Natural Solution to FraCaS Entailment Problems

Lasha Abzianidze
TiLPS, Tilburg University, the Netherlands
L.Abzianidze@uvt.nl

Abstract

Reasoning over several premises is not a common feature of RTE systems as it usually requires deep semantic analysis. On the other hand, FraCaS is a collection of entailment problems consisting of multiple premises and covering semantically challenging phenomena. We employ the tableau theorem prover for natural language to solve the FraCaS problems in a *natural* way. The expressiveness of a type theory, the transparency of natural logic and the schematic nature of tableau inference rules make it easy to model challenging semantic phenomena. The efficiency of theorem proving also becomes challenging when reasoning over several premises. After adapting to the dataset, the prover demonstrates state-of-the-art competence over certain sections of FraCaS.

1 Introduction

Understanding and automatically processing the natural language semantics is a central task for computational linguistics and its related fields. At the same time, inference tasks are regarded as the best way of testing an NLP systems's semantic capacity (Cooper et al., 1996, p. 63). Following this view, recognizing textual entailment (RTE) challenges (Dagan et al., 2005) were regularly held which evaluate the RTE systems based on the RTE dataset. The RTE data represents a set of text-hypotheses pairs that are human annotated on the inference relations: *entailment*, *contradiction* and *neutral*. Hence it attempts to evaluate the systems on human reasoning. In general, the RTE datasets are created semi-automatically and are often motivated by the scenarios found in the applications like question answering, relation extraction, information retrieval and summarization (Dagan et al., 2005; Dagan et al., 2013). On the other hand, the semanticists are busy designing theories that account for the valid logical relations over natural language sentences. These theories usually model reasoning that depends on certain semantic phenomena, e.g., Booleans, quantifiers, events, attitudes, intensionality, monotonicity, etc. These types of reasoning are weak points of RTE systems as the above mentioned semantic phenomena are underrepresented in the RTE datasets.

In order to test and train the weak points of an RTE system, we choose the FraCaS dataset (Cooper et al., 1996). The set contains complex entailment problems covering various challenging semantic phenomena which are still not fully mastered by RTE systems. Moreover, unlike the standard RTE datasets, FraCaS also allows multi-premised problems. To account for these complex entailment problems, we employ the theorem prover for higher-order logic (Abzianidze, 2015a), which represents the version of formal logic motivated by *natural logic* (Lakoff, 1970; Van Benthem, 1986). Though such expressive logics usually come with the inefficient decision procedures, the prover maintains efficiency by using the inference rules that are specially tailored for the reasoning in natural language. We introduce new rules for the prover in light of the FraCaS problems and test the rules against the relevant portion of the set. The test results are compared to the current state-of-the-art on the dataset.

The rest of the paper is structured as follows. We start with introducing a tableau system for natural logic (Muskens, 2010). Section 3 explores the FraCaS dataset in more details. In Section 4, we describe the process of adapting the theorem prover to FraCaS, i.e. how specific semantic phenomena are modeled with the help of tableau rules. Several premises with monotone quantifiers in-

*Proceedings of the Fifth Joint Conference on Lexical and Computational Semantics (*SEM 2016)*, pages 64–74,
Berlin, Germany, August 11-12, 2016.

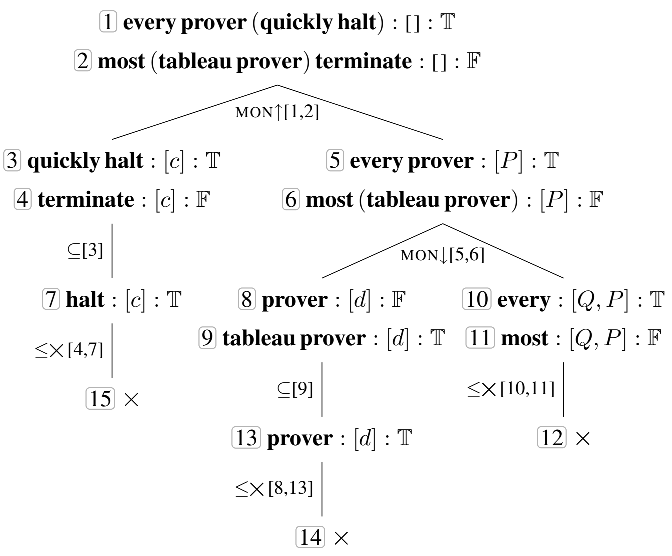

Figure 1: A closed tableau proves that *every prover halts quickly* entails *most tableau provers terminate*. Each branch growth is marked with the corresponding rule application.

crease the search space for proofs. In Section 5, we present several rules that contribute to shorter proofs. In the evaluation part (Section 6), we analyze the results of the prover on the relevant FraCaS sections and compare them with the related RTE systems. We end with possible directions of future work.

2 Tableau theorem prover for natural language

Reasoning in formal logics (i.e., a formal language with well-defined semantics) is carried out by automated theorem provers, where the provers come in different forms based on their underlying proof system. In order to mirror this scenario for reasoning in natural language, Muskens (2010) proposed to approximate natural language with a version of natural logic (Lakoff, 1970; Van Benthem, 1986; Sánchez-Valencia, 1991) while a version of analytic tableau method (Beth, 1955; Hintikka, 1955; Smullyan, 1968), hereafter referred to as *natural tableau*, is introduced as a proof system for the logic. The version of natural logic employed by Muskens (2010) is higher-order logic formulated in terms of the typed lambda calculus (Church, 1940).[1] As a result, the logic is much more expressive (in the sence of modeling certian phenomena in an intuitive way) than first-order logic, e.g., it can naturally account for generalized quantifiers (Montague, 1973; Barwise and Cooper, 1981), monotonicity calculus (Van Benthem, 1986; Sánchez-Valencia, 1991; Icard and Moss, 2014) and subsective adjectives.

What makes the logic *natural* are its terms, called Lambda Logical Forms (LLFs), which are built up only from variables and lexical constants via the functional application and λ-abstraction. In this way the LLFs have a more natural appearance than, for instance, the formulas of first-order logic. The examples of LLFs are given in the nodes of the tableau proof tree in Figure 1, where the type information for terms is omitted. A tableau node can be seen as a statement of truth type which is structured as a triplet of a main LLF, an argument list of terms and a truth sign. The semantics associated with a tableau node is that the application of the main LLF to the terms of an argument list is evaluated according to the truth sign. For instance, the node $\boxed{9}$ is interpreted as the term **tableau prover** d being true, i.e. d is in the extension of **tableau prover**. Notice that LLFs not only resemble surface forms in terms of lexical elements but most of their constituents are in correspondence too. This facilitates the automatized generation of LLFs from surface forms.

The natural tableau system of (Muskens, 2010), like any other tableau systems (D'Agostino et al., 1999), tries to prove statements by refuting them. For instance, in case of an entailment proof, a tableau starts with the counterexample where the premises are true and the conclusion is false. The proof is further developed with the help of schematic inference rules, called tableau rules (see Figure 2). A tableau is closed if all its branches are closed, i.e. are marked with a closure (×) sign. A tableau branch intuitively corresponds to a situation while a closed branch represents an inconsistent situation. Refutation of a statement fails if a closed tableau is obtained. Hence the closed tableau serves as a proof for the statement. The proof of an entailment in terms of the closed tableau is demonstrated in Figure 1. The tableau starts with the counterexample ($\boxed{1}$,$\boxed{2}$) of the entailment. It is further developed by applying the rule (MON↑) to $\boxed{1}$ and $\boxed{2}$, taking into account that

[1] More specifically, the logic is two-sorted variant of Russell's type theory, which according to Gallin (1975) represents a more general and neat formulation of Montague (1970)'s intensional logic. For theorem proving, we employ one-sorted type theory, i.e. with the entity e and truth t types, and hence omit a type s for world-time pairs.

$$\frac{\begin{array}{c} G\,A : [\vec{C}] : \mathbb{T} \\ H\,B : [\vec{C}] : \mathbb{F} \end{array}}{\begin{array}{ll} A : [\vec{d}\,] : \mathbb{T} & G : [P, \vec{C}] : \mathbb{T} \\ B : [\vec{d}\,] : \mathbb{F} & H : [P, \vec{C}] : \mathbb{F} \end{array}}\,\text{MON}\!\uparrow$$

G or H is mon↑ and $\vec{d}$ and P are fresh

$$\frac{\begin{array}{c} G\,A : [\vec{C}] : \mathbb{T} \\ H\,B : [\vec{C}] : \mathbb{F} \end{array}}{\begin{array}{ll} A : [\vec{d}\,] : \mathbb{F} & G : [P, \vec{C}] : \mathbb{T} \\ B : [\vec{d}\,] : \mathbb{T} & H : [P, \vec{C}] : \mathbb{F} \end{array}}\,\text{MON}\!\downarrow$$

G or H is mon↓ and $\vec{d}$ and P are fresh

$$\frac{A\,N : [\vec{C}] : \mathbb{T}}{N : [\vec{C}] : \mathbb{T}}\subseteq \text{ where } A \text{ is subsective}$$

$$\frac{\begin{array}{c} A : [\vec{C}] : \mathbb{T} \\ B : [\vec{C}] : \mathbb{F} \end{array}}{\times}\leq\!\times \;\begin{array}{l}\text{where } A \text{ entails } B \\ \text{written as } A \leq B\end{array}$$

Figure 2: The tableau rules employed by the tableau proof in Figure 1

every is upward monotone in the second argument position. The rule application is carried out until all branches are closed or no new rule application is possible. In the running example, all the branches close as ($\leq\times$) identifies inconsistencies there; for instance, $\boxed{4}$ and $\boxed{7}$ are inconsistent according to ($\leq\times$) assuming that a knowledge base (KB) provides that *halting* entails *termination*, i.e. **halt** $\leq$ **terminate**.

The natural tableau system was succesfully applied to the SICK textual entailment problems (Marelli et al., 2014) by Abzianidze (2015a). In particular, the theorem prover for natural language, called LangPro, was implemented that integrates three modules: the parsers for Combinatory Categorial Grammar (CCG) (Steedman, 2000), LLFgen that generates LLFs from the CCG derivation trees, and the natural logic tableau prover (NLogPro) which builds tableau proofs. The pipeline architecture of the prover is depicted in Figure 3: the sentences of an input problem are first parsed, then converted into LLFs, which are further processed by NLogPro. For a CCG parser, there are at least two options, C&C (Clark and Curran, 2007; Honnibal et al., 2010) and EasyCCG (Lewis and Steedman, 2014). The inventory of rules (IR) of NLogPro is a crucial component for the prover; it contains most of the rules found in (Muskens, 2010) and also additional rules that were collected from SICK. In order to make theorem proving robust, LangPro employs a conservative extension of the type theory for accessing the syntactic information of terms (Abzianidze, 2015b): in addition to the basic semantic types e and t, the extended type theory incorporates basic syntactic types $\mathtt{n}, \mathtt{np}, \mathtt{s}$ and $\mathtt{pp}$ corresponding to the primitive categories of CCG.

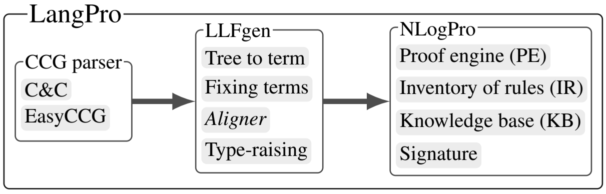

Figure 3: The architecture of LangPro

Abzianidze (2015a) shows that on the unseen portion of SICK LangPro obtains the results comparable to the state-of-the-art scores while achieving an almost perfect precision. Based on this inspiring result, we decide to adapt and test LangPro on the FraCaS problems, from the semantics point of view much more harder than the SICK ones.[2]

3 FraCaS dataset

The FraCaS test suite (Cooper et al., 1996) is a set of 346 test problems. It was prepared by the FraCaS consortium as an initial benchmark for semantic competence of NLP systems. Each FraCaS problem is a pair of premises and a yes-no-unknown question that is annotated with a gold judgment: *yes* (entailment), *no* (contradiction), or *unknown* (neutral). The problems mainly consist of short sentences and resemble the problems found in introductory logic books. To convert the test suite into the style of RTE dataset, MacCartney and Manning (2007) translated the questions into declarative sentences. The judgments were copied from the original test suite with slight modifications.[3] Several problems drawn from the obtained FraCaS dataset are presented in Table 1.

Unlike other RTE datasets, the FraCaS problems contain multiple premises (45% of the total

[2] An online version of LangPro is available at: `http://lanthanum.uvt.nl/labziani/tableau/`

[3] More details about the conversion, including information about several *noisy* problems (e.g., a problem missing a premise or hypothesis, or having a non-standard gold answer) can be found in MacCartney (2009). The FraCaS RTE dataset is available at: `http://www-nlp.stanford.edu/~wcmac/downloads/fracas.xml`

problems) and are structured in sections according to the semantic phenomena they concern. The sections cover generalized quantifiers (GQs), plurals, anaphora, ellipsis, adjectives, comparatives, temporal reference, verbs and attitudes. Due to the challenging problems it contains, the FraCaS dataset can be seen as one of the most complex RTE data from the semantics perspective. Unfortunately, due to its small size the dataset is not representative enough for system evaluation purposes. The above mentioned facts perhaps are the main reasons why the FraCaS data is less favored for developing and assessing the semantic competence of RTE systems. Nevertheless, several RTE systems (MacCartney and Manning, 2008; Angeli and Manning, 2014; Lewis and Steedman, 2013; Tian et al., 2014; Mineshima et al., 2015) were trained and evaluated on (the parts of) the dataset. Usually the goal of these evaluations is to show that specific theories/frameworks and the corresponding RTE systems are able to model deep semantic reasoning over the phenomena found in FraCaS. Our aim is also the same in the rest of the sections.

ID	FraCaS entailment problem
6 no	**P**: No really great tenors are modest. **C**: There are really great tenors who are modest.
26 yes	**P1**: Most Europeans are resident in Europe. **P2**: All Europeans are people. **P3**: All people who are resident in Europe can travel freely within Europe. **C**: Most Europeans can travel freely within Europe.
44 yes	**P1**: Few committee members are from southern Europe. **P2**: All committee members are people. **P3**: All people who are from Portugal are from southern Europe. **C**: There are few committee members from Portugal.
56 unk	**P1**: Many British delegates obtained interesting results from the survey. **C**: Many delegates obtained interesting results from the survey.
76 yes	**P1**: Few committee members are from southern Europe. **C**: Few female committee members are from southern Europe.
85 no	**P1**: Exactly two lawyers and three accountants signed the contract. **C**: Six lawyers signed the contract.
99 yes	**P1**: Clients at the demonstration were all impressed by the system's performance. **P2**: Smith was a client at the demonstration. **C**: Smith was impressed by the system's performance.
100 yes	**P**: Clients at the demonstration were impressed by the system's performance. **C**: Most clients at the demonstration were impressed by the system's performance.
211 no	**P1**: All elephants are large animals. **P2**: Dumbo is a small elephant. **C**: Dumbo is a small animal.

Table 1: Samples of the FraCaS problems

4 Modeling semantic phenomena

Modeling a new semantic phenomenon in the natural tableau requires introduction of special rules. The section presents the new rules that account for certain semantic phenomena found in FraCaS.

FraCaS Section 1, in short FrSec-1, focuses on GQs and their monotonicity properties. Since the rules for monotonicity are already implemented in LangPro, in order to model monotonicity behavior of a new GQ, it is sufficient to define its monotonicity features in the signature. For instance, *few* is defined as $\textbf{few}_{\mathtt{n}\downarrow,\mathtt{vp}\downarrow,\mathtt{s}}$ while *many* and *most* are modeled as $\textbf{many}_{\mathtt{n},\mathtt{vp}\uparrow,\mathtt{s}}$ and $\textbf{most}_{\mathtt{n},\mathtt{vp}\uparrow,\mathtt{s}}$ respectively.[4] The contrast between monotonicity properties of the first arguments of **few** and **many** is conditioned solely by the intuition behind the FraCaS problems: *few* is understood as an absolute amount while *many* as proportional (see Fr-56 and 76 in Table 1). Accounting for the monotonicity properties of *most*, i.e. $\textbf{most}_{\mathtt{n},\mathtt{vp}\uparrow,\mathtt{s}}$, is not sufficient for fully capturing its semantics. For instance, solving Fr-26 requires more than just upward monotonicity of *most* in its second argument. We capture the semantics, concerning *more than a half*, of *most* by the following new rule:

$$\frac{\begin{array}{l}\textbf{most}_{\mathtt{q}}\ N\ A : [\,] : \mathbb{T}\\ \textbf{most}_{\mathtt{q}}\ N\ B : [\,] : \mathbb{X}\end{array}}{\begin{array}{c}A : [c_e] : \mathbb{T}\\ B : [c_e] : \mathbb{X}\\ N : [c_e] : \mathbb{T}\end{array}}\ \textsc{most},\quad \begin{array}{l}\text{where } \mathtt{q} \equiv (\mathtt{n}, \mathtt{vp}, \mathtt{s})\\ \text{and } \mathbb{X} \text{ is either } \mathbb{T} \text{ or } \mathbb{F}\end{array}$$

With (MOST), now it is possible to prove Fr-26 (see Figure 4). The rule efficiently but partially captures the semantics of *most*. Modeling its complete semantics would introduce unnecessary inefficiency in the theorem proving.[5]

FrSec-1 involves problems dedicated to the conservativity phenomenon (1). Although we have

[4] Following the conventions in (Sánchez-Valencia, 1991), we mark the argument types with monotonicity properties associated with the argument positions. In this way, $\textbf{few}_{\mathtt{n}\downarrow,\mathtt{vp}\downarrow,\mathtt{s}}$ is downward monotone in its noun and VP arguments, where $\mathtt{vp}$ abbreviates $(\mathtt{np}, \mathtt{s})$.

[5] For complete proof-theoretic semantics of *most* wrt *same* and *all* in syllogistic logic see Endrullis and Moss (2015). Similar rules that account for additional semantics of *few* and *many* are presented in Section 5 as they coincide with efficient rules for other quantifiers.

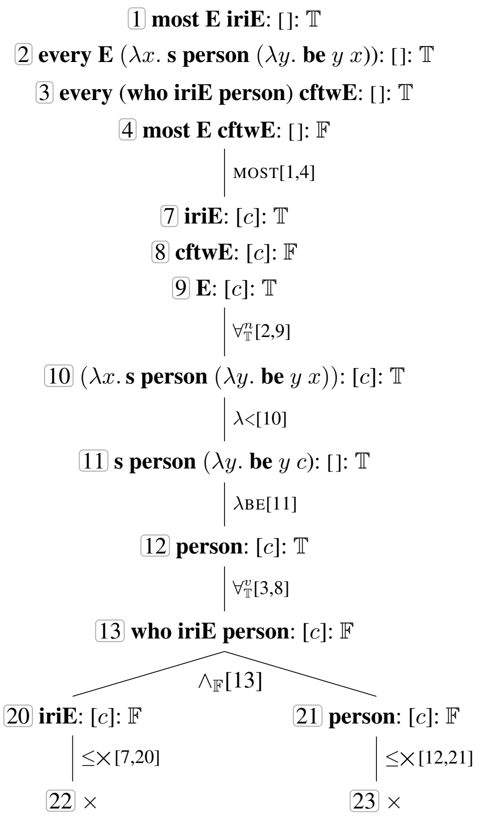

Figure 4: The tableau proof, generated by LangPro, classifies Fr-26 as entailment. The abbreviations **cftwE**, **iriE** and **E** stand for the LLFs of *can freely travel within Europe*, *is resident in Europe* and *European*, respectively. The nodes that do not contribute to the closure of the tableau are omitted. The proof also employs the admissible rules ($\forall^n_{\mathbb{T}}$) and ($\forall^v_{\mathbb{T}}$) from Section 5.

not specially modeled the conservativity property of GQs in LangPro, it is able to solve all 16 poblems about conservativity except one. The reason is that conservativity is underrepresented in FraCaS. Namely, the problems cover conservativity in the form of (2) instead of (1) (see Fr-6).

$$Q\ A \text{ are } B \leftrightarrow Q\ A \text{ are } A \text{ who are } B \tag{1}$$

$$Q\ A \text{ are } B \leftrightarrow \text{There are } Q\ A \text{ who are } B \tag{2}$$

We capture (2) with the help of the existing rules for GQs and (THR×), from (Abzianidze, 2015b), which treats the expletive constructions, like *there is*, as a universal predicate, i.e., any entity not satisfying it leads to inconsistency (×).

$$\frac{\textbf{be}\ c\ \textbf{there} : [\,] : \mathbb{F}}{\times}\ \text{THR}\times$$

But these rules are not enough for solving Fr-44 because the monotonicity rules cannot lead to the solution when applied to the following nodes representing P1 and C of Fr-44, respectively.

$$\textbf{few}\ M\ (\textbf{be from}\ S) : [\,] : \mathbb{T} \tag{3}$$

$$\textbf{few}\ (\textbf{from}\ P\ M)\ (\lambda x.\ \textbf{be}\ x\ \textbf{there}) : [\,] : \mathbb{F} \tag{4}$$

To solve Fr-44, we introduce a new tableau rule (THR_PP) which acts as a paraphrase rule. After the rule is applied to (4), (MON↓) can be applied to the resulted node and (3) which contrasts *being from southern Europe* to *being from Portugal*.

$$\frac{Q\ (p_{\mathtt{np,n,n}} A\ N)(\lambda x.\ \textbf{be}\ x\ \textbf{there}) : [\,] : \mathbb{X}}{Q\ N\ (\textbf{be}\ (p\ A)) : [\,] : \mathbb{X}}\ \text{THR\_PP}$$

FrSec-2 covers the problems concerning plurals. Usually the phrases like bare plurals, definite plurals and definite descriptions (e.g., *the dog*) do not get special treatment in wide-coverage semantic processing and by default are treated as indefinites. Since we want to take advantage of the expressive power of the logic and its proof system, we decide to separately model these phrases. We treat bare plurals and definite plurals as GQs of the form $\mathbf{s}_{\mathtt{n,vp,s}}\ N_{\mathtt{n}}$, where **s** stands for the plural morpheme. The quantifier **s** can be ambiguous in LLFs due to the ambiguity related to the plurals: they can be understood as *more than one*, *universal* or *quasi-universal* (i.e. *almost every*). Since most of the problems in FraCaS favor the latter reading, we model **s** as a quasi-universal quantifier. We introduce the following lexical knowledge, $\mathbf{s} \leq \mathbf{a}$ and $\mathbf{s} \leq \mathbf{most}$, in the KB and allow the existential quantification rules (e.g., $\exists_{\mathbb{T}}$) to apply the plural terms $\mathbf{s}\,N$. With this treatment, for instance, the prover is able to prove the entailment in Fr-100.

We model the definite descriptions as generalized quantifiers of the form **the** N, where the rules make **the** act as the universal and existential quantifiers when marked with $\mathbb{T}$ and as the existential quantifier in case of $\mathbb{F}$. Put differently, ($\forall_{\mathbb{T}}$), ($\exists_{\mathbb{T}}$) and ($\exists_{\mathbb{F}}$) allow the quantifier in their antecedent nodes to match **the**.

$$\frac{g_{\mathtt{q}}\ N\ V : [\,] : \mathbb{T}}{N : [c_e] : \mathbb{F} \quad V : [c_e] : \mathbb{T}}\ \forall_{\mathbb{T}}$$
$$g \in \{\textbf{every}, \textbf{the}\} \text{ and } c_e \text{ is old}$$

$$\frac{g_{\mathtt{q}}\ N\ V : [\,] : \mathbb{F}}{N : [c_e] : \mathbb{F} \quad V : [c_e] : \mathbb{F}}\ \exists_{\mathbb{F}}$$
$$g \in \{\textbf{a}, \textbf{the}\} \text{ and } c_e \text{ is old}$$

$$\frac{g_{\mathtt{q}}\ N\ V : [\,] : \mathbb{T}}{\begin{array}{c} N : [c_e] : \mathbb{T} \\ V : [c_e] : \mathbb{T} \end{array}}\ \exists_{\mathbb{T}}$$
$$g \in \{\textbf{a}, \textbf{s}, \textbf{the}\} \text{ and } c_e \text{ is fresh}$$

This choice guarantees that, for example, *the demonstration* in the premises of Fr-99 co-refer and allow the proof for entailment. This approach also maintains the link if there are different surface forms co-referring, e.g., *the demonstration* and *the presentation*, in contrast to the approach in Abzianidze (2015a).

FrSec-2 also involves several problems with contrasting cardinal phrases like **exactly** n and m, where $n < m$ (see Fr-85). We account for these problems with the closure rule ($\times$EXCT), where the type $\mathtt{q}$, the predicate $\mathtt{greater}/2$ and the domain for E act as constraints.

$$\frac{\begin{array}{c} E_{\mathtt{q},\mathtt{q}}N_{\mathtt{q}} : [\vec{C}] : \mathbb{T} \\ M_{\mathtt{q}} : [\vec{C}] : \mathbb{T} \end{array}}{\times}\,\times\text{EXCT} \qquad \begin{array}{l} \text{such that} \\ E \in \{\textbf{just}, \textbf{exactly}\} \\ \text{and } \mathtt{greater}(M, N) \end{array}$$

FrSec-5 contains RTE problems pertaining to various types of adjective. First-order logic has problems with modeling *subsective* or *privative* adjectives (Kamp and Partee, 1995), but they are naturally modeled with higher-order terms. A subsective term, e.g., $\mathbf{small}_{\mathtt{n},\mathtt{n}}$, is a relation over a *comparison class* and an entity, e.g., $\mathbf{small}_{\mathtt{n},\mathtt{n}}\ \mathbf{animal}_{\mathtt{n}}\ c_e$ is of type t as $\mathtt{n}$ is a subtype of et according to the extended type theory (Abzianidze, 2015b). The rule ($\subseteq$) in Figure 2 accounts for the subsective property. With the help of it, the prover correctly identifies Fr-211 as contradiction (see Figure 5). In case of the standard first-order *intersective* analysis, the premises of Fr-211 would be translated as:

$$\mathtt{small(dumbo)} \land \mathtt{elephant(dumbo)} \land$$
$$\forall\mathtt{x}\big(\mathtt{elephant(x)} \to (\mathtt{large(x)} \land \mathtt{animal(x)})\big)$$

which is a contradiction given that $\mathtt{small}$ and $\mathtt{large}$ are contradictory predicates. Therefore, due to the *principle of explosion* everything, including the conclusion and its negation, would be entailed from the premises.

$$\begin{array}{c}
\boxed{1}\ \textbf{every elephant}\ (\lambda x.\,\mathbf{s}\ (\textbf{large animal})\ (\lambda y\,\mathbf{be}\ y\ x)) : [\,] : \mathbb{T} \\
\boxed{2}\ \mathbf{a}\ (\textbf{small elephant})\ (\lambda x.\,\mathbf{be}\ x\ \mathbf{dumbo}) : [\,] : \mathbb{T} \\
\boxed{3}\ \mathbf{a}\ (\textbf{small animal})\ (\lambda x.\,\mathbf{be}\ x\ \mathbf{dumbo}) : [\,] : \mathbb{T} \\
\lambda\text{BE}[3]\ \Big| \\
\boxed{4}\ \textbf{small animal} : [\mathbf{dumbo}] : \mathbb{T} \\
\lambda\text{BE}[2]\ \Big| \\
\boxed{5}\ \textbf{small elephant} : [\mathbf{dumbo}] : \mathbb{T} \\
\subseteq[5]\ \Big| \\
\boxed{6}\ \textbf{elephant} : [\mathbf{dumbo}] : \mathbb{T} \\
\forall^{n}_{\mathbb{T}}[1{,}6]\ \Big| \\
\boxed{7}\ \lambda x.\,\mathbf{s}\ (\textbf{large animal})\ (\lambda y.\,\mathbf{be}\ y\ x) : [\mathbf{dumbo}] : \mathbb{T} \\
\lambda{<}[7]\ \Big| \\
\boxed{8}\ \mathbf{s}\ (\textbf{large animal})\ (\lambda y.\,\mathbf{be}\ y\ \mathbf{dumbo}) : [\,] : \mathbb{T} \\
\lambda\text{BE}[8]\ \Big| \\
\boxed{9}\ \textbf{large animal} : [\mathbf{dumbo}] : \mathbb{T} \\
{>}[4{,}9]\ \Big| \\
\boxed{10}\ \mathbf{small} : [\mathbf{animal}, \mathbf{dumbo}] : \mathbb{T} \\
\boxed{11}\ \mathbf{large} : [\mathbf{animal}, \mathbf{dumbo}] : \mathbb{T} \\
\times|\,[10{,}11]\ \Big| \\
\boxed{12}\ \times
\end{array}$$

Figure 5: The closed tableau by LangPro proves Fr-211 as contradiction.

FrSec-9, about attitudes, is the last section we explore. Though the tableau system of (Muskens, 2010) employs intensional types, LangPro only uses extensional types due to simplicity of the system and the paucity of intensionality in RTE problems. Despite the fact, with the proof-theoretic approach and extensional types, we can still account for a certain type of reasoning on attitude verbs by modeling entailment properties of the verbs in the style of Nairn et al. (2006) and Karttunen (2012). For example, *know* has (+/+) property meaning that when it occurs in a positive embedding context, it entails its sentential complement with a positive polarity. Similarly, *manage to* is (+/+) and (-/-) because *John managed to run* entails *John run* and *John did not manage to run* entails *John did not run*. We accommodate the entailment properties in the tableau system in a straightforward way, e.g., terms with (+/+) property, like **know** and **manage**, are modeled via the rule (+/+) where $?p$ is an optional prepositional or particle term. The rest of the three entailment properties for attitude verbs are captured in the similar way.

$$\frac{h^{++}_{\alpha,\mathtt{vp}}(?p_{\alpha,\alpha}\ V_\alpha) : [d] : \mathbb{T}}{V_\alpha : [\vec{E}] : \mathbb{T}}\ {+/+}$$
$$\text{such that if } \alpha = \mathtt{vp}\text{, then } \vec{E} = d;$$
$$\text{otherwise } \alpha = \mathtt{s} \text{ and } \vec{E} \text{ is empty}$$

We also associate the entailment properties with the phrases *it is true that* and *it is false that* and model them via the corresponding tableau rules.

Our account for intensionality with the extensional types represents a syntactic approach rather than semantic. From the semantics perspective, the extensional types license John knowing all true statement if he knows at least one of them. But using the proof system, a syntactic machinery, we avoid such unwanted entailments with the absence of rules. In future, we could incorporate intensional types in LangPro if there is representative RTE data for the intensionality phenomenon.

The rest of the FraCaS sections were skipped during the adaptation phase for several reasons. FrSec-3 and FrSec-4 are about anaphora and ellipsis respectively. We omitted these sections as recently pronoun resolution is not modeled in the natural tableau and almost all sentences involving ellipsis are wrongly analyzed by the CCG parsers. In the current settings of the natural tableau, we treat auxiliaries as vacuous, due to this reason LangPro cannot properly account for the problems in FrSec-8 as most of them concern the aspect of verbs. FrSec-6 and FrSec-7 consists of problems with comparatives and temporal reference respectively. To account the latter phenomena, the LLFs of certain constructions needs to be specified further (e.g., for comparative phrases) and additional tableau rules must be introduced that model *calculations* on time and degrees.

5 Efficient theorem proving

Efficiency in theorem proving is crucial as we do not have infinite time to wait for provers to terminate and return an answer. Smaller tableau proofs are also easy for verifying and debugging. The section discusses the challenges for efficient theorem proving induced by the FraCaS problems and introduces new rules that bring efficiency to some extent.

The inventory of rules is a main component of a tableau method. Usually tableau rules are such inference rules that their consequent expressions are not larger than the antecedent expressions and are built up from sub-parts of the antecedent expressions. The natural tableau rules also satisfy these properties which contribute to the termination of tableau development. But there is still a big chance that a tableau does not terminate or gets unnecessarily large. The reasons for this is a combination of branching rules, δ-rules (introducing fresh entity terms), γ-rules (triggered for each entity term), and non-equivalent rules (the antecedents of which must be accessible by other rules too).[6] Efficeint theorem proving with LangPro becomes more challenging with multi-premised problems and monotonic GQs. More nodes in a tableau give rise to more choice points in rule applications and monotonic GQs are usually available for both monotonic and standard semantic rules.

To encourage short tableau proofs, we introduce eight *admissible* rules — the rules that are redundant from completeness point of view but represent *smart* shortcuts of several rule applications.[7] Half of the rules for the existential (e.g., *a* and *the*) and universal (e.g., *every*, *no* and *the*) quantifiers are γ-rules.[8] To make application of these rules more efficient, we introduce two admissible rules for each of the γ-rules. For instance, ($\forall^n_{\mathbb{T}}$) and ($\forall^v_{\mathbb{T}}$) are admissible rules which represent the efficient but incomplete versions of ($\forall_{\mathbb{T}}$):

$$\frac{\begin{array}{c} q\,N\,V : [\,] : \mathbb{T} \\ N : [c] : \mathbb{T} \end{array}}{V : [c] : \mathbb{T}}\,\forall^n_{\mathbb{T}} \qquad \frac{\begin{array}{c} q\,N\,V : [\,] : \mathbb{T} \\ V : [c] : \mathbb{F} \end{array}}{N : [c] : \mathbb{F}}\,\forall^v_{\mathbb{T}}$$
$$\text{where } q \in \{\textbf{every}, \textbf{the}\}$$

Their efficiency is due to choosing a relevant entity c_e, rather than any entity like ($\forall_{\mathbb{T}}$) does: ($\forall^n_{\mathbb{T}}$) chooses the entity that satisfies the noun term while ($\forall^v_{\mathbb{T}}$) picks the one not satisfying the verb term. Moreover, the admissible rules are not branching unlike their γ counterparts. Other four admissible rules account for *a* and *the* in a false context and *no* in a true context in the similar way.

The monotonicity rules, (MON↑) and (MON↓), are inefficient as they are branching δ-rules. On the other hand, the rules for GQs are also inefficient for being a γ or δ-rule. Both types of rules are often applicable to the same GQs, e.g., *every* and *a*, as most of GQs have monotonicity properties. Instead of triggering these two types of rules separately, we introduce two admissible rules, (∃FUN↑) and (∅FUN↓), which trigger them in tandem:

$$\frac{\begin{array}{c} g_{\mathsf{q}}\,N\,A : [\,] : \mathbb{T}\ \boxed{1} \\ g_{\mathsf{q}}\,N\,B : [\,] : \mathbb{F}\ \boxed{2} \end{array}}{\begin{array}{c} A : [c_e] : \mathbb{T}\ \boxed{3} \\ B : [c_e] : \mathbb{F}\ \boxed{4} \\ N : [c_e] : \mathbb{T}\ \boxed{5} \end{array}}\,\exists\text{FUN}\uparrow \qquad \frac{\begin{array}{c} h_{\mathsf{q}}\,N\,A : [\,] : \mathbb{F} \\ h_{\mathsf{q}}\,N\,B : [\,] : \mathbb{T} \end{array}}{\begin{array}{c} A : [c_e] : \mathbb{T} \\ B : [c_e] : \mathbb{F} \\ N : [c_e] : \mathbb{T} \end{array}}\,\emptyset\text{FUN}\downarrow$$
$$g \in \{\textbf{a}, \textbf{s}, \textbf{many}, \textbf{every}\} \qquad h \in \{\textbf{no}, \textbf{few}\}$$

[6] For instance, (MON↑) and (MON↓) in Figure 2 are both branching and δ. They are also non-equivalent since their consequents are semantically weaker than their antecedents; this requires that after their application, the antecedent nodes are still reusable for further rule applications. On the other hand, ($\forall_{\mathbb{T}}$) is non-equivalent and γ; for instance, for any entity term c_e, it is applicable to **every dog bark** : [ ] : $\mathbb{T}$ and asserts that either c is not **dog** or c does **bark**.

[7] In other words, if a closed tableau makes use of an admissible rule, the tableau can still be closed with a different rule application strategy that ignores the admissible rule.

[8] Remember from Section 4 that *the* is treated like the universal and existential quantifiers in certain cases.

ID	FraCaS entailment problem
64 unk	**P**: At most ten female commissioners spend time at home. **C**: At most ten commissioners spend time at home.
88 unk	**P**: Every representative and client was at the meeting. **C**: Every representative was at the meeting.
109 no	**P**: Just one accountant attended the meeting. **C**: Some accountants attended the meeting.
215 unk	**P1**: All legal authorities are law lecturers. **P2**: All law lecturers are legal authorities. **C**: All competent legal authorities are competent law lecturers.

Table 2: Problems with false proofs

For instance, if g = **every**, a single application of ($\exists$FUN$\uparrow$) already yields the fine-grained semantics: there is c_e that is A and N but not B. If the nodes were processed by the rules for **every**, ($\forall_\mathbb{F}$) would first entail $\boxed{4}$ and $\boxed{5}$ from $\boxed{2}$ and then ($\forall_\mathbb{T}$) or ($\forall^n_\mathbb{T}$) would introduce $\boxed{3}$ from $\boxed{1}$. ($\exists$FUN$\uparrow$) also represents a more specific version of the admissible rule (FUN$\uparrow$) of Abzianidze (2015a), which itself is an efficient and partial version of (MON$\uparrow$).

($\exists$FUN$\uparrow$) and ($\emptyset$FUN$\downarrow$) not only represent admissible rules but they also model semantics of *few* and *many* not captured by the monotonicity rules. For instance, if $\textbf{few dog bark} : [\,] : \mathbb{F}$ and $\textbf{few dog bite} : [\,] : \mathbb{T}$, then a set of entities that are **dog** and **bark**, denoted by $[\![\textbf{dog}]\!] \cap [\![\textbf{bark}]\!]$, is strictly larger than $[\![\textbf{dog}]\!] \cap [\![\textbf{bite}]\!]$ (despite the absolute or relative readings of *few*). Due to this set relation, there is an entity in $[\![\textbf{dog}]\!] \cap [\![\textbf{bark}]\!]$ and not in $[\![\textbf{bite}]\!]$. Therefore, we get the inference encoded in ($\emptyset$FUN$\downarrow$). Similarly, it can be shown that *many* satisfies the inference in ($\exists$FUN$\uparrow$).

6 Evaluation

After adapting the prover to the FraCaS sections for GQs, plurals, adjectives and attitudes, we evaluate it on the relevant sections and analyze the performance. Obtained results are compared to related RTE systems.

We run two version of the prover, ccLangPro and easyLangPro, that employ CCG derivations produced by C&C and EasyCCG respectively. In order to abstract from the parser errors to some extent, the answers from both provers are aggregated in LangPro: a proof is found iff one of the parser-specific provers finds a proof. The evaluation results of the three versions of LangPro on the relevant FraCaS sections are presented in Table 3 along with the confusion matrix for LangPro.

Meas%	ccLP	eLP	**LP**
Prec	**94**	93	**94**
Rec	73	71	**81**
Acc	80	79	**85**

Gold\LP	YES	NO	UNK
YES	**60**	0	14
NO	1	**14**	2
UNK	4	0	**47**

Table 3: Measures of ccLangPro (ccLP), easyLangPro (eLP) and LangPro (LP) on FraCaS sections 1, 2, 5, 9 and the confusion matrix for LP.

The results show that LangPro performs slightly better with C&C compared to EasyCCG. This is due to LLFgen which is mostly tuned on the C&C derivations. Despite this bias, easyLangPro proves 8 problems that were not proved by ccLangPro. In case of half of these problems, C&C failed to return derivations for some of the sentences while in another half of the problems the errors in C&C derivations were crucial, e.g., in the conclusion of Fr-44 *committee members* was not analyzed as a constituent. On the other hand, ccLangPro proves 10 problems unsolved by easyLangPro, e.g., Fr-6 was not proved because EasyCCG analyzes *really* as a modifier of *are* in the conclusion, or even more unfortunate, the morphological analyzer of EasyCCG cannot get the lemma of *clients* correctly in Fr-99 and as a result the prover cannot relate **clients** to **client**.

The precision of LangPro is high due to its sound inference rules. Fr-109 in Table 2 was the only case when entailment and contradiction were confused: plurals are not modeled as strictly more than one.[9] The false proves are mostly due to a lack of knowledge about adjectives. LangPro does not know a default comparison class for *clever*, e.g., *clever person→clever* but *clever politician↛clever*). Fr-215 was proved as entailment because we have not modeled intensionality of adjectives. Since EasyCCG was barely used during adaptation (except changing most of NP modifiers into noun modifiers), it analyzed *at most* in Fr-64 as a sentential modifier which was not modeled as downward monotone in the signature. Hence, by default, it was considered as upward monotone leading to the proof for entailment.

There are several reasons behind the problems that were not proved by the prover. Several problems for adjectives were not proved as they con-

[9] Moreover, Fr-109 is identical to Fr-107 which has *yes* as a gold answer. Another inconsistency in gold answers of Fr-87 and Fr-88 (due to the ambiguous premise) is a reason for a false proof. While Fr-87 was correctly proved by the prover, obviously Fr-88 was misclassified automatically.

Sec (Sing/All)	Single-premised (Acc %)							Multi-premised (Acc %)					Overall (Acc %)				
	BL	NL07,08	LS P/G	NLI	T14a,b	M15	**LP**	BL	LS P/G	T14a,b	M15	**LP**	BL	LS P/G	T14a,b	M15	**LP**
1 GQs (44/74)	45	84 **98**	70 89	95	80 93	82	93	57	50 80	80 **97**	73	93	50	62 85	80 **95**	78	93
2 Plur (24/33)	58	~~42~~ **75**	-	~~38~~	-	67	**75**	67	-	-	67	67	61	-	-	67	**73**
5 Adj (15/22)	40	60 80	-	**87**	-	**87**	**87**	43	-	-	29	43	41	-	-	68	**73**
9 Att (9/13)	67	~~56~~ 89	-	~~22~~	-	78	**100**	50	-	-	**75**	**75**	62	-	-	77	**92**
1,2,5,9 (92/142)	50	- **88**	-	-	-	78	**88**	56	-	-	66	**80**	52	-	-	74	**85**

Table 4: Comparison of RTE systems tested on FraCaS: NL07 (MacCartney and Manning, 2007), NL08 (MacCartney and Manning, 2008), LS (Lewis and Steedman, 2013) with Parser and Gold syntax, NLI (Angeli and Manning, 2014), T14a (Tian et al., 2014), T14b (Dong et al., 2014) and M15 (Mineshima et al., 2015). BL is a majority (*yes*) baseline. Results for non-applicable sections are strikeout.

tained comparative constructions, not covered by the rules. Some problems assume the universal reading of plurals. A couple of problems involving *at most* were not solved as the parsers often analyze the phrase in a wrong way.[10]

We also check the FraCaS sections how representative they are for higher-order GQs (HOGQs). After replacing all occurrences of **most**, **several**, **many**, **s** and **the** with the indefinite **a** in LLFs, LangPro$^{-\text{HOGQ}}$ (without the HOGQs) achieves an overall accuracy of 81% over FrSec-1,2,5,9. Compared to LangPro only 6 problems, including Fr-56, 99, were misclassified while Fr-26, 100 were solved. This shows that the dataset is not representative enough for HOGQs.

In Table 4, the current results are compared to the RTE systems that have been tested on the single or multi-premised FraCaS problems.[11] According to the table, the current work shows that the natural tableau system and LangPro are successful in deep reasoning over multiple premises.

The natural logic approach in MacCartney and Manning (2008) and Angeli and Manning (2014) models monotonicity reasoning with the exclusion relation in terms of the string edit operations over phrases. Since the approach heavily hinges on a sequence of edits that relates a premise to a conclusion, it cannot process multi-premised problems properly. Lewis and Steedman (2013) and Mineshima et al. (2015) both base on first-order logic representations. While Lewis and Steedman (2013) employs distributional relation clustering to model the semantics of content words, Mineshima et al. (2015) extends first-order logic with several higher-order terms (e.g., for *most, believe, manage*) and augments first-order inference of Coq with additional inference rules for the higher-order terms. Tian et al. (2014) and Dong et al. (2014) build an inference engine that reasons over abstract denotations, formulas of relational algebra or a sort of description logic, obtained from Dependency-based Compositional Semantic trees (Liang et al., 2011). Our system and approach differ from the above mentioned ones in its unique combination of expressiveness of high-order logic, *naturalness* of logical forms (making them easily obtainable) and flexibility of a semantic tableau method. All these allow to model surface and deep semantic reasoning successfully in a single system.

7 Future work

We have modeled several semantic phenomena in the natural tableau theorem prover and obtained high results on the relevant FraCaS sections. Concerning the FraCaS dataset, in future work we plan to account for the comparatives and temporal reference in the natural tableau. After showing that the natural tableau can successfully model deep reasoning (e.g., the FraCaS problems) and (relatively) wide-coverage and surface reasoning (e.g., the SICK dataset), we see the RTE datasets, like RTE-1 (Dagan et al., 2005) and SNLI (Bowman et al., 2015), involving texts obtained from newswire or crowd-scouring as a next step for developing the theory and the theorem prover.

Acknowledgments

The author thanks the anonymous reviewers for their valuable comments and feedback. The research is a part of the project “Towards Logics that Model Natural Reasoning” supported by the NWO grant (project number 360-80-050).

[10] Tableau proofs of the FraCaS problems are available at: `http://lanthanum.uvt.nl/langpro/fracas`

[11] Since the FraCaS data is small and usually the problems are seen during the system development, the comparison should be understood in terms of an expressive power of a system and the underlying theory.

References

Lasha Abzianidze. 2015a. A tableau prover for natural logic and language. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2492–2502, Lisbon, Portugal, September. Association for Computational Linguistics.

Lasha Abzianidze. 2015b. Towards a wide-coverage tableau method for natural logic. In Tsuyoshi Murata, Koji Mineshima, and Daisuke Bekki, editors, *New Frontiers in Artificial Intelligence: JSAI-isAI 2014 Workshops, LENLS, JURISIN, and GABA, Kanagawa, Japan, October 27-28, 2014, Revised Selected Papers*, pages 66–82. Springer Berlin Heidelberg, Berlin, Heidelberg, June.

Gabor Angeli and Christopher D. Manning. 2014. Naturalli: Natural logic inference for common sense reasoning. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Jon Barwise and Robin Cooper. 1981. Generalized quantifiers and natural language. *Linguistics and Philosophy*, 4(2):159–219.

Evert W. Beth. 1955. Semantic Entailment and Formal Derivability. *Koninklijke Nederlandse Akademie van Wentenschappen, Proceedings of the Section of Sciences*, 18:309–342.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal, September. Association for Computational Linguistics.

Alonzo Church. 1940. A formulation of the simple theory of types. *Jurnal of Symbolic Logic*, 5(2):56–68, June.

Stephen Clark and James R. Curran. 2007. Wide-coverage efficient statistical parsing with ccg and log-linear models. *Computational Linguistics*, 33.

Robin Cooper, Dick Crouch, Jan Van Eijck, Chris Fox, Josef Van Genabith, Jan Jaspars, Hans Kamp, David Milward, Manfred Pinkal, Massimo Poesio, Steve Pulman, Ted Briscoe, Holger Maier, and Karsten Konrad. 1996. *FraCaS: A Framework for Computational Semantics*. Deliverable D16.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The pascal recognising textual entailment challenge. In *Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment*.

Ido Dagan, Dan Roth, Mark Sammons, and Fabio Massimo Zanzotto. 2013. *Recognizing Textual Entailment: Models and Applications*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers.

Marcello D'Agostino, Dov M. Gabbay, Reiner Hhnle, and Joachim Posegga, editors. 1999. *Handbook of Tableau Methods*. Springer.

Yubing Dong, Ran Tian, and Yusuke Miyao. 2014. Encoding generalized quantifiers in dependency-based compositional semantics. In *Proceedings of the 28th Pacific Asia Conference on Language, Information, and Computation*, pages 585–594, Phuket,Thailand, December. Department of Linguistics, Chulalongkorn University.

Jörg Endrullis and Lawrence S. Moss. 2015. Syllogistic logic with "most". In Valeria de Paiva, Ruy de Queiroz, S. Lawrence Moss, Daniel Leivant, and G. Anjolina de Oliveira, editors, *Logic, Language, Information, and Computation: 22nd International Workshop, WoLLIC 2015, Bloomington, IN, USA, July 20-23, 2015, Proceedings*, pages 124–139. Springer Berlin Heidelberg, Berlin, Heidelberg.

Daniel Gallin. 1975. *Intensional and Higher-Order Modal Logic: With Applications to Montague Semantics*. American Elsevier Pub. Co.

Jaakko Hintikka. 1955. *Two Papers on Symbolic Logic: Form and Content in Quantification Theory and Reductions in the Theory of Types*. Number 8 in Acta philosophica Fennica. Societas Philosophica.

Matthew Honnibal, James R. Curran, and Johan Bos. 2010. Rebanking ccgbank for improved np interpretation. In *Proceedings of the 48th Meeting of the Association for Computational Linguistics (ACL 2010)*, pages 207–215, Uppsala, Sweden.

Thomas F. Icard and Lawrence S. Moss. 2014. Recent progress on monotonicity. *Linguistic Issues in Language Technology*, 9.

Hans Kamp and Barbara Partee. 1995. Prototype theory and compositionality. *Cognition*, 57(2):129–191.

Lauri Karttunen. 2012. Simple and phrasal implicatives. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 124–131, Montréal, Canada, 7-8 June. Association for Computational Linguistics.

George Lakoff. 1970. Linguistics and natural logic. In Donald Davidson and Gilbert Harman, editors, *Semantics of Natural Language*, volume 40 of *Synthese Library*, pages 545–665. Springer Netherlands.

Mike Lewis and Mark Steedman. 2013. Combined distributional and logical semantics. *Transactions of the Association for Computational Linguistics (TACL)*, 1:179–192.

Mike Lewis and Mark Steedman. 2014. A* CCG parsing with a supertag-factored model. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 990–1000, Doha, Qatar, October. Association for Computational Linguistics.

P. Liang, M. I. Jordan, and D. Klein. 2011. Learning dependency-based compositional semantics. In *Association for Computational Linguistics (ACL)*, pages 590–599.

Bill MacCartney and Christopher D. Manning. 2007. Natural logic for textual inference. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, RTE '07, pages 193–200, Stroudsburg, PA, USA. Association for Computational Linguistics.

Bill MacCartney and Christopher D. Manning. 2008. Modeling semantic containment and exclusion in natural language inference. In Donia Scott and Hans Uszkoreit, editors, *COLING*, pages 521–528.

Bill MacCartney. 2009. *Natural language inference*. Phd thesis, Stanford University.

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 Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland. European Language Resources Association (ELRA).

Koji Mineshima, Pascual Martínez-Gómez, Yusuke Miyao, and Daisuke Bekki. 2015. Higher-order logical inference with compositional semantics. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2055–2061, Lisbon, Portugal, September. Association for Computational Linguistics.

R. Montague. 1970. English as a formal language. In Bruno et al. (eds.) In Visentini, editor, *Linguaggi nella società e nella tecnica. Milan: Edizioni di Comunità.*, pages 188–221.

Richard Montague. 1973. The proper treatment of quantification in ordinary English. In K. J. J. Hintikka, J. Moravcsic, and P. Suppes, editors, *Approaches to Natural Language*, pages 221–242. Reidel, Dordrecht.

Reinhard Muskens. 2010. An analytic tableau system for natural logic. In Maria Aloni, Harald Bastiaanse, Tikitu de Jager, and Katrin Schulz, editors, *Logic, Language and Meaning*, volume 6042 of *Lecture Notes in Computer Science*, pages 104–113. Springer Berlin Heidelberg.

Rowan Nairn, Cleo Condoravdi, and Lauri Karttunen. 2006. Computing relative polarity for textual inference. In *Proceedings of the Fifth International Workshop on Inference in Computational Semantics (ICoS-5)*.

Víctor Sánchez-Valencia. 1991. Categorial grammar and natural reasoning. ILTI Publication Series for Logic, Semantics, and Philosophy of Language LP-91-08, University of Amsterdam.

Raymond M. Smullyan. 1968. *First-order Logic*. Springer-Verlag.

Mark Steedman. 2000. *The Syntactic Process*. MIT Press, Cambridge, MA, USA.

Ran Tian, Yusuke Miyao, and Takuya Matsuzaki. 2014. Logical inference on dependency-based compositional semantics. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 79–89, Baltimore, Maryland, June. Association for Computational Linguistics.

Johan Van Benthem. 1986. *Essays in Logical Semantics*, volume 29 of *Studies in Linguistics and Philosophy*. Springer Netherlands.