Applied Temporal Analysis: A Complete Run of the FraCaS Test Suite

Jean-Philippe Bernardy **Stergios Chatzikyriakidis**
Centre for Linguistic Theory and Studies in Probability
Department of Philosophy, Linguistics and Theory of Science
University of Gothenburg
firstname.lastname@gu.se

Abstract

In this paper, we propose an implementation of temporal semantics that translates syntax trees to logical formulas, suitable for consumption by the Coq proof assistant. The analysis supports a wide range of phenomena including: temporal references, temporal adverbs, aspectual classes and progressives. The new semantics are built on top of a previous system handling all sections of the FraCaS test suite except the temporal reference section, and we obtain an accuracy of 81 percent overall and 73 percent for the problems explicitly marked as related to temporal reference. To the best of our knowledge, this is the best performance of a logical system on the whole of the FraCaS.

1 Introduction

The semantics of tense and aspect has been a long standing issue in the study of formal semantics since the early days of Montague Grammar and a number of different ideas have been put forth to deal with them throughout the years. Recent proposals include the works of the following authors: Dowty (2012); Prior and Hasle (2003); Steedman (2000); Higginbotham (2009); Fernando (2015). The semantics of tense and aspect have been also considered in the study of Natural Language Inference (NLI). The various datasets for NLI that have been proposed by the years contain examples that have some implicit or explicit reliance on inferences related to tense and aspect. One of the early datasets used to test logical approaches, the FraCaS test suite (Cooper et al., 1996) contains a whole section dedicated to temporal and aspectual inference (section 7 of the dataset). This part of the FraCaS test suite has been difficult to tackle. That is, so far, no computational system has been capable to deal with it in its entirety: when authors report accuracy over the FraCaS test suite they skip this section. In fact, they also often skip the anaphora and ellipsis sections, the exception being the system presented by Bernardy and Chatzikyriakidis (2017, 2019), which includes support for anaphora and ellipsis but still omit the temporal section.[1] In this paper, we take up the challenge of providing a computationally viable account of tense and aspect to deal with the section 7 of the FraCaS test suite. Our account is not meant to be a theoretically extensive account of tense and aspect, but rather an account that is driven by the need to cover the test suite in a way that is general enough to capture the test suite examples, *while still covering the rest of the FraCaS test suite*.

The account is evaluated on the entailment properties of various temporal and aspectual examples, as given by the test suite. As such, we are not getting into the discussion of how tense and aspect might affect grammaticality or infelicitousness of various sentences. We assume that the sentences of the FraCaS suite are syntactically and semantically correct, and strive to produce accurate logical representations given that assumption. We further assume that the entailment annotations of various problems are valid, and we use those to evaluate the correctness of the logical representations of sentences.

The paper is structured as follows: in Section 2, we give a brief summary of the computational frameworks whose various subsystems rely on. In particular, the Grammatical Framework is used to construct the syntactic parser, the Coq proof assistant checks all the reasoning and a monad-based dynamic semantics deals with Montague-style se-

[1] One can consider that MacCartney and Manning (2007) have made a run against the whole test suite. However, they do not deal with multi-premise cases. Consequently only 36/75 cases in the temporal section are attempted. The general accuracy of the system is .59, and .61 for the temporal section. Our system, as shown Table 1, presents considerable improvements in coverage and accuracy over that of MacCartney and Manning.

mantics, and references (anaphora). We also provide some brief remarks on temporal semantics. In Section 3, we discuss the main aspects of the compositional semantics of our system, using various examples from the FraCaS suite to illustrate its effectiveness. In Section 5, we evaluate how our system performs with respect to the FraCaS suite. We ran the system across the whole suite: our system is thus the first which is capable of handling the complete FraCaS test suite. Yet, we are interested in particular in the performance on the temporal section. In Section 6, we conclude and discuss avenues for future work.

2 Temporal-Semantics in a Logic-based NLI System

Our temporal analysis places itself in the context of a complete NLI system – which is why we can test it on the FraCaS suite. In this section we give a brief overview of the phases of the system, referring the reader to published work for details.

GF The first phase of the system, parsing, is taken care of by the Grammatical Framework (GF, Ranta (2004)), which is a powerful parser generator for natural languages, based on type-theoretical abstract grammars. The present work leverages a syntactic representation of the FraCaS test suite in GF abstract syntax, in effect a GF FraCaS treebank (Ljunglöf and Siverbo, 2011). Thanks to this, we skip the parsing phase and avoid any syntactic ambiguity.

For the purpose of this paper, the important feature of GF syntax is that it aims at a balance of sufficient abstraction to provide a semantically-relevant structure, but at the same time it embeds sufficiently many syntactic features to be able to reconstruct natural-language text. That is, the parse trees generally satisfy the homomorphism requirement of Montague (1970, 1974), and we can focus on the translation of syntactic trees to logical forms. Consequently, the system presented here does not aim at textual natural language understanding, but rather provides a testable, systematic formal semantics of temporal phenomena. Example (1) shows an example abstract syntax tree and its realisation in English.

Dynamic Semantics Parse trees are then processed by a dynamic semantic component. Its role is essentially to support (non-temporal) anaphora, using a monadic-based dynamic semantics, generally following the state of the art in this matter (Unger, 2011; Charlow, 2015, 2017). Our particular implementation has weaknesses in certain areas (including group readings and counting; see Bernardy et al. (2020) for details) but non-temporal anaphoroi in the testsuite are generally resolved as they should be: on the whole accuracy is not affected significantly by issues in this subsystem.

As it is the case for other basic phenomena, there is not much interaction between our treatment of time and non-temporal anaphora. Critical exceptions are discussed in Section 3 and Section 5.

Montagovian Semantics Non-withstanding special support for anaphora, the core of the translation of syntax trees to logical form follows a standard montagovian semantics. In brief, sentences are interpreted as propositions, verbs and noun-phrases as predicates. We use type-raising of noun-phrases, to support quantifiers (Montague, 1974).

We support additionally the basic constructions and phenomena present in the testsuite, including adjectives, adverbs, nouns, verbs, anaphora, etc. The method is outlined by Montague (1970, 1973), but we direct the reader to our previous work for details Bernardy and Chatzikyriakidis (2017, 2019) , but the particular treatment of such phenomena is essentially independent from our treatment of time: in this paper we simply ignore these aspects beyond the fact that they are handled correctly in the FraCaS testsuite, except in a few pathological cases.

Inference using Coq Logical forms are then fed to the Coq interactive theorem prover (proof assistant). Coq is based on the calculus of co-inductive constructions (Werner, 1994) We do not use any co-induction (or even induction) in this paper, relying on the pure lambda-calculus inner core of Coq. Coq is a very powerful reasoning engine that makes it fit for implementing natural language semantics. Coq also supports dependent typing and subtyping. Both concepts are instrumental in expressing NL semantics (Chatzikyriakidis and Luo, 2014). Besides, on a more practical side, it works well for the the task of NLI, when the latter is formalised as a theorem proving task: its many tactics mean that many tasks in theorem proving are trivialised. In particular, all problems of time-intervals inclusion, which occur in every temporal problems, are solved with Coq's linear arithmetic tactic.

3 Our Treatment of Time

In montagovian semantics, (intransitive) verbs are one-place predicates; in types, they are functions from entities to propositions ($e \to t$). Our basic approach is to generalise the interpretation of verbs, so that it takes two additional time parameters, one corresponding to the starting time of the action and one corresponding to its stopping time ($(e \times time \times time) \to t$). For example, if John walked between t_0 and t_1, we would have: $walk(john, t_0, t_1)$. From now on we will call an interval of time points $[t_0, t_1]$ a timespan, where t_0 and t_1 are elements of the $time$ type, which is represented in Coq as an abstract ordered ring. Every timespan $[t_0, t_1]$ has the property $t_0 \leq t_1$: it starts no later than it stops. (We are thus using a simple Newtonian model of time, corresponding to a layman intuition of a linear constant flow of time.)

In principle, common nouns and adjectives should undergo the same procedure. For simplicity we will however only consider verbs from now on. (In fact, even in our implementation we chose not to extend nouns nor adjectives with timespan parameters. This choice limits the increase in complexity of the formulas compared to non-temporal semantics, at the expense of inaccuracy for a couple of problems in the FraCaS test suite: problems 271 and 272 use a an adjective as a copula which is subject to temporal reasoning.)

(271) **A** unknown
P1 Smith was present.
P2 Jones was present.
P3 Smith was present after Jones was present.
H Jones was present before Smith was present.

Temporal Context We adjust the montagovian semantics so that the interpretation of every category (propositions, verb phrases, etc.) takes a *temporal context* as an additional parameter, which serves as a time reference for the interpretation of all time-dependent semantics within the phrase. (While some categories do not need this temporal context, we pass it everywhere for consistency.) This context propagates through the compositional interpretation down to lexical items with atomic representation (verbs). By default, every interpretation passes the temporal context down to its components without changing it. However some key elements will act on it on nontrivial ways, which we proceed to detail below.

This temporal context is an *optional* timespan. That is, it can be a timespan or an explicitly unspecified context. The timespan in the context is optional because, in certain situations, the semantics is different depending on whether a timespan has been specified externally or not, as we explain below. A non-present timespan will be represented as $-$. If a semantic function does not depend on the temporal context at all, we will write $*$ instead.

Tenses The principal non-trivial manipulators of timespans are tense markers. In our syntax, inherited from GF, tenses are represented syntactically as an attribute of clauses. An illustration of a past-tense clause and its interpretation follows in Example (1). Notice in particular the $past$ argument to the $useCl$ constructor.

(1) A scandinavian won the nobel prize.

$$\begin{array}{l} useCl\, past\, pPos \\ (predVP\, (detCN\, (detQuant\, indefArt\, numSg) \\ \qquad scandinavian\_CN) \\ \quad (complSlash\, (slashV2a\, win\_V2) \\ \quad (detCN\, (detQuant\, indefArt\, numSg) \\ \qquad nobel\_prize\_CN))) \end{array}$$

In our semantics we deal only with present and past tenses (simple and continuous). Indeed we find that FraCas does not exercise additional specific tenses. (When a more complicated tense is used, the additional information is also carried by adverbs or adverbial phrases, in a more specific way). While we believe that many other tenses can be captured under the same general framework, we leave a detailed study to further work.

Even though we discuss a refinement to handle the past continuous at the end of this section, the procedure to handle tense annotations is as follows:

- If the tense is the past, and the temporal context is unspecified, then we locally quantify over a time interval $[t_0, t_1]$, such that $t_1 < now$, where now is a logical constant representing the current timepoint. The temporal context then becomes this interval.
- If the tense is the present and the temporal context is unspecified, then the temporal context becomes the simple (now, now) interval.
- If the temporal context is specified (for example due to the presence of an adverb or an adverbial clause, such as "before James swam"),

then the tense does not create a new interval, but it may constrain it. Typically, a past tense adds the constraint that the temporal context ends before the timepoint now.

Temporal Adverbs The other single most important source of interesting timespans are adverbs. Most of the temporal adverbs fall in either of the following categories:

exact For such adverbs, an exact interval is provided. In fact, such adverbs typically specify a single point in time (so the start and the end of the interval coincide).

$$[\![\text{at 5 pm}, s]\!](*) = [\![s]\!](5pm, 5pm)$$

existentially quantifying The majority of temporal adverbs existentially quantify over a timespan. Examples include "since 1991", "in 1996", "for two years", etc. The common theme is to introduce the interval and then restrict its bounds or its duration in some way. Sometimes the restriction is an equality, as in "for exactly two hours". In the following example we show the inclusion constraint, for "in 1992".

$$[\![\text{in 1992}, s]\!](*) = \exists t_1, t_2.[t_1, t_2] \subseteq 1992, [\![s]\!](t_1, t_2)$$

In the FraCaS test suite, we normally do not find several time-modifying adverbs modifying a single verb phrase. Indeed, sentences such as "in 1992, in 1991 john wrote a novel" are infelicitous. This justifies ignoring the input timespan in the above interpretation – we are in particular not interested in modelling felicity with our semantics, only giving an accurate semantics when the input is felicitous.

universally quantifying A few adverbs introduce intervals via a universal quantification (sometimes with a constraint). Examples include "always" and "never".

If there is no explicit time context, then "always" has no constraint on the interval, otherwise the quantified interval must be included in it:

$$[\![\text{always } s]\!](t_0, t_1) = \forall t'_0, t'_1.[t'_0, t'_1] \subseteq [t_0, t_1], [\![s]\!](t'_0, t'_1)$$

Note that here we *do* use the input interval, resulting in a correct interpretation for phrases such as "In 1994, Itel was always on time." .

Aside: aspectual classes in the literature In this paper we borrow several notions from classical temporal semantics such as "stative", "achievement", "activity", etc., even though our definitions do not perfectly match the classical ones. We explain our precise meaning for these terms in the body of the paper. Nevertheless, we refer the reader to Steedman (2000) for an extensive review of formal temporal semantics.

For the *cognoscenti*, we can already point out some differences in terminology: we use the term activity as a general term which encompasses the three classical notions of activites, achievements and accomplishments. Indeed, insofar as the test suite is concerned, we find that these three categories can be collapsed into a single one (they are subject to Eq. (1)). That is, it is sufficient for the testsuite to distinguish between events and states. (In this paper, we always assume that the problems in the FraCaS testsuite are correctly annotated.)

Time references and aspectual classes A common theme in the testsuite is the reference to previous occurrences of an event:

(262) **P1** Smith left after Jones left.
P2 Jones left after Anderson left.
H Did Smith leave after Anderson left?

To be able to conclude that there is entailment, as the testsuite expects, we have to make sure that the two occurrences of "Jones left" (in **P1** and **P2**) refer to the same time intervals. For this purpose we postulate *unicity of action* for certain time-dependent propositions:

$$\mathsf{unicity}_P : P(t_1, t_2) \to P(t_3, t_4) \to (t_1 = t_3) \wedge (t_2 = t_4) \quad (1)$$

Unicity of action holds only if the aspectual class of the proposition P is *activity* (Steedman, 2000) (which, for our purposes, includes *achievements* and *accomplishments* as well).

(The difference between activity and accomplishments on the one hand and achievement on the other hand is that for the latter, time intervals can be assumed to be of nil duration. In reality, this is an oversimplification as achievements are usually of short duration, but not nil. However, this plays little role in our analysis. As far as we can tell the FraCaS test suite does exercise temporal semantics to such a level of precision.)

Unicity of action plays the role of event coreference in (neo-)Davidsonian accounts (Parsons,

1990). It is also a fine-grained principle, allowing coreference to take into account certain arguments when referencing. As we detail below, taking arguments into account yields is critical to handle repeatability of achievements.

Unicity of action appears to be a non-logical principle. Indeed, it is quite possible that "Jones left" several times. However, it seems that this principle is never contradicted by the testsuite. As such, even though unicity of action is only a pragmatic rule, it can be taken as a valid one *by default*: it is only when we have a sufficiently constrained situation that one should reject it. Consider the following discourse:

(1) Smith left at 1pm.
(2) Smith went to his appointment with the lawyer.
(3) Smith left at 4pm.

One would normally not say that there is contradiction. However if the middle sentence were not present, a contradiction should be flagged. We leave such discourse analysis as future work, and simply apply unicity of action everywhere: it is valid uniformly in the FraCaS test suite for activity aspect classes.

Statives *A contrario*, if P is *stative*, then we get a time-interval subsumption property:

$$\mathsf{subsumption}_P : [t_3, t_4] \subseteq [t_1, t_2] \rightarrow P(t_1, t_2) \rightarrow P(t_3, t_4)$$

This principle is used to reason about problem (314), below (note that "Smith" is used as a surname in the FraCaS and can take both feminine and masculine values):

(314) **P1** Smith arrived in Paris on the 5th of May, 1995.
P2 Today is the 15th of May, 1995.
P3 She is still in Paris.
H Smith was in Paris on the 7th of May, 1995.

Indeed, from **P3** we get that Smith was in Paris between May 5th and May 15th. Because "being in Paris" is stative, we also get that Smith was in Paris in any sub-interval. Contrary to unicity of action, subsumption is always valid.

Class-modifying adverbs It should be noted that some adverbs can locally disable the application of subsumption. For example, problem 299 features the sentence "Smith lived in Birmingham for exactly a year". Even though "live" is normally stative, one can no longer apply subsumption in the context of "exactly a year" — this can be done by propagating another context flag in the montagovian semantics (in addition to the temporal context).

(Un)repeatable Achievements The principle of using unicity of action interacts well with the usual interpretation of existential quantifiers (and anaphora). Indeed, using it, we can refute problem (279), as expected by the testsuite:

(279) **P1** Smith wrote a novel in 1991.
H Smith wrote it in 1992.

Indeed, following our account, the above (contradictory) inference problem is to be interpreted as

$$\begin{array}{l}\forall x.novel(x) \wedge \\ \exists t_1, t_2.[t_1, t_2] \subseteq 1991 \wedge write(smith, x, t_1, t_2) \wedge \\ \exists t_3, t_4.[t_3, t_4] \subseteq 1992 \wedge write(smith, x, t_3, t_4) \\ \longrightarrow \bot\end{array} \quad (2)$$

Note here that the scope for the existential is extended beyond the scope of **P1**, and its polarity switched (to universal). This extension can follow the account of Unger (2011), and our implemented analysis of anaphora(Bernardy et al., 2020; Bernardy and Chatzikyriakidis, 2019).

Thanks to the unicity of action of $write(smith, x, ...)$ (the subject and direct object are fixed) we find $[t_1, t_2] = [t_3, t_4]$, and due to the years 1991 and 1992 being disjoint we obtain contradiction. In sum, no special notion of accomplishment is needs to be invoked: we only need the principle of unicity of action.

Yet, the testsuite instructs that we should *not* be able to refute problem (280), with the justification that "wrote a novel" is a repeatable accomplishment:

(280) **P1** Smith wrote a novel in 1991.
H Smith wrote a novel in 1992.

Here our interpretation is:

$$(\exists x.novel(x) \wedge$$
$$\exists t_1, t_2.[t_1, t_2] \subseteq 1991 \wedge write(smith, x, t_1, t_2)) \wedge$$
$$(\exists y.novel(y) \wedge$$
$$\exists t_3, t_4.[t_3, t_4] \subseteq 1992 \wedge write(smith, y, t_3, t_4))$$
$$\longrightarrow \bot$$

Our analysis does not need to treat this last case specially. Indeed, even if $write(smith, x, ., .)$ is an activity and thus subject to unicity of action, in (280), x is quantified existentially; we have two *different* actions: $write(smith, x, t_1, t_2)$ and $write(smith, y, t_3, t_4)$, because $x \neq y$, and thus we cannot deduce equality of the intervals t_1, t_2 and t_3, t_4. In turn, the hypothesis cannot be refuted.

Action-modification Verbs The final class of lexemes carrying a temporal-dependent semantics are verbs taking a proposition as argument, like "finish", "start", etc. These verbs modify the temporal context in non-trivial ways. Consider for example "finish to ...". The timespan of the argument of "finish" should end within the timespan of the finishing action:

$$[\![\text{finish to } s]\!](t_0, t_1) = \exists (t'_0, t'_1).t'_1 \in [t_0, t_1] \wedge [\![s]\!](t'_0, t'_1)$$

Progressive Aspect We treat verbs in the progressive form as different semantically from the non-progressive form. For example, "John was writing a book" is encoded as $\exists (t_1, t_2).t_1 \leq t_2, t_2 \leq now, PROG\_write(John, book, t_1, t_2)$, while "John wrote a book" is encoded as $\exists (t_1, t_2).t_1 \leq t_2, t_2 \leq now, write(John, book, t_1, t_2)$. This distinction is necessary because in our analysis the progressive form ($PROG\_write$) is subject to subsumption. That is, if John is writing in the interval $[t_1, t_2]$ then he is writing in any sub-interval of $[t_1, t_2]$. This interpretation corresponds to the idea that the action takes place continuously over the whole interval. However, the same cannot be said of the non-continuous form ($write$): the end-points of the interval indicate the time needed to complete the achievement. (For example, "John wrote a book in 1993" neither entails "John wrote a book in January 1993" nor "John wrote a book in December 1993".) (In fact, *write*, in the non-progressive from, is on the contrary subject to unicity.) Finally, we also have $write(x, y, t_1, t_2) \rightarrow PROG\_write(x, y, t_1, t_2)$. That is, the achievement (or *activity* in our terminology) variant implies the stative variant, for the same interval. Consequently we get the entailment from "John wrote a book in 1993" to "John was writing a book in 1993", but not the other way around.

We note however that this interpretation differs only slightly from the usual accounts of the progressive in the literature. Ogihara (2007) summarises the position of Bennett and Partee (1978) as follows: a progressive sentence is true at an interval $[t_0, t_1]$ iff there is an interval $[t'_0, t'_1]$ such that $[t_0, t_1]$ is a non-final subinterval of $[t'_0, t'_1]$ and the progressive sentence is true at $[t'_0, t'_1]$. This is very similar to our approach (subsumption for the progressive form only), but there is a difference regarding final intervals. Yet in our view this difference is hard to justify: we cannot see why "John was writing a book in 1993" entails that he was writing it January, February, etc. but not in December.

Ogihara (2007) argues that this simple account of the progressive fails to reject a sentence such as "Lee is resembling Terri." while "Lee is walking" is acceptable. We argue instead that the latter should be rejected for pragmatic reasons. Indeed, when a predicate holds for a very long interval, one typically uses the simple present tense in English. Therefore the continuous form pragmatically implies that the predicate holds for a limited interval. But, without further context, the predicate "resemble Terri" does not vary over time (while "walk" generally does). Therefore the continuous form "Lee is resembling Terri" is confusing: one implies a limited interval, but the semantics of resembling normally yield an unlimited interval. Because we do not account for pragmatics, we prefer to retain the simplest account based on the subinterval property (which we call subsumption here).

Finally we stress that not all verbs are subject to the stative/achievement distinction induced by the progressive. For example, the phrases "John ran" and "John was running" appear to be logically equivalent, for entailment purposes.

4 Worked out example

To give a sense of the additional details necessary to deal with the precision demanded by a proof-assistant such as Coq we show how problem (279) is worked out in full details.

We start with input trees in GF format, given by Ljunglöf and Siverbo (2011). They can be rendered

as follows:

```
s_279_1_p=
sentence
(useCl past pPos
 (predVP
   (usePN (lexemePN "smith_PN"))
   (advVP
     (complSlash
      (slashV2a (lexemeV2 "write_V2")))
     (detCN (detQuant indefArt numSg)
       (useN (lexemeN "novel_N"))))
     (lexemeAdv "in_1991_Adv")))
s_279_3_h=
sentence
(useCl past pPos
 (predVP (usePN
   (lexemePN "smith_PN"))
   (advVP
     (complSlash
      (slashV2a (lexemeV2 "write_V2"))
      (usePron it_Pron))
     (lexemeAdv "in_1992_Adv"))))
```

Of particular note is the use of the pronoun "it", and the fact that adverbial expressions such that "in 1992" are lexicalized. We also follow the GF convention to postfix lexical items with the name of their category. Most of the other categories follow usual naming conventions. We remind the reader that "slash" categories are used to swap the order of arguments (compared to non-slashed categories of similar names).

Our dynamic and temporal semantics gives the following interpretation for `s_279_1_p` implies `s_279_3_h`.

```
FORALL (fun a=>novel_N a)
(fun a=>(exists (b: Time),
((exists (c: Time),
(IS_INTERVAL Date_19910101 b /\
 IS_INTERVAL c Date_19911231 /\
 IS_INTERVAL b c /\
 appTime b c (write_V2 a)
   (PN2object smith_PN))))) ->
Not (exists (f: Time),
((exists (g: Time),
 (IS_INTERVAL Date_19920101 f /\
  IS_INTERVAL g Date_19921231 /\
  IS_INTERVAL f g /\
  appTime f g (write_V2 a)
    (PN2object smith_PN)))))).
```

In the above, one should remark the top-level quantification over the novel (as explained in Section 3), the quantification over time intervals as individual timepoints, and the use of custom operators for several constructions (`FORALL`, `Not`, `IS_INTERVAL`, `appTime`). This use of custom operators is useful for several generalisations (for example, we have quantifiers such as `MOST` in addition to `FORALL` — see Bernardy and Chatzikyriakidis (2017))

Unfolding the definitions for these operators yield the following proposition:

```
forall x : object,
novel_N x ->
(exists b c : Z,
   Date_19910101 <= b /\
   c <= Date_19911231 /\
   b <= c /\ write_V2 x SMITH b c) ->
(exists f g : Z,
   Date_19920101 <= f /\
   g <= Date_19921231 /\
   f <= g /\ write_V2 x SMITH f g) ->
False
```

This is very close to our idealised representation of the problem Eq. (2). One difference is the use of abstract Coq integers for timepoints. Using a discrete time allows us to use predefined Coq tactics. The discrete nature of integers does not interfere with the reasoning.

Finally, we can show a Coq proof for the above proposition:

```
Theorem  problem279 : Problem279aFalse.
cbv.
intros novel isSmithsNovel P1 H.
destruct P1 as
   [t0 [t1 [ct1 [ct2 [ct3 P1]]]]].
destruct H as
   [u0 [u1 [cu1 [cu2 [cu3 H]]]]].
specialize writeUnique
  with (x := novel)(y := SMITH) as A.
unfold UniqueActivity in A.
specialize (A _ _ _ _ P1 H) as B.
lia.
Qed.
```

The intros and destruct tactics serve bookkeeping purposes. The critical part is the use of the `writeUnique` axiom, which witnesses the aspectual class of the predicate `write_V2`. The proof is completed by the use of the `lia` tactic, which is embeds a decision procedure for linear arithmetic problems[2]. Fortunately, `lia` can take care of all the problems which arise in the FraCaS testsuite.

5 Results and Evaluation

Our target is the FraCaS testsuite, which aims at covering a wide range of common natural-language phenomena. The suite is structured according to the semantic phenomena involved in the inference process for each example, and contains nine sections: Quantifiers, Plurals, Anaphora, Ellipsis, Adjectives, Comparatives, Temporal, Verbs and Attitudes. The system described here focuses on the Temporal section. However, it also supports the other eight sections. To our knowledge this is the first system which attempts to target the temporal section in full. But in fact, our system even provides support for all the other sections. Thus, a couple of decades

[2]It solves linear goals over rings by searching for linear refutations and cutting planes

Section	#FraCaS	This	FC2	FC	MINE	Nut	LP
Quantifiers	75	.93 74	.96 74	.96	.77	.53	.93 44
Plurals	33	.79	.82	.76	.67	.52	.73 24
Anaphora	28	.79	.86	-	-	-	-
Ellipsis	52	.81	.87	-	-	-	-
Adjectives	22	.95 20	.95 20	.95	.68	.32	.73 12
Comparatives	31	.65	.87	.56	.48	.45	-
Temporal	75	.73	-	-	-	-	-
Verbs	8	.75	.75	-	-	-	-
Attitudes	13	.85	.92	.85	.77	.46	.92 9
Total	337	.81 329	.89 259	.83 174	.69 174	.50 174	.85 89

Table 1: Accuracy of our system compared to others. "This" refers to the approach presented in this paper. When a system does not handle the nominal number of test cases (shown in the second column), the actual number of test cases attempted is shown below the accuracy figure, in smaller font. "FC" refers to the work of Bernardy and Chatzikyriakidis (2017), and "FC2" its followup (Bernardy and Chatzikyriakidis, 2019). "MINE" refers to the approach of Mineshima et al. (2015), "NUT" to the CCG system that utilises the first-order automated theorem prover *nutcracker* (Bos, 2008), and "LP" to the system presented by Abzianidze (2015). A dash indicates that no attempt was made for the section.

after its formulation, we propose a first attempt at covering the whole suite. As such, there it is no other system to compare our system with, in all aspects. We can however compare with systems which target parts of the FraCaS testsuite, as shown in Table 1.

Interaction with anaphora One reason explaining the lower performance of our system on some sections of the testsuite is that our interpretation of time interacts imperfectly with anaphora and ellipsis. Consider the following example:

(232) **P1** ITEL won more orders than APCOM did.
P2 APCOM won ten orders.
H ITEL won at least eleven orders.

In the first premise, our system essentially resolves the ellipsis to get the following reading: "ITEL won X orders and APCOM won Y orders and $X > Y$.". One would need each of the verb phrases "won X orders" and "won Y orders" to introduce their own timespans with existential quantifiers. However, the organisation of our system is such that the existentials are introduced before the ellipsis is expanded. Consequently we get a wrong interpretation and the inference cannot be made.

6 Conclusions and Future Work

We have presented a first attempt for a computational approach dealing with the temporal section of the FraCaS test suite. To do this, we have provided a simplified taxonomy of aspectual classes for verb phrases, guided by the applicability of the unicity of action and temporal subsumption properties. While part of this simplification is accidental (conflation of activity and accomplishment), we find that other parts (the automatic distinction between repeatable and unrepeatable achievements) constitute theoretical improvements.

Besides inference, formal interpretation of tense is found in natural-language interfaces to databases. Of note is the work of Androutsopoulos et al. (1998), which handles many of the time-aware adverbial clauses that we address. However, we cover many more logical aspects of inference, such as coreference via unity of action and interaction with quantifiers.

Bernardy and Chatzikyriakidis (2019) presented a logical system for handling 8 of the 9 sections of the FraCaS test suite, but excluded section 7, suggesting that it requires many examples that need an *ad hoc* treatment. Here, we took up this challenge and have shown that a system similar to theirs can be extended to cover the remaining section of the test suite, without considerably decreasing the performance of the rest of the sections. This is indeed a common problem with logical approaches, namely the fact that one can have theoretically motivated implementations of individual phenomena, e.g. anaphora, ellipsis, quantifiers, temporal reference etc., but when one tries to put all these together into a unified system, this proves to be a daunting task. We believe that this paper presents an exception, and provides a system that can deal with all these different semantic phenomena under a unified system with very good results. We use the same combination of a number of well-studied tools as Bernardy and Chatzikyriakidis (2019) : type theory, parsing using the Grammatical Framework (GF), Monadic Dynamic Semantics and proof assistant technology (Coq). The system achieves an accuracy of 0.73 on the Temporal Section and 0.81 overall. The whole system, including data sets, is available at the following url: `https://github.com/GU-CLASP/FraCoq/tree/iwcs2021`.

One of the things to be looked at is fixing the issues associated with parts of the test suite that "broke" when the temporal analysis was introduced. Some of these have been already mentioned: interaction of the temporal variables with anaphora.

Another extension of this work is to reflect more temporal semantic inference properties in an extended test suite. Indeed, there as properties which are not captured in the FraCaS test suite, such as fine-grained examples of lexical and grammatical aspect, as well as the interaction between those two, for example cases where one needs to actually distinguish between achievements and accomplishments on the basis of their inferential properties:

(∗1) **P1** John found his keys.
H John was finding his keys (UNK).

(∗2) **P1** John wrote a book.
H John was writing a book (YES).

In the first of the two examples involving an achievement verb, the inference is UNK, since there is no guarantee that the action is non-instantaneous. To the contrary, for accomplishment verbs, the inference follows.

Further cases to be included in an extended FraCaS future suite involve examples where the interaction between different tenses needs to be captured:[3]

(∗3) **P1** When the phone rang, John had entered the house.
H John entered the house before the phone rang (YES).

Finally it would be desirable to improve automation of the system, and evaluate it on a larger test set. As it stands Coq fully *checks* the proof of entailment for each (provable) problem. However, the construction of such proofs has demanded human intervention. It would be desirable to fully automate the proof construction step. For this to make sense however we need a much larger test suite, properly separated into a development and a (secret) test set. Otherwise, only the limited power of the logic prevents us (or any followup work) to fine-tune the rules of the system until one gets full coverage. This kind of observation holds in general of any rule-based system, and thus applies not only to the proof-construction phase, but also to the underlying dynamic semantics and parsing phase (which is limited only by the power of the language and frameworks used for its implementation). In sum, contrary to statistical approaches to language understanding, the value of the present work lies not in the bare accuracy numbers which we are able to achieve, but in the details of *how* we do so: the of set of rules which we use, which is described in detail here and in the work which we base ourselves upon (Bernardy et al., 2020; Bernardy and Chatzikyriakidis, 2019).

[3]While this work was completed, the work by (Vashishtha et al., 2020) was published. The authors present a five datasets to be used for the training of neural models' ability to capture temporal reasoning. It would be interesting to check the amount of data covered, most specifiaclly the level of fine-grainedness of temporal reasoning needed to capture those examples, as compared to what we have been discussing in this paper. We thank an anonymous reviewer for bringing this work to our attention.

Acknowledgements

The research reported in this paper was supported by grant 2014-39 from the Swedish Research Council, which funds the Centre for Linguistic Theory and Studies in Probability (CLASP) in the Department of Philosophy, Linguistics, and Theory of Science at the University of Gothenburg. We are grateful to our colleagues in CLASP for helpful discussion of some of the ideas presented here. We also thank anonymous reviewers for their useful comments on an earlier draft of the paper.

References

Lasha Abzianidze. 2015. A tableau prover for natural logic and language. In *Proceedings of EMNLP15*.

Ion Androutsopoulos, Graeme D Ritchie, and Peter Thanisch. 1998. Time, tense and aspect in natural language database interfaces. *arXiv preprint cmp-lg/9803002*.

Michael Bennett and Barbara Hall Partee. 1978. *Toward the logic of tense and aspect in English*, volume 84. Indiana University Linguistics Club Bloomington.

Jean-Philippe Bernardy and Stergios Chatzikyriakidis. 2017. A type-theoretical system for the fracas test suite: Grammatical framework meets coq. In *IWCS 2017-12th International Conference on Computational Semantics-Long papers*.

Jean-Philippe Bernardy and Stergios Chatzikyriakidis. 2019. A wide-coverage symbolic natural language inference system. In *Proceedings of the 22nd Nordic Conference on Computational Linguistics*. ACL.

Jean-Philippe Bernardy, Stergios Chatzikyriakidis, and Aleksandre Maskharashvili. 2020. A computational

treatment of anaphora and its algorithmic implementation: Extended version. Available on the first author's homepage: https://jyp.github.io/pdf/phoroi.pdf or online https://bit.ly/2xQ4G2M.

Johan Bos. 2008. Wide-coverage semantic analysis with boxer. In *Semantics in text processing. Step 2008 conference proceedings*, pages 277–286.

Simon Charlow. 2015. Monadic dynamic semantics for anaphora. *Ohio State Dynamic Semantics Workshop*.

Simon Charlow. 2017. A modular theory of pronouns and binding. In *Logic and Engineering of Natural Language Semantics (LENLS) 14*. Springer.

Stergios Chatzikyriakidis and Zhaohui Luo. 2014. Natural language inference in coq. *Journal of Logic, Language and Information*, 23(4):441–480.

R. Cooper, D. Crouch, J. van Eijck, C. Fox, J. van Genabith, J. Jaspars, H. Kamp, D. Milward, M. Pinkal, M. Poesio, and S. Pulman. 1996. Using the framework. Technical report LRE 62-051r, The FraCaS consortium. ftp://ftp.cogsci.ed.ac.uk/pub/FRACAS/del16.ps.gz.

David R Dowty. 2012. *Word meaning and Montague grammar: The semantics of verbs and times in generative semantics and in Montague's PTQ*, volume 7. Springer Science & Business Media.

Tim Fernando. 2015. The semantics of tense and aspect. *The Handbook of Contemporary Semantic Theory*, pages 203–236.

James Higginbotham. 2009. *Tense, aspect, and indexicality*, volume 26. OUP Oxford.

P. Ljunglöf and M. Siverbo. 2011. A bilingual treebank for the FraCas test suite. Clt project report, University of Gothenburg.

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

Koji Mineshima, Yusuke Miyao, and Daisuke Bekki. 2015. Higher-order logical inference with compositional semantics. In *Proceedings of EMNLP*.

Richard Montague. 1970. English as a formal language. In *Linguaggi nella Societa e nella Tecnica*.

Richard Montague. 1973. The proper treatment of quantification in ordinary english. In *Approaches to natural language*, pages 221–242. Springer.

Richard Montague. 1974. The proper treatment of quantification in ordinary english. In Richmond Thomason, editor, *Formal Philosophy*. Yale UP, New Haven.

Toshiyuki Ogihara. 2007. Tense and aspect in truth-conditional semantics. *Lingua*, 117(2):392–418.

Terence Parsons. 1990. *Events in the Semantics of English*, volume 5. MIT press Cambridge, MA.

Arthur N Prior and Per FV Hasle. 2003. *Papers on time and tense*. Oxford University Press on Demand.

Aarne Ranta. 2004. Grammatical framework. *Journal of Functional Programming*, 14(2):145–189.

Mark Steedman. 2000. The productions of time. *Draft. Available at http://www. cogsci. ed. ac. uk/steedman/papers. html*.

Christina Unger. 2011. Dynamic semantics as monadic computation. In *JSAI International Symposium on Artificial Intelligence*, pages 68–81. Springer.

Siddharth Vashishtha, Adam Poliak, Yash Kumar Lal, Benjamin Van Durme, and Aaron Steven White. 2020. Temporal reasoning in natural language inference. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4070–4078, Online. Association for Computational Linguistics.

Benjamin Werner. 1994. *Une théorie des constructions inductives*. PhD thesis, Université de Paris 7.