Semantic parsing and reasoning in LFG: The case of gradable adjectives

Mark-Matthias Zymla
University of Konstanz

Farai Grenzdörffer
University of Konstanz

Kascha Kruschwitz
University of Konstanz

Paul Zodl
University of Konstanz

Proceedings of the LFG'25 Conference

Miriam Butt, Jamie Y. Findlay and Ida Toivonen (Editors)

2025

PubliKon

lfg-proceedings.org

Abstract

In this paper, we present an extension to XLE+Glue that puts the mapping from form to meaning under scrutiny by incorporating a reasoning component based on the Vampire theorem prover. We illustrate various aspects of this system by means of an analysis of gradable adjectives, particularly their positive and comparative uses. Crucially, we show how our system deals with compositional and non-compositional semantic information. As part of this effort, we also present a newly developed XLE+Glue grammar that combines co-descriptive semantics with an additional layer of description-by-analysis. We propose that this approach allows us to better handle certain kinds of context sensitivity and align compositional and non-compositional aspects of meaning. Furthermore, we show that such a layered approach can be extended to the reasoning process.

1 Introduction

This paper presents an analysis of gradable adjectives that is supported by a computational implementation spanning from syntax to semantic interpretation. Thus, its contributions are two-fold: i) it provides an implementation of gradable adjectives at the syntax/semantics interface based on a hybrid approach to Glue semantics, combining local co-description and description-by-analysis, and ii) it highlights the validity of this approach by integrating a reasoning engine into XLE+Glue (Dalrymple et al. 2020; Zymla et al. 2025) and showing that the analysis leads to the correct prediction of various inferences underlying gradable adjectives and comparatives.

Gradable adjectives are an interesting test case at the intersection of compositional and contextual meaning. Following the spirit of Zymla (2017, 2024b), we elaborate on a layered approach to interpreting gradable adjectives, following the assumption that reasoning itself is layered and can be modeled correspondingly as a system that grows in complexity based on the requirements of the context. Thus, for example, (1) can be interpreted correctly without giving the POSITIVE use of *great* an articulated semantics.

(1) Some great tenors are Swedish.
———————————————
Are there great tenors who are Swedish? → YES

However, in example (2), it is crucial to understand the relation between the POSITIVE uses of *small* and *large*. This relation is context-dependent, as the meaning of these adjectives is affected by the intermediate linguistic context, but also requires further contextual information. Concretely, *small* is (essentially) interpreted as *small for an animal* and *large* is interpreted as *large for a mouse* (Kennedy & McNally 2005). Additionally, we need to ascertain that the antonym relation holds between *small* and *large* without the fact that Mickey is a large mouse obscuring the matter.

(2) All mice are small animals.
Mickey is a large mouse.
———————————————
Is Mickey a large animal? → NO

Our analysis relies on a layered compositional approach to gradable adjectives consisting of a co-descriptive layer and a description-by-analysis layer, and a method to automatically trigger additional axioms that do not compositionally interact with the meaning of a sentence, allowing us to capture both the linguistic and contextual aspects of gradable adjectives.

Another example that fails when gradable adjectives are interpreted as properties of entities is given in (3), illustrating a COMPARATIVE use of *fast*. Comparatives serve as a syntactically and semantically complex test case for the analysis in the present paper.

(3) The PC-6082 is faster than every ITEL computer.
The ITEL-ZX is an ITEL computer.
———
Is the PC-6082 faster than the ITEL-ZX? → YES

Figure 1 describes the overall architecture of our system. We use XLE+Glue, an interface between the XLE (Crouch et al. 2017) and the Glue Semantics Workbench (GSWB; Meßmer & Zymla 2018), to provide a co-descriptive analysis capturing core aspects of syntax and semantics (Dalrymple et al. 2020). To deal with the contextual nature of adjectives, the semantics is enriched by LiGER (Linguistic Graph Expansion and Rewriting; Zymla et al. 2025), which is also used to infer non-compositional axioms. Our semantics are rendered in DRT based on Blackburn & Bos (2005), referred to here as BB DRT. The DRT representations are translated to first-order logic (DRT2FOL) and interpreted by the Vampire theorem prover (Kovács & Voronkov 2013).

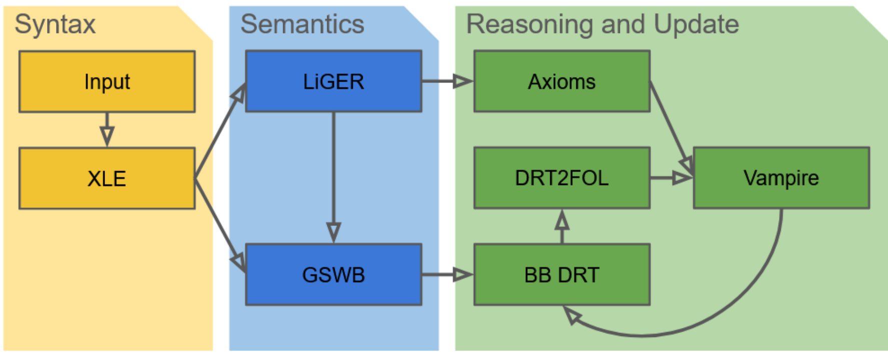


Figure 1: XLE+Glue pipeline

The paper is structured as follows: In the next section, we begin by describing the syntax and semantics of adjectives commonly assumed in (computational) LFG, as a baseline for our approach. In section 3, we discuss the semantics of gradable adjectives. Then, in section 4, we couch the semantics in a syntax/semantics interface based on LFG's Glue semantics. Further, we discuss the computational implementation of the analysis in XLE+Glue. Section 5 describes the inference system used for interpreting the semantics in the context of natural language inference. Finally, section 6 concludes.

2 Adjectives in Lexical Functional Grammar

We assume a fairly standard recursive treatment of adjectival modifiers in the phrase structure of noun phrases. Thus, we posit the c-structure rules in (4), following, e.g., Andrews (1983); Findlay & Haug (2022). However, unlike Andrews (2018), we maintain a flat f-structure.

(4) *Rules:*

$$\begin{array}{llcc} \text{NP} & \rightarrow & \text{(D)} & \text{N}' \\ & & \uparrow = \downarrow & \uparrow = \downarrow \end{array}$$

$$\begin{array}{llccccccc} \text{N}' & \rightarrow & \text{AP} & \{ & \text{N}' & \mid & \text{N} & \} \\ & & \downarrow \in (\uparrow \textsc{adj}) & & \uparrow = \downarrow & & \uparrow = \downarrow & \end{array}$$

As is traditionally done, we distinguish between attributive uses of adjectives, captured by the rules above, and predicative uses. In comparison to attributive adjectives, predicative uses of adjectives first combine with a copula in English and then subsequently modify the subject (see (7)).

In (5), we illustrate lexical entries for attributive and predicative uses of adjectives.[1] The main difference here is that, in the predicative use, the adjective explicitly selects for a SUBJ, enabling the specification of its meaning as a simple property of that GF. In contrast, attributive uses *modify* their dominating f-structure, resulting in a more complex type (see Dalrymple 2001: Ch. 10 for discussion). This also captures the intuition that while attributive adjectives can be recursively added to the set of modifiers, this is not the case for predicative uses (except via adjectival coordination).

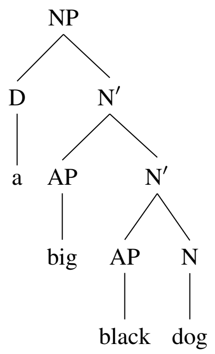
(5) *Lexicon:*

$$\begin{array}{lll} \text{big} & \text{A} & (\uparrow \textsc{pred}) = \text{`big'} \\ & & \lambda P.big(P) : ((\textsc{adj} \uparrow)_e \multimap (\textsc{adj} \uparrow)_t) \multimap \\ & & \qquad (\textsc{adj} \uparrow)_e \multimap (\textsc{adj} \uparrow)_t \end{array}$$

$$\begin{array}{lll} \text{tall} & \text{A} & (\uparrow \textsc{pred}) = \text{`tall} < (\uparrow \textsc{subj}) > \text{'} \\ & & \lambda x.tall(x) : (\uparrow \textsc{subj})_e \multimap \uparrow_t \end{array}$$

As a result of this analysis, the scope of attributive adjectives is not determined by their linear order. Putting all of this together, we get the representation for a complex NP like *a big black dog* in (6) and a predicative structure like *Jordan is tall* in (7).

(6) **a big black dog**



$$\mathbf{1}\begin{bmatrix} \text{PRED `dog'} \\ \text{ADJ} \left\{ \begin{bmatrix}\text{PRED `big'}\end{bmatrix}, \begin{bmatrix}\text{PRED `black'}\end{bmatrix} \right\} \end{bmatrix}$$

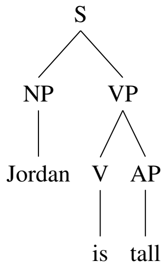
$\lambda P.big(P)$:
$(\mathbf{1}_e \multimap \mathbf{1}_t) \multimap (\mathbf{1}_e \multimap \mathbf{1}_t)$

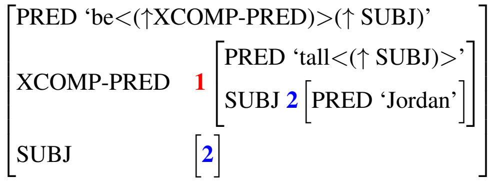
$\lambda P.black(P)$:
$(\mathbf{1}_e \multimap \mathbf{1}_t) \multimap (\mathbf{1}_e \multimap \mathbf{1}_t)$

$dog : (\mathbf{1}_e \multimap \mathbf{1}_t)$

[1] We generally use subscripts to indicate types. However, we assume first-order Glue (Kokkonidis 2008) as the underlying logic (i.e., types take indices as arguments). The present notation is a notational variant.

(7) **Jordan is tall.**





$\lambda x.tall(x) : (\mathbf{2}_e \multimap \mathbf{1}_t)$

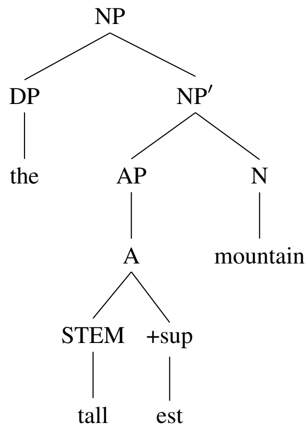
$Jordan : \mathbf{2}_e$

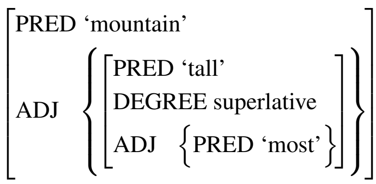
Gradable adjectives can be used to express POSITIVE, COMPARATIVE, and SUPERLATIVE meanings. In English, these meanings are reflected in the morphology:

(8) a. The dog is **big**.
b. The cat is **faster** than the dog.
c. Jordan climbed the **tallest** mountain.

We follow the ParGram approach (i.e., a computational LFG approach) to analyzing morphology (Butt et al. 1999: Ch. 12). This approach is realizational (Beesley & Karttunen 2003), such that a single morpheme can realize multiple grammatical and/or semantic distinctions. Words are decomposed via lexical rules. These are rules that can parse regular languages (i.e., those covered by finite-state automata). This is illustrated in (9), where the adjective *tallest* is decomposed into a stem *tall* and a superlative morpheme *-est* which expresses the DEGREE feature SUPERLATIVE. Additionally, it also realizes a PRED feature indicating its semantic import (i.e., a covert *most*):

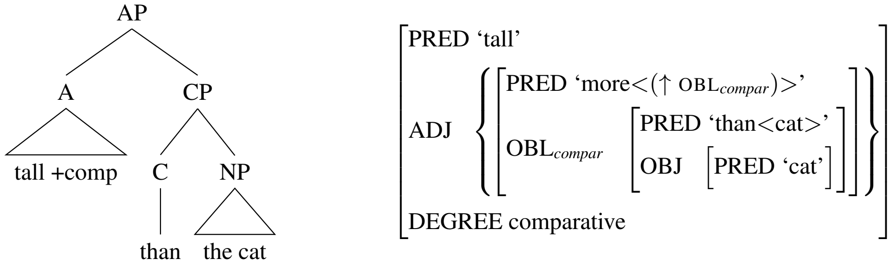
(9) **the tallest mountain**





The comparative morpheme also realizes such a PRED feature. In the case of the comparative, it is not *most* but *more*. This proposal is broadly built on Bresnan (1973). These additional PRED features play an integral role in handling the possibility of further modification with a *than*- or *as*-clause. This modifying clause is encoded via the ability of these forms to take an optional OBL$_{compar}$ at f-structure:

(10) The dog is **taller than the cat**.



3 Gradable adjectives

So far, we have simply treated adjectives as properties. This corresponds to a plain translation of the adjective *tall* to a symbol in our non-logical vocabulary. When we look more closely at the kinds of meanings epxressed by certain adjectives, namely gradable ones, we find that their meaning is scalar in nature. Adjectives like *tall* or *heavy* indicate some scale, e.g. height and weight, on which they are perceived. This scale is measured in terms of degrees, e.g. degrees of height or degrees of weight. Consequently, we have to extend our ontology of types to also incorporate degrees. Corresponding systems are broadly covered by the term DEGREE SEMANTICS (see e.g. Cresswell 1976; Kennedy & McNally 2005; Schwarzschild 2008).

Here we assume that gradable adjectives express a relation between an individual and a scale. By uttering (11) we locate Jordan on a scale of height and we identify this degree as being 175cm:

(11) a. Jordan is 175cm tall.
b. $tall(j, \delta) \wedge \delta = 175cm$

Besides expressing degrees overtly, we can also simply imply where a certain individual lies on a scale. To model this, a comparison class is employed. We use θ to indicate the use of contextually provided comparison classes. For (12) to be felicitous, Jordan needs to exceed some contextually supplied standard, here, the one for the height of humans, as indicated by the argument of the comparison class θ. In the case of predicative adjectives, the subscript is inferred via the context. In the case of an attributive use, comparison classes are (at least partially) determined compositionally (e.g., in (12c), Jordan might be tall with respect to a contextually salient subset of teachers).

(12) a. Jordan is tall (for a human).
b. $tall(j, \delta) \wedge \theta_{tall}(human) < \delta$
c. Jordan is a tall teacher.

Additionally, the interpretation of gradable adjectives is modulated by certain axioms. Firstly, we assume that degrees of some property P are monotonic:

(13) $\forall P[\forall x[\forall \delta[P(\delta)(x) \rightarrow \forall \delta'[\delta' < \delta \rightarrow P(\delta')(x)]]]]$ *monotonicity*

This captures the intuition that if an individual reaches a certain height, e.g. that they are 175cm tall, they are also tall to every smaller degree. Secondly, we assume that degrees have an upper bound. Thus, there is a unique degree that identifies the maximal point on the scale of the corresponding property.

(14) $\forall P[\forall x[\exists \delta[P(\delta)(x) \rightarrow \forall \delta'[\delta' > \delta \rightarrow \neg P(\delta')(x)]]]]$ *upper bound*

For now, we simply assume that these axioms are contextually provided, but we will clarify their contribution to reasoning in section 5.

3.1 Comparatives

Traditionally, comparatives are treated as comparing maximal degrees between the modified noun and the *than*-complement, i.e., an explicit comparison class (Cresswell 1976; Kennedy & McNally 2005). This is illustrated in the following example:[2]

(15) The elephant is taller than the mouse.

a. $\exists \delta, \delta', x, y[elephant(x) \wedge max_{height}(x, \delta) \wedge$
$mouse(y) \wedge max_{height}(y, \delta') \wedge \delta > \delta']$

A different implementation was proposed by Klein (1980). Here, we do not need to compute maximal values.[3] The intuition behind this proposal is that we are looking for a degree that is held by the degree holder but not by the object of comparison. This analysis is also referred to as the P-not-P analysis, since we are looking for a threshold that the degree holder exceeds, which the object of comparison does not (Schwarzschild 2008). For the sentence in (15), we arrive at the following representation:

(16) $\exists \delta, x, y[elephant(x) \wedge height(x, \delta) \wedge mouse(y) \wedge \neg height(y, \delta)]$

We can treat equative comparatives in a similar way. Here, we are employing universal quantification, such that for every degree that the degree holder holds, the object of comparison holds it, too.[4]

(17) The elephant is as tall as the house.

a. $\exists x, y[\forall \delta[house(x) \wedge height(x, \delta) \rightarrow elephant(y) \wedge height(y, \delta)]]$

Similar representations have been used for computational purposes by Haruta et al. (2022). Our decision to maintain degrees in our ontology is also inspired by their work. To reconcile Klein's original analysis with a semantics of degrees, we adopt Haruta et al.'s (2022) consistency postulate (CP). This axiom states that in a P-not-P configuration, the degree holder holds any degree of the underlying adjective that the object of comparison holds. Thus, this axiom subsumes monotonicity and the upper bound when a comparative construction is involved.[5]

[2]We abstract away from the semantics of the definite for sake of simplicity. We assume a presuppositional analysis following Van der Sandt (1992), which accommodates existence out of the blue.

[3]Note that in a Klein-style analysis (see also Klein 1982) we do not strictly need degrees in our ontology. Rather, gradable adjectives are partial functions that map individuals to the adjective's positive or negative extension. To keep with the more recent literature, we still employ degrees (see, e.g. Schwarzschild 2008). This allows for a unified treatment of positive and comparative uses of adjectives.

[4]As a reviewer points out, this means the elephant is *at least* as tall as the house. A more strict semantics might require the use of max_{height} similar to (15a).

[5]Consequently, it is only relevant when a comparative construction is used. Thus, we require a method

(18) (CP) $\forall x \forall y[\exists\delta[A(x,\delta) \wedge \neg A(y,\delta)] \rightarrow \forall\delta[A(y,\delta) \rightarrow A(x,\delta)]]$,
where A is an arbitrary gradable adjective.
(Haruta et al. 2022: p. 148)

Note that our semantics for the comparative does not employ the contextual standard familiar from the positive. This correctly captures the fact that in examples like (19), the degree holder does not necessarily have to fulfill this standard:

(19) The tick is heavier than the amoeba.

3.2 Clausal and phrasal comparatives

In the literature, two types of comparatives are distinguished:

(20) a. The mouse is smaller than [$_{\text{DP}}$ the dog ~~is small~~]
b. The mouse is smaller than [$_{\text{IP}}$ the dog is ~~small~~]

The sentence in (20a) is referred to as a phrasal comparative while (20b) is referred to as a clausal comparative. We do not assume ellipsis in the syntax for comparatives, but rather resolve it via a semantic treatment of ellipsis (see section 4.5). Furthermore, in contrast to, e.g., Bhatt & Takahashi (2011), we only assume one kind of semantics for the comparative, namely a 2-place operator. Consequently, both sentences in (20) receive the semantics in (21) in our analysis.

(21) $\exists x, y, \delta[mouse(x) \wedge dog(y) \wedge small(x,\delta) \wedge \neg small(y,\delta)]$

In both cases, the elided adjective is thus only represented in the semantics but not in either c- or f-structure. As explained in more detail in the next section, we achieve this kind of analysis by employing description-by-analysis. We show that the analysis is computationally convenient, but we also acknowledge that the situation is ultimately more complicated and needs to be explored in more detail in future research.

4 Deriving the compositional semantics via XLE+Glue

In this section, we couch the semantics presented above in a syntax/semantics interface based on LFG's Glue semantics (Dalrymple 1999). Our analysis is implemented in XLE+Glue (Dalrymple et al. 2020) and make use of recent developments allowing for the specification of hybrid grammars combining co-descriptive semantics with description-by-analysis (DBA; Zymla et al. 2025). Thus, we present a computational grammar accompanied by a set of rewrite rules for extracting compositional semantic information as well as axiomatic knowledge required for reasoning about degrees. Before describing the technical details, we first briefly introduce the FraCaS testsuite, the source of our linguistic data.

---

for modulating the use of the correct and necessary axioms in different linguistic contexts. This is explained in section 5.

4.1 The FraCaS testsuite

Our grammar is developed on the basis of the FraCas testsuite (Cooper et al. 1996), a test set for natural language inference (NLI), focusing on compositional and functional semantic aspects (e.g., tense/aspect, quantifiers). It is designed to require little lexical knowledge, allowing us to work almost entirely with our grammar, without requiring external semantic resources. Each example in the testsuite consists of one or more premise sentences, a conclusion sentence and their entailment status. The conclusion can either be an entailment, a contradiction or neutral given the premise(s).

In (22) the premise set consists of two sentences. Given this premise set, there are three conclusions in (22) illustrating the possible entailment statuses: *entailment*, *contradiction*, and *neutral*. (22a) is an entailment, as the conclusion is necessarily true given the premises. (22b) is thus a contradiction, as the ITEL-XZ being faster than the PC-6082 can never be true if the inverse in (22a) is true. Lastly, because comparative sentences do not use a contextual standard, the premises, more specifically that the PC-6082 is faster than every ITEL computer, do not necessitate that the PC-6082 is fast in general. Thus, (22c) is neither an entailment nor a contradiction and thus *neutral*.

(22) The PC-6082 is faster than every ITEL computer
The ITEL-XZ is an ITEL computer

---

a. The PC-6082 is faster than the ITEL-XZ $\rightarrow$ Entailment
b. The ITEL-XZ is faster than the PC-6082 $\rightarrow$ Contradiction
c. The PC-6082 is fast. $\rightarrow$ Neutral

The FraCaS suite also minimizes effects of discourse relations. Thus, we parse sentences individually providing a corresponding testsuite. From this testsuite, we build NLI items and process them with the reasoning component. (see section 5).

4.2 The XLE grammar

Syntactically, our grammar aligns its analyses with the English ParGram grammar (Butt et al. 2002) and the specifications in Butt et al. (1999), illustrated in section 2. We decided to re-implement the relevant parts to provide a self-contained semantic grammar that can serve as reference for semantic parsing in XLE.[6] We particularly focus on parsing attributive and predicative adjectives, as well as the related comparative and equative constructions, but the grammar also generally covers a range of syntactic constructions like PPs and relative clauses on a basic level.

Semantically, the grammar provides a core for semantic parsing that captures predicate/argument structure, as well as some other highly compositional aspects, mainly scope taking operators. The core fragment is constrained to a fundamental ontology consisting of entities, events, and truth values. Thus, for this implementation, we have decided to mediate ontological extensions via LiGER (see section 4.5). This design decision allows us to implement a core semantics that is largely operational on its own but is extensible via description-by-analysis. Thus, we essentially pursue a layered approach to semantic parsing (Zymla 2017, 2024b).[7]

[6] Grammar and code are available at: https://github.com/Mmaz1988/xleplusglue/tree/2024_inference
[7] We further motivate this in section 5.

4.3 Co-descriptive semantics

The co-descriptive part of the grammar assumes a standard Neo-Davidsonian event semantics (Parsons 1990; Asudeh & Giorgolo 2012), such that verbs introduce an event predicate and thematic arguments can be added as modifiers of the predicate. We render our semantics in λ-DRT. More generally, we use a semantics close to that used in the Boxer system (Bos 2008).[8] Accordingly, all the DRSs we specify can be translated to first-order logic without loss of information.[9]

For example, in the lexical entry of a verb in our grammar, two meaning constructors (MCs) are specified: the base verbal semantics (23a) (an event predicate) and the existential closure in (23b). Arguments of a verb are treated as modifiers, as shown in (23c), and can be configured according to the logic presented in Asudeh & Giorgolo (2012). We enumerate arguments starting from 1 assuming some form of argument hierarchy, so these argument roles can serve as interface between GFs and thematic roles given an appropriate framework (see Findlay 2020 for discussion).[10]

(23) **Components of the event description:**

a. $\lambda e_v.\begin{array}{|c|}\hline \\ \hline \text{appear}(e) \\ \hline \end{array} : (\uparrow_\sigma \text{EV})_v \multimap (\uparrow_\sigma \text{EV})_t$

b. $\lambda P_{vt}.\begin{array}{|c|}\hline e \\ \hline \\ \hline \end{array} \oplus P(e) : ((\uparrow_\sigma \text{EV})_v \multimap (\uparrow_\sigma \text{EV})_t) \multimap \uparrow_{\sigma_t}$

c. $\lambda P_{vt}.\lambda x_e.\lambda e_v.P(v) \oplus \begin{array}{|c|}\hline \\ \hline \text{ARG1}(e) = x \\ \hline \end{array} :$

$(\uparrow_\sigma \text{EV})_v \multimap (\uparrow_\sigma \text{EV})_t \multimap \%arg1_e \multimap (\uparrow_\sigma \text{EV})_v \multimap (\uparrow_\sigma \text{EV})_t,$
where $\%arg1 = (\uparrow \text{SUBJ})_\sigma$ (determined by linking)

(24) **Components in FOL with instantiated MCs:**

a. $\lambda e_v.appear(e) : \quad e_v \multimap e_t$

b. $\lambda P_{vt}.\exists e[P(e)] : \quad (e_v \multimap e_t) \multimap f_t$

c. $\lambda P_{vt}.\lambda x_e.\lambda e_v.P(e) \wedge \text{ARG1}(e) = x : \quad (e_v \multimap e_t) \multimap g_e \multimap (e_v \multimap e_t)$

Our grammar builds event descriptions around a dedicated EV-index in the s-structure: $(\uparrow_\sigma \text{EV})$ (see, e.g, Haug 2008). This allows for a simple treatment of event closure, which must be obligatorily narrow. In the present example, we build the event description in (25) by adding the ARG1 modifier and then saturating the argument position.

(25) **Building the event description:**

$$\dfrac{\dfrac{\begin{array}{c}\lambda P_{vt}.\lambda x_e.\lambda e_v.P(x) \wedge \text{ARG1}(e) = x : \\ (e_v \multimap e_t) \multimap g_e \multimap (e_v \multimap e_t)\end{array} \qquad \lambda e_v.appear(e) : e_v \multimap e_t}{\lambda x_e.\lambda e_v.appear(e) \wedge \text{ARG1}(e) = x : g_e \multimap (e_v \multimap e_t)} \qquad X_i : [g_e]^i}{\lambda e_v.appear(e) \wedge \text{ARG1}(e) = X_i : (e_v \multimap e_t)}$$

[8] In the following examples, we use $\oplus$ for the merge operation.

[9] This is needed for the reasoning pipeline. Here, we first present the MCs in the DRT format, but construct proofs with their FOL counterparts to highlight the correspondence and for reasons of conciseness.

[10] This is not relevant for the examples covered here, even when reasoning is involved.

Event closure then raises the derivation from modifying the event predicate to the (semantics of the) f-structure, allowing it to interact with other components of the proof (essentially, the semantics of noun phrases).

(26) **Event closure and discharging the assumption linked to the argument:**

$$\dfrac{\dfrac{\lambda P_{vt}.\exists e_v[P(e)] : (e_v \multimap e_t) \multimap f_t \qquad \lambda e_v.appear(e) \wedge \text{ARG}1(e) = X_i : (e_v \multimap e_t)}{\exists e_v[appear(e) \wedge \text{ARG}1(e) = X_i] : f_t}}{\lambda x_e.\exists e_v[appear(e) \wedge \text{ARG}1(e) = x] : g_e \multimap f_t}$$

(27) **Resulting entry for the intransitive verb (uninstantiated):**

$$\lambda x_e.\begin{array}{|c|}\hline e \\ \hline \text{appear}(e) \\ \text{ARG}1(e) = x \\ \hline \end{array} : \%arg1_e \multimap\uparrow_{\sigma_t}$$

Using the MCs in (28) contributed by the NP and more specifically their combination in (29), the meaning of the sentence “a dog appeared" can be derived as in (30), with the equivalent DRS in (31).

(28) **Components of the quantified noun phrase:**

a. $\lambda x_e.\begin{array}{|c|}\hline \\ \hline \text{dog}(x) \\ \hline \end{array} : (\uparrow \text{SUBJ})_{\sigma_e} \multimap (\uparrow \text{SUBJ})_{\sigma_t}$

b. $\lambda P_{et}.\lambda Q_{et}.\begin{array}{|c|}\hline x \\ \hline \\ \hline \end{array} \oplus P(x) \oplus Q(x) :$

$((\uparrow \text{SUBJ})_{\sigma_e} \multimap (\uparrow \text{SUBJ})_{\sigma_t}) \multimap ((\uparrow \text{SUBJ})_{\sigma_e} \multimap \%scope_t) \multimap \%scope_t$,
where $\%scope = (\text{GF}^{+} \uparrow)_{\sigma} \approx f$.[11]

(29) **Constructing quantified noun phrase meaning:**

$$\dfrac{\begin{array}{c}\lambda P_{et}.\lambda Q_{et}.\exists x[P(x) \wedge Q(x)] : \\ (g_e \multimap g_t) \multimap (g_e \multimap f_t) \multimap f_t\end{array} \qquad \lambda x.dog(x) : g_e \multimap g_t}{\lambda Q_{et}.\exists x[dog(x) \wedge Q(x)] : (g_e \multimap f_t) \multimap f_t}$$

(30) **Finishing up the derivation:**

$$\dfrac{\begin{array}{c}\lambda Q_{et}.\exists x[dog(x) \wedge Q(x)] : \\ (g_e \multimap f_t) \multimap f_t\end{array} \qquad \begin{array}{c}\lambda x_e.\exists e_v[appear(e) \wedge \text{ARG}1(e) = x] : \\ g_e \multimap f_t\end{array}}{\exists x_e[\exists e_v[appear(e) \wedge dog(x) \wedge \text{ARG}1(e) = x]] : f_t}$$

(31)
$$\begin{array}{|c|}\hline e,x \\ \hline \text{dog}(x) \\ \text{appear}(e) \\ \text{ARG}1(e) = x \\ \hline \end{array}$$

[11]Reading: The typical case is that the scope of a quantifier corresponds to the f-structure index of the verbal spine (f in our case), but as functional uncertainty is used here, other nodes are in principle possible in more complex proofs.

4.4 Adjectives

Following the analysis presented in section 2, predicative adjectives are analysed as taking the subject as an argument and are themselves so-called XCOMP-PRED arguments of *to be*, thus generally following a raising analysis of predicatives. Attributives are treated as modifiers of a noun collected in an adjunct set. Under this analysis, adjectives are lexically ambiguous, captured by the template in Figure 3. This template highlights the novel Glue notation presented in Zymla et al. (2025), which allows for a "literal" notation of meaning constructors (i.e., not as AVM as in Dalrymple et al. 2020). They are delimited by an initial marker **:** $ and ended by a comma or by a final dot.

The main difference conveyed in this template is that an attributive adjective modifies a property of entities, whereas a predicative adjective modifies a property of events (essentially, the event predicate that identifies the event variable introduced by the copula).[12] The two possibilities are constrained by an ATYPE attribute that is associated with the underlying syntactic positions. To pass down the attribute from the syntax to the lexicon, we use parameterized rules. The NP rule in Figure 2 illustrates this. Analogously, the VP rule introduces the PREDICATIVE parameter.

```
NP -->  (D: ^=!)
        AP[attributive]*: ! $ (^ ADJ);
        (NMod: ! $ (^ MOD))
        N: ^=!;
```

Figure 2: Parameterized NP rule in XLE notation

In section 2, we explained that the comparative marker is represented as a modifier (adjunct) of the adjective in the f-structure. This adjunct can further take an oblique comparative (OBL_{compar}) as an argument, which is contributed by an optional *than*- or *as*-clause.[13] These OBL_{compar} are defined in the CPComp rule in (32). The CComp is either *than* for comparatives or *as* for equatives. For the latter, the first instance of *as* is a different word class, namely an ADVComp, which attaches as an adjunct before the adjective and takes the OBL_{compar} as an argument. Further, as mentioned before, comparatives syntactically allow both phrasal ("than the dog") and clausal ("than the dog is tall") complements. This is reflected in the rule in (32), where either can be the object of the CComp.

(32)
$$\text{CPComp} \rightarrow \underset{\uparrow = \downarrow}{\text{CComp}} \left\{ \begin{array}{c} \text{NP} \\ (\uparrow \text{OBJ}) = \downarrow \\ (\downarrow \text{CASE}) = \text{ACC} \end{array} \middle| \begin{array}{c} \text{S} \\ (\uparrow \text{OBJ}) = \downarrow \\ \end{array} \right\}$$

[12]This analysis patterns with examples like the following.

(i) Sam is a beautiful dancer.

There, *beautiful* can be interpreted intersectively, or as modifying the underlying event. By adding a corresponding axiom linking the subject of the event to the adjectival description, we assume both (rather than an ambiguity). We leave a more refined analysis for future work.

[13]We treat them as complement clauses, but nothing hinges on this exact classification.

```
DEFAULT-ADJ-SEM(P)  = { @(PRED P)
                  (^ ATYPE) =c attributive
                  s::(ADJ $ ^) = %s
                  :$ lam(X,drs([],[pred('P',X)])) : (s::^_e -o s::^_t),
                  :$ lam(Q,lam(R,lam(X,merge(app(Q,X),app(R,X))))) :
                  ((s::^_e -o s::^_t) -o ((%s_e -o %s_t) -o
                  (%s_e -o %s_t))) || noscope,
                  |
                  (^PRED) = 'P<(^SUBJ)>'
                  (^ ATYPE) =c predicative
                  (s::(XCOMP-PRED ^) EV) = %s
                  :$ lam(X,drs([],[pred('P',X)])) : (s::^_v -o s::^_t),
                  :$ lam(Q,lam(R,lam(X,merge(app(Q,X),app(R,X))))) :
                  ((s::^_v -o s::^_t) -o ((%s_v -o %s_t) -o
                  (%s_v -o %s_t))) || noscope,
                  }.
```

Figure 3: Co-descriptive entry for adjectives in XLE notation

4.5 Enriching adjective semantics via description-by-analysis

The presented grammar is designed to deal with the compositional aspects of semantic meaning. However, as explained in section 3, adjectives, particularly gradable ones, are highly context sensitive. We partially capture this by adding a layer of description-by-analysis (DBA) rules, the benefits of which are further explained in section 4.7. For this, we use LiGER (Zymla et al. 2025), which models DBA as a set of ordered rewrite rules, similar to Crouch & King's (2006) rewrite semantics. These rules are used on the one hand to establish additional semantic structure (s-structure) and on the other hand to introduce meaning constructors. LiGER treats the output of XLE as a directed (acyclic) graph and the left-hand side of a rule (which we call a query) must match this graph to apply. `#a...#z` are variables over graph nodes, usually f-structure indices, but also possibly indices of, e.g., s-structure. `%a...%z` are variables over strings, usually attribute values (including PRED values), which are encoded as AVMs attached to the underlying graph's nodes. Thus, the following query matches with an f-structure that has an XCOMP-PRED and that has a semantic structure specifying the index of its event variable (see the f-structure in Figure 7).

(33) `#x XCOMP-PRED #a & #x s:: #y EV #z &`
`#a ATYPE 'predicative'`

To analyze a degree adjective, we begin with type-raising it to a higher type including degrees, i.e., from $< e, t >$ to $< d, et >$. This is illustrated in Figure 4. The antecedent of this rule searches for a a sub-graph of an f-structure that describes a predicative use of a degree adjective (as in (33)). Given that such a structure is found, the rule specifies a semantic structure (indicated via `s::`) for the adjective consisting of a DEGREE attribute and a DEGREE-HOLDER attribute. Furthermore, it adds a meaning constructor that consumes the corresponding semantic description and introduces a new one carrying a degree variable.[14] On the meaning side of the MC, we use the LiGER-

[14]We implement type-raising in this way to keep the the core semantics and the DBA rules independent.

specific predicate `strip/1` which, given a PRED-value, returns a string of the same PRED-value without subcategorization information, i.e., just the PRED-value's stem.

```
//predicative
#x XCOMP-PRED #a & #x s:: #y EV #z &
#a ATYPE 'predicative' & #a SEMTYPE 'degree' & #a PRED %a &  #a s:: #s
==> & #s DEGREE #d & #s DEGREE-HOLDER #z &
    #p GLUE lam(P,lam(D,lam(X,drs([],[rel(strip(%a),X,D)])))) :
    ((#s_v -o #s_t) -o (#d_d -o (#s_v -o #s_t))).
```

Figure 4: DBA-rule for type-raising predicative adjectives

Next, we can specify rules that correspond to the different possible realizations of gradable adjectives, i.e., their POSITIVE, COMPARATIVE, and SUPERLATIVE use.[15] Generally, these rules serve as degree closure, binding the degree variable of a gradable adjective. However, the three closure operations are mutually exclusive. For example, the POSITIVE use of a gradable adjective is closed off by the rule in 5.

```
//predicative
#x XCOMP-PRED #a PRED %a & #x SUBJ #y PRED %y &
#a s:: #b DEGREE #d & #b DEGREE-HOLDER #e
==> #d GLUE lam(P,merge(drs([D:d],[]),merge(drs([],[eq(th_strip(%a)
    ('strip(%y)'),D)]),app(P,D)))) : ((#d_d -o #e_t) -o #e_t) ||
    noscope.
```

Figure 5: Rule for positive interpretation of gradable adjectives

This rule applies just in case there exists a predicative adjective that has a DEGREE attribute and a holder of that degree in its s-structure. In the case of the predicative use, the degree is held by the event variable rather than an entity (see Figure 4). This means that a predicatively used adjective introduces an event predicate corresponding to a potentially temporary state during which the adjective is true of the subject of the predicative adjective.[16] The DEGREE and DEGREE-HOLDER attribute have been introduced by the previously discussed type-raising rule, highlighting the incremental nature of these rules. Consequently, this rule only triggers if a suitable degree expression requires POSITIVE closure. Generally, the closure rules check for the information contributed by the morphological component for a given adjective (see section 2). As discussed in section 3.1, the semantics for the POSITIVE closure rule state that the degree in question is

[15]In this paper, we do not provide an analysis of the superlative, which is compositionally challenging due to its inherent ambiguity regarding its comparison class. This ambiguity is arguably mediated by focus, further complicating the analysis.

[16]This approach is necessary (for our semantics) to permit the ambiguity in sentences like "Every dog is faster than a cat". More generally, if a predicative adjective would modify the subject directly, it would always be trapped in the restrictor of the corresponding quantifier. This leads to a situation where the corresponding quantifier cannot outscope a quantifier in a potential comparative clause. The relation between the subject of the predicative and the adjective is established via a meaning postulate that asserts the following correspondence: $adj(e) \wedge be(e) \wedge arg1(e) = x \rightarrow adj(x)$. If a predicative eventuality is modified by an adjective, then the first argument of that eventuality has the property described by the adjective. Admittedly, this is a technical trick, rather than a sophisticated compositional analysis.

larger than a contextual standard, namely the one which is contributed by the modified noun. Thus, in *fast computer*, the comparison class is described by $\theta_{fast}(computer)$. For the sake of our implementation, we assume this to be a constant that can be ordered relative to other standards.[17]

The simplified rule in Figure 6 exemplifies how we deal with comparatives.[18] Similar to the POSITIVE closure-rule, this rule searches for an f-structure of an adjective marked for comparative use with a corresponding s-structure containing a DEGREE attribute and a DEGREE-HOLDER attribute. It also checks whether the comparison class has been explicitly specified via a comparative clause by searching for an OBL$_{compar}$ in the f-structure. In this context, the basic MC for comparative constructions according to the P-not-P analysis is applied.

As already stated, this rule competes with the previously introduced rule for POSITIVE closure (recall also example (19) and the accompanying discussion in section 3.1). This is achieved by both of them aiming to attach an MC to the degree node of the modified adjective (in our rules `#d`). This ensures that positive uses and comparative uses are mutually exclusive.[19]

```
#h DEGREE 'comparative' & #h s:: #b DEGREE #d & #b DEGREE-HOLDER #e &
& #h ADJ #a in_set #o OBL-COMPAR #i OBJ #j
==> #d GLUE \R.\x.\y.Ed[R(d)(x)&~R(d)(y)] :
    ((#g_d -o (#g_e -o #f_t)) -o (#g_e -o (#j_e -o #f_t)))
```

Figure 6: Simplified rule for the comparative

4.6 Example derivation

We present the analysis of the comparative sentence in (34). The semantic derivation process is parallel to the one presented in section 4.3. Thus, we first build the semantics of the predicative construction. The subject or degree holder is saturated by an assumption that is later discharged during combination with the NP *the PC-6082*, making sure that *the* has wide scope.[20] The predicative component of the proof is built twice, once for the subject and once for the comparison class which requires saturation by the object of the OBL$_{compar}$. Furthermore, its event variable is also saturated by the comparative semantics. Finally, the comparative semantics existentially close their degree variables.

[17]Fixing the constant in this way helps avoid the Sorites paradox , but it leaves underspecified the exact nature of these constants and how they are calculated (see Kennedy 2007 for discussion)

[18]In section 4.6, we present the DRT version of the meaning constructor, which is more complex as it is couched in an event semantics. However, the overall idea remains the same. It is noteworthy that the DRT meaning constructor is more general, as the semantics for the main clause and the comparative clause are constructed separately, rather than sharing the predication of the adjective. This is necessary anyway, as discussed in section 3.1 for cases like the following:

(i) The elephant is taller than the house is wide.

[19]To be precise, the comparative closure rule overwrites the positive closure rule as it is more specific by virtue of checking for COMPARATIVE morphology in the f-structure.

[20]For *the* this is not strictly necessary as the wide scope of *the* could also be modeled via presupposition resolution in the sense of Van der Sandt (1992).

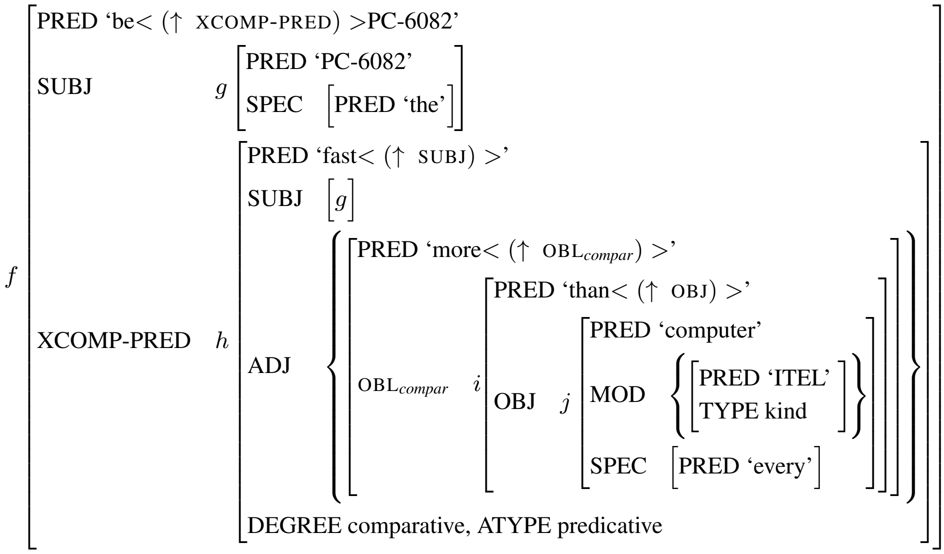


Figure 7: F-structure for *The PC-6082 is faster than every ITEL computer*

(34) The PC-6082 is faster than every ITEL computer.

(35) a. $\textbf{fast}_{\textbf{PC-6082}}: \lambda\delta.\lambda e.\begin{array}{|c|}\hline \\ \hline \text{be}(e) \\ \text{fast}(e,\delta) \\ \text{ARG1}(e)\text{=X} \\ \hline\end{array} : g_d \multimap e_v \multimap h_t$

b. $\textbf{fast}_{\textbf{EVERY ITEL COMPUTER}}: \lambda\delta.\lambda e.\begin{array}{|c|}\hline \\ \hline \text{fast}(e,\delta) \\ \text{ARG1}(e)\text{=Y} \\ \hline\end{array} : j_d \multimap c_v \multimap i_t$

c. $\lambda P.\lambda Q.\lambda e.\begin{array}{|c|}\hline \delta \\ \hline \\ \hline\end{array} \oplus P(\delta)(e) \oplus \begin{array}{|c|}\hline \\ \hline \neg \begin{array}{|c|}\hline e' \\ \hline \\ \hline\end{array} \oplus Q(\delta)(e') \\ \hline\end{array} :$

$(g_d \multimap e_v \multimap h_t) \multimap (j_d \multimap c_v \multimap i_t) \multimap (e_v \multimap h_t)$

Event closure is applied to the result of combining these elements, after which the quantifiers can be applied. As mentioned above, the definite takes wide scope, while the universal, here modeled via implication (see Bos 2008), takes narrow scope. However, in general, our analysis is compatible with quantifier ambiguity, as the semantics of the comparative are simply part of a complex event description.

The resulting DRS, as produced by the computational system, is shown in Figure 8. It states that there is an instance (`x2`; an event) of being fast to a degree `x5` for the PC-6082 (`x3`), and that there is no instance of an event such that an ITEL computer (`x1`) is fast to that degree. In section 5, we describe the reasoning component on the basis of FraCaS example that it belongs to:

```
 ____________________________________________
| x3                                         |
|--------------------------------------------|
| x3 = pc-6082                               |
|  ______________       ____________________ |
| | x1           |     | x2 x5              | |
| |--------------|     |--------------------| |
| | kind(itel,x1) | ==> | fast(x2,x5)        | |
| | computer(x1)  |     |      ____________  | |
| |_______________|     |     | x4          | | |
|                       | __  |-------------| | |
|                       |   | | fast(x4,x5) | | |
|                       |     | arg1(x4,x1) | | |
|                       |     |_____________| | |
|                       | be(x2)              | |
|                       | arg1(x2,x3)         | |
|                       |_____________________| |
|_____________________________________________|
```

Figure 8: DRS for *The PC-6082 is faster than every ITEL computer*

(36)
The PC-6082 is faster than every ITEL computer.
The ITEL-ZX is an ITEL computer.
———
Is the PC-6082 faster than the ITEL-ZX? → YES

4.7 Hybrid grammars for analysis prototyping

In principle, both DBA and co-descriptive semantics can be used to build the semantic representations presented here. However, adding meaning constructors to a grammar can considerably increase its complexity, leading to potential challenges during debugging. One such case is meaning constructors that introduce functional uncertainties, such as quantifiers. As in XLE+Glue (Dalrymple et al. 2020; Zymla et al. 2025), meaning constructors are essentially resolved in the same way as f-structures. Thus, adding meaning constructors thoughtlessly can make resolving functional annotations untenable.[21] By disentangling f-structure and meaning constructor creation, the two systems are computationally easier to maintain. Furthermore, since the rewrite system, LiGER, can then operate on fully assembled f-structures, the search space for its queries is much smaller than for functional annotations in an XLE+Glue grammar. In other words, it is easier to resolve a query in LiGER than to resolve a complex set of functional and semantic annotations to find corresponding f-structures with attached MCs.[22] Given all this, we briefly want to highlight that using DBA to test analyses can help analyze generalizations without having to implement large-scale grammar changes, making DBA

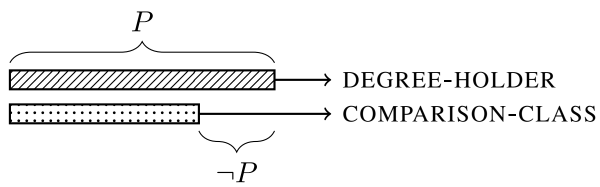
[21]One could alternatively opt for quantified linear logic (i.e., first-order Glue; Kokkonidis 2008), but that leads to different issues in devising efficient Glue provers (Lev 2007; Zymla 2024a).

[22]Yet another way to look at this is via the notion of constructive vs. non-constructive constraints (see Crouch et al. 2017 on function application and uncertainty).

ideal for testing the computational implementation of new analyses, and helping to disentangle complexities by making certain MCs non-constructive.

5 Reasoning with XLE+Glue

As indicated in section 3, reasoning about degrees requires additional non-compositional axioms to fall out correctly. To make this clearer, consider the comparative semantics we employ. It is based on the P-not-P-analysis, intuitively indicating that the subject holds a degree that the comparison class does not. However, this alone is true whether the subject's degree is larger or smaller than that of the comparison class. More precisely, we want to encode that the subject *reaches* a degree that the comparison class does not reach. Example (37) (adapted from Haruta et al. 2022) illustrates this. Essentially, the comparative is true if there is a degree in *P* that describes the degree the subject holds that is in the $\neg P$-range for the comparison class.

(37)


As explained in section 3, to express this we need to capture that degrees are *monotonic* and have an *upper bound*. As described in Haruta et al. (2022), these properties can be reduced to a single postulate, the consistency postulate , which we trigger via the following rewrite rule when a comparative is present.

(38) `#a PRED %adj #a DEGREE 'comparative' ==>`
`#a AXIOM` $\forall x \forall y[\exists \delta[\%\text{adj}(x,\delta) \wedge \neg\%\text{adj}(y,\delta)] \rightarrow \forall \delta[\%\text{adj}(y,\delta) \rightarrow \%\text{adj}(x,\delta)]].$

Given a comparative adjective, this rule attaches the axiom to the linguistic structure of the input sentence that triggers it. More concretely, axioms are stored in a separate AXIOM attribute that, similar to the GLUE attribute, flags the information for collection in an axiom set. Note that the axiom is not higher-order as we use the adjective's PRED-value to instantiate the axiom to the case of the adjective present in the input.

Similar to the different types of degree closure discussed in section 4, we also introduce different axioms for the positive use of gradable adjectives as they are not covered by the consistency postulate, but require the more fine-grained *monotonicity* and *upper bound* axioms introduced in section 3. More generally, different syntactic constructions may require different axioms for reasoning. Another way of conceptualizing this is as an activation process, where different aspects of knowledge are primed depending on the problem that is reasoned about. Thus, while it would be possible to have a massive static database of axioms that is called upon for every reasoning step, modeling axiomatic knowledge dynamically makes individual proofs and reasoning steps easier to track and reduces the search space for the automated theorem prover. Together with the DBA-component of the grammar, this also allows us to develop targeted grammars that are able to verify certain kinds of linguistic expressions and their effect on reasoning independently of obfuscating factors. For example, our grammar is heavily specialized to deal with degrees, but currently completely ignores tense and aspect. Thus, having corresponding information in the online processing of our semantic parser would be unnecessary and makes the outputs used, e.g for debugging, less easy to evaluate.

When an input is parsed in our semantic parser, the initially relevant axioms are controlled by LiGER and collected in a set that is attached to linguistic structure of the input sentence. For example, (38) attaches the axiom directly to the f-structure, but it could also be stored in the s-structure or even a separate axiom structure. During reasoning, the individual sentences are combined semantically and in terms of the relevant axioms. Essentially, the axiom set for reasoning is the union of the axiom sets of the individual sentences. Storing the axioms in a set provides a natural way to avoid duplicates.

5.1 Inference via discourse checks

Following Blackburn & Bos (2005), we decompose the NLI task into discourse checks for consistency and informativity. They are modeled as logic problems that are resolved via automated theorem proving with Vampire (Kovács & Voronkov 2013). Example (39) explains how the checks are formalized. In Blackburn & Bos (2005), the positive checks rely on model checking, i.e., they succeed when a model builder finds a model.[23] While we adopt this idea for the core semantics, semantics with non-finite domains, like degree semantics (which, here, make use of integers) are not compatible with it. In this case, discourse checks may remain undefined or only partially defined.[24]

(39) For some (set of conjoined) premise(s) p and a hypothesis q:

a.	$\neg(p \rightarrow q)$	(if model exists) +informative
b.	$p \rightarrow q$	−informative
c.	$p \wedge q$	(if model exists) +consistent
d.	$p \rightarrow \neg q$	−consistent

To solve this issue, we propose a more fine-grained heuristic for interpreting discourse checks that resolves *undefined* values via the pragmatic notion of *relevance*. Essentially, we assume that new contributions to a discourse are *consistent* and *informative* if we only have partial proof for it or no proof for the opposite.

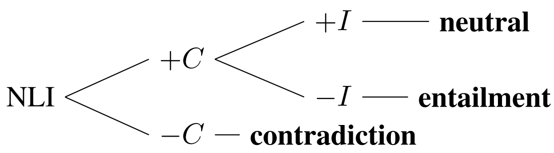
We also take ambiguities into consideration. Concretely, we determine consistency and informativity by generalizing over all readings, accumulating average scores. Thus, the present approach selects the most likely interpretation to be the one that coincides with the most individual readings. The approach works well in a controlled setting such as the one presented here. However, we anticipate the need to refine this system in future work, e.g., by assigning probabilities to readings and pruning the dialog as it develops.

In example (40), we show how these checks map onto NLI labels: inconsistency is interpreted as a contradiction and if an utterance is consistent but not informative, it is entailed. Thus, it is straightforward to operationlize the discourse checks for reasoning.

[23] Vampire provides different proof search strategies, including model building, and proof by refutation.

[24] If a proof is undefined, it means that neither a negative nor a positive solution was found. Usually because the prover times out.

(40)



5.2 Putting it all together

We alluded to treating NLI as a discourse update problem. When applying this to (36), we get two updates. The first update contributed by the second premise is neutral with respect to the initial premise, i.e., it is supposed to be informative and consistent. However, as explained in the previous section, due to the circumstances imposed by reasoning with degrees, the output, as shown in Table 1, is not fully specified ($+$ indicates satisfiability, $-$ indicates refutation, and ? indicates that the result is undefined). From the result of the prover we cannot infer either consistency or inconsistency as satisfiability of the negative consistency check merely means that inconsistency is a possibility not a necessity (consequently, the negative consistency check is really only informative when it fails). However, we have partial results for determining informativity. Concretely, we show that the prover finds a proof indicating that the second premise does not follow from the first.[25] As previously explained, we rely on the maxim of relevance to intrepret such partial results. In this case, we accept the update. Thus, we conjoin the two premises to serve as single premise for the second update (i.e., the original problem from the FraCaS testsuite).

	+consistent	−consistent	+informative	−informative
Termination reason	?	+	+	?
SZS status	?	+	+	?

Table 1: Update 1

The second update, the conclusion of (36), should be uninformative (and, thus entailed). This is partially captured by the theorem prover, as shown in Table 2. Thus, we graciously interpret the update as intended, providing the correct NLI label.

	+consistent	−consistent	+informative	−informative
Termination reason	?	?	−	?
SZS status	?	?	−	?

Table 2: Update 2

As this example highlights, theorem proving faces challenges when dealing with complex semantic domains. By adapting the underlying heuristic, we can capture these examples. However, a quantitative exploration is necessary to make sure that our system

[25]In the future, we want to explore additional or alternative tests for different reasoning domains as well as adding a confidence value based on how expressive the prover's outputs are. Generally, refutation of a discourse check is the most expressive result, as it makes the strongest generalization over possible situations described by a proof. Satisfiability is less expressive, as it does not generalize over situations, and it is clear that an undefined result is impossible to interpret on logical grounds.

generally makes the correct predictions. As proposed here, we believe that pragmatic principles are a fruitful avenue for further research (see also Blackburn & Bos 2005).

6 Conclusion

In the past few years, XLE+Glue has spurred the exploration of semantic parsing based on LFG principles and formalizations. However, the focus has been strictly on the syntax/semantics interface. While this has contributed to the development of novel analyses regarding the syntax/semantics interface (e.g., Przepiórkowski & Patejuk 2023), the adequacy of the resulting semantics remains untested. This is particularly unsatisfying from a computational perspective, but arguably goes against the holistic approach to LFG as a form-to-meaning mapping more generally.

The present paper aims to close this gap by interfacing XLE+Glue with an inference component. The system highlights the interaction between compositional and non-compositional meaning, which can be seen as a necessary step to incorporate pragmatics into the LFG framework. Correspondingly, we base our approach to reasoning on the discourse checks of *consistency* and *informativity*. We have illustrated how their expressiveness could be extended to deal with complex reasoning problems by invoking the maxim of relevance, although this warrants some more rigorous testing.

The work presented here is implemented as part of XLE+Glue. Correspondingly, we provide an XLE+Glue grammar of English that illustrates the system with a focus on the interpretation of gradable adjectives, as a test case for more complex reasoning. The result is a layered approach to semantic parsing that shows how XLE+Glue grammars can be incrementally extended without violating certain core properties, providing various avenues for future work.

The analysis of gradable adjectives presented in this paper is targeted towards examples from the FraCaS testsuite. However, there is certainly more to be said there. A particularly interesting avenue for future work is the superlative construction, which is challenging due to its subtle interaction with context.

All in all, we lay the foundation for task-oriented semantic parsing within some recently developed computational LFG tools. Thus, we provide an analysis that takes the mapping from form to meaning seriously and that opens up promising new directions for research.

References

Andrews, Avery. 1983. A note on the constituent structure of modifiers. *Linguistic Inquiry* 14(4). 695–697. https://www.jstor.org/stable/4178355.

Andrews, Avery D. 2018. Sets, heads, and spreading in LFG. *Journal of Language Modelling* 6(1). 1–53. https://doi.org/10.15398/jlm.v6i1.175.

Asudeh, Ash & Gianluca Giorgolo. 2012. Flexible composition for optional and derived arguments. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG'12 Conference*, 64–84. Stanford: CSLI Publications. https://typo.uni-konstanz.de/lfg-proceedings/LFGprocCSLI/LFG2012/papers/lfg12asudehgiorgolo.pdf.

Beesley, Kenneth R. & Lauri Karttunen. 2003. *Finite state morphology*. Stanford: CSLI Publications.

Bhatt, Rajesh & Shoichi Takahashi. 2011. Reduced and unreduced phrasal comparatives. *Natural Language & Linguistic Theory* 29(3). 581–620. https://doi.org/doi:10.1007/s11049-011-9137-1.

Blackburn, Patrick & Johan Bos. 2005. *Representation and inference for natural language: A first course in computational semantics*. Stanford: Center for the Study of Language and Information.

Bos, Johan. 2008. Wide-coverage semantic analysis with Boxer. In Johan Bos & Rodolfo Delmonte (eds.), *Semantics in text processing. STEP 2008 conference proceedings*, 277–286. College Publications. https://aclanthology.org/W08-2222/.

Bresnan, Joan W. 1973. Syntax of the comparative clause construction in English. *Linguistic Inquiry* 4(3). 275–343. url:https://www.jstor.org/stable/4177775.

Butt, Miriam, Helge Dyvik, Tracy Holloway King, Hiroshi Masuichi & Christian Rohrer. 2002. The Parallel Grammar project. In *COLING-02: Grammar engineering and evaluation*. https://aclanthology.org/W02-1503/.

Butt, Miriam, Tracy Holloway King, María-Eugenia Niño & Frédérique Segond. 1999. *A grammar writer's cookbook*. Stanford: CSLI Publications.

Cooper, Robin, Dick Crouch, Jan van Eijck, Chris Fox, Josef van Genabith, Jan Jaspars, Hans Kamp, David Milward, Manfred Pinkal, Massimo Poesio & Steve Pulman. 1996. Using the framework. Tech. Rep. LRE 62-051 D-16 The FraCaS Consortium.

Cresswell, Max J. 1976. The semantics of degree. In Barbara H. Partee (ed.), *Montague grammar*, 261–292. Elsevier.

Crouch, Dick, Mary Dalrymple, Ronald M. Kaplan, Tracy Holloway King, John T. Maxwell III & Paula Newman. 2017. *XLE documentation*. Palo Alto Research Center, https://ling.sprachwiss.uni-konstanz.de/pages/xle/doc/xle_toc.html.

Crouch, Richard & Tracy Holloway King. 2006. Semantics via f-structure rewriting. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG'06 conference*, 145–165. Stanford: CSLI Publications. https://typo.uni-konstanz.de/lfg-proceedings/LFGprocCSLI/LFG2006/pdfs/lfg06crouchking.pdf.

Dalrymple, Mary. 1999. *Semantics and syntax in Lexical Functional Grammar: The resource logic approach*. Cambridge, MA: The MIT Press.

Dalrymple, Mary. 2001. *Lexical Functional Grammar* (Syntax and Semantics 34). New York: Academic Press. https://doi.org/10.1163/9781849500104.

Dalrymple, Mary, Agnieszka Patejuk & Mark-Matthias Zymla. 2020. XLE+Glue – a new tool for integrating semantic analysis in XLE. In Miriam Butt & Ida Toivonen (eds.), *Proceedings of the LFG'20 Conference*, 89–108. Stanford, CA: CSLI Publications. https://web.stanford.edu/group/cslipublications/cslipublications/LFG/LFG-2020/lfg2020-dpz.pdf.

Findlay, Jamie Y. 2020. Mapping Theory and the anatomy of a verbal lexical entry. In Miriam Butt & Ida Toivonen (eds.), *Proceedings of the LFG'20 Conference*, 127–147. Stanford: CSLI Publications. https://typo.uni-konstanz.de/lfg-proceedings/LFGprocCSLI/LFG2020/lfg2020-findlay.pdf.

Findlay, Jamie Y. & Dag T. T. Haug. 2022. Managing scope ambiguities in Glue via multistage proving. In Miriam Butt, Jamie Y Findlay & Ida Toivonen (eds.), *Proceedings of the LFG'22 Conference*, 144–163. Konstanz: PubliKon. https://lfg-proceedings.org/lfg/index.php/main/article/view/18.

Haruta, Izumi, Koji Mineshima & Daisuke Bekki. 2022. Implementing Natural Language Inference for comparatives. *Journal of Language Modelling* 10(1). 139–191. https://doi.org/10.15398/jlm.v10i1.294.

Haug, Dag. 2008. Tense and aspect for Glue semantics: the case of participial XADJs. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG'08 Conference*, 291–311. Stanford, CA: CSLI Publications. http://web.stanford.edu/group/cslipublications/cslipublications/LFG/13/papers/lfg08haug.pdf.

Kennedy, Christopher. 2007. Vagueness and grammar: The semantics of relative and absolute gradable adjectives. *Linguistics and Philosophy* 30(1). 1–45. https://doi.org/10.1007/s10988-006-9008-0.

Kennedy, Christopher & Louise McNally. 2005. Scale structure, degree modification, and the semantics of gradable predicates. *Language* 81(2). 345–381. https://doi.org/10.1353/lan.2005.0071.

Klein, Ewan. 1980. A semantics for positive and comparative adjectives. *Linguistics and Philosophy* 4(1). 1–45. https://doi.org/10.1007/BF00351812.

Klein, Ewan. 1982. The interpretation of adjectival comparatives. *Journal of Linguistics* 18(1). 113–136. https://doi.org/10.1017/S0022226700007271.

Kokkonidis, Miltiadis. 2008. First-Order Glue. *Journal of Logic, Language and Information* 17(1). 43–68. https://doi.org/doi:10.1007/s10849-006-9031-0.

Kovács, Laura & Andrei Voronkov. 2013. First-order theorem proving and VAMPIRE. In Natasha Sharygina & Helmut Veith (eds.), *International Conference on Computer Aided Verification (CAV 2013)* (Lecture Notes in Computer Science 8044), 1–35. Springer. https://doi.org/10.1007/978-3-642-39799-8_1.

Lev, Iddo. 2007. *Packed computation of exact meaning representations*. Ph.D. thesis, Stanford University.

Meßmer, Moritz & Mark-Matthias Zymla. 2018. The Glue semantics workbench: a modular toolkit for exploring linear logic and Glue semantics. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG'18 Conference*, 249–263. Stanford, CA: CSLI Publications. https://web.stanford.edu/group/cslipublications/cslipublications/LFG/LFG-2018/lfg2018-messmer-zymla.pdf.

Parsons, Terence. 1990. *Events in the semantics of English*. Cambridge, MA: The MIT Press.

Przepiórkowski, Adam & Agnieszka Patejuk. 2023. Filling gaps with Glue. In Miriam Butt, Jamie Y. Findlay & Ida Toivonen (eds.), *Proceedings of the LFG'23 Conference*, 223–240. Konstanz: PubliKon. https://lfg-proceedings.org/lfg/index.php/main/article/view/41.

Schwarzschild, Roger. 2008. The semantics of comparatives and other degree constructions. *Language and Linguistics Compass* 2(2). 308–331. https://doi.org/10.1111/j.1749-818X.2007.00049.x.

Van der Sandt, Rob A. 1992. Presupposition projection as anaphora resolution. *Journal of Semantics* 9(4). 333–377. https://doi.org/10.1093/jos/9.4.333.

Zymla, Mark-Matthias. 2017. Cross-linguistically viable treatment of tense and aspect in parallel grammar development. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG'17 Conference*. Konstanz: CSLI Publications. https://typo.uni-konstanz.de/lfg-proceedings/LFGprocCSLI/LFG2017/lfg2017-zymla-sulger.pdf.

Zymla, Mark-Matthias. 2024a. Ambiguity management in computational Glue semantics. In Miriam Butt, Jamie Y. Findlay & Ida Toivonen (eds.), *Proceedings of the LFG'24 Conference*, 285–310. Konstanz: PubliKon. https://lfg-proceedings.org/lfg/index.php/main/article/view/59/41.

Zymla, Mark-Matthias. 2024b. *Tense and aspect in multilingual semantic construction*. Ph.D. thesis, University of Konstanz.

Zymla, Mark-Matthias, Mary Dalrymple & Agnieszka Patejuk. 2025. Computational semantic tools for Glue semantics. In *Proceedings of the 16th International Conference on Computational Semantics (IWCS 2025)*, 189–207. Düsseldorf: Association for Computational Linguistics. https://aclanthology.org/2025.iwcs-main.18/.