Submitted version; for published version see J. of Logic, Language and Information (2023). https://doi.org/10.1007/s10849-023-09412-

Authors: Gene Louis Kim, Mandar Juvekar, Junis Ekmekciu, Viet Duong, and Lenhart Schubert

Monotonic Inference with Unscoped Episodic Logical Forms: from Principles to System

Abstract

We describe the foundations and the systematization of natural logic-like monotonic inference using Unscoped Episodic Logical Forms (ULFs) that Kim et al. [1, 2] introduced and first evaluated. In addition to providing a more detailed explanation of the theory and system, we present results from extending the inference manager to address a few of the limitations that Kim et al.'s [2] naive system has. Namely, we add mechanisms to incorporate lexical information from the hypothesis (or goal) sentence, enable the inference manager to consider multiple possible scopings for a single sentence, and match against the goal using English rather than the ULF.*

Keywords: Monotonic Inference, Natural Logic, Episodic Logic, Unscoped Logical Form, Automated Reasoning

1 Introduction

Unscoped Logical Form (ULF) is an underspecified form of Episodic Logic (EL), an extended first-order logic designed to closely match the form and expressivity of natural language [3, 4]. ULFs fully capture the semantic type structure of corresponding EL formulas while leaving scope, anaphora, and word sense ambiguities unresolved. While presenting ULF, Kim and Schubert [3] proposed that ULF is suitable for five classes of inferences, namely monotonic inferences, inferences based on clause-taking verbs, inferences based on counterfactuals, inferences from questions, and inferences from requests. The suitability of ULF for each of these classes of inferences has since been experimentally demonstrated by Kim et al. [2], who demonstrated the monotonic inferences on the FraCaS dataset [5], and Kim et al. [6], who demonstrated the remaining categories of inferences in a dataset discourse-oriented sentences that regularly give rise to these phenomena.

Kim et al.'s [2] monotonic inference system for ULFs was based on a proof-based formalism described by Kim et al. [1]. This approach was formalized via correspondence to Sánchez Valencia's [7] treatment of natural logic which uses Lambek cum Permutation Calculus [8], ensuring that the ULF-based inference method can handle the same set of natural logic inferences as this system from past literature. The full picture of the ULF monotonic inference system is fragmented across two workshop papers, one describing the theoretical inference framework with many details tucked away in the appendix [1], and another focusing on how a feasible inference system was built based on the theoretical framework [2]. Many details of the relationship between the theoretical rules and the implemented system are left implied by citations and the accompanying codebase due to limitations of the publication format.

In this paper, we present a unified description of the theoretical framework and the implementation of Kim et al.'s [1] monotonic inference system making clear the correspondence between the theoretical rules and how they are realized in practice. In addition, we present methods to overcome a few limitations in this proof-of-concept inference system—steps that are necessary (but not sufficient) to turn this inference system into a more mature, competitive inference engine. We measure the effect of these changes on the performance of the same dataset as Kim et al. [2] and discuss what these results suggest about the next steps in developing a robust ULF monotonic inference system.

2 Background

Natural logic is an approach to logical inference based on directly accessing natural language—more specifically, using the syntactic structure and knowledge of the semantic properties of the lexical items and local constructions [9, 10]. Monotonicity calculus is an important fragment of natural logic that uses that systematically applies the knowledge of language elements that act as monotone functions over the syntactic structure of a sentence to identify global polarity contexts. These polarity contexts then mediate specific entailment conditions based on the properties of the monotone functions. Figure 1 shows the three basic cases of monotonicity inference, upward, downward, and non-monotone contexts leading to different entailment conditions. A

Upward	*The train* (*departed the station according to schedule*)$^{\blacktriangle}$ $\Rightarrow$ *The train departed the station*
Downward	*I have never* (*cycled*)$^{\blacktriangledown}$ *before* $\Rightarrow$ *I have never cycled in the Tour de France before*
Non-monotone	*Exactly 5 friends went on a* (*cruise*)$^{\blacksquare}$ $\not\Leftrightarrow$ *Exactly 5 friends went on a* (*transpacific cruise*)$^{\blacksquare}$

Fig. 1: Examples of sentence pairs with upward entailment $(\dots)^{\blacktriangle}$, downward entailment $(\dots)^{\blacktriangledown}$, and non-monotone $(\dots)^{\blacksquare}$ relationships.

monotone function operates over partially ordered sets and either preserves or inverts the ordering of argument values. More precisely, for an upward monotone function f, $x \leq y$ implies $f(x) \leq f(y)$. Similarly, for a downward monotone function g, $x \leq y$ implies $g(x) \geq g(y)$. If neither of these implications hold for a function h, h is said to be non-monotone. When the ordering of the arguments and function results describe subset relations and entailments of expressions in language, monotonicity can be a tool for making natural language inferences. For instance, consider the second example in Figure 1. *Never* is downward monotone in entailment since it flips the entailment ordering of (1) *I have cycled in the Tour de France before* entails (2) *I have cycled before* to (2′) *I have* ***never*** *cycled before* entails (1′) *I have* ***never*** *cycled in the Tour de France before*.

Kim et al.'s [6] discourse-related inferences using ULFs are made from manually annotated ULFs using symbolic meta-axioms generalized to handle syntactic idiosyncrasies and achieved reasonable precision on a multi-genre dataset. Kim et al. [2] then complemented this by demonstrating Natural Logic-like inferences using ULFs on the general quantifiers section of the FraCaS dataset, also incorporating an automatic ULF parser. Unlike most other work in these domains, both of these systems perform forward inferences, rather than goal-directed inference. This distinction is important because of the authors' interest in generating inferences spontaneously in discourse where the desired inferences are not known beforehand. Furthermore, they focus on demonstrating that such inferences are possible using ULFs, not on building competitive systems for specific benchmarks, so their inference systems have some significant limitations. We investigate methods for overcoming the following two limitations of Kim et al.'s [2] system.

Limitation 1. The inference manager only considers the premises when determining the relevant lexical relationships to introduce into the search process.

Limitation 2. The inference manager only considers a single possible scoping of the unscoped logical form.

Limitation 1 is a consequence of the authors' commitment to a forward inference strategy, which cannot possibly anticipate all ways in which new vocabulary might appear in the entailments of the premises. For example, given the appearance of *few committee members* in the premises, the forward inference process cannot be expected to enumerate various modified versions of this phrase, such as *few female committee*

members from Scandinavia, etc. Therefore we make use of modifier information from the goal hypothesis in the updated experiments (see Section 5.1).

Limitation 2 is a more substantive limitation. This assumes that the polarity management mechanisms, both the initial labeling and the propagation method, are free of mistakes. Conceptually, supporting multiple polarity annotations in the inference process can be as simple as running the inference process over ULF–scoping pairs rather than only the ULFs. However in practice, the engineering optimizations in the implementation of the inference system (which are incompatible with this change) requires the development of similar optimizations to have a computationally feasible system (see Section 5.2).

2.1 Automated Monotonicity Inference

Automated natural logic inference systems development—distinct from general natural language inference—is an active area of research [11–18]. In order to evaluate our monotonicity-specific inference system with minimal external resources fairly, we focus on the FraCaS dataset [5]. FraCaS was carefully curated to include specific, technical, inference relationships, including monotonicity-based entailments. In Section 7, we summarize Kim et al.'s [2] results against a few notable systems that were previously evaluated on the same parts of the FraCaS dataset: Mineshima et al. [13], Abzianidze [14], Hu et al. [15], and Haruta et al. [16] to contextualize the system performance in the wider literature, despite this not being a state-of-the-art (SOTA) system. This sets the stage to investigate the scale of improvements that we get from overcoming specific limitations in the naive ULF inference system.

Mineshima et al. [13] and Abzianidze [14] extend first-order lambda logical forms with higher-order terms (e.g., most, many, half of, etc.) and augment first-order inference with rules geared towards those terms. Haruta et al. [16] achieve SOTA performance by employing degree and event semantics to approximate key higher-order logic features presented in different linguistic phenomena. Hu et al. [15] differs from the others by running directly on the natural language text, with a combinatory categorical grammar (CCG)-based monotonicity labeling system. Apart from our reliance on forward inference, our approach most resembles Hu et al. [15] based on our shared use of relatively compact sets of monotonic inference rules that operate over logical forms that resemble the form and expressiveness of natural language.

Monotonicity inference is also being investigated in the context of large neural models—whether large neural models effectively encode monotonicity relations. This includes both those trained on natural language inference tasks [19] and those pretrained on large corpora of unstructured text [20]. They find that despite their statistically strong performance, neural models of both types fail to reliably capture polarity information and monotonicity inferences.

3 Theoretical Inference Method

Kim et al.'s [2] system is based on a proof-based inference method described by Kim et al. [1] that uses ULF as the base semantic representation. This proof-based method uses inference rules for ULFs that correspond directly to inference rules in

Sánchez-Valencia's [10] formulation of Natural Logic. Here we list the inference rules formulated by Kim et al. [1].[1] Please note that where polarity contexts are necessitated by operators present in the formulas, the polarity markings are omitted for clarity, e.g., every.d imposes a negative polarity on its restrictor and a positive polarity on its body.

Theory Rule 1 (Monotonicity, UMI).

$$\text{(1.1)}\qquad \frac{\phi[P1^{\blacktriangle}],\ ((\text{every.d } P1)\ (\text{be.v } (=\ (\text{a.d } P2))))}{\phi[P2]}$$

$$\text{(1.2)}\qquad \frac{\phi[P2^{\blacktriangledown}],\ ((\text{every.d } P1)\ (\text{be.v } (=\ (\text{a.d } P2))))}{\phi[P1]}$$

This is the main rule for making monotonicity-based inferences.

Theory Rule 2 (Conversion, UCI).

$$\frac{((\delta_1\ P)\ (\text{be.v } (=\ (\delta_2\ Q))))}{((\delta_1\ Q)\ (\text{be.v } (=\ (\delta_2\ P))))} \quad \begin{array}{l}\text{where } \delta_1 \in \{\text{some.d, a.d, no.d}\}\\ \text{and } \delta_2 \in \{\text{some.d, a.d}\}.\end{array}$$

Theory Rule 3 (Polarity Marking, PM).

$$\frac{\phi[\psi]}{\phi[\psi^{\#}]} \quad \begin{array}{l}\text{where `\#' } \in \{\blacktriangle, \blacktriangledown, \blacksquare\} \text{ based on the polarity marking from}\\ \text{the corresponding SLF, } \phi'[\psi'^{\#}].\end{array}$$

Polarities are computed respective to specific scopings of ULFs—in the form of scoped logical forms (SLFs)—then propagated back to the ULFs to enable inferences that are contingent on the polarity context. When converting a ULF to an SLF, each scoping operator must be lifted to a valid position for a scope operator. These positions are any that are operating on well-formed formulas, embedded or not, and which are not within the original operand of the scoping operator position in the ULF (that is it must be "lifted"). For most scoping operators, this process simply places the operator in a new position. Determiners, however, have a more complicated scoping process. For a ULF, ψ, which contains a quantified expression $\wr$ of form the $(\delta\ \pi)$ where δ is a determiner and π is a predicate, the corresponding formula with $(\delta\ \pi)$ at the top-level scope is $(\delta\ x\text{: } (x\ \pi)\ \psi[\wr/x])$. That is, the restrictor must be lifted alongside the determiner, and the original position of $\wr$ is replaced by a newly introduced, unique variable. Each possible permutation of scope operator placements results in a unique SLF corresponding to a ULF.

From the SLF, the global polarity contexts are computed. Using lexical knowledge regarding monotonicity properties, SLFs are marked with local entailment contexts (the local entailment direction is based only on each expression's parent). From this, the global polarity marking is computed using a simple recursive algorithm.

1. Node a has non-monotone polarity ($\blacksquare$) if any node in the path from the root to a is marked with "$\circ$" (the local analog of $\blacksquare$).
2. Else, node a has negative polarity ($\blacktriangledown$) if there are an odd number of nodes between the root and a (inclusive) marked with "$-$" (the local analog of $\blacktriangledown$).
3. Otherwise, node a has positive polarity ($\blacktriangle$).

[1] For the correspondence of these rules to Sánchez-Valencia's [10] natural logic, please see Kim et al.'s [1] explanation.

Notice that two different SLFs may (but not necessarily) result in differing polarity annotations for the same ULF depending on how the monotonicity operators scope in the two SLFs.

Theory Rule 4 (Negation Introduction/Elimination, NI/NE).

$$(4.1)\quad \frac{\phi}{(\text{not } (\text{not } \phi))} \qquad (4.2)\quad \frac{(\text{not } (\text{not } \phi))}{\phi}$$

Theory Rule 5 (Negation-Quantifier Identities, NQ).

$$(5.1)\quad \frac{\phi[(\text{not } ((\text{some.d } P1)\ P2))]}{\phi[((\text{no.d } P1)\ P2)]} \qquad (5.2)\quad \frac{\phi[((\text{no.d } P1)\ P2)]}{\phi[(\text{not } ((\text{some.d } P1)\ P2))]}$$

$$(5.3)\quad \frac{\phi[(\text{not } ((\text{a.d } P1)\ P2))]}{\phi[((\text{no.d } P1)\ P2)]} \qquad (5.4)\quad \frac{\phi[((\text{no.d } P1)\ P2)]}{\phi[(\text{not } ((\text{a.d } P1)\ P2))]}$$

In addition to basic negation introduction and elimination rules, Kim et al. [1] provide useful rules regarding the interactions between negation and some common quantifiers. The exact rules that the implemented inference system uses and a discussion of their relationship to the theory rules presented in this section is given in Section 4.4.

Below is a basic inference example from Kim et al. [1], which demonstrates the use of SLF for polarity computation and a simple use of the UMI inference rule as described in this section.[2]

Basic Monotonicity Example with ULF

1. (|Abelard| (see.v (a.d carp.n)))	Assumption
2. ((every.d carp.n) (be.v (= (a.d fish.n))))	Assumption
3. (a.d x: (x carp.n)▲ (|Abelard| (see.v x)▲)▲)	SLF of 1. w/ polarity
4. (|Abelard| (see.v (a.d carp.n▲)))	Polarity marking 1.,3.
5. (|Abelard| (see.v (a.d fish.n)))	UMI 2.,4.

See Kim and Schubert [3] or Kim et al. [1] for explanations of the syntactic conventions of ULF, such as the type-designating suffixes (e.g., .d, .v, .n, etc.).

4 Baseline System Description

We start our discussion of Kim et al.'s [2] system with an inference example that demonstrates some of the ways in which the system differs from the underlying theoretical framework presented by Kim et al. [1]. This example is problem 24 from the FraCaS dataset. The formulas and annotations are based on the actual output of the automatic system. The UMI rules (Theory Rule 1) generalize *every A is a B* to equivalent universal quantification forms (in this case *all As are Bs*), our polarity marking method circumvents the need for SLFs (Sections 4.3 and 4.4.1), and we have rules to generate monotonicity relations from intersective predicate modification (Section 4.4).

[2] The computation of global polarity via local entailment context propagation and irrelevant polarity marking symbols are omitted for brevity and clarity.

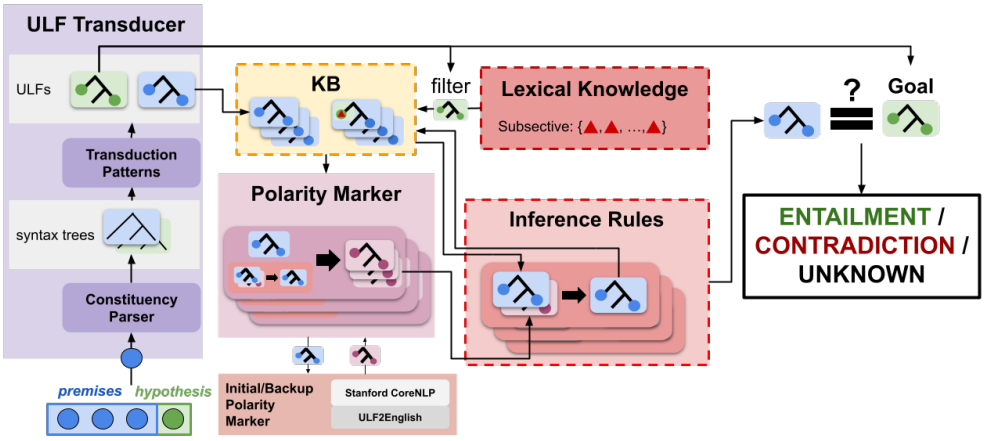

Fig. 2: A diagram of the inference system component dependencies. The notable differences when compared to Kim et al.'s [2] system are (1) the introduction of goal-filtered lexical knowledge to the knowledge base (Section 5.1) and (2) the production of multiple possible polarity markings for each ULF formula in polarity marker component (Section 5.2).

Inference Example (FraCaS Problem 24)

1. *Many delegates obtained interesting results from the survey*	Premise (English)
2. *Many delegates obtained results from the survey*	Hypothesis (English)
3. ((many.d (plur delegate.n)) ((past obtain.v) (k (interesting.a (plur result.n))) (adv-a (from.p (the.d survey.n)))))	Premise (ULF), 1.
4. ((many.d (plur delegate.n)) ((past obtain.v) (k (plur result.n)) (adv-a (from.p (the.d survey.n)))))	Hypothesis (ULF), 2.
5. ((all.d (interesting.a (plur result.n))) ((pres be.v) (= (k (plur result.n)))))	Extract intersective modifier relation, 3.
6. ((many.d (plur delegate.n)■) ((past obtain.v) (k (interesting.a (plur result.n))▲) (adv-a (from.p (the.d survey.n)))))	Polarity marking 3.
7. ((many.d (plur delegate.n)) ((past obtain.v) (k (plur result.n)) (adv-a (from.p (the.d survey.n)))))	UMI 5.,6.
8. **Entailment**	Exact Match, 4.,7.

As demonstrated by this example, the inference system starts with a set of premise sentences and a hypothesis sentence in English which are first automatically converted to ULF before starting the proof-based inference process to determine whether there is an *entailment*, *contradiction*, or *unknown* relationship between the premises and the hypothesis. This inference process uses a forward search from the premises.

This system simplifies the theoretical framework in two ways to reduce the search space and to make for a sufficiently fast inference process. For one, only one SLF and

polarity annotation is computed for each ULF which is assumed to be the correct scoping. In addition, variations of the monotonicity and conversion inference rules are introduced to support specific ULF macros and common syntactic constructions. Such variations lead to a more direct inference process by using a single inference step to handle a two-step inference in the theoretical framework.

The generalization of Theory Rule 1 (UMI) to other forms of universal quantification is a simple example of this second simplification. A more interesting example is the simplification of the monotonicity relation extraction rules, which themselves are optimizations to the search process as explained in Section 4.4. The intersective prenominal modification extraction rule (System Rule 9) identifies an intersective prenominal modification, such as the phrase *a sneaky panther*, and extracts the monotonicity relations *every sneaky panther is a panther* and *every sneaky panther is sneaky*. The most basic theoretical form of this would first require an expansion of *a sneaky panther* to *something that is sneaky and that is a panther*, from which the monotonicity relations can be extracted based on properties of logical conjunction. The system instead directly extracts the entailments upon recognition of the intersective nature of the modification.

Component 1. Heuristic-based inference search (Section 4.1)

Component 2. A ULF transducer (Section 4.2)

Component 3. A global polarity marking function (Section 4.3)

Component 4. Inference rules with polarity propagators (Section 4.4)

Algorithm 1 Heuristic search. Inference rules map a set ULF premises to a set of ULF inferences.

Inputs: Φ, a set of premises; ψ, a goal ULF; h, a heuristic function; M, a search depth limit.
Outputs: The entailment classification.
Global Constants: U, a list of unary rules; B, a list of binary rules; ε, a small positive number; c, a step count for search method change.
Procedure:
Initialize $n \leftarrow 0$, KB $\leftarrow \Phi$.
Initialize $Q_{\text{h}} \leftarrow$ empty priority queue.
Initialize $Q_{\text{bfs}} \leftarrow$ empty basic queue.
Initialize $Q \leftarrow Q_{\text{h}}$.
Initialize $Q_{\text{other}} \leftarrow Q_{\text{bfs}}$.
loop
  If $n > M$ or $Q = \varnothing$, **return** UNKNOWN.
  If $\psi \in$ KB, **return** ENTAILMENT.
  If $\neg\psi \in$ KB, **return** CONTRADICTION.
  $\nu \leftarrow Q$.pop().
  $t_{\text{unary}} \leftarrow U \times \nu$.
  $t_{\text{binary}} \leftarrow B \times ((\nu \times \nu) \cup (\nu \times \text{KB}) \cup (\text{KB} \times \nu))$.
  Push all results x of computing the tuples in t_{unary} and t_{binary} that are not contained in KB to Q_{h} with key $h(x) + n\varepsilon$ and Q_{bfs}.
  KB $\leftarrow$ KB $\cup\, \nu$.
  $n \leftarrow n + 1$.
  if $n \mod c = 0$ **then**
    tmp $\leftarrow Q$.
    $Q \leftarrow Q_{\text{other}}$.
    $Q_{\text{other}} \leftarrow$ tmp.
  end if
end loop

Figure 2 shows a diagram of the component dependencies. While most of the inference system is symbolic, the initial constituency parses and initial polarity marking—used for ULF transduction and scope selection, respectively—are computed using NN and

ML methods. Furthermore, the ML-based polarity marking is used when the symbolic polarity propagation methods fail or take too long.

4.1 Search Process

The inference process is guided by a heuristic-based forward search. Algorithm 1 shows this process in detail. The heuristic-guided search uses a heuristic of the distance between an arbitrary formula and the goal formula. While this heuristic search typically runs quickly, it does not guarantee completeness of the search process and its performance is highly dependent on the quality of the heuristic function. Rather than investing time into finding the optimal heuristic function, Kim et al. [2] chose to interleave heuristic search steps with breadth-first search steps (BFS). The heuristic function estimates the distance from an arbitrary formula, u, to the goal formula. The search process additionally gives preference to formulas reached earlier in the search process in cases of heuristic ties by adding a small positive number, ϵ, to the heuristic value at each step. A hyperparameter, c, sets the frequency at which the search process alternates between heuristic search and BFS. The specific values used in the experiments are detailed in Section 6.

4.2 ULF Transducer

The English-to-ULF parsing is done by using a constituency parser then transducing the constituency tree into a ULF. Specifically, Kim et al. [2] use the Berkeley neural parser [21] to obtain constituency trees.[3] The following transduction into ULF follows a series of simple correspondences from the phrase structure and POS tags to ULF expressions. This process resembles the parsing processing in the initial stages of past transduction-based EL parsers [22–25].[4] The existing neural network ULF parser [27] was not used here because sentences in monotonicity datasets tend to be fairly short and follow written English syntax. This plays into the strengths of the transduction parser which can reliably handle simple cases and have more predictable errors that the inference system may recover from.

4.3 Polarity Marking

Kim et al. [2] rely on the Natlog and NaturalLI systems [11, 28] for the initial polarity marking problem as well as a recovery tool in cases of polarity propagation failure.[5] ULFs first are converted to English because Natlog and NaturalLI operate on English, not ULFs or general tree structures. This is done using the ULF2English system [6] and an alignment between ULF expressions and the English words are computed using subroutines of the ULF2English system.

The polarity markings on the English words are mapped to the ULFs using the alignment. This however only provides polarity markings at the word level. In order to extrapolate the polarity marking for every ULF subexpression, all SLF possibilities of

[3] The version 0.2.0 release and the `benepar_en3` model available at https://github.com/nikitakit/self-attentive-parser/.

[4] The transduction rules are written in a combination of the tree-to-tree transduction language [26] and a simplified variant.

[5] This is available through the Natural Logic component of Stanford CoreNLP.

the ULF are generated, the corresponding polarity markings are inferred for each SLF using the process described in Section 3 and a manually curated set of negative polarity operators. The SLF which leads to the fewest polarity annotation discrepancies when compared to the NatLog polarity labels is selected. The inference rules propagate the polarities so this is typically only performed on the input sentences (Section 4.4.1).

Computing possible SLFs requires an account of island constraints. These constraints are imperfectly modeled with the following rule: *Scoping operators cannot scope outside of ancestors that are ULF type-shifters.* This rule handles complex modifiers (which are shifted from predicates to modifiers) and reified clausal complements (e.g., *I believe that everyone thinks.*) and is simple to implement within the ULF type system. This rule leads to a loss in expressive capacity as it ignores known exceptions [29] and the *de dicto* / *de re* distinction for clausally-embedded indefinite quantifiers [30, 31].

4.4 System Inference Rules

The inference rules in the system fall under five broad categories. There are 9 total inference rules when accounting for specializations for macros—though some of these inference rules themselves include several distinct transduction patterns to account for less substantial syntactic variations. Here we list the broad categories and the specific inference rules in our system for each category.

A. **Monotonicity Substitution.** This is the core monotonicity inference. Given the premise *Every A is a B*, *B* is substituted for *A* in positive polarity contexts and *A* is substituted for *B* in negative polarity contexts. In order to reduce the proof lengths, we suppress ULF macro expansion rules and extract monotonicity relations directly from macro instances.

   System Rule 1 (Positive Monotonicity Substitution, UMI▲). Direct implementation of Theory Subrule 1.1. UMI where the replaced expression is in a positive polarity context.

   System Rule 2 (Negative monotonicity substitution, UMI▼). Direct implementation of Theory Subrule 1.2. UMI where the replaced expression is in a negative polarity context.

   System Rule 3 (Monotonicity-based quantifier substitution). Some quantifiers also have relationships that can lead to polarity-dependent substitutions. For example, *the* and *both* can be replaced with *a* in positive polarity contexts and the reverse is true in negative polarity contexts. In our system, we generalize this to *all*, *every*, *each* $\leq$ *a* under the assumption that the domain of individuals is not empty.

B. **Conversion.** Inferences that swap the restrictor and body predicates in existential quantification.

   System Rule 4 (Conversion). Direct implementation of Theory Rule 2. For example, *Some artwork is a painting* $\Leftrightarrow$ *Some painting is an artwork.*

C. **Conservativity.** This category of inferences is based on the widespread conservativity feature in natural language quantifiers. The FraCaS dataset includes examples targeting this inference and this inference rule is also commonly used for introducing and eliminating relative clauses in simple quantified expressions.

System Rule 5 (Conservativity Transductions). We implement this rule using two different transduction patterns. One based on relative clauses, δ *As are Bs* $\Leftrightarrow$ δ *As are As that/who are Bs*, where δ is a determiner. And another based on existential "there" constructions, δ *As are Bs* $\Rightarrow$ *there is an A that is a B.*

D. **Equivalences.** Substituting equivalent words or constructions for each other.

System Rule 6 ("No" to Negated Indefinite). Replacing ϕ[no.d] with (not $\phi[d1]$) where $d1 \in \{\text{some.d}, \text{a.d}, \text{an.d}\}$.

System Rule 7 (Equivalent Quantifier Substitution). This inference rule substitutes between the equivalent forms of existential (some, a), universal (all, every, each, *simple generics*), and numerical (one.d == a.d) quantifiers. For example, *Every genre has merit* $\Leftrightarrow$ *All genres have merit*)

E. **Search Optimizations.** A couple of inference rules were introduced as search optimizations. They allow us to avoid producing possible intersective relationships for the entire known lexicon. Instead focusing in on only intersective relationships that appear in our problem. We also avoid expanding out intersective modifiers as lambda functions of conjunctions and using separate inference rules to process logical conjuncts. Intersective modifications are identified in formulas in the knowledge base (premises + inferred formulas) and formulas describing these intersective relationships are directly constructed. These formulas are then used with UMI to generate equivalent inferences to a generic conjunction-based system without the need to introduce rules for managing explicit predicate conjunctions.

System Rule 8 (Intersective Postnominal Modifier Rule Extraction). This rule identifies post-nominal modifiers that are intersective in a formula and extracts a formula that can be used in conjunction with UMI to generate appropriate inferences. For example, from the postmodified noun phrase *a mouse in the walls* this rule directly extracts the entailments *every mouse in the walls is a mouse* and *every mouse in the walls is in the walls.*

System Rule 9 (Intersective Prenominal Modifier Rule Extraction). This rule is the prenominal modification analog of System Rule 8. For example, from the phrase *a sneaky panther*, this rule extracts the entailments *every sneaky panther is a panther* and *every sneaky panther is sneaky.*

The intersective modifiers are identified using a non-subsective adjective list [32] expanded to all members of WordNet [33] synsets that match a word in the list.

4.4.1 Polarity Propagation

Polarity propagation functions, and even more so polarity marking (Section 4.3) are costlier than running the inference rules themselves. The polarity marking process

of converting ULF to English, running the Natlog system, generating possible SLFs, inferring polarity markings, and finally identifying the best match becomes a huge bottleneck on the inference engine. To avoid this cost, Kim et al. [2] added polarity propagation functions corresponding to each inference rule. These functions consider a single inference step, taking the premise ULF formulas, their polarity markings, and the conclusion as arguments, and compute the polarity marking of the conclusion.

Consider the UMI inference rule from FraCaS problem 24 described in Section 3. The premises are *many delegates obtained interesting results from the survey* and *all interesting results are results* and the conclusion is *many delegates obtained results from the survey*. The polarity markings for our premises are

(all.d (interesting.a (plur result.n))$^{\blacktriangledown}$ ((pres be.v) (= (k (plur result.n)$^{\blacktriangle}$)))).

and

((many.d (plur delegate.n)$^{\blacksquare}$) ((past obtain.v) (k (interesting.a (plur result.n))$^{\blacktriangle}$)
(adv-a (from.p (the.d survey.n))))).

The propagation function uses a simple structural replay of the original inference rule to find that (plur result.n)$^{\blacktriangle}$ is the polarized subexpression that we substituted for (interesting.a (plur result.n)). Thus, most polarity markings are maintained from *many delegates obtained interesting results from the survey* except the marking for (plur result.n) in the new subexpression. This leads to the following relevant polarity marking of the conclusion.

((many.d (plur delegate.n)$^{\blacksquare}$) ((past obtain.v) (k (plur result.n)$^{\blacktriangle}$)
(adv-a (from.p (the.d survey.n))))).

Most of these propagation functions never consider SLFs because the inference context and scope island constraints [34–37] eliminate the possibility of the scoping affecting the expressions participating in the substitution. This leads to a considerable speed advantage over the polarity marking functions.

The monotonicity substitution of determiners is an important exception where the new determiner may induce different entailment contexts than the replaced determiner in its restrictor and body. Thus the SLF is needed to properly propagate these relations to global polarities.[6]

5 System Extensions

In accord with the earlier discussion, we extend the baseline system by Kim et al. [2] in two ways. We introduce lexical relationships that may be necessary for inference based on the hypothesis sentence and enable the inference system to consider multiple scoping choices for each formula in the inference process.

5.1 Hypothesis-based Lexical Relationships

In their qualitative analysis, Kim et al. [2] showed a minimal pair of examples that demonstrates a key limitation in the system. By chaining System Rule 9 and System Rule 1 the system succeeds on a key part of example 59 from the FraCaS dataset.

[6] For example, in positive contexts, *the* may be replaced with *a*, as in, *I saw the dog* ⇒ *I saw a dog*. *The* imposes a flat entailment context on its restrictor whereas *a* imposes a positive entailment context which warrants a fresh computation of the global polarity markings.

(*A few*)$^{\blacktriangle}$ (*female committee members*)$^{\blacktriangle}$ *are from Scandinavia*
⇒ *At least a few committee members are from Scandinavia*

System Rule 9 extracts *Every female committee member is a committee member* after identifying that *female* is an intersective modifier and adds it to the knowledge base. This is then used to replace *female committee members* in positive context with *committee members*. Surprisingly, the same system fails on FraCaS example 76.

Few (*committee members*)$^{\blacktriangledown}$ *are from southern Europe*
⇒ *Few female committee members are from southern Europe*

On its surface, this is the same inference process but in reverse and in the opposite polarity context. However, the system is unable to solve this problem because it is completely limited to forward inference. That is, none of the inference rules, notably here System Rule 9, can be run on the ULF for the hypothesis sentence. Thus, the intersective relationship between *female* and *committee member* cannot be established.

We have a few options to overcome this limitation. One possibility is to introduce backward inference from the hypothesis. This has the added benefit of typically reducing the number of inference rules needed to find a solution. However, due to the way the inference rules are currently implemented, this would require a significant overhaul of the inference system. Instead, we could use a lexicon of words that act as intersective modifiers to generate all possible combinations of these words with all possible ULF expressions that can be modified in the knowledge base. This would be computationally infeasible as due to an explosion of unnecessary inferences in the search process—though, it would preserve the forward inference framework and its merits.

We opt for a method that is a combination of the two. We filter the lexicon of intersective modifiers by the ULF expressions present in the hypothesis formula. In practice, this is a simple extension of the existing inference system. Before starting the inference process, we run the intersective modifier extraction rules (System Rule 8 and System Rule 9) on the hypothesis ULF and add the results to the knowledge base.

5.2 Multiple Scope Choices

Another key limitation in Kim et al.'s [2] system is that the system determines a single compatible scoping and polarity annotation pair for each ULF and fully commits to such a scoping. This means that we cannot recover from any failures in the polarity annotation and scoping computation, problems that are far from fully solved.

We modify the inference system to generate all possible SLFs corresponding to each ULF, compute the resulting polarity annotations, and score the quality of the SLF is based on the disagreement between the resulting polarity annotations and the externally computed polarity annotation (a la Section 4.3). The disagreement score between two polarity annotated ULFs is the sum of mismatched polarities at the ULF leaves. This score is undefined if the underlying ULFs are different. We then incorporate this value into the search heuristic cost with the following equation (where lower values are preferred).

$$H'(u, s, h) = H(u, h) + \lambda_p * \text{Disagreement}(P_{\text{base}}(u), P_{\text{SLF}}(u, s))$$

where u is the current ULF, s is a specific scoping, h is the hypothesis ULF, $H()$ is the original heuristic function, P_{base} is the base polarity annotation, and P_{SLF} is the scoping based polarity annotation. λ_p is the hyperparameter for determining the relative weight of the original heuristic value and the polarity disagreement. As $\lambda_p \to \infty$ we effectively ignore all but the best scoping and end up with the same system we had before. If $\lambda_p = 0$, we treat every scoping equally and completely ignore how they relate to the baseline polarity annotation system.

While this is conceptually quite simple, Kim et al.'s [2] system optimizations make this change tricky. They used the fact that each ULF had only a single possible polarity annotation to keep track of a global hash table from ULFs to polarity annotations, thus deferring polarity annotation or propagation until the polarity is needed for an inference rule. This optimization avoids polarity marking for the large frontier of inferences that are used to guide the search process, most of which will never be used to make an additional inference.

When each ULF can have multiple polarity annotations, a global hash table is no longer sufficient. We instead separate out the polarity resolution step separately in the inference search process, effectively reintroducing Theory Rule 3 to the system inferences. Upon applying an inference rule, we now construct a closure of the corresponding polarity resolution function for the premise ULFs, premise polarity markings, and the resulting ULF. This closure is then lazily evaluated if and when the inference search process opts to expand the frontier to this formula and needs the polarity.

5.3 English-based Goal Matching

By strictly matching the goals using ULFs, we lose some of the flexibility of performing inference directly over natural language. To bring us closer to natural language-based inference, when checking for an entailment or conclusion, we translate the ULFs into English and make the comparison using those strings. This makes the system more robust to minor parsing errors but risks making conclusions that are unwarranted due to meaningfully different semantic structures.

6 Experimental Setup

The FraCaS dataset is a small hand-curated set of 346 entailment questions related to specific semantic phenomena [5]. Kim et al. [2] focus on section 1 of the dataset set, regarding generalized quantifiers (GQ) as this is the most relevant section. It is also the largest section, making up almost a quarter of the dataset. FraCaS is a small, focused

Accuracy %									
Baselines		**Ours**				**Other RTE**			
MC	KM	+H	+S	+E	+HSE	MN	LP	HU	HR
50	70	76	72	73	73	78	93	88	99

Table 1: FraCaS performance of our system (Ours) compared against a majority class (ENT) baseline (MC), Kim et al.'s [2] (KM), and several notable RTE systems: MN [13], LP [14], HU [15], and HR [16]. +H: hypothesis based lexical relationships, +S: multiple scope choices, +E: English-based goal checking, and +HSE for all three.

set of challenging problems, making it a useful testbed for specific capacities—such as monotonic inference—without also solving the generalization problem of larger datasets which takes away from the task focus. As such Kim et al. [2] use it to demonstrate the capacity to use ULFs as the basis for monotonic inferences, rather than present a system to compete with the state-of-the-art on entailment tasks.

Kim et al. [2] use a leaf label F1 heuristic (LL-F1) which alternates with BFS every 5 inference steps. LL-F1 computes the F1 score between the leaf labels of the formula in question and the goal, ignoring order but preserving repetitions. This is turned into a cost ranging from 0 to 1 by subtracting it from 1. They set a maximum of 50 inference steps, after which the system produces an "unknown" output.

We focus on the same set of problems from the FraCaS dataset when evaluating of our system extensions so that we can compare against the baseline model. When only introducing hypothesis-based lexical relationships (Section 5.1) or English-based goal matching (Section 5.3) we use the exact same hyperparameters as the baseline model. When allowing multiple scope choices, we increase the maximum inference steps to 2,000. We found that the additional scoping alternatives caused the search process to branch out for many more inference steps before finding the solution. $\lambda_p = 0.3$.

G\P	ENT	CON	UNK
ENT	**22**	0	15
CON	0	**0**	5
UNK	0	0	**32**

(a) Baseline [2]

G\P	ENT	CON	UNK
ENT	**24**	0	13
CON	0	**0**	5
UNK	0	0	**32**

(b) Rel. from hypothesis

G\P	ENT	CON	UNK
ENT	**22**	0	15
CON	0	**0**	5
UNK	1	0	**21**

(c) Scoping choices

G\P	ENT	CON	UNK
ENT	**22**	0	15
CON	1	**0**	4
UNK	0	0	**32**

(d) English goal match

G\P	ENT	CON	UNK
ENT	**23**	0	14
CON	1	**0**	4
UNK	1	0	**31**

(e) All extensions

Table 2: Confusion matrices for the baseline and extended models. G\P is short for "Gold\Prediction".

7 Results

Table 1 shows the accuracy of the extensions compared to the baseline system and other systems that were evaluated on this dataset. We find the hypothesis-based lexical relationship provides the largest boost in performance of 6 points. Matching based on English rather than ULFs gives us a 3-point improvement. Considering multiple scope choices provides a small boost in performance (2 points). When we combine the methods, we do not get the best of all methods, rather we get close to the average of each improvement.

Table 2 shows the confusion matrix for each extension which gives us a better picture of where the performance gains are coming from. None of the changes, even adding all three, seem to make major changes to the overall inference patterns. All of them generally have high precision, but as we might expect, matching based on English and supporting multiple scoping choices both cause imprecise entailment

predictions. As noted by Kim et al. [2], the failure to correctly identify any contradictions is not an inherent limitation of the system. Parser errors led to the inability to generate the necessary inferences in the 5 problems with contradiction labels.

Table 3 shows the number of inferences by each of these systems. We see that extracting rules from the hypothesis and matching goals with English makes almost no difference in the number of inference steps. Introducing scoping possibilities, however, greatly increase the number of inference steps necessary to find the same entailment relations. Considering multiple scope possibilities leads to about 10x the number of steps to find entailment relations.

	KM	+H	+S	+E	+HSE
Av. Steps	37.9	37.3	-	37.3	1,412.6
Av. Steps (conv.)	7.0	8.7	-	6.9	108.0

Table 3: Average numbers of inference steps for each of our extensions against the baseline. Avg. Steps (conv.) is the number of steps when we only consider problems where the inference method converged to an answer other than "unknown".

+S numbers were lost due to a software bug.

The three table results suggest that introducing scoping possibilities will tend to do more harm than good. Of course, the size and nature of this dataset limit how certainly we can make such a claim. It may simply be that the polarity annotations and scoping procedures are uncharacteristically performant in this small dataset.

8 Conclusion

We presented a comprehensive description of the monotonic inference system using ULFs, making clear the relationship between Kim et al.'s [1] theoretical framework and the realized system by Kim et al. [2]. In addition, we presented three mechanisms to manage notable limitations in the inference system. Our evaluation on the FraCaS dataset shows that these improvements make small improvements to the system, though sometimes at the cost of precision and inference speed.

References

[1] Kim, G., Juvekar, M., Schubert, L.: Monotonic inference for underspecified episodic logic. In: Proceedings of the 1st and 2nd Workshops on Natural Logic Meets Machine Learning (NALOMA), pp. 26–40. Association for Computational Linguistics, Groningen, the Netherlands (online) (2021). https://aclanthology.org/2021.naloma-1.5

[2] Kim, G., Juvekar, M., Ekmekciu, J., Duong, V., Schubert, L.: A (mostly) symbolic system for monotonic inference with unscoped episodic logical forms. In: Proceedings of the 1st and 2nd Workshops on Natural Logic Meets Machine Learning (NALOMA), pp. 71–80. Association for Computational Linguistics, Groningen, the Netherlands (online) (2021). https://aclanthology.org/2021.naloma-1.9

[3] Kim, G.L., Schubert, L.: A type-coherent, expressive representation as an initial step to language understanding. In: Proceedings of the 13th International

Conference on Computational Semantics - Long Papers, pp. 13–30. Association for Computational Linguistics, Gothenburg, Sweden (2019). https://doi.org/10.18653/v1/W19-0402 . https://aclanthology.org/W19-0402

[4] Schubert, L.K.: The situations we talk about. In: Minker, J. (ed.) Logic-based Artificial Intelligence, pp. 407–439. Kluwer Academic Publishers, Norwell, MA, USA (2000)

[5] Cooper, R., Crouch, D., Eijck, J.V., Fox, C., Genabith, J.V., Jaspars, J., Kamp, H., Milward, D., Pinkal, M., Poesio, M., Pulman, S.: Using the framework. Technical Report LRE 62-051 D-16, The FraCaS Consortium (1996)

[6] Kim, G., Kane, B., Duong, V., Mendiratta, M., McGuire, G., Sackstein, S., Platonov, G., Schubert, L.: Generating discourse inferences from unscoped episodic logical formulas. In: Proceedings of the First International Workshop on Designing Meaning Representations, pp. 56–65. Association for Computational Linguistics, Florence, Italy (2019). https://doi.org/10.18653/v1/W19-3306 . https://aclanthology.org/W19-3306

[7] Sánchez Valencia, V.: Categorial Grammar and Natural Logic. ILTI Prepublication: Logic, Philosophy and Linguistics (LP) Series (1991)

[8] Lambek, J.: Categorial and categorical grammars. In: Categorial Grammars and Natural Language Structures, pp. 297–317. Springer, ??? (1988)

[9] Van Benthem, J., *et al.*: Essays in Logical Semantics. Springer, ??? (1986)

[10] Sánchez-Valencia, V.: Studies on natural logic and categorial grammar. PhD thesis, University of Amsterdam (1991)

[11] Angeli, G., Manning, C.D.: NaturalLI: Natural logic inference for common sense reasoning. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 534–545. Association for Computational Linguistics, Doha, Qatar (2014). https://doi.org/10.3115/v1/D14-1059 . https://aclanthology.org/D14-1059

[12] Tian, R., Miyao, Y., Matsuzaki, T.: Logical inference on dependency-based compositional semantics. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 79–89. Association for Computational Linguistics, Baltimore, Maryland (2014). https://doi.org/10.3115/v1/P14-1008 . https://aclanthology.org/P14-1008

[13] Mineshima, K., Martínez-Gómez, P., Miyao, Y., Bekki, D.: Higher-order logical inference with compositional semantics. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 2055–2061 (2015)

[14] Abzianidze, L.: Natural solution to fracas entailment problems. In: Proceedings of

the Fifth Joint Conference on Lexical and Computational Semantics, pp. 64–74 (2016)

[15] Hu, H., Chen, Q., Moss, L.: Natural language inference with monotonicity. In: Proceedings of the 13th International Conference on Computational Semantics - Short Papers, pp. 8–15. Association for Computational Linguistics, Gothenburg, Sweden (2019). https://doi.org/10.18653/v1/W19-0502 . https://aclanthology.org/W19-0502

[16] Haruta, I., Mineshima, K., Bekki, D.: Combining event semantics and degree semantics for natural language inference. In: Proceedings of the 28th International Conference on Computational Linguistics, pp. 1758–1764. International Committee on Computational Linguistics, Barcelona, Spain (Online) (2020). https://doi.org/10.18653/v1/2020.coling-main.156 . https://aclanthology.org/2020.coling-main.156

[17] Kalouli, A.-L., Crouch, R., Paiva, V.: Hy-NLI: a hybrid system for natural language inference. In: Proceedings of the 28th International Conference on Computational Linguistics, pp. 5235–5249. International Committee on Computational Linguistics, Barcelona, Spain (Online) (2020). https://doi.org/10.18653/v1/2020.coling-main.459 . https://aclanthology.org/2020.coling-main.459

[18] Chen, Z., Gao, Q., Moss, L.S.: NeuralLog: Natural language inference with joint neural and logical reasoning. In: Proceedings of *SEM 2021: The Tenth Joint Conference on Lexical and Computational Semantics, pp. 78–88. Association for Computational Linguistics, Online (2021). https://doi.org/10.18653/v1/2021.starsem-1.7 . https://aclanthology.org/2021.starsem-1.7

[19] Rozanova, J., Ferreira, D., Thayaparan, M., Valentino, M., Freitas, A.: Decomposing natural logic inferences for neural NLI. In: Proceedings of the Fifth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP, pp. 394–403. Association for Computational Linguistics, Abu Dhabi, United Arab Emirates (Hybrid) (2022). https://aclanthology.org/2022.blackboxnlp-1.33

[20] Chen, Z., Gao, Q.: Probing linguistic information for logical inference in pre-trained language models. Proceedings of the AAAI Conference on Artificial Intelligence **36**(10), 10509–10517 (2022) https://doi.org/10.1609/aaai.v36i10.21294

[21] Kitaev, N., Klein, D.: Constituency parsing with a self-attentive encoder. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 2676–2686. Association for Computational Linguistics, Melbourne, Australia (2018). https://doi.org/10.18653/v1/P18-1249 . https://aclanthology.org/P18-1249

[22] Schubert, L.: Can we derive general world knowledge from texts? In: Proceedings of the Second International Conference on Human Language Technology Research. HLT '02, pp. 94–97. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA

(2002). http://dl.acm.org/citation.cfm?id=1289189.1289263

[23] Schubert, L., Tong, M.: Extracting and evaluating general world knowledge from the brown corpus. In: Proceedings of the HLT-NAACL 2003 Workshop on Text Meaning, pp. 7–13 (2003). https://aclanthology.org/W03-0902

[24] Gordon, J., Schubert, L.: Quantificational sharpening of commonsense knowledge. In: Proceedings of the AAAI 2010 Fall Symposium on Commonsense Knowledge (2010)

[25] Schubert, L.: From treebank parses to episodic logic and commonsense inference. In: Proceedings of the ACL 2014 Workshop on Semantic Parsing, pp. 55–60. Association for Computational Linguistics, Baltimore, MD (2014). http://www.aclweb.org/anthology/W14-2411

[26] Purtee, A., Schubert, L.: TTT: A tree transduction language for syntactic and semantic processing. In: Proceedings of the Workshop on Applications of Tree Automata Techniques in Natural Language Processing. ATANLP '12, pp. 21–30. Association for Computational Linguistics, Stroudsburg, PA, USA (2012)

[27] Kim, G., Duong, V., Lu, X., Schubert, L.: A transition-based parser for unscoped episodic logical forms. In: Proceedings of the 14th International Conference on Computational Semantics (IWCS), pp. 184–201. Association for Computational Linguistics, Groningen, The Netherlands (online) (2021). https://aclanthology.org/2021.iwcs-1.18

[28] MacCartney, B., Manning, C.D.: Modeling semantic containment and exclusion in natural language inference. In: Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008), pp. 521–528. Coling 2008 Organizing Committee, Manchester, UK (2008). https://aclanthology.org/C08-1066

[29] Barker, C.: Rethinking scope islands. Linguistic Inquiry, 1–55 (2021) https://doi.org/10.1162/ling_a_00419 https://direct.mit.edu/ling/article-pdf/doi/10.1162/ling_a_00419/1889100/ling_a_00419.pdf

[30] Donnellan, K.S.: Reference and definite descriptions. The philosophical review **75**(3), 281–304 (1966)

[31] Burge, T.: Belief de re. The Journal of Philosophy **74**(6), 338–362 (1977)

[32] Nayak, N., Kowarsky, M., Angeli, G., Manning, C.D.: A dictionary of nonsubsective adjectives. Technical Report CSTR 2014-04, Department of Computer Science, Stanford University (October 2014). https://hci.stanford.edu/cstr/reports/2014-04.pdf

[33] Miller, G.A.: WordNet: A lexical database for english. Communications of the ACM **38**(11), 39–41 (1995) https://doi.org/10.1145/219717.219748

[34] Fodor, J., Sag, I.: Referential and quantificational indefinites. Linguistics and Philosophy **5**, 355–398 (1982)

[35] Park, J.C.: Quantifier scope and constituency. In: 33rd Annual Meeting of the Association for Computational Linguistics, pp. 205–212. Association for Computational Linguistics, Cambridge, Massachusetts, USA (1995). https://doi.org/10.3115/981658.981686 . https://aclanthology.org/P95-1028

[36] Ruys, E.G., Winter, Y.: Quantifier scope in formal linguistics. In: Handbook of Philosophical Logic, pp. 159–225. Springer, ??? (2011). https://doi.org/10.1007/978-94-007-0479-4_3

[37] Barker, C.: Scope. In: Lappin, S., Fox, C. (eds.) Handbook of Contemporary Semantics, 2nd edn., pp. 40–76. Wiley Blackwell, ??? (2015). Chap. 2