Polarity Computations in Flexible Categorial Grammar

Hai Hu *huhai@indiana.edu*, Lawrence S. Moss *lsm@cs.indiana.edu*

Indiana University

Abstract

This paper shows how to take parse trees in CCG and algorithmically find the polarities of all the constituents. Our work uses the well-known polarization principle corresponding to function application, and we have extended this with principles for type raising and composition. We provide an algorithm, extending the polarity marking algorithm of van Benthem. We discuss how our system works in practice, taking input from the C&C parser.

Main Objective

Polarize sentences to get inferences. For example:

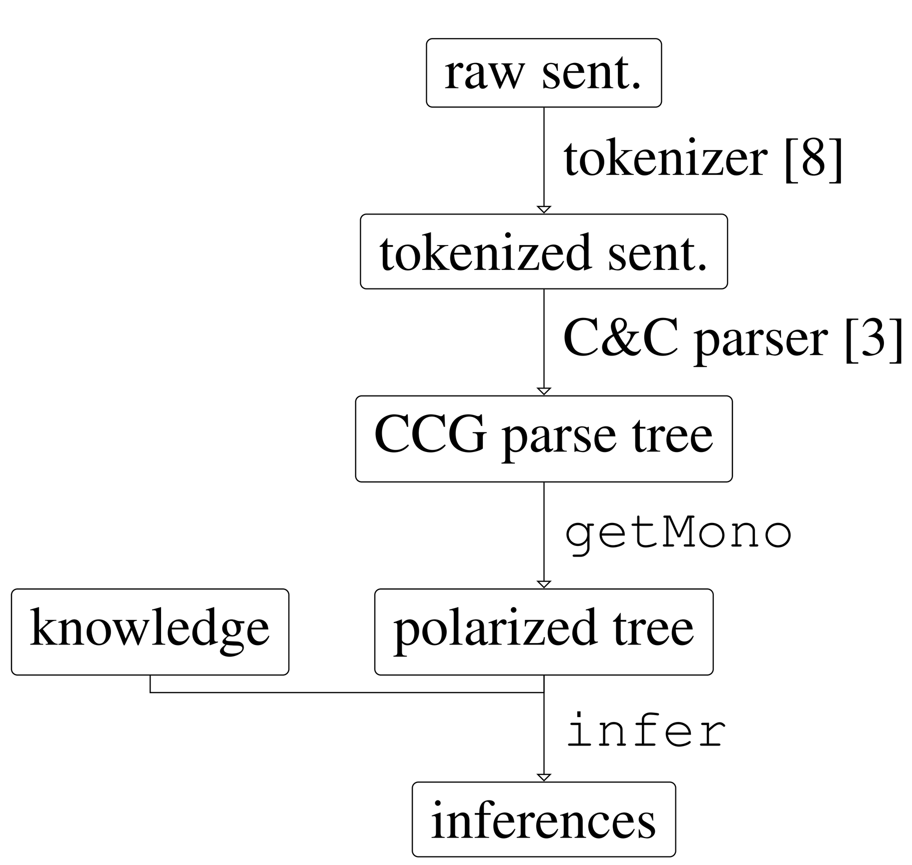
Raw: *Every dog scares at least two cats.*

Polarized: *Every dog*$^{\downarrow}$ *scares*$^{\uparrow}$ *at least two*$^{\downarrow}$ *cats*$^{\uparrow}$.

Knowledge base: *cats* $\leq$ *animals*, *beagles* $\leq$ *dogs*, *scares* $\leq$ *startles*.

Inference: *Every beagle startles at least one animal.*

Pipeline



This work: `getMono` and `infer`

Theory

1. **Meaning of d: $\uparrow$ and $\downarrow$**

$\mathbb{P}$ and $\mathbb{Q}$ are *preorders* as used in [9, 6].

A function $f : \mathbb{P} \to \mathbb{Q}$ is *monotone* ($\uparrow$ or *order preserving*) if $p \leq q$ in $\mathbb{P}$ implies $f(p) \leq f(q)$ in $\mathbb{Q}$. And f is *antitone* ($\downarrow$ or *order inverting*) if $p \leq q$ in $\mathbb{P}$ implies $f(q) \leq f(p)$ in $\mathbb{Q}$.

E.g. *every dog*$^{\downarrow}$ *barks*$^{\uparrow}$ means:
For all models $\mathcal{M}$, all $m_1 \leq m_2$ in $\mathbb{P}_{et}$ (for *dog*), and all $n_1 \leq n_2$ in $\mathbb{P}_{(et)t}$ (for *barks*), we have in $\mathbb{2}$ that $[\![every]\!]\ m_2\ n_1 \leq [\![every]\!]\ m_1\ n_2$.

2. **Meaning of m: $+$ and $-$**

We incorporate monotonicity information into the types. Our lexicon comes with order-enriched **semantic** types, e.g.:

every : $N \xrightarrow{-} NP^{+}$; *no* : $N \xrightarrow{-} NP^{-}$

some : $N \xrightarrow{+} NP^{+}$; *most* : $N \xrightarrow{\cdot} NP^{+}$

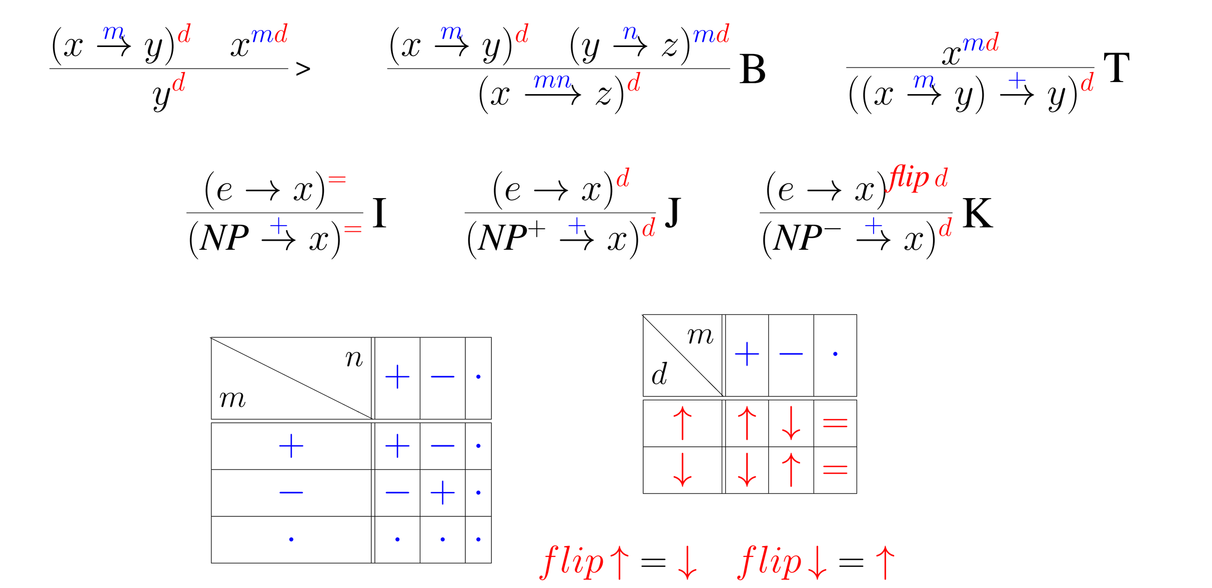
where $N = e \to t$, $NP^{+} = (e \to t) \xrightarrow{+} t$

Rules

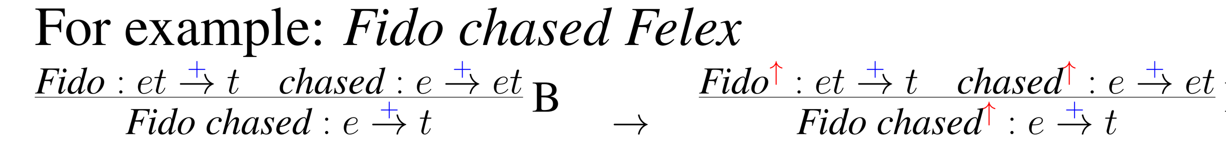
Our algorithm `getMono` has two steps, similar to van Benthem's algorithm [2]:

1. `mark()`: leaves $\to$ root (going *down*).
2. `polarize()`: root $\to$ leaves (going *up*).

Both operations follow the rules below.



For example: *Fido chased Felex*



Inference: An Example

Suppose we have this polarized sentence S:

every$^{\uparrow}$ *man*$^{\downarrow}$ *chased*$^{\uparrow}$ *some*$^{\uparrow}$ *cat*$^{\uparrow}$.$^{\uparrow}$

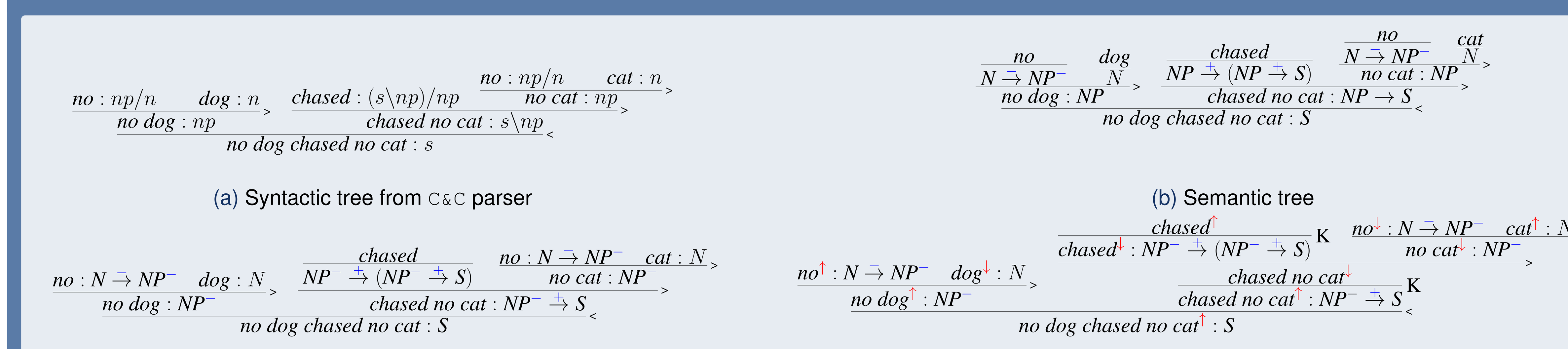
and a knowledge base K:

cat $\leq$ *animal*	*every man* $\leq$ *John* $\leq$ *some man*
old dog $\leq$ *dog*	*chased some cat* $\leq$ *liked every dog*
young man $\leq$ *man*	*every* $\leq$ *most*

Using InferBySubstitution, we can get:

After 1 substitution	every$^{\uparrow}$ young$^{\downarrow}$ man$^{\downarrow}$ chased$^{\uparrow}$ some$^{\uparrow}$ cat$^{\uparrow}$ most$^{\uparrow}$ man$^{=}$ chased$^{\uparrow}$ some$^{\uparrow}$ cat$^{\uparrow}$ John$^{\uparrow}$ chased$^{\uparrow}$ some$^{\uparrow}$ cat$^{\uparrow}$ every$^{\uparrow}$ man$^{\downarrow}$ liked$^{\uparrow}$ every$^{\uparrow}$ dog$^{\downarrow}$
After 2 substitutions	every$^{\uparrow}$ young$^{\downarrow}$ man$^{\downarrow}$ chased$^{\uparrow}$ some$^{\uparrow}$ animal$^{\uparrow}$ John$^{\uparrow}$ liked$^{\uparrow}$ every$^{\uparrow}$ dog$^{\downarrow}$ every$^{\uparrow}$ man$^{\downarrow}$ liked$^{\uparrow}$ every$^{\uparrow}$ old$^{\downarrow}$ dog$^{\downarrow}$
After 3 substitutions	some$^{\uparrow}$ man$^{\uparrow}$ chased$^{\uparrow}$ some$^{\uparrow}$ animal$^{\uparrow}$ John$^{\uparrow}$ liked$^{\uparrow}$ every$^{\uparrow}$ old$^{\downarrow}$ dog$^{\downarrow}$ every$^{\uparrow}$ young$^{\downarrow}$ man$^{\downarrow}$ liked$^{\uparrow}$ every$^{\uparrow}$ old$^{\downarrow}$ dog$^{\downarrow}$

A Complete Example: *no dog chased no cat*



(a) Syntactic tree from C&C parser

(b) Semantic tree

(c) After `mark()`

(d) After `polarize()`

Current Capabilities

No$^{\uparrow}$ *man*$^{\downarrow}$ *walks*$^{\downarrow}$

Every$^{\uparrow}$ *man*$^{\downarrow}$ *and*$^{\uparrow}$ *some*$^{\uparrow}$ *woman*$^{\uparrow}$ *sleeps*$^{\uparrow}$

Every$^{\uparrow}$ *man*$^{\downarrow}$ *and*$^{\uparrow}$ *no*$^{\uparrow}$ *woman*$^{\downarrow}$ *sleeps*$^{=}$

If$^{\uparrow}$ *some*$^{\downarrow}$ *man*$^{\downarrow}$ *walks*$^{\downarrow}$, *then*$^{\uparrow}$ *no*$^{\uparrow}$ *woman*$^{\downarrow}$ *runs*$^{\downarrow}$

Every$^{\uparrow}$ *man*$^{\downarrow}$ *does*$^{\downarrow}$ *n't*$^{\uparrow}$ *hit*$^{\downarrow}$ *every*$^{\downarrow}$ *dog*$^{\uparrow}$

No$^{\uparrow}$ *man*$^{\downarrow}$ *that*$^{\downarrow}$ *likes*$^{\downarrow}$ *every*$^{\downarrow}$ *dog*$^{\uparrow}$ *sleeps*$^{\downarrow}$

Most$^{\uparrow}$ *men*$^{=}$ *that*$^{=}$ *every*$^{=}$ *woman*$^{=}$ *hits*$^{=}$ *cried*$^{\uparrow}$

Every$^{\uparrow}$ *young*$^{\downarrow}$ *man*$^{\downarrow}$ *that*$^{\uparrow}$ *no*$^{\uparrow}$ *young*$^{\downarrow}$ *woman*$^{\downarrow}$ *hits*$^{\uparrow}$ *cried*$^{\uparrow}$

Inference [5]

Input: a polarized sentence S, a *knowledge base* K.
Output: inferences of S based on K.
Knowledge base K: a set of $\leq$ pairs:

cat $\leq$ *animal*   *old dog* $\leq$ *dog*

Algorithm 0.1: InferBySubstitution(S, K)

for each *Constituent* C $\in$ S
do { **if** C $\in$ *pair* P *in* K *where polarity matches*
then { *replace* C *with* C′ *in* P; *add new sentence* S′ *to* `S.inferences` }

Conclusion

We have shown how to polarize a CCG parse tree and make simple inferences based on the result. Our paper relates to other work with similar aims, but not in the CCG context, e.g. [7, 10], as well as other work on natural logic [4, 11, 1].

Main References

[1] Lasha Abzianidze. "LangPro: Natural Language Theorem Prover". In: *CoRR* abs/1708.09417 (2017).

[2] Johan van Benthem. *Essays in Logical Semantics*. Vol. 29. Studies in Linguistics and Philosophy. Dordrecht: D. Reidel Publishing Co., 1986, pp. xii+225.

[3] Stephen Clark and James R Curran. "Wide-coverage efficient statistical parsing with CCG and log-linear models". In: *Computational Linguistics* 33.4 (2007), pp. 493–552.

[4] Yaroslav Fyodorov, Yoad Winter, and Nissim Fyodorov. "Order-based inference in natural logic". In: *Log. J. IGPL* 11.4 (2003). Inference in computational semantics: the Dagstuhl Workshop 2000, pp. 385–417.

[5] Hai Hu, Thomas Icard, and Larry Moss. "Automated Reasoning from Polarized Parse Trees". In: *Proceedings of the Fifth Workshop on Natural Language and Computer Science*. 2018.

[6] Thomas F. Icard and Lawrence S. Moss. "Recent Progress on Monotonicity". In: *Linguistic Issues in Language Technology* 9.7 (2014), pp. 167–194.

[7] Bill MacCartney and Christopher D. Manning. "An extended model of natural logic". In: *IWCS-8, Proceedings of the Eighth International Conference on Computational Semantics*. 2009, pp. 140–156.

[8] Pascual Martínez-Gómez et al. "ccg2lambda: A Compositional Semantics System". In: *Proceedings of ACL 2016 System Demonstrations*. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 85–90.

[9] Lawrence S. Moss. "The Soundness of Internalized Polarity Marking". In: *Studia Logica* 100.4 (2012), pp. 683–704.

[10] Rowan Nairn, Cleo Condoravdi, and Lauri Karttunen. "Computing relative polarity for textual inference". In: *Proceedings of ICoS-5 (Inference in Computational Semantics)*. Buxton, UK, 2006.

[11] Anna Zamansky, Nissim Francez, and Yoad Winter. "A 'natural logic' inference system using the Lambek calculus". In: *Journal of Logic, Language, and Information* 15.3 (2006), pp. 273–295.

Our program can be found at: https://github.com/huhailinguist/ccg2mono

Hai Hu is supported by China Scholarship Council.