).

Towards that end, we have implemented a prototype shift-reduce parser in Prolog that uses the unary and binary combination rules defined in Figure 2 together with additional rules defined in

Forward Application with η

$$\dfrac{\begin{matrix} X/Y \\ f : M\alpha \to \beta \end{matrix} \quad \begin{matrix} Y \\ a : \alpha \end{matrix}}{\begin{matrix} X \\ fa^{\eta} : \beta \end{matrix}} >^{\eta}$$

Backward Application with η

$$\dfrac{\begin{matrix} Y \\ a : \alpha \end{matrix} \quad \begin{matrix} X\backslash Y \\ f : M\alpha \to \beta \end{matrix}}{\begin{matrix} X \\ fa^{\eta} : \beta \end{matrix}} <^{\eta}$$

(a) Base CCG Combinators with Monadic Type Coercion (not exhaustive)

Lower Right

$$\dfrac{\begin{matrix} \dfrac{\vdots}{A} \\ a : M\alpha \to \beta \end{matrix} \quad \begin{matrix} \dfrac{\vdots}{B} \\ b : \gamma \end{matrix}}{\begin{matrix} \dfrac{\vdots}{C} \\ c : \beta \end{matrix}} \downarrow\mathbf{R}$$

Lower Left

$$\dfrac{\begin{matrix} \dfrac{\vdots}{B} \\ b : \gamma \end{matrix} \quad \begin{matrix} \dfrac{\vdots}{A} \\ a : M\alpha \to \beta \end{matrix}}{\begin{matrix} \dfrac{\vdots}{C} \\ c : \beta \end{matrix}} \downarrow\mathbf{L}$$

if

$$\dfrac{\begin{matrix} \dfrac{\vdots}{B} \\ b : \gamma \end{matrix}}{\begin{matrix} B' \\ b' : M\alpha \end{matrix}} \downarrow$$

and

$$\dfrac{\begin{matrix} \dfrac{\vdots}{A} \\ a : M\alpha \to \beta \end{matrix} \quad \begin{matrix} B' \\ b' : M\alpha \end{matrix}}{\begin{matrix} \dfrac{\vdots}{C} \\ c : \beta \end{matrix}} \qquad \dfrac{\begin{matrix} B' \\ b' : M\alpha \end{matrix} \quad \begin{matrix} \dfrac{\vdots}{A} \\ a : M\alpha \to \beta \end{matrix}}{\begin{matrix} \dfrac{\vdots}{C} \\ c : \beta \end{matrix}}$$

(b) Monadic Type–Driven Lowering

Figure 4: Dynamic Continuized CCG: Monadic Arguments

Section 5 for implementing Charlow's monadic dynamic semantics.[11] When implementing the parser, we found it paid off to only invoke lowering where necessary, as discussed in Section 5, rather than simply invoking lowering whenever possible. Moreover, in Section 7, we will see how enforcing scope islands keeps tower heights under control, while also providing an opportunity to define normal form constraints that limit spurious ambiguity, another important practical consideration. To test the implementation, we developed an initial test suite of 40 examples of average length 6.7 words, roughly comparable in size and complexity to Baldridge's (2002) OpenCCG[12] test suite. With the normal form constraints, parsing time was 60ms per item on a laptop, similar to OpenCCG on the same hardware. By contrast, with the normal form constraints turned off, the parsing time increased to 4.6s per item, nearly two orders of magnitude slower.

7 Normal Form Constraints

A normal form parse is the simplest parse in an equivalence class of parses yielding the same interpretation. Normal form constraints can play an important role in practical CCG parsing by eliminating derivations leading to spurious ambiguities without requiring expensive pairwise equivalence checks on λ-terms (Eisner, 1996; Clark and Curran, 2007; Hockenmaier and Bisk, 2010; Lewis and Steedman, 2014). Continuized CCG can employ existing CCG normal form constraints at the base level. The main additional source of spurious ambiguity is illustrated in Figure 7.[13] In the figure, the two towers at the upper left are combined via $\uparrow\mathbf{R}$ and $\uparrow\mathbf{L}$ to yield a three-level tower, which potentially allows an operator to subsequently take scope between any scopal elements present in the left and right input signs. However, if this three-level tower is subsequently lowered without any operator taking intermediate scope, the derivation will yield an interpretation that is equivalent to the one yielded by the simpler derivation that just combines the two signs in their surface scope order (i.e., without yielding a three-level tower).[14] As such, the lowering operations triggered by scope islands or sentence boundaries provide an opportunity to recursively detect and eliminate such non–normal form derivations, as follows:

Trigger If a sign is created using a lower operation, check the input sign for a spurious $\uparrow\mathbf{R}, \uparrow\mathbf{L}$ combination.

Base A sign constructed via $\ldots, \uparrow\mathbf{R}, \uparrow\mathbf{L}, \ldots$ has a spurious $\uparrow\mathbf{R}, \uparrow\mathbf{L}$ combination.

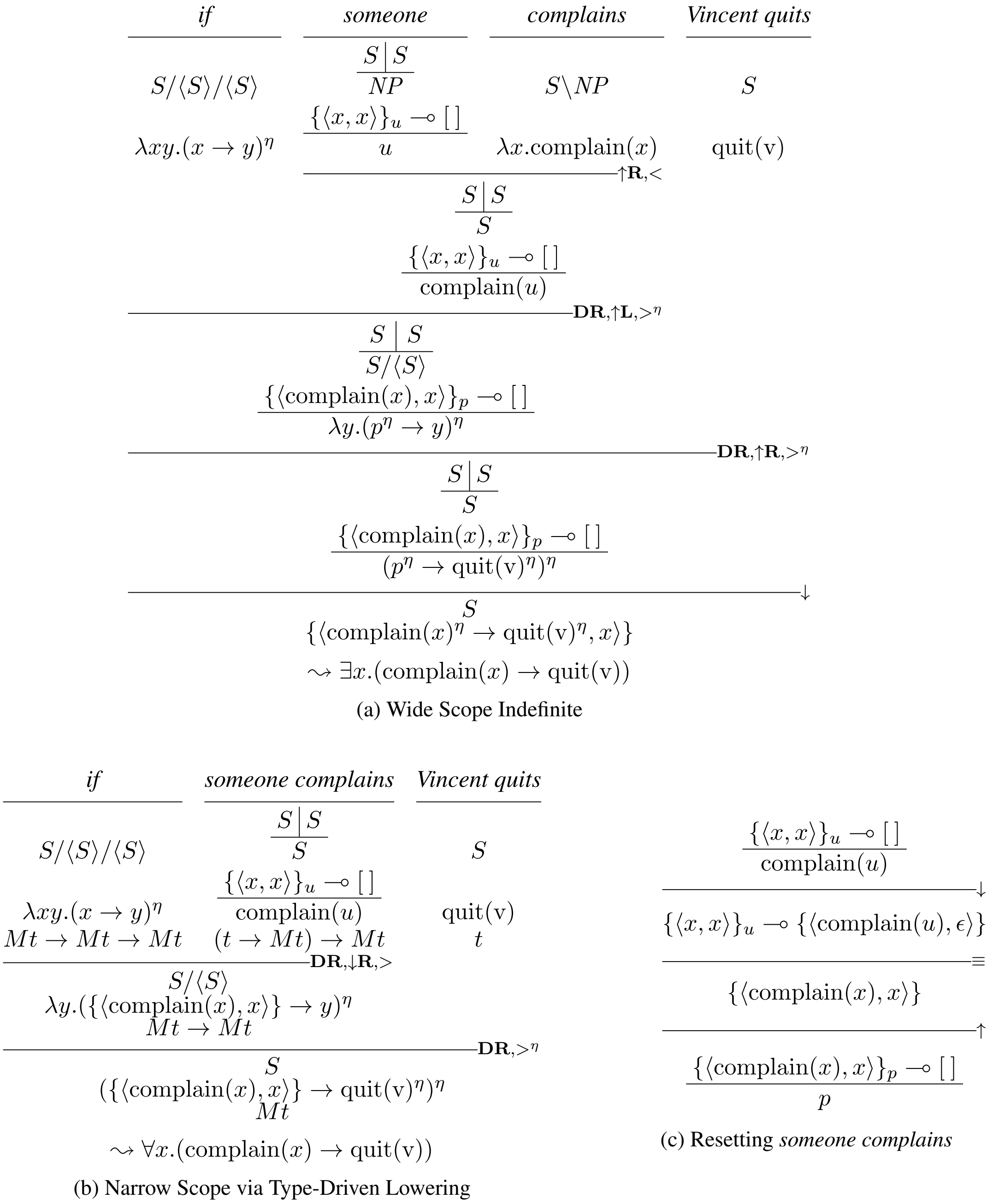
Non-Scopal A sign that is derived from a non-scopal input sign—i.e., one whose category

[11] https://github.com/mwhite14850/dyc3g

[12] http://openccg.sourceforge.net/

[13] Spurious ambiguity can also arise from the inversion of two indefinites; we leave this issue for future work.

[14] As noted in Section 5, it remains for future work to add the lowering rules for multi-level towers that enable Charlow's treatment of selective exceptional scope; the normal form constraints will need to be augmented accordingly.


if
someone
complains
Vincent quits
$S/\langle S\rangle/\langle S\rangle$
$\lambda xy.(x \to y)^\eta$
$\frac{S|S}{NP}$
$\frac{\{\langle x,x\rangle\}_u \multimap [\,]}{u}$
$S\backslash NP$
$\lambda x.\mathrm{complain}(x)$
S
$\mathrm{quit}(\mathrm{v})$
↑R,<
$\frac{S|S}{S}$
$\frac{\{\langle x,x\rangle\}_u \multimap [\,]}{\mathrm{complain}(u)}$
DR,↑L,>η
$\frac{S|S}{S/\langle S\rangle}$
$\frac{\{\langle \mathrm{complain}(x),x\rangle\}_p \multimap [\,]}{\lambda y.(p^\eta \to y)^\eta}$
DR,↑R,>η
$\frac{S|S}{S}$
$\frac{\{\langle \mathrm{complain}(x),x\rangle\}_p \multimap [\,]}{(p^\eta \to \mathrm{quit}(\mathrm{v})^\eta)^\eta}$
↓
S
$\{\langle \mathrm{complain}(x)^\eta \to \mathrm{quit}(\mathrm{v})^\eta, x\rangle\}$
$\rightsquigarrow \exists x.(\mathrm{complain}(x) \to \mathrm{quit}(\mathrm{v}))$
if
someone complains
Vincent quits
$S/\langle S\rangle/\langle S\rangle$
$\lambda xy.(x \to y)^\eta$
$Mt \to Mt \to Mt$
$\frac{S|S}{S}$
$\frac{\{\langle x,x\rangle\}_u \multimap [\,]}{\mathrm{complain}(u)}$
$(t \to Mt) \to Mt$
S
$\mathrm{quit}(\mathrm{v})$
t
DR,↓R,>
$S/\langle S\rangle$
$\lambda y.(\{\langle \mathrm{complain}(x),x\rangle\} \to y)^\eta$
$Mt \to Mt$
DR,>η
S
$(\{\langle \mathrm{complain}(x),x\rangle\} \to \mathrm{quit}(\mathrm{v})^\eta)^\eta$
Mt
$\rightsquigarrow \forall x.(\mathrm{complain}(x) \to \mathrm{quit}(\mathrm{v}))$
$\frac{\{\langle x,x\rangle\}_u \multimap [\,]}{\mathrm{complain}(u)}$
↓
$\{\langle x,x\rangle\}_u \multimap \{\langle \mathrm{complain}(u),\epsilon\rangle\}$
≡
$\{\langle \mathrm{complain}(x),x\rangle\}$
↑
$\frac{\{\langle \mathrm{complain}(x),x\rangle\}_p \multimap [\,]}{p}$

(a) Wide Scope Indefinite

(b) Narrow Scope via Type-Driven Lowering

(c) Resetting *someone complains*

Figure 5: Conditional with Indefinite Example

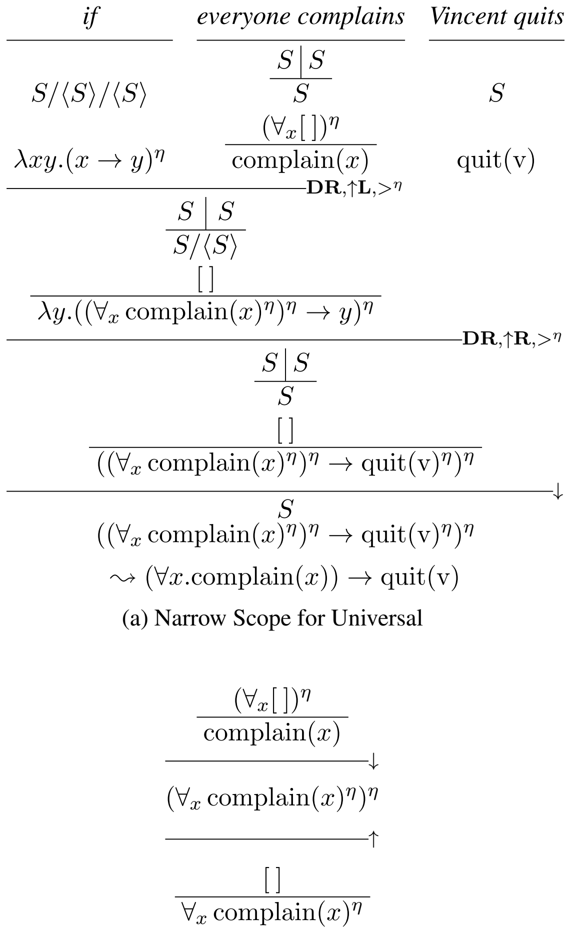
has no tower top—has a spurious ↑**R**, ↑**L** combination if the other input sign has a spurious ↑**R**, ↑**L** combination. This case is illustrated in Figure 7, where H is such a nonscopal input sign.

Inversion A sign that is derived by a ↑**L**, ↑ **R** inversion has a spurious ↑**R**, ↑**L** combination if either input sign has a spurious ↑**R**, ↑ **L** combination.

Recurse Right A sign that is derived by a **C**, ↑**L** has a spurious ↑**R**, ↑**L** combination if the right input sign has a spurious ↑**R**, ↑**L** combination. Note that ↑**L**, **C** can derive intermediate scope for the left input sign.

Recurse Left A sign that is derived by a ↑**R**, **C** has a spurious ↑**R**, ↑**L** combination if the left input sign has a spurious ↑**R**, ↑**L** combination. Note that **C**, ↑**R** can derive intermediate scope for the right input sign.

These rules have been tested for safety in the reference implementation by ensuring that all six (3!) desired interpretations result from a ditransi-



(a) Narrow Scope for Universal

(b) Resetting *everyone complains*

Figure 6: Conditional with Universal Example

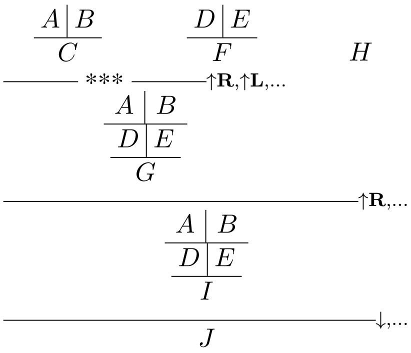


Figure 7: Non–Normal Form Derivation

tive verb combined with three scopal arguments, and all 4! desired interpretations result from a 4-argument verb in combination with four scopal arguments. With the ditransitive verb, all spurious ambiguity is eliminated, reducing 78 derivations in an otherwise unambiguous sentence down to just the six normal form derivations. The rules are not quite complete though, as six spuriously equivalent derivations remain with the 4-argument verb, where 525 derivations are whittled down to 30; safely filtering the remaining six spuriously equivalent derivations would require more complex rules that track the level at which the ↑**R**, ↑**L** operations apply in the base case, which may not be worth the added complexity in practice.[15]

8 Conclusion

We have presented a method of parsing with Dynamic Continuized CCG that for the first time derives the exceptional scope behavior of indefinites in a principled and plausibly practical way. Our approach (i) extends Barker and Shan's (2014) method of parsing with continuized grammars to only invoke Charlow's (2014) monadic lifting and lowering where necessary, (ii) integrates Steedman's (2000) CCG for deriving basic predicate-argument structure and enriches it with a practical method of lexicalizing scope island constraints, and (iii) takes advantage of the resulting scope islands in defining novel normal form constraints for efficient parsing. We have argued that the account (i) improves upon Steedman's (2012) approach to quantifier scope in terms of theoretical perspicuity by taking advantage of a dynamic semantics for indefinites independently needed for anaphora, and (ii) offers better empirical coverage by allowing quantifiers to take scope from medial positions and some subordinate clauses. At the same time, by respecting the Principle of Adjacency, only combining overtly realized adjacent constituents, our approach is easy to use with well-studied parsing algorithms, as with Steedman's CCG. Although the normal form constraints are quite effective in small-scale experiments, it remains for future work to verify quantitatively whether these constraints suffice for practical parsing in conjunction with statistical filtering techniques. It also remains for future work to computationally explore the novel analyses made possible by this framework, including order-sensitivity in negative polarity items (Barker and Shan, 2014) and selective exceptional scope for indefinites and focus alternatives (Charlow, 2014). Towards that end, we have made available online an open source prototype implementation suitable for testing out grammatical analyses.

[15] Normal form constraints need not be complete to be practically useful, as any remaining ambiguity can be handled by pairwise checks.

Acknowledgments

We thank Carl Pollard, Scott Martin, Mark Steedman, the OSU Clippers and Synners Groups, the Midwest Speech and Language Days 2016 audience and the anonymous reviewers for helpful comments and discussion. This work was supported in part by a Targeted Investment in Excellence Grant from OSU Arts & Sciences and by NSF grant IIS-1319318.

References

Yoav Artzi and Luke Zettlemoyer. 2013. Weakly supervised learning of semantic parsers for mapping instructions to actions. *Transactions of the Association for Computational Linguistics* 1(1):49–62.

Jason Baldridge. 2002. *Lexically Specified Derivational Control in Combinatory Categorial Grammar*. Ph.D. thesis, University of Edinburgh.

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

Chris Barker and Chung-chieh Shan. 2006. Types as graphs: Continuations in type logical grammar. *Journal of Logic, Language and Information* 15:331–370.

Chris Barker and Chung-chieh Shan. 2014. *Continuations and Natural Language*. Oxford Studies in Theoretical Linguistics.

Johan Bos. 2003. Implementing the binding and accommodation theory for anaphora resolution and presupposition projection. *Computational Linguistics* 29(2):179–210. https://aclweb.org/anthology/J/J03/J03-2002.pdf.

Simon Charlow. 2014. *On the semantics of exceptional scope*. Ph.D. thesis, New York University.

Stephen Clark and James R. Curran. 2007. Wide-Coverage Efficient Statistical Parsing with CCG and Log-Linear Models. *Computational Linguistics* 33(4):493–552. https://aclweb.org/anthology/J/J07/.

Ann Copestake, Dan Flickinger, Carl Pollard, and Ivan Sag. 2005. Minimal recursion semantics: An introduction. *Research on Language and Computation* 3:281–332.

Jason Eisner. 1996. Efficient normal-form parsing for Combinatory Categorial Grammar. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Santa Cruz, California, USA, pages 79–86. https://doi.org/10.3115/981863.981874.

Donka F. Farkas and Anastasia Giannakidou. 1996. How clause-bounded is the scope of universals? In *Proceedings of Semantics and Linguistic Theory*. Cornell University, volume 6, pages 35–52.

Danny Fox and Uli Sauerland. 1996. Illusive scope of universal quantifiers. In *Proceedings of the North Eastern Linguistic Society (NELS)*. volume 26, pages 71–86.

Claire Gardent and Laura Kallmeyer. 2003. Semantic construction in feature-based TAG. In *Proceedings of EACL-03*.

Julia Hockenmaier and Yonatan Bisk. 2010. Normal-form parsing for Combinatory Categorial Grammars with generalized composition and type-raising. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*. Coling 2010 Organizing Committee, Beijing, China, pages 465–473. http://www.aclweb.org/anthology/C10-1053.

Graham Hutton and Erik Meijer. 1996. Monadic Parser Combinators. Technical Report NOTTCS-TR-96-4, Department of Computer Science, University of Nottingham.

Hans Kamp and Uwe Reyle. 1993. *From Discourse to Logic: An Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and DRT*. Kluwer, Dordrecht, The Netherlands.

Alexander Koller, Joachim Niehren, and Stefan Thater. 2003. Bridging the gap between underspecification formalisms: Hole semantics as dominance constraints. In *Proceedings of EACL-03*.

Richard Larson. 1985. Quantifying into NP. MIT Manuscript.

Kenton Lee, Mike Lewis, and Luke Zettlemoyer. 2016. Global neural CCG parsing with optimality guarantees. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Austin, Texas, pages 2366–2376. https://aclweb.org/anthology/D16-1262.

Roger Levy and Galen Andrew. 2006. Tregex and Tsurgeon: tools for querying and manipulating tree data structures. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*. European Language Resources Association (ELRA). http://aclweb.org/anthology/L06-1311.

Mike Lewis and Mark Steedman. 2014. Improved CCG parsing with semi-supervised supertagging. *Transactions of the Association of Computational Linguistics* 2:327–338. http://aclweb.org/anthology/Q14-1026.

Rebecca Nesson and Stuart M. Shieber. 2006. Simpler TAG semantics through synchronization. In *Proceedings of the 11th Conference on Formal Grammar*.

Sylvain Pogodalla and Florent Pompigne. 2012. Controlling extraction in abstract categorial grammars. In *Formal Grammar*. Springer, pages 162–177.

Tanya Reinhart. 1997. Quantifier scope: How labor is divided between QR and choice functions. *Linguistics and philosophy* 20(4):335–397.

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

Chung-chieh Shan and Chris Barker. 2006. Explaining crossover and superiority as left-to-right evaluation. *Linguistics and Philosophy* 29(1):91–134.

Mark Steedman. 2000. *The syntactic process*. MIT Press, Cambridge, MA, USA.

Mark Steedman. 2012. *Taking Scope: The Natural Semantics of Quantifiers*. MIT Press, Cambridge, MA, USA.

Kristen Syrett. 2015. Experimental support for inverse scope readings of finite-clause-embedded antecedent-contained-deletion sentences. *Linguistic Inquiry* .

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

$$
\begin{aligned}
\left(\frac{m_\nu \multimap [\,]}{f\;\nu}\right)^{\downarrow\uparrow} &= (m_\nu \multimap (f\;\nu)^\eta)^\uparrow && \downarrow \\
&= \frac{(m_\nu \multimap (f\;\nu)^\eta)_u \multimap [\,]}{u} && \uparrow \\
&= \frac{m_\nu \multimap (f\;\nu)_u^\eta \multimap [\,]}{u} && \text{Assoc} \\
&= \frac{m_\nu \multimap [\,]}{f\;\nu} && \text{LeftID}
\end{aligned}
$$

Figure 8: Side Effects Not Affected By Reset (Charlow, 2014, Fact 4.1)

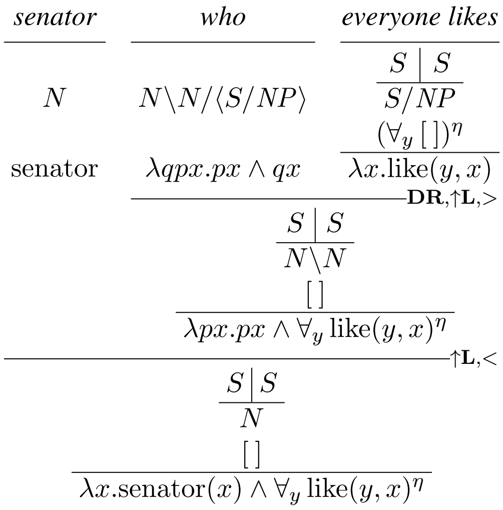


Figure 9: Relative Clause Example

A Supplemental Material

A.1 Exceptional Scope in the Penn Treebank

As noted in Section 2, the empirical status of scope islands remains unsettled, with further corpus-based and experimental work necessary to adequately characterize the distribution of true quantifiers. Nevertheless, a search of the Penn Treebank reveals that if scope islands do not represent hard constraints, then violations are at least very rare. We used Tregex (Levy and Andrew, 2006) to search the Wall Street Journal portion of the Penn Treebank with the pattern

```
SBAR << /MD|VBD|VBP|VBZ/ << /^every|^each/
```

and found that only 385 finite subordinate clauses contain (a form of) *every* or *each*, including 80 relative clauses and just 9 conditionals, with none showing clear evidence of the universal scoping out of the finite clause. There were, however, a couple of potential counter-examples, such as 7, that appear amenable to an analysis involving functional readings, rather than exceptional scope; these deserve further study.

(7) Tandy said its experience during the shortage didn't merit the $5 million to $50 million investment$_i$ <U.S. Memories is seeking from each$_i$ investor>.

By contrast, exceptionally scoping indefinites are quite easy to find.

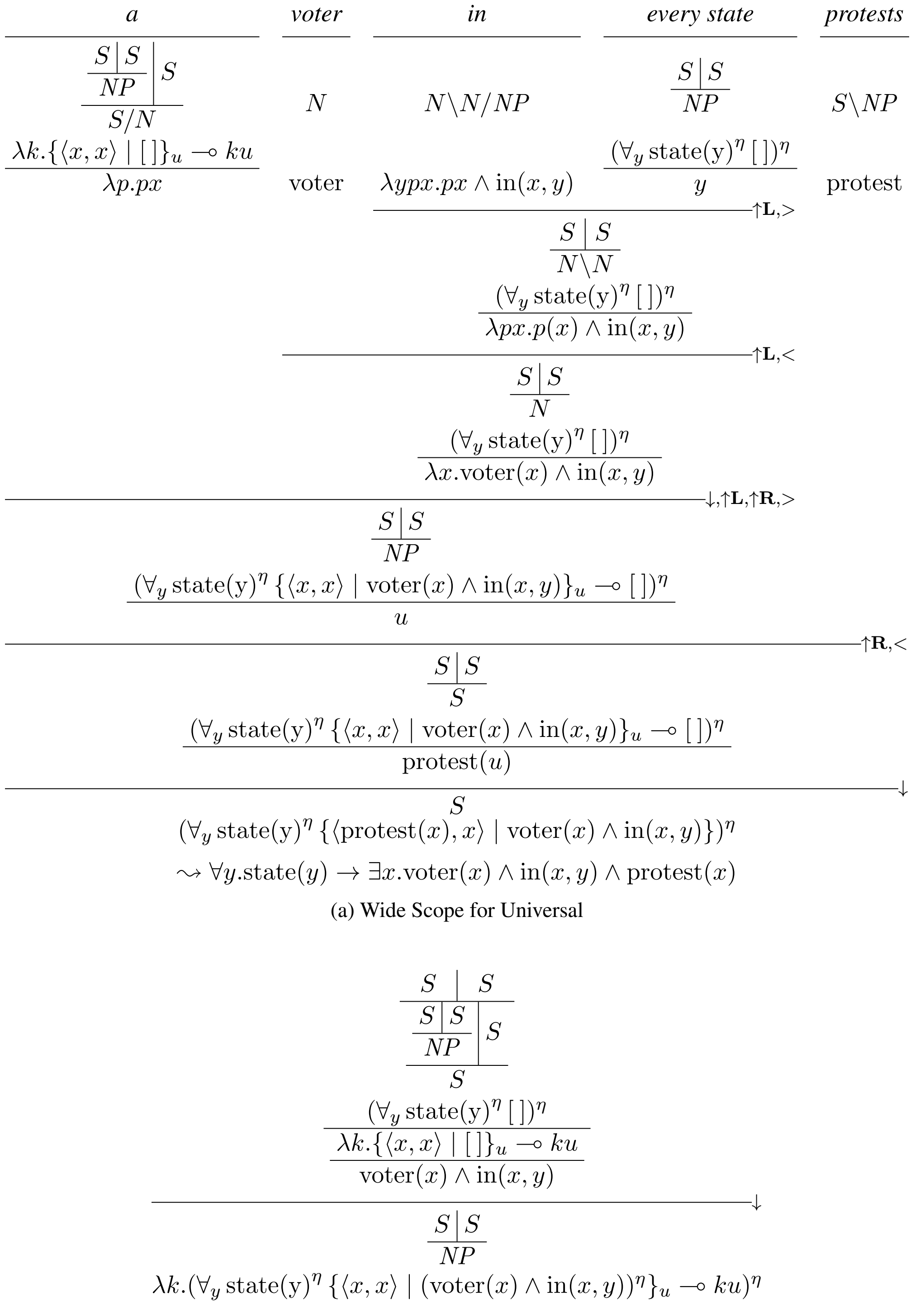
A.2 Side Effects and Reset

Figure 8 reproduces Charlow's (2014) proof that in the general case, side effects in an underlying monad are not affected by reset if the lift and lower operations in the continuized grammar are identifed with the monad's sequencing ($\multimap$) and injection (η) operators.

A.3 Relative Clauses and Inverse Linking

Figure 9 gives an example of a relative clause scope island. The category for the relative pronoun requires its clausal argument to be delimited, triggering a reset via the Delimit Right (**DR**) rule, which closes off the semantic scope for *everyone*. Not shown is the derivation of the base category S/NP for *everyone likes*, which can be derived using standard CCG rules on the bottom without invoking empty string elements. The lowering rule for incomplete clauses is required in order for this base category to be lowerable.

By contrast, Figure 10 shows how inverse scope goes through for the nominal PP *in every state*, since the preposition category does not require its argument to be delimited. One-fell-swoop result lowering implements Larson's (1985) constraint barring interleaved inverse scope out of NPs while preserving the ability of the universal to bind subsequent pronouns. Although Steedman's (2012) account handles examples such as the one in Figure 10, where the inversely linked PP is right peripheral, his treatment—unlike the present one—cannot handle examples such as *few voters$_i$ [in every state] who$_i$ supported Trump participated in the protests* where the inversely linked PP is in medial position. (Note that the relative clause here must be interpreted restrictively, and thus is not tenable as an appositive, contra Steedman's suggested analysis of related examples.)



a *voter* *in* *every state* *protests*
$\frac{\frac{S|S}{NP}\,\big|\,S}{S/N}$ N $N\backslash N/NP$ $\frac{S|S}{NP}$ $S\backslash NP$
$\frac{\lambda k.\{\langle x,x\rangle \mid [\,]\}_u \multimap ku}{\lambda p.px}$ voter $\lambda ypx.px \wedge \text{in}(x,y)$ $\frac{(\forall_y\,\text{state(y)}^\eta\,[\,])^\eta}{y}$ protest
↑L,>
$\frac{S|S}{N\backslash N}$ $\frac{(\forall_y\,\text{state(y)}^\eta\,[\,])^\eta}{\lambda px.p(x) \wedge \text{in}(x,y)}$
↑L,<
$\frac{S|S}{N}$ $\frac{(\forall_y\,\text{state(y)}^\eta\,[\,])^\eta}{\lambda x.\text{voter}(x) \wedge \text{in}(x,y)}$
↓,↑L,↑R,>
$\frac{S|S}{NP}$ $\frac{(\forall_y\,\text{state(y)}^\eta\,\{\langle x,x\rangle \mid \text{voter}(x) \wedge \text{in}(x,y)\}_u \multimap [\,])^\eta}{u}$
↑R,<
$\frac{S|S}{S}$ $\frac{(\forall_y\,\text{state(y)}^\eta\,\{\langle x,x\rangle \mid \text{voter}(x) \wedge \text{in}(x,y)\}_u \multimap [\,])^\eta}{\text{protest}(u)}$
↓
S
$(\forall_y\,\text{state(y)}^\eta\,\{\langle \text{protest}(x),x\rangle \mid \text{voter}(x) \wedge \text{in}(x,y)\})^\eta$
$\rightsquigarrow \forall y.\text{state}(y) \rightarrow \exists x.\text{voter}(x) \wedge \text{in}(x,y) \wedge \text{protest}(x)$
$\frac{S \,\big|\, S}{\frac{\frac{S|S}{NP}\,\big|\,S}{S}}$
$\frac{(\forall_y\,\text{state(y)}^\eta\,[\,])^\eta}{\frac{\lambda k.\{\langle x,x\rangle \mid [\,]\}_u \multimap ku}{\text{voter}(x) \wedge \text{in}(x,y)}}$
↓
$\frac{S|S}{NP}$
$\lambda k.(\forall_y\,\text{state(y)}^\eta\,\{\langle x,x\rangle \mid (\text{voter}(x) \wedge \text{in}(x,y))^\eta\}_u \multimap ku)^\eta$

(a) Wide Scope for Universal

(b) Lowering Result Tower

Figure 10: Inverse Linking Example