Monotonic Inference for Underspecified Episodic Logic

Gene Louis Kim[♡], **Mandar Juvekar**[♢], **and Lenhart Schubert**[♣]
University of Rochester
Department of Computer Science
{gkim21[♡],schubert[♣]}@cs.rochester.edu
mjuvekar[♢]@u.rochester.edu

Abstract

We present a method of making natural logic inferences from Unscoped Logical Form of Episodic Logic. We establish a correspondence between inference rules of scope-resolved Episodic Logic and the natural logic treatment by Sánchez Valencia (1991a), and hence demonstrate the ability to handle foundational natural logic inferences from prior literature as well as more general nested monotonicity inferences.

1 Introduction

Natural Logic is an approach to generating inferences from language directly over the grammatical structure through knowledge of entailment monotonicity in the lexicon. Monotonicity is a characteristic of functions within an ordering, i.e. f is upward monotone if $x \leq y$ implies $f(x) \leq f(y)$ (and downward monotone for the opposite). For example, not is downward monotone in entailment since it flips the entailment ordering of "Fido is a dog" entails "Fido is an animal" to "Fido is *not* an animal" entails "Fido is *not* a dog". Natural Logic can be seen as an extension of Aristotelian syllogistic reasoning (Van Benthem et al., 1986) and was first formally related to higher-order logic entailments by Sánchez Valencia (1991a). Icard and Moss (2014) and Icard et al. (2017) later construed Natural Logic as a formal system of its own, independent of a separate logical formalism.

Unscoped Logical Form (ULF) of Episodic Logic was developed with the aim to integrate machine learning into automatic natural language inference by simplifying the semantic parsing task that presupposes symbolic inference (Kim and Schubert, 2019). ULFs retain certain ambiguities in the sentence while strictly defining the core semantic type structure that is necessary to specify the compositional structure. This results in a parsing task that is similar in form and complexity to constituency parsing, for which the community has built effective parsers (Mrini et al., 2019; Zhou and Zhao, 2019). Promising preliminary results show parsability of ULF from a small dataset and minimal representation-specific knowledge (Kim, 2019). Automatic inference generation from ULF has been demonstrated for dialogue-focused structural inferences which correspond to simple presuppositions and implicatures over questions, requests, counterfactual constructions, and clause-taking verbs (Kim et al., 2019).

Here we present a proof-based Natural Logic inference formalism for ULF. We show that this method covers inferences presented by Sánchez Valencia (1991a) and can support Rule Instantiation from Episodic Logic (EL) inference for nested polarity inference. The contributions of this paper are two-fold: (1) this marks the first formalized inference procedure for ULF and (2) we present an alternative to parsing full logical formulas in symbolic Natural Logic inference through the use of an underspecified representation. An implementation of the inference procedure that we describe is beyond the scope of this paper, but is an important next step for empirically evaluating the efficacy of this approach against existing Natural Logic inference systems. Due to space limitations, we leave fully formalized definitions and proofs to the appendix and use the main document for condensed explanations and demonstrative examples.

2 Natural Logic

We will limit our discussion of Natural Logic to that presented by Sánchez Valencia (1991a) and follow his notation and terminology.[1] The semantics of Sánchez-Valencia's Natural Logic is rooted in an undirected, typed lambda calculus constructed

[1]Much of the recent work in Natural Logic uses the terminology and notation introduced by Icard and Moss (2014).

Proceedings of the 1st and 2nd Workshops on Natural Logic Meets Machine Learning (NALOMA), pages 26–40
June 16, 2021. ©2021 Association for Computational Linguistics

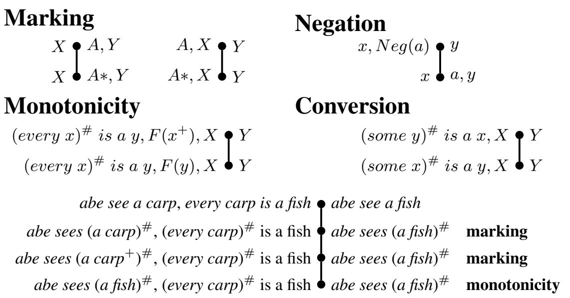

Figure 1: The basic inference rules for Sánchez-Valencia's natural logic proof system and an example.

from derivations of Lambek cum Permutation Calculus (Lambek, 1988). The primitives semantic types are e and t (for entities and truth values) which denote sets, with complex types of the form $\langle a, b \rangle$ where a and b are semantic types in the language. There is one inference rule over these, $\langle \alpha, \beta \rangle, \alpha \rightarrow \beta$, where the order of the functor and the argument do not matter.

Sánchez-Valencia formalizes the monotonicity of polarity lexical terms from the linguistic literature of Natural Logic in relation to this semantic framework in order to make soundness claims of predicate substitutions within positive and negative polarity contexts. Polarity contexts are determined by counting the number of downward entailing arguments that lie in between a constituent and the root of the Lambek derivation.

Inference Reasoning is done with a tableau proof system (Beth, 1955) starting with a node with the premises, a_i, on the left and conclusion, b, on the right like so, $a_1, ..., a_n \bullet b$, where all statements are in plain English. This is accompanied by the Lambek analyses for each of the statements, which supply grammatical information (scoping and polarity) to the proof. The tableau is closed when all paths in the proof tree are closed and a path is closed when the leaf of the path has the same statement (including scope marking) on both sides of the node. A^+ and A^- mark positive and negative polarity, respectively, and $(A)^\#$ marks the outermost operator scope. The inference rules in Sánchez's proof system needed for demonstrating basic monotonic inference and an example are displayed in Figure 1.

3 Episodic Logic: Unscoped Logical Form

Episodic Logic (EL) is an extended FOL that closely matches the form and expressivity of natural language, using type-shifters and a liberal ontology of individuals (e.g. basic individuals, situations, propositions, kinds, etc.) to keep the logic first-order while allowing for intensionality, general quantifiers, etc. (Schubert, 2000). EL supports deductive and uncertain inference, including forward and goal-chaining inference that uses polarity-based substitution in a Natural Logic-like manner (Hwang and Schubert, 1993; Morbini and Schubert, 2009; Schubert, 2014). The forward inference rules are basically the same as nested inference rules proposed by (Traugott, 1986).

ULF fully specifies the semantic type structure of EL by specifying the types of the atoms and all of the predicate-argument relationships while leaving operator scope, anaphora, and word sense unresolved (Kim and Schubert, 2019). The name, Unscoped Logical Form, is a label for its stage in the interpretation process of EL and does not mean that scoping is the only unresolved aspect of the logical form. Kim and Schubert (2019) describe the role of ULF in the interpretation process.

The types of ULF atoms that correspond to surface words and are not logical or macro symbols are marked with suffixed tags resembling the part-of-speech (e.g. .v, .n, .pro, .d for verbs, nouns, pronouns, and determiners). Case-sensitive symbols such as names and titles are marked with pipes (e.g. |John|). Pipe-marked symbols may be left without a type tag in which case they default to having an entity type. A closed set of logical and macro symbols have unique types so the type marking is omitted. Each suffix indicates a set of possible semantic denotations, e.g. .pro always denotes an *entity* and .v denotes an *n-ary predicate* where *n* can vary.

Type shifters in ULF maintain coherence of the semantic type structure. For example, the type shifter adv-a maps a predicate into a predicate modifier as in the prepositional phrase *"with a dog"* in Figure 2, as opposed to its predicative use *"I am*

(|Ali| (do.aux-s **not** (know.v (that
  (i.pro (work.v (adv-a (with.p **(a.d dog.n)**))))))))

↓ **scoping**

(not (|Ali| (do.aux-s (know.v (that
  (a.d x: (x dog.n)
    (i.pro (work.v (adv-a (with.p **x**)))) **)**)))) **)**

Figure 2: Example of a ULF scoping into an SLF for the sentence *Ali does not know that I work with a dog.*

with my dog".

The syntactic structure is closely reflected in ULF even under syntactic movement through the use of simple rewriting *macros* which explicitly mark these occurrences and upon expansion make available the exact semantic argument structure.

The ordering of operator-argument relations in ULF can have the operator in the first or second position, disambiguated by the types of the participating expressions. The EL type system only allows function application for combining types, $\langle A, B \rangle, A \rightarrow B$, much like Montagovian semantics (Montague, 1970) without type-raising.

Scoped Logical Form (SLF) SLF is ULF with explicit scoping. Since polarity propagates through scoped operator relations, scopes must be fully specified before adding polarities. While inferences will interface with ULFs, auxiliary SLFs are necessary to model the polarities and bookkeep scope-related assumptions in the inferences. Scoped operator orderings are represented using parentheses, and are lifted around the sentence that it scopes around. Scoped determiners are represented as (δ ν: ϕ ψ) where δ is a determiner, ν is a variable, ϕ is the restrictor wff, and ψ is the scope wff. Figure 2 shows examples of the scoping process.

4 Inference with ULF

Scope marking Rather than using a Lambek analysis for identifying operator scopes and hence polarities, we use SLFs.[2] The scoping of determiners leads to decoupled representations of the scoped constituent, so we must define a correspondence that allows us to mark the scoping of the ULF based on a fixed realization of the scoping.

For a ULF, ψ, that contains a quantified expression ♀ of form the (δ π), where δ is a determiner and π is a predicate, the corresponding formula with (δ π) at the top-level scope is (δ x: (x π) ψ[♀/x]).

Top-level scope marking process Given the SLF that defines the scope ordering, the constituent of the form (δ π) in ψ at the position of x in ψ[♀/x] is marked with # as the top scope of ψ.[3] Below is

[2] In accordance with Sánchez-Valencia's treatment, we do not address the possible scoping complexities of including sentential modifiers, tenses, and aspect as scoped operators.

[3] ♀ is not an alias for the pattern (δ π). Rather it refers to a unique constituent of ψ that has the form (δ π). This is an important distinction in order to properly handle sentences with multiple constituents of the same form, e.g., *"A dog greets a dog"*.

Scoping Operators (S1)	Determiners (S2)
not$_{(-)}$, never.adv$_{(-)}$	a.d$_{(+,+)}$, every.d$_{(-,+)}$, some.d$_{(+,+)}$, many.d$_{(+,+)}$, most.d$_{(+,\circ)}$
Verbs	
know.v$_{(+,\circ)}$	

Figure 3: Examples of lexical monotonicity markings.

an example to help illustrate the mapping.

"Abelard sees a carp"
SLF (a.d x: (x carp.n) (|Abelard| (see.v x)))
Marked ULF (|Abelard| (see.v (a.d carp.n)$^{\#}$))
$\boldsymbol{\delta}$: a.d, $\boldsymbol{\pi}$: carp.n

Polarity marking We perform polarity marking in a two stage process that mirrors the process used by Sánchez Valencia (1991a). First we classify lexical entries according to their monotonicity properties—in what entailment contexts they place their arguments—and mark them in the SLF with parenthesized subscripts. The possible entailment options are $+$ for upward, $-$ for downward, and $\circ$ for none. Figure 3 provides a few examples.[4]

Using the lexical annotations, we mark the local entailment direction of the constituents in the SLF using subscripts *without parentheses*. Finally, the global polarity is derived from these local entailment directions and marked with superscripts. The global polarity is computed by traversing the SLF from the root and counting the number of occurrences of negative and flat entailments, with the following rules.

1. Node a has no polarity if any node in the path from the root to a is marked with $\circ$.
2. Else, node a has negative polarity if there are an odd number of nodes between the root and a (inclusive) marked with $-$.
3. Otherwise, node a has positive polarity.

Figure 4 shows all of these markings in a tree format. We then mark the global polarity in the ULF according to the corresponding SLF to get.
((no.d^{+} scientist.n^{-})
(know.v^{-} (every.d^{-} (scientific.a^{+} fact.n^{+})$^{+}$))$^{-}$)

Inference Rules Figure 5 lists the ULF versions of the monotonicity and conversion rules from Figure 1, but in a standard rule of inference format. Sánchez-Valencia's Marking and Negation rules are specific to the tableau system and not relevant

[4] Unmarked lexical entries are assumed to have upward entailment on all of their arguments.

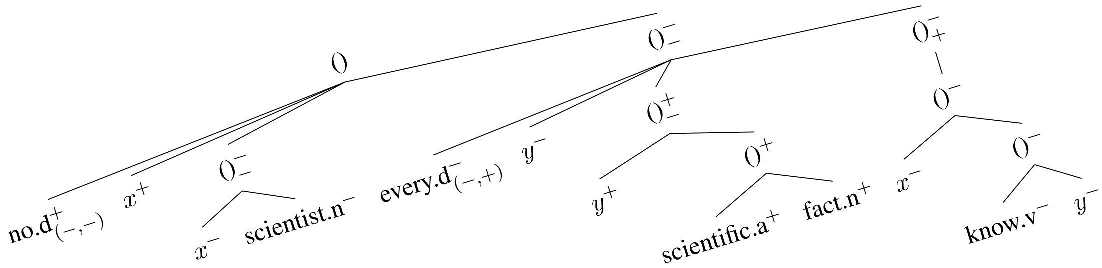

Figure 4: A tree representation of the SLF for "No scientist knows every scientific fact." with all lexical monotonicity, local entailment context, and global polarity markings.

Monotonicity (UMI)

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

where δ is a determiner.

Conversion (UCI)

$$\frac{((d1\ P)\ (\text{be.v}\ (=\ (d2\ Q))))}{((d1\ Q)\ (\text{be.v}\ (=\ (d2\ P))))}$$ where $d1 \in \{\text{some.d, a.d, no.d}\}$ and $d2 \in \{\text{some.d, a.d}\}$.

Figure 5: Inference rules in ULF corresponding to basic inference rules for Sánchez-Valencia's natural logic proof system.

as a general logical inference rule. Derivations and proofs are available in Appendix B.

We can also define the corresponding monotonicity rule for the negative polarity context. The monotonicity rule in ULF handles the explicit copula, through the transparent semantic interpretation of 'be.v'.

Example Now we use these ULF rules to perform the inferences from Figure 1.[5]

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)$^{+}$))	Pol marking 1.,3.
5.	(|Abelard| (see.v (a.d fish.n)))	UMI 2.,4.

It turns out that the monotonicity rules so far are special cases of EL Rule Instantiation, which operates on substitution under arbitrarily nested polarity contexts (Schubert and Hwang, 2000).

RI-1
$$\frac{MAJ(\phi^{-}), MIN(\phi'^{+})}{MAJ_{\sigma}(\neg MIN_{\sigma}(\bot^{+})^{-})}$$

RI-2
$$\frac{MAJ(\phi^{-}), MIN(\phi'^{+})}{MIN_{\sigma}(MAJ_{\sigma}(\top^{-})^{+})}$$

[5]Appendix D demonstrates how to handle all traditional Aristotelian syllogisms.

where RI-1 is sound if the only variables in the matching expression (ϕ') of the minor premise (MIN) are "matchably bound,"—bound within ϕ' or by a universal quantifier in positive polarity context or existential quantifier in negative polarity context—and RI-2 is sound if the only variables in the matching expression (ϕ) of the major premise (MAJ) are "matchably bound."

UMI is a special case of RI-2 and the negative polarity version is a special case of RI-1. These can handle inferences where the major premise is a more complex construction than *every p is a q*. RI-2 can be used to conclude *Something is a cap or pretty if Little Red Riding Hood wears it* from *Every dress or hood that Little Red Riding Hood wears is pretty* and *Something is a cap or a hood*. See Appendix C for a thorough discussion on Rule Instantiation.

5 Integration with Machine Learning

While a working inference system is beyond the scope of this paper, in this section we discuss some ways in which machine learning can be leveraged in conjunction with the inference formalism that we describe in this paper. An obvious and important role for machine learning in building a ULF-based inference system is to train a semantic parser to provide ULFs for English sentences. Our preliminary work in this direction (Kim, 2019) using an annotated dataset has shown promising results. Our current method is to train an LSTM to parse action sequences for a cache transition parser (Gildea et al., 2018). Including contextual embeddings such as BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) as inputs to such as model will allow the parser to use the representational power of these embeddings to select the most appropriate parse.

Similarly, we can expect polarity labeling algorithms to improve with the introduction of contextual embeddings, though we are unaware of any work that has tried to do this. This labeling could also have a collaborative effect with a symbolic polarity labeling. With a partially complete lexicon of negative polarity inducing operators, a ULF could verify parts of the sequentially labeled polarities or correct them if inconsistencies are found in the graph where lexical knowledge is available.

For tasks like SICK (Marelli et al., 2014) which rely largely on lexical specializations, we envision using a lexical resource like WordNet (Miller, 1995). There still remains the issue of word sense, which is not resolved in ULF. Again distributional word representations could be used here to select the most appropriate word sense or set of word senses. Tasks that provide all the necessary relationships such as FraCaS (Cooper et al., 1996) do not require any additional axioms beyond inference rules for basic logical operators and for introducing and eliminating macro operators. For example, modeling relative clauses, which appear frequently in FraCaS, simply requires properly handling the relativizer and post-nominal modification macros to get a monotonicity ordering between predicates "A" and "A that B" that is fully modeled by logical conjunction "A" and "(λ x: ((x A) $\wedge$ (x B)))".

The near-syntactic nature of ULF allows accurate generation of English sentences corresponding to formulas (Kim et al., 2019). An interesting question is whether generative language models could be used to enhance inference generation. Such models are trained to learn the patterns of language use, and as such do not necessarily reflect valid entailments. But anchoring the use of language models to a symbolic representation like ULF would potentially enable constraining inferences to interpretable ones.

6 Conclusion

We have presented a proof-based formalism for making natural logic inferences from ULFs with in-proof scoping assumption declarations using less ambiguous, scoped LFs. This inferential capacity of ULF, in conjunction with its ease of parsing, positions ULF as a promising representational basis for automatically generating natural logic inferences. Machine learning tools can then be deployed for semantic parsing (ULFs) and sequence labeling (polarities), both well-researched paradigms, rather than building a model of Natural Logic directly on top of statistical tools.

Acknowledgments

This work was supported by NSF EAGER grant NSF IIS-1908595, DARPA CwC subcontract W911NF-15-1-0542, and a Sproull Graduate Fellowship from the University of Rochester. We are grateful to Hannah An, Sapphire Becker and the anonymous reviewers for their helpful feedback.

References

Evert Willem Beth. 1955. Semantic entailment and formal derivability. In *Proceedings of the Section of Sciences*, 18, page 309–342. Koninklijke Nederlandse Akademie van Wentenschappen.

Robin Cooper, Dick Crouch, Jan Van Eijck, Chris Fox, Johan Van Genabith, Jan Jaspars, Hans Kamp, David Milward, Manfred Pinkal, Massimo Poesio, and Steve Pulman. 1996. Using the framework. Technical Report LRE 62-051 D-16, The FraCaS Consortium.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Daniel Gildea, Giorgio Satta, and Xiaochang Peng. 2018. Cache transition systems for graph parsing. *Computational Linguistics*, 44(1):85–118.

C.H. Hwang and L.K. Schubert. 1993. Episodic Logic: A situational logic for natural language processing. In P. Aczel, D. Israel, Y. Katagiri, and S. Peters, editors, *Situation Theory and its Applications 3 (STA-3)*, pages 307–452. CSLI.

Thomas Icard, Lawrence Moss, and William Tune. 2017. A monotonicity calculus and its completeness. In *Proceedings of the 15th Meeting on the Mathematics of Language*, pages 75–87, London, UK. Association for Computational Linguistics.

Thomas F Icard and Lawrence S Moss. 2014. Recent progress on monotonicity. In *Linguistic Issues in Language Technology*. Citeseer.

Gene Kim, Benjamin Kane, Viet Duong, Muskaan Mendiratta, Graeme McGuire, Sophie Sackstein, Georgiy Platonov, and Lenhart Schubert. 2019. Generating discourse inferences from unscoped episodic logical formulas. In *Proceedings of the First International Workshop on Designing Meaning Representations*, pages 56–65, Florence, Italy. Association for Computational Linguistics.

Gene Kim and Lenhart Schubert. 2019. A type-coherent, expressive representation as an initial step to language understanding. In *Proceedings of the 13th International Conference on Computational Semantics*, Gothenburg, Sweden. Association for Computational Linguistics.

Gene Louis Kim. 2019. Towards parsing unscoped episodic logical forms with a cache transition parser. In *the Poster Abstracts of the Proceedings of the 32nd International Conference of the Florida Artificial Intelligence Research Society*.

Joachim Lambek. 1988. Categorial and categorical grammars. In *Categorial grammars and natural language structures*, pages 297–317. Springer.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. A SICK cure for the evaluation of compositional distributional semantic models. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 216–223, Reykjavik, Iceland. European Language Resources Association (ELRA).

George A. Miller. 1995. WordNet: A lexical database for english. *Communications of the ACM*, 38(11):39–41.

Richard Montague. 1970. Universal grammar. *Theoria*, 36(3):373–398.

Fabrizio Morbini and Lenhart Schubert. 2009. Evaluation of Epilog: A reasoner for Episodic Logic. In *Proceedings of the Ninth International Symposium on Logical Formalizations of Commonsense Reasoning*, Toronto, Canada.

Khalil Mrini, Franck Dernoncourt, Trung Bui, Walter Chang, and Ndapa Nakashole. 2019. Rethinking self-attention: An interpretable self-attentive encoder-decoder parser.

Victor Sánchez Valencia. 1991a. Categorial grammar and natural logic. ILTI Prepublication: Logic, Philosophy and Linguistics (LP) Series.

Victor Sánchez Valencia. 1991b. *Studies on natural logic and categorial grammar*. Ph.D. thesis, University of Amsterdam.

Lenhart Schubert. 2014. From treebank parses to Episodic Logic and commonsense inference. In *Proceedings of the ACL 2014 Workshop on Semantic Parsing*, pages 55–60, Baltimore, MD. Association for Computational Linguistics.

Lenhart K. Schubert. 2000. The situations we talk about. In Jack Minker, editor, *Logic-based Artificial Intelligence*, pages 407–439. Kluwer Academic Publishers, Norwell, MA, USA.

Lenhart K. Schubert and Chung Hee Hwang. 2000. Episodic Logic meets Little Red Riding Hood: A comprehensive natural representation for language understanding. In Lucja M. Iwańska and Stuart C. Shapiro, editors, *Natural Language Processing and Knowledge Representation*, pages 111–174. MIT Press, Cambridge, MA, USA.

Jonathan Traugott. 1986. Nested resolution. In *8th International Conference on Automated Deduction*, pages 394–402, Berlin, Heidelberg. Springer Berlin Heidelberg.

Johan Van Benthem et al. 1986. *Essays in Logical Semantics*. Springer.

Junru Zhou and Hai Zhao. 2019. Head-driven phrase structure grammar parsing on Penn treebank. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2396–2408, Florence, Italy. Association for Computational Linguistics.

Appendix A Sánchez-Valencia's Treament of Natural Logic

This appendix lays out the formal treatment of Sánchez-Valencia's Natural Logic which is described in Section 2. The semantics of Sánchez-Valencia's Natural Logic is rooted in an undirected, typed lambda calculus constructed from derivations of Lambek cum Permutation Calculus (Lambek, 1988). The primitives semantic types are e and t (for entities and truth values) which denote sets, with complex types of the form $\langle a, b\rangle$ where a and b are semantic types in the language.There is one inference rule over these, $\langle \alpha, \beta\rangle$, $\alpha \rightarrow \beta$, where the order of the functor and the argument do not matter.

Definition A.1. Monotonicity is defined over the partial ordering relation $\leq_a$ which is defined as follows, where a is a semantic type and $\mathcal{D}_a$ is the corresponding set:

- If $\alpha, \beta \in \mathcal{D}_e$ then $\alpha \leq_e \beta$ iff $\alpha = \beta$.
- If $\alpha, \beta \in \mathcal{D}_t$ then $\alpha \leq_t \beta$ iff $\alpha = \bot$ or $\beta = \top$.
- If $\alpha, \beta \in \mathcal{D}_{(c,d)}$ then $\alpha \leq_{(c,d)} \beta$ iff for each $\kappa \in \mathcal{D}_c$, $\alpha(\kappa) \leq_d \beta(\kappa)$.

Definition A.2. Monotonicity for functions $f \in \mathcal{D}_{(a,b)}$ is defined over this ordering as follows:

- f is *upward monotone* iff for all $x, y \in \mathcal{D}_a$, $x \leq_a y$ entails $f(x) \leq_b f(y)$.
- f is *downward monotone* iff for all $x, y \in \mathcal{D}_a$, $x \leq_a y$ entails $f(y) \leq_b f(x)$.
- f is *non-monotone* if it is neither upward or downward monotone.

Definition A.3. Monotonicity of an occurrence M in N is defined relative to its semantic interpretation, where I is the interpretation function such that:

- M is *upward monotone* in N iff $I(M) \leq I(M')$ entails $I(N) \leq I(N\{M/M'\})$ for all models and assignments.
- M is *downward monotone* in N iff $I(M') \leq I(M)$ entails $I(N\{M/M'\}) \leq I(N)$ for all models and assignments.[6]

Using this, Sánchez-Valencia proves that positive and negative polarity items from the prior Natural Logic literature corresponds to upward and downward monotone occurrences. From this correspondence the soundness of substituting supersets for subsets in positive polarities and vice versa is realized.

Appendix B Detailed Inference System Correspondence

Sánchez-Valencia reasons using a tableau proof system (Beth, 1955) with nodes of the form

$$a_1, a_2, ..., a_n \bullet b_1, b_2, ..., b_m$$

where $a_i, 1 \leq i \leq n$ and $b_j 1 \leq j \leq m$ are English expressions with corresponding Lambek derivations a'_i and b'_j. The proof starts with the premises on the left side (a_i) and the desired conclusions on the right side (b_j). The proof concludes when all paths of the proof tree are closed. A path is closed when the leaf node of the path has the same statement (including scope markings) on both sides of the node. For those unaware of the notation of tableau systems, this node can be interpreted as the following well-formed formula.

$$(A_1 \wedge A_2 \wedge ... \wedge A_n) \rightarrow (B_1 \vee B_2 \vee ... \vee B_m)$$

A tableau step, e.g.

$$\begin{array}{c} a \bullet b \\ a' \bullet b' \end{array}$$

can be interpreted as the formula

$$(A \rightarrow B) \longleftrightarrow (A' \rightarrow B')$$

[6] $N\{M/M'\}$ is shorthand for M' substitutes for M in N.

B.1 Marking in NLog

Sánchez-Valencia's (1991a) monotonicity and scope marking rules are

$$\begin{array}{c} X \bullet A, Y \\ X \bullet A^*, Y \end{array} \qquad \begin{array}{c} A, X \bullet Y \\ A^*, X \bullet Y \end{array}$$

where A^* is a monotonicity or scope marking of A provided by the fixed Lambek analysis of A. Monotonicity can take values $+$ and $-$ and scoping is marked with $()^{\#}$ where the parentheses circumscribe the words associated with the top-level scope.

B.2 Scope Marking in ULF

Rather than using a Lambek analysis for identifying the operator scopes and as a result the polarities, scoped logical forms (SLFs) are used, which are ULFs with disambiguated scopes. The conversion from ULF to SLF can be denoted mid-proof so that a specific scoping does not need to be committed to at the start of the proof.

B.2.1 Mapping ULFs to SLFs

Scoping for ULFs comes in two flavors:

(S1) **Independent scoped operators.** An independent scoped operator is one that simply raises up to any wff level and introduces some information to only that scope of the overall formula. This includes tense, aspect, and sentence-level adverbials operators. These operators add temporal, locative, or general additional contextual information to the wff.

(|Abelard| ((past see.v) him.pro yesterday.adv-e))

↓ scoping

(past (yesterday.adv-e (|Abelard| (see.v him.pro))))

(S2) **Determiners with restrictors.** When determiners are scoped, they bring with them the restrictor predicate. A variable is introduced which is placed in the position of the lifted constituent and this variable is quantified with the lifted determiner and restricted by the restrictor predicate.

(|Abelard| (see.v (a.d carp.n)))

↓ scoping

(a.d x: (x carp.n) (|Abelard| (see.v x)))

B.2.2 Scope Marking with SLFs

In accordance with Sánchez-Valencia's treatment, we will only perform scope marking on the (S2) classes of scoping operators. Fortunately, this is the more interesting one from a structural perspective. The scoping of ULFs with (S2) classes leads to a more decoupled representation of the constituent, so we must define a correspondence between these components that allows us to still mark the scoping of the ULF based on the fixed scoped realization.

The key to making a correspondence between the (S2)-type constituent in ULF and SLF is the quantified variable of the SLF. For a ULF, ψ which contains a quantified expression ♀ 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 $(\delta\ x$: $(x\ \pi)\ \psi[♀/x])$.

Top-level scope marking process Knowing this correspondence, we have a path to marking the ULF quantified expression constituent that is the top-level scope. Given the SLF which defines the scope ordering, the constituent of the form $(\delta\ \pi)$ in psi at the position of x in $\psi[♀/x]$ is marked as the top scope of ψ. Note that ♀ does is not an alias for the pattern $(\delta\ \pi)$. Rather it refers to a unique constituent of ψ which has the form $(\delta\ \pi)$. This is an important distinction in order to properly handle sentences with multiple constituents of the same form, e.g. *"A dog greets a dog"*. Below is an example to help illustrate the mapping.

"Abelard sees a carp"
SLF (a.d x: (x carp.n) (|Abelard| (see.v x)))
Marked ULF (|Abelard| (see.v (a.d carp.n)$^{\#}$))
$\boldsymbol{\delta}$: a.d, $\boldsymbol{\pi}$: carp.n

In practice, we will not use the actual natural logic marking for inference since we don't use the tableau method for inference. Rather, we use this process to identify the ULF constituent with the top-level scope on the fly using the process which retains the same inferential capacity to the marking in tableau method.

B.3 Polarity marking in ULF

We perform polarity marking in a two stage process that mirrors the process used by (Sánchez Valencia, 1991a). First we classify lexical entries according to their monotonicity properties—in which direction they place entailment contexts on their arguments—and mark them in the SLF with subscripts. For example, the determiner no.d, which has negative downward entailment on both the restrictor and scope is marked as no.d$_{(-,-)}$. Here are a few lexical polarity annotated items. The possible entailment options are $+$ for upward, $-$ for downward, and $\circ$ for flat. Figure 6 provides more examples.

Scoping Operators (S1)
not$_{(-)}$, never.adv$_{(-)}$

Verbs
know.v$_{(+,\circ)}$

Determiners (S2)
a.d$_{(+,+)}$, every.d$_{(-,+)}$, some.d$_{(+,+)}$, many.d$_{(+,+)}$, most.d$_{(+,\circ)}$

Figure 6: Examples of lexical monotonicity markings.

Unmarked lexical entries are assumed to have upward entailment on all of their arguments. Using the lexical annotations, we mark the local entailment direction of the argument constituents in the SLF using subscripts again. For example, the SLF for "no scientist knows every scientific fact"

(no.d$_{(-,-)}$ x: (x scientist.n)
(every.d$_{(-,+)}$ y: (y (scientific.a fact.n))
(x (know.v y))))

gets its arguments marked as follows.

(no.d$_{(-,-)}$ x: (x scientist.n)$_{-}$
(every.d$_{(-,+)}$ y: (y (scientific.a fact.n))$_{-}$
(x (know.v y))$_{+}$)$_{-}$)

Finally, the global polarity is derived from these local entailment directions and marked with superscripts. The global polarity is derived by traversing the SLF from the root and counting the number of occurrences of negative and flat entailments. The global polarity is computed form this using the following rules, applied in order.

1. If node a has no polarity if any node in the path from the root to a is marked with $\circ$ (flat local entailment).

2. Else, if node a has negative polarity if there are an odd number of nodes between the root and a (including a) marked with $-$ (downward local entailment).

3. Otherwise, node a has positive polarity.

Following these rules the argument marked SLF gets marked with global polarity as (limiting global polarity marking to just nodes with local entailment marking for readability)

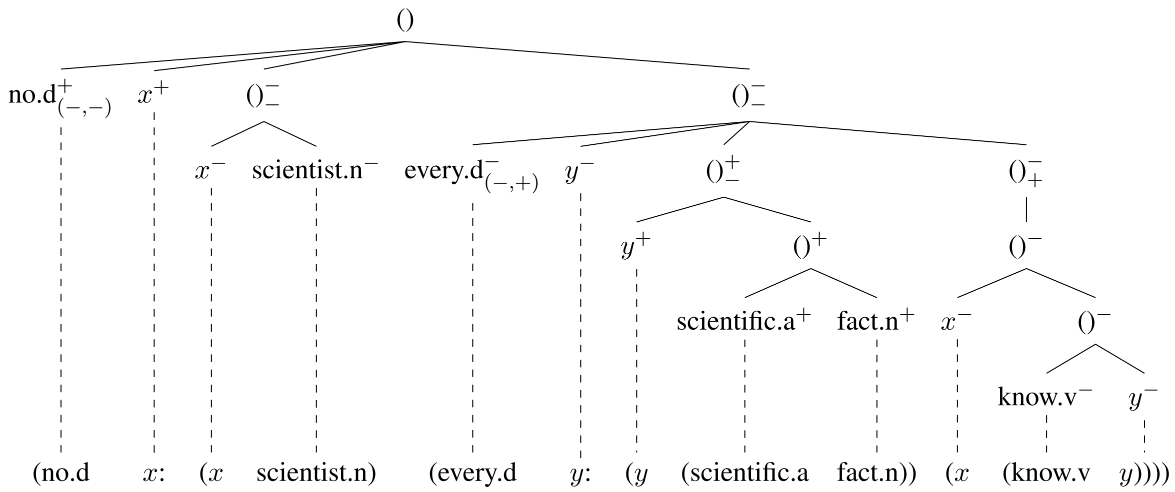

Figure 7: A tree representation of the SLF for "No scientist knows every scientific fact." with all lexical monotonicity, local entailment context, and global polarity markings.

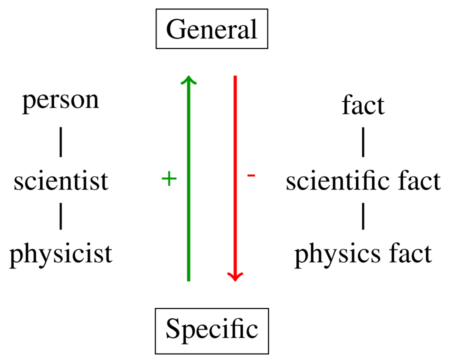

Figure 8: Monotonicity orderings of the predicates that are involved in the polarity inference example. Lower predicates are subsets of predicates above them. Predicates can be replaced with those above them in positive polarity and below them in negative polarity.

$$\text{(no.d}_{(-,-)}\ x\text{: }(x\ \text{scientist.n})^{-}_{-}\ \text{(every.d}_{(-,+)}\ y\text{: }(y\ \text{(scientific.a fact.n)})^{+}_{-}\ (x\ (\text{know.v}\ y))^{-}_{+})^{-}_{-})$$

The propagation of the polarity is easier to see when the SLF is written in a tree diagram, Figure 7.

Then we can mark the global polarity in the ULF according to the corresponding constituent in the SLF. The following is the resulting ULF for the above SLF, only marking constituents that correspond to predicates in predicative position in the SLF.

((no.d scientist.n^{-})
(know.v^{-} (every.d (scientific.a fact.n)$^{+}$)))

We now show that these markings result in inferences that follow our intuition. From the sentence *"No scientists knows every scientific fact"* we want to infer

(1) a. No physicist knows every scientific fact

b. No scientist knows every fact

but not

(2) a. *No person knows every scientific fact

b. *No scientist knows every physics fact

Figure 8 shows the monotonicity orderings of the predicates that are involved in this example and the direction of warranted inference in each polarity.

In positive polarity, we are warranted inference upward on this diagram, and for negative polarity downward. That is, 'scientist' is in negative polarity which warrants replacement by 'physicist', which is indeed intuitively warranted in English. As is desired from the motivating sentences, "scientist" is in negative polarity and can be replaced with "physicist" (1a) and "scientific fact" is in positive polarity and can be replaced with "fact" (1b).

B.4 Inferences with ULFs

B.4.1 Negation

Sánchez-Valencia's (1991a) negation rule is

$$\frac{x, Neg(a) \bullet y}{x \bullet a, y}$$

In simple logical terms, this rule is

$$((X \wedge \neg A) \rightarrow Y) \leftrightarrow (X \rightarrow (A \vee Y)) \qquad (1)$$

Negation of ULFs is performed by applying the logical 'not' operator so no special rule is necessary.

For instance, the negation of the ULF (|Abelard| walk.v) is (not (|Abelard| walk.v)).

Some identities that are useful for inferences are listed below, in the form of inference rules. Proofs for these identities can be given using the formal definitions of the respective generalized quantifiers.

$$\frac{\text{(not (not } \phi))}{\phi} \qquad \frac{\phi}{\text{(not (not } \phi))}$$

$$\frac{\text{(not (some.d } \nu\text{: } \psi\ \phi))}{\text{(no.d } \nu\text{: } \psi\ \phi)} \qquad \frac{\text{(no.d } \nu\text{: } \psi\ \phi)}{\text{(not (some.d } \nu\text{: } \psi\ \phi))}$$

$$\frac{\text{(not (a.d } \nu\text{: } \psi\ \phi))}{\text{(no.d } \nu\text{: } \psi\ \phi)} \qquad \frac{\text{(no.d } \nu\text{: } \psi\ \phi)}{\text{(not (a.d } \nu\text{: } \psi\ \phi))}$$

B.4.2 Handling 'be.v'

While 'be.v' are included in ULFs for simplifying the interface to natural language since the copula can act as an anchor for modifications from adverbial phrases and temporal information from its conjugation, we can consider it to be semantically void with respect to its arguments.

SLF Rule 1 (be.v Elimination).

$$\frac{\text{(a.d } y\text{: } (y\ P)\ (x \text{ (be.v (= } y))))}{(x\ P)}$$

where P is an arbitrary unary predicate and x is an arbitrary term.

Proof

1.	(a.d y: $(y\ P)$ (x (be.v (= y))))	Assumption
2.	I((a.d y: $(y\ P)$ (x (be.v (= y)))))	Interp. fn.
3.	There exists $d \in \mathcal{D}$ s.t. $d \in I(P)$ and $I((x \text{ (be.v (= } y))))^{\mathcal{U}_{y:d}}$	Satisfaction conds of $\exists$
4.	There exists $d \in \mathcal{D}$ s.t. $d \in I(P)$ and $I((x\ (= y)))^{\mathcal{U}_{y:d}}$	Def of be.v
5.	There exists d s.t. $d \in I(P)$ and $(I(x) = d)$	I(=)
6.	$I(x) \in I(P)$	Variable substitution
7.	$(x\ P)$	Interp. fn. of predication

With this inference rule we can easily derive a predicate subset defining inference rule which is necessary for polarity inferences.

$$\frac{\text{(every.d } x\text{: } (x\ P1) \text{ (a.d } y\text{: } (y\ P2)\ (x \text{ (be.v (= } y)))))}{\text{(every.d } x\text{: } (x\ P1)\ (x\ P2))}$$

where $P1$ and $P2$ are arbitrary unary predicates.

For example, from the initial SLF for "every carp is a fish" we get a nice relationship between the predicates 'carp.n' and 'fish.n'.

(every.d x: (x carp.n) (a.d y: (y fish.n) (x (be.v (= y)))))

$\Downarrow$

(every.d x: (x carp.n) (x fish.n))

B.4.3 Monotonicity Inference

The basic monotonicity inference rule in Natural Logic takes a subset relationship between two predicates, $P1 \subseteq P2$, and a formula, f, where something of type $P1$ occurs in positive polarity. Then we can assert f' which is the same as f except that $P2$ is substituted for $P1$. We can state a direct analog of this rule using SLFs.

We also formulate a similar rule which takes the subset relationship $P1 \subseteq P2$ and a formula g, where something of type $P2$ appears in negative polarity. In this case we can assert g' which is the same as g except that $P1$ is substituted for $P2$. Using SLFs the inferences looks as follows.

SLF Rule 2 (Monotonicity Inference, SMI).

$$\frac{(\delta\ x\text{: } (x\ P1)^{+}\ \phi(x)),\ \text{(every.d } y\text{: } (y\ P1)\ (y\ P2))}{(\delta\ x\text{: } (x\ P2)\ \phi(x))}$$

$$\frac{(\delta\ x\text{: } (x\ P2)^{-}\ \phi(x)),\ \text{(every.d } y\text{: } (y\ P1)\ (y\ P2))}{(\delta\ x\text{: } (x\ P1)\ \phi(x))}$$

where δ is a determiner.

The SLFs are necessary for keeping track of the outer scope and determining the polarities, but the core inference can be written using ULFs, closer to surface form, with the SLFs acting as auxiliary information to ensure consistency of the formulas. Using ULFs and chaining SMI and be.v elimination we get the following inferences.

ULF Rule 1 (Monotonicity Inference, UMI).

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

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

where δ is a determiner.

It is worth noting that if the restrictor of a determiner is a conjunction of predicates restricting the variable, then due to the upward entailing nature of the $\wedge$ operator we can propagate the polarity induced by the determiner on its restrictor to each term in the conjunction. Also, we know that the $\wedge$ operator preserves subset relations, that is if x satisfies predicates $P1$ and $P2$ and if every element of $P1$ is also in Q, then x must satisfy the predicates Q and $P2$. Therefore, in a case where a variable is restricted by a conjunction of predicates, it is possible to use the monotonicity inference rule on individual predicates in the restrictor. This is particularly useful when dealing with extensionally modified predicates (see B.4.4).

Example 1. Now we will use the presented ULF/SLF marking and inference rules to perform an inference over generalized quantifiers that Sánchez Valencia (1991a) demonstrated: from "Abelard sees a carp" and "Every carp is a fish" we will conclude "Abelard sees a fish". Before we start the inference, we walk through the scoping and polarity derivation of assumption (1), which will be used in the inference. This derivation takes the place of the Lambek derivations used by Sánchez-Valencia to get polarity and scoping information into the proof.

Scoping and Polarity Derivation

1.	(|Abelard| (see.v (a.d carp.n)))	ULF
2.	(a.d x: (x carp.n) (|Abelard| (see.v x)))	Only possible scoping
3.	(a.d$_{(+,+)}$ x: (x carp.n) (|Abelard| (see.v x)))	a.d lexical monotonicity
4.	(a.d x: (x carp.n)$_+$ (|Abelard| (see.v x))$_+$)	Local entail. context
5.	(a.d x: (x carp.n)$^+$ (|Abelard| (see.v x)$^+$)$^+$)	Global polarity

Now for the actual proof.

Proof

1.	(|Abelard| (see.v (a.d carp.n)))	Assumption
2.	((every.d carp.n) (be.v (= (a.d fish.n))))	Assumption
3.	(|Abelard| (see.v (a.d carp.n)$^+$))	Polarity marking, 1.
4.	(|Abelard| (see.v (a.d fish.n)))	UMI, 2.,3.

B.4.4 Inferences with Predicate Modifiers

Let P' be an extensional modification of a predicate P, and let P_m be the modifying predicate. Since the modification is extensional, we know that an entity x satisfies P' if and only if it satisfies both P and P_m. Hence we get the following rule.

SLF Rule 3.

$$\frac{(\delta\ x\text{: } (x\ P')\ \phi(x))}{(\delta\ x\text{: } ((x\ P) \wedge (x\ P_m))\ \phi(x))}$$

where P' is an extensional modification of the predicate P with modifying predicate P_m, and δ is a determiner.

This rule can then be combined with monotonicity inference to get

ULF Rule 2.

$$\frac{\phi[(\delta\ (M\ P1))^+],\ ((\text{every.d}\ P1)\ (\text{be.v}\ (=\ (\text{a.d}\ P2))))}{\phi[(\delta\ (M\ P2))]}$$

where M is an extensional modifier and δ is a determiner. A similar rule for negative polarities can be written as well.

This allows us to make another inference demonstrated by Sánchez Valencia (1991a):

Example 2. From "Abelard sees a male carp" and "Every carp is a fish", we will conclude "Abelard sees a male fish".

Scoping and Polarity Derivation

1.	(|Abelard| (see.v (a.d (male.a carp.n))))	ULF
2.	(a.d x: (x (male.a carp.n)) (|Abelard| (see.v x)))	Only possible scoping
3.	(a.d x: ((x male.a) $\wedge$ (x carp.n)) (|Abelard| (see.v x)))	Assume intersective modification
4.	(a.d$_{(+,+)}$ x: ((x male.a) $\wedge$ (x carp.n)) (|Abelard| (see.v x)))	a.d lexical monotonicity
5.	(a.d x: ((x male.a) $\wedge$ (x carp.n))$_+$ (|Abelard| (see.v x))$_+$)	Local entail. context
6.	(a.d x: ((x male.a)$_+$ $\wedge$ (x carp.n)$_+$) (|Abelard| (see.v x))$_+$)	Upward entail. of $\wedge$
7.	(a.d x: ((x male.a)$^+$ $\wedge$ (x carp.n)$^+$) (|Abelard| (see.v x)$^+$)$^+$)	Global polarity

Proof

1.	(|Abelard| (see.v (a.d (male.a carp.n))))	Assumption
2.	((every.d carp.n) (be.v (= (a.d fish.n))))	Assumption
3.	(|Abelard| (see.v (a.d (male.a carp.n$^+$))))	Polarity marking, 1.
4.	(|Abelard| (see.v (a.d (male.a fish.n))))	UMI, 2.,3.

The need for addressing the intersective nature of the modification in *male fish* brings up a benefit of using ULFs as a basis for the inferences. Since ULF is explicitly underspecified, the assumptions made during the inference process must be stated. The corresponding proof presented by Sánchez-Valencia hides the assumption of intersective modification in the lexical monotonicity marking of *male* (as $((e,t)^+,(e,t))$). In ULF, modifications are assumed to be intensional unless otherwise assumed, so the intersective nature of the modifier male.n must be explicitly stated.

B.4.5 Conversion Rules

Sánchez-Valencia's (1991a) conversion rule is

$$\frac{(some\ y)^{\#}\ is\ a\ x, X \bullet Y}{(some\ x)^{\#}\ is\ a\ y, X \bullet Y}$$

Before stating the corresponding rule for ULFs, we note that the rule also works for the determiners a.d and no.d. Thus we state the ULF conversion rules as follows:

SLF Rule 4 (Conversion).

$$(d\ x\text{: } (x\ P)\ (x\ Q)) \leftrightarrow (d\ y\text{: } (y\ Q)\ (y\ P))$$

where $d \in \{$some.d, a.d, no.d$\}$.

Correctness of this rule can be argued using the definitions of the generalized quantifiers 'some', 'a', and 'no' and subset relations under interpretation.

Using ULFs and chaining this rule with the be.v inference, we get the following rule.

ULF Rule 3 (Conversion).

(($d1$ P) (be.v (= ($d2$ Q))))
$\leftrightarrow$ (($d1$ Q) (be.v (= ($d2$ P))))

where $d1 \in \{$some.d, a.d, no.d$\}$ and $d2 \in \{$some.d, a.d$\}$.

B.5 Boolean Connectives

Sánchez-Valencia (1991a) handles generalized boolean connectives by allowing connectives (*and*, *or*) to have the type $(a, (a, a))$, where a is any complex category ending in t. Then the inference rules appropriately substitute one of the connective constituents for the entire phrase. The rules, which we will not list here, have a few versions depending on the position of the connective due to the left side of the tableau nodes being interpreted as connected with conjunctions and the right side with disjunctions.

ULF handles this similarly, without the tableau-specific details by allowing connectives to be interpreted as $\langle A, \langle A, A\rangle\rangle$ for an arbitrary type A. They are interpreted as generalized lambda expressions.

Definition B.1 (ULF Generalized Connective).

$$(A\ \chi\ B) \leftrightarrow$$
$$(\lambda\ x_1, ..., x_n\text{: } ((A\ x_1...x_n)\ \chi\ (B\ x_1...x_n)))$$

where $\chi \in \{$and.cc, or.cc$\}$ and both A and B are prefix operators with arity n. Infix operators are defined in the equivalent way while respecting the predicate position relative to the arguments.

This along with the observation that the following formulas hold true through the intersective and unionistic nature of conjunction and disjunction, respectively, allow us to use simple monotonicity rules in the context of boolean connectives.

$$(\lambda\ x_1, ..., x_n\text{: } ((A\ x_1...x_n) \text{ and.cc } (B\ x_1...x_n))) \subseteq A, B$$
$$A, B \subseteq (\lambda\ x_1, ..., x_n\text{: } ((A\ x_1...x_n) \text{ or.cc } (B\ x_1...x_n)))$$

Appendix C Polarity-based EL Inference

EL supports two forward inference rules and two goal-based inference rules that operate on substitutions under appropriate polarity contexts (Schubert and Hwang, 2000). Here we present a couple of examples and connect it to the inference rules in ULF. First, the forward inference rules, called Rule Instantiation (RI):

RI-1
$$\frac{MAJ(\phi^-), MIN(\phi'^+)}{MAJ_\sigma(\neg MIN_\sigma(\bot^+)^-)}$$

RI-2
$$\frac{MAJ(\phi^-), MIN(\phi'^+)}{MIN_\sigma(MAJ_\sigma(\top^-)^+)}$$

where RI-1 is sound if the only variables in the matching expression (ϕ') of the minor premise (MIN) are "matchably bound,"—bound within ϕ' or by a universal quantifier in positive polarity context or existential quantifier in negative polarity context—and RI-2 is sound if the only variables in the matching expression (ϕ) of the major premise (MAJ) are "matchably bound."

It turns out that the monotonicity rule presented by Sánchez Valencia (1991a) is a special case of RI-2. Here is an example to demonstrate a monotonicity inference over SLFs, which for this inference is sufficiently disambiguated.

Proof

1.	*Every carp is a fish*	Assumption
2.	*Abelard sees a carp*	Assumption
3.	(every.d x: (x carp.n) (a.d y: (y fish.n) (be.v (= y))))	SLF for 1.
4.	(a.d x: (y carp.n) (|Abelard| (see.v y)))	SLF for 2.
5.	(every.d x: (x carp.n) (x fish.n))	be.v Elim, 3.
6.	(every.d x: (x carp.n)$^-$ (x fish.n)$^+$)	Polarity marking, 5.
7.	(a.d y: (y carp.n)$^+$ (|Abelard| (see.v y))$^+$)	Polarity marking, 4.
8.	(a.d y: (y fish.n)$^+$ (|Abelard| (see.v y))$^+$)	RI-2, 6.,7. (see C)
9.	*Abelard sees a fish*	English for 8.

Step-by-step RI-2 application

1. (every.d x: (x carp.n)$^-$ (x fish.n)$^+$) MAJ

2. (a.d y: (y carp.n)$^+$ (|Abelard| (see.v y))$^+$) — MIN

3. $\top \rightarrow$ (y fish.n)$^+$ — Converted MAJ, $\{x/y\}$, 1.

4. $\bot \vee$ (y fish.n)$^+$ — $\rightarrow$ Def, 3.

5. (y fish.n)$^+$ — $\bot$ Elim, 4.

6. (a.d y: (y fish.n)$^+$ (|Abelard| (see.v y))$^+$) — Subst. of converted MAJ, 2.,5.

Notice that this proof holds for an arbitrary predicates in place of *carp* and *fish* and an arbitrary sentence where *carp* occurs in positive polarity context in place of *Abelard sees a carp*. Thus, RI-2 is a generalization of Sánchez's monotonicity rule.

$$(every\ x)^{\#}\ is\ a\ y, F(x^+), X \bullet Y$$
$$(every\ x)^{\#}\ is\ a\ y, F(y), X \bullet Y$$

Note that RI-2 can also handle inferences where the major premise is a more complex construction than "every p is a q". In episodic logic, RI-2 can be used to conclude *Something is a cap or pretty if Little Red Riding Hood wears it* from *Every dress or hood that Little Red Riding Hood wears is pretty* and *Something is a cap or a hood* (Schubert and Hwang, 2000).

Additionally, RI-1 is a generalization of the reverse inference: substituting in more specific predicates when in negative polarity.

Step-by-step RI-1 application

1. (every.d x: (x carp.n)$^-$ (x fish.n)$^+$) — MIN

2. (no.d y: (y fish.n)$^-$ (|Abelard| (see.v y))$^-$) — MAJ

3. (no.d x: (x fish.n)$^-$ (|Abelard| (see.v x))$^-$) — Converted MAJ, $\{y/x\}$, 2.

4. $\neg$((x carp.n)$^- \rightarrow \bot^+$) — Converted MIN, 1.

5. $\neg$($\neg$(x carp.n)$^- \vee \bot^+$) — $\rightarrow$ Def, 4.

6. $\neg\neg$(x carp.n)$^- \wedge \top^+$ — de Morgan, 5.

7. (x carp.n)$^- \wedge \top^+$ — $\neg$ Elim, 6.

8. (x carp.n)$^-$ — $\top$ Annih, 7.

9. (no.d x: (x carp.n)$^-$ (|Abelard| (see.v x))$^-$) — Subst. of converted MIN, 3.,8.

We've already shown that RI subsumes the specialized monotonicity inference presented by Sánchez Valencia (1991a). Now, we will show that in first order contexts RI also subsumes a more general presentation of natural logic inference by Sánchez Valencia (1991b). The upward monotonicity inference in positive contexts is written by Sánchez Valencia as

$$\frac{[\![M]\!] \leq [\![M']\!] \quad [\![N(M)]\!]}{[\![N(M')]\!]}$$

Where $[\![]\!]$ is the denotation function and $\leq$ is the monotonicity ordering from Definition A.1. We will refer to this rule as SVMI. First, we show that the RI-2 inference in first order contexts can be interpreted in this form.

Since RI-2 substitutes $MAJ_\sigma(\top)$ for ϕ'^+ in $MIN(\phi'^+)$, if we can show that $[\![\phi']\!] \leq [\![MAJ_\sigma(\top)]\!]$, then RI-2 can be justified through SVMI. As $MAJ(\phi)$ is assumed in RI-2, if $\phi = \top$, then for all models satisfying the assumptions, $MAJ(\top) = \top$.[7] Basically, $MAJ(\phi), (\phi = \top) \rightarrow MAJ(\top)$. This by definition of $\leq$ (A.1) satisfies $[\![\phi]\!] \leq [\![MAJ(\top)]\!]$: in any case where ϕ is true, so is $MAJ(\top)$. Since ϕ and ϕ' are matchably bound, their differences are irrelevant in the above justification and can be substituted for each other. Thus, for any application of RI-2 $[\![\phi']\!] \leq [\![MAJ_\sigma(\top)]\!]$ holds, and therefore the inference can be justified through SVMI.

Now, we show that for any SVMI inference where M and M' are wffs, it can be written in the form of RI-2. $[\![M]\!] \leq [\![M']\!]$ can be restated as $(\forall_x M \rightarrow M')$, which we identify as $MAJ(\phi^-)$, where M is *phi* and we know that M is in a negative polarity context due to $\forall$. $N(M)$ is identified as $MIN(\phi'^+)$ where M is ϕ' and we know is in a positive polarity context by assumption. $MAJ(\top) = \top \rightarrow M' = M'$ so $MIN(MAJ(\top)) = N(M')$.

We conjecture that this generalizes to all M and M' that are not wffs. The monadic predicate case seems simple enough through a connection with modus ponens, but proofs for cases such as determiners and variables are more elusive.

Appendix D Traditional Aristotelian Syllogisms in ULF

In this appendix, we show that similar to Sánchez-Valencia's (1991a) Natural Logic, the inference system described for ULFs can explain traditional syllogistic inference. We will give proofs for the

[7] This step requires the first-order context. In intensional contexts, the substitution via equality is not justified.

syllogisms of the first figure using ULFs. Since all other syllogisms can be derived from these, this is sufficient to show that all traditional syllogisms can be derived.

D.1 Scoping and Polarity Derivations

Before proving the syllogisms we go through the scoping and polarity derivations of the propositions used in the syllogisms.

Proposition i. "Every X is a Y"

1.	((every.d X) (be.v (= (a.d Y))))	ULF
2.	(every.d x: (x X) (x (be.v (= (a.d Y)))))	Scope every.d
3.	(every.d x: (x X) (a.d y: (y Y) (x (be.v (= y)))))	Scope a.d
4.	(every.d$_{(-,+)}$ x: (x X) (a.d$_{(+,+)}$ y: (y Y) (x (be.v (= y)))))	Lexical monotonicity
5.	(every.d x: (x X)$_-$ (a.d y: (y Y)$_+$ (x (be.v (= y)))$_+$)$_+$)	Local entail. context
6.	(every.d x: (x X)$^-$ (a.d y: (y Y)$^+$ (x (be.v (= y)))$^+$)$^+$)	Global polarity

Proposition ii. "Some X is a Y"

1.	((some.d X) (be.v (= (a.d Y))))	ULF
2.	(some.d x: (x X) (a.d y: (y Y) (x (be.v (= y)))))	Scoping
3.	(some.d$_{(+,+)}$ x: (x X) (a.d$_{(+,+)}$ y: (y Y) (x (be.v (= y)))))	Lexical monotonicity
4.	(some.d x: (x X)$_+$ (a.d y: (y Y)$_+$ (x (be.v (= y)))$_+$)$_+$)	Local entail. context
5.	(some.d x: (x X)$^+$ (a.d y: (y Y)$^+$ (x (be.v (= y)))$^+$)$^+$)	Global polarity

Proposition iii. "No X is a Y"

1.	((no.d X) (be.v (= (a.d Y))))	ULF
2.	(no.d x: (x X) (a.d y: (y Y) (x (be.v (= y)))))	Scoping
3.	(no.d$_{(-,-)}$ x: (x X) (a.d$_{(+,+)}$ y: (y Y) (x (be.v (= y)))))	Lexical monotonicity
4.	(no.d x: (x X)$_-$ (a.d y: (y Y)$_+$ (x (be.v (= y)))$_+$)$_-$)	Local entail. context
5.	(no.d x: (x X)$^-$ (a.d y: (y Y)$^-$ (x (be.v (= y)))$^-$)$^-$)	Global polarity

Proposition iv. "Not every X is a Y"

1.	(not (every.d X) (be.v (= (a.d Y))))	ULF
2.	(not (every.d x: (x X) (a.d y: (y Y) (x (be.v (= y))))))	Scoping
3.	(not$_{(-)}$ (every.d$_{(-,+)}$ x: (x X) (a.d$_{(+,+)}$ y: (y Y) (x (be.v (= y))))))	Lexical monotonicity
4.	(not (every.d x: (x X)$_-$ (a.d y: (y Y)$_+$ (x (be.v (= y)))$_+$)$_+$)$_-$)	Local entail. context
5.	(not (every.d x: (x X)$^+$ (a.d y: (y Y)$^-$ (x (be.v (= y)))$^-$)$^-$)$^-$)	Global polarity

D.2 Deriving the Syllogisms

Syllogism 1 (Barbara). "Every M is a P" and "Every S is a M" entail "Every S is a P".

Proof

1.	((every.d M) (be.v (= (a.d P))))	Assumption
2.	((every.d S) (be.v (= (a.d M))))	Assumption
3.	((every.d M)$^-$ (be.v (= (a.d P))))	Polarity marking, 1.
4.	((every.d S) (be.v (= (a.d P))))	UMI, 2.,3.

Syllogism 2 (Darii). "Every M is a P" and "Some S is a M" entail "Some S is a P".

Proof

1.	((every.d M) (be.v (= (a.d P))))	Assumption
2.	((some.d S) (be.v (= (a.d M))))	Assumption
3.	((some.d S) (be.v (= (a.d M)$^+$)))	Polarity marking, 2.
4.	((some.d S) (be.v (= (a.d P))))	UMI, 1.,3.

Syllogism 3 (Celarent). "No M is a P" and "Every S is a M" entail "No S is a P".

Proof

1.	((no.d M) (be.v (= (a.d P))))	Assumption
2.	((every.d S) (be.v (= (a.d M))))	Assumption
3.	((no.d M)$^-$ (be.v (= (a.d P))))	Polarity marking, 1.
4.	((no.d S) (be.v (= (a.d P))))	UMI, 2.,3.

Syllogism 4 (Ferio). "No M is a P" and "Some S is a M" entail "Not every S is a P".

Using the logical interpretation of Sánchez-Valencia's Negation Rule twice (1), we see that the syllogism is true iff "Every S is a P" and "Some S is a M" entail "Some M is a P". We prove this as follows.

Proof

1.	((every.d S) (be.v (= (a.d P))))	Assumption
2.	((some S) (be.v (= (a.d M))))	Assumption
3.	((some S)$^+$ (be.v (= (a.d M))))	Polarity marking, 2.
4.	((some P) (be.v (= (a.d M))))	UMI, 1.,3.
5.	((some M) (be.v (= (a.d P))))	Conversion, 4.

This could alternatively be proved by contradiction without the use of the equivalent of the Negation Rule, where "Not every S is a P" becomes negated to "every S is a P", after which, applying UMI and Conversion leads to a contradiction.