Temporal Meaning Representations in a Natural Language Front-End

Ion Androutsopoulos

Software and Knowledge Engineering Laboratory
Institute of Informatics and Telecommunications
National Centre for Scientific Research "Demokritos"
153 10 Ag. Paraskevi, Athens, Greece
e-mail: ionandr@iit.demokritos.gr

Abstract

Previous work in the context of natural language querying of temporal databases has established a method to map automatically from a large subset of English time-related questions to suitable expressions of a temporal logic-like language, called TOP. An algorithm to translate from TOP to the TSQL2 temporal database language has also been defined. This paper shows how TOP expressions could be translated into a simpler logic-like language, called BOT. BOT is very close to traditional first-order predicate logic (FOPL), and hence existing methods to manipulate FOPL expressions can be exploited to interface to time-sensitive applications other than TSQL2 databases, maintaining the existing English-to-TOP mapping.

1 Introduction

Time is an important research issue in linguistics (e.g. [7], [8], [19]), logics (e.g. [12], [28]), and computer systems (e.g. temporal databases [26] [27]). In [3] and [4] a framework that integrates ideas from these three areas was proposed in the context of natural language querying of temporal databases. This framework consists of: (i) a formally defined logic-like language, dubbed TOP, (ii) a systematic mapping from a large and rich in temporal phenomena subset of English to TOP, based on the widely used HPSG grammar theory [20] [21], and (iii) an algorithm to translate from TOP to TSQL2, TSQL2 being a recent temporal extension of the SQL database language that has been proposed by the temporal databases community [24]. The framework allows written time-related English questions to be answered automatically, by converting them into TOP and then TSQL2 expressions, and executing the resulting TSQL2 queries. The framework improves on previous approaches to natural language querying of temporal databases (e.g. [6], [10]), mainly in terms of linguistic coverage, existence of formal definitions, and implementation (see [3] and [4] for details).[1]

This paper shows how TOP expressions can be translated automatically into a simpler logic-like representation language, called BOT. BOT is very close to traditional first-order predicate logic (FOPL). Hence, existing methods to manipulate FOPL expressions can be exploited, to interface to time-sensitive applications other than TSQL2 databases (e.g. hybrids

[1]A prototype implementation of this framework is freely available from http://www.dai.ed.ac.uk/groups/nlp.

of standard SQL and Prolog databases [5] [11] [15], or planners [9]), maintaining the existing English-to-TOP linguistic front-end. The mapping from TOP to BOT is also expected to make the linguistic front-end easier to interface to forthcoming new temporal SQL versions [25], as BOT is much simpler than TOP, and hence establishing a mapping from BOT to a new database language is easier than from TOP.

This paper focuses on TOP, BOT, and the mapping from TOP to BOT. Information about other aspects of the work mentioned above, including the English-to-TOP mapping, can be found in [3] and [4]. The remainder of this paper is organised as follows: Section 2 introduces TOP, Section 3 presents BOT, Section 4 describes the TOP-to-BOT mapping, and Section 5 concludes. Formal definitions of TOP and BOT can be found in Appendices A and B respectively. Appendix C provides a full list of the translation rules that are used in the TOP-to-BOT mapping.

2 The TOP language

This section introduces informally TOP, the logic-like language English questions are initially translated into. A formal definition of the syntax and semantics of TOP can be found in Appendix A. TOP was designed to support the systematic representation of English time-related semantics, rather than inferencing (contrary to the logics of e.g. [1], [14], [16], [17]). Hence, although in many ways similar to traditional temporal logics, TOP is not a full logic, as it provides no proof theory.

TOP atomic formulae are constructed by applying predicate symbols to constants and variables. More complex formulae are constructed using conjunctions and temporal operators (TOP is an acronym for Temporal OPerators). For example, (1) is represented by the TOP formula (2). The "^{v}" suffix marks variables, and free variables (e.g. the e^v in (2)) are treated as quantified by an implicit existential quantifier with scope over the entire formula. TOP currently provides no disjunction, negation, or explicit quantification mechanisms, as these were not needed for the linguistic phenomena that the work being reported here focused on. Such mechanisms can be added easily in future TOP versions.

(1) Was tank 5 empty on 1/1/98?

(2) $At[1/1/98, Past[e^v, empty(tank5)]]$

Roughly speaking, the verb tense in (1) introduces a *Past* operator, which requires $empty(tank5)$ to have been empty at some past time e^v, and the *"at 1/1/98"* adverbial introduces an *At* operator, which requires that past time to fall within 1/1/98. (Unlike Priorean operators [22], TOP's *Past* and *At* operators do not shift the time where their argument is expected to hold. They simply accumulate restrictions on what this time can be. This is explained further below.) Assuming that 1/1/98 falls in the past, the *Past* operator of (2) is actually redundant, since any time that falls within 1/1/98 will also belong to the past. It is important to realise, however, that the mapping from English to TOP is carried out automatically. This mapping introduces a *Past* operator when encountering the past tense, to ensure that a sentence like (3), where no adverbial is present, is represented correctly (as in (4)).

(3) Tank 5 was empty.

(4) $Past[e^v, empty(tank5)]]$

The combination of *At* and *Past* operators in (2) also accounts for the oddity of (1) when uttered before 1/1/98. The oddity of (1) can be attributed to the fact that in this case the *At* and *Past* operators introduce incompatible restrictions (no past time can fall within 1/1/98 if the question is uttered before 1/1/98).

Temporal operators are used in TOP (much as in [9]) to introduce compact chunks of semantics, in a manner that makes it easier to track the semantic contribution of each linguistic mechanism. No claim is made that TOP is more expressive than (or even as expressive as) other temporal representation formalisms (e.g. [22]), though it should be noted that TOP is part of a complete path from English to an application formalism (TSQL2), which is not available with most other temporal representation formalisms.

Time in TOP is linear, discrete and bounded [12] [28]. Following Reichenbach [23], formulae are evaluated with respect to three times: *speech time* (st), *event time* (et), and *localisation time* (lt). Intuitively, st is the time when the question is submitted, et is the time when the situation of the formula holds, and lt is a temporal window that contains et. (TOP's lt is different from Reichenbach's reference time, and closer to the "location time" of [13].) st is a time-point, while et and lt are generally periods. "Period" is used here to refer to what logicians usually call "intervals", i.e. convex sets of time-points.

In (1), the answer will be affirmative if (2) evaluates to true. When evaluating a formula, lt initially covers the entire time, but it can be narrowed down by temporal operators. In (2), the *At* and *Past* operators narrow lt to its intersection with 1/1/98 and $[t_{first}, st)$ respectively, where t_{first} is the beginning of time. Assuming that 1/1/98 lies entirely in the past, the resulting lt is 1/1/98. The formula evaluates to true iff there is an et where $empty(tank5)$ is true, and $et \sqsubseteq lt$. (p_1 is a subperiod of p_2, written $p_1 \sqsubseteq p_2$ iff p_1, p_2 are periods and $p_1 \subseteq p_2$.)

The semantics of TOP guarantee that TOP predicates are always *homogeneous*, meaning that if a predicate is true at some et, it will also be true at any other $et' \sqsubseteq et$. (A similar notion of homogeneity is used in [2].) In (2), if tank 5 was empty from 30/12/97 to 10/1/98 (dates are shown in the dd/mm/yy format), $empty(tank5)$ will be true at any et that is a subperiod of that period. Hence, there will be an et that is a subperiod of 1/1/98 (the lt) where $empty(tank5)$ holds, and (2) will evaluate to true. The reading of (1) that requires the tank to have been empty *throughout* 1/1/98, which is easier to grasp in the affirmative (5), is expressed as (6). The *Fills* operator requires et to cover the entire lt.

(5) Tank 5 was empty on 1/1/98.

(6) $At[1/1/98, Past[e^v, Fills[empty(tank5)]]]$

The remainder of this section illustrates the use of some of TOP's temporal operators, narrowing the discussion to the representation of yes/no single-clause questions. To save space, some of TOP's mechanisms, including those that are used to represent wh-questions (e.g. *"Which tanks were empty on 1/1/98?"*) and multiple clauses (e.g. *"Which flights were circling while BA737 was landing?"*), are not covered (see [3] and [4] for the full details).

With verbs that refer to situations with inherent climaxes [18] [29], non-progressive tenses introduce an additional *Culm* operator, which requires et to be the period from the point where the situation first started to the point where the situation last stopped, and the situation to reach its climax at the end of et. For example, (7) and (9) are mapped to (8) and (10) respectively. (10) requires the building to have been completed, and the entire building to have taken place within 1997. In contrast, (8) simply requires part of the bulding to have been ongoing in 1997.

(7) Was HouseCorp building bridge 2 in 1997?

(8) $At[1997, Past[e^v, buiding(housecorp, bridge2)]]$

(9) Did HouseCorp build bridge 2 in 1997?

(10) $At[1997, Past[e^v, Culm[building(housecorp, bridge2)]]]$

Questions referring to present situations are represented using the *Pres* operator, which simply requires *st* to fall within *et*. (11), for example, is represented as (12).

(11) Is tank 5 empty?

(12) $Pres[empty(tank5)]$

The *Perf* operator is used to express the perfective aspect of questions like (13). The *Perf* operator introduces a new event time (denoted by $e2^v$ in (14)) that must precede the original one ($e1^v$). In (14), the inspection time ($e2^v$) must precede another past time ($e1^v$). The latter corresponds to Reichenbach's *reference time* [23], a time that serves as a view-point.

(13) Had J. Adams inspected BA737?

(14) $Past[e1^v, Perf[e2^v, Culm[inspecting(jadams, ba737)]]]$

In (15), the *"on 1/1/95"* may refer to either the inspection time or the reference time. The two readings are represented by (16) and (17) respectively (the English to TOP mapping generates both).

(15) Had J. Adams inspected BA737 on 1/1/95?

(16) $Past[e1^v, Perf[e2^v, At[1/1/95, Culm[inspecting(jadams, ba737)]]]]$

(17) $At[1/1/95, Past[e1^v, Perf[e2^v, Culm[inspecting(jadams, ba737)]]]]$

The *Ntense* operator (borrowed from [9]) is useful in questions like (18), where *"the president"* may refer to either the present or the 1995 president. The two readings are captured by (19) and (20) respectively. In (20), the e^v arguments of the *Past* and *Ntense* operators are used to ensure that both $president(p^v)$ and $visiting(p^v, athens)$ hold at the same time.[2]

(18) Did the president visit Athens in 1995?

(19) $Ntense[now, president(p^v)] \wedge At[1995, Past[e^v, visiting(p^v, athens)]]$

(20) $Ntense[e^v, president(p^v)] \wedge At[1995, Past[e^v, visiting(p^v, athens)]]$

The reading of (21) that asks if tank 5 was empty at some time after 5:00 pm is represented as (22). The *Part* operator forces f^v to range over 5:00pm-times, and the *After* operator requires the past *et* where tank 5 is empty to follow f^v.

(21) Was tank 5 empty after 5:00pm?

[2]The e^v arguments of the *Past* and *Perf* operators are also used in time-asking questions. Consult [3] and [4] for related discussion.

(22) $\mathit{Part}[\mathit{5{:}00pm}, f^v] \wedge \mathit{After}[f^v, \mathit{Past}[e^v, \mathit{empty}(\mathit{tank}5)]]$

In practice, (21) would be uttered in a context where previous discourse has established a temporal window that contains a single 5:00pm-time, and *"at 5:00pm"* would refer to that time. This anaphoric use of *"at 5:00pm"* can be captured by setting the initial value of *lt* to the discourse-defined window, rather than the entire time-axis.

Finally, durations can be specified with the *For* operator. (23) is mapped to (24), which requires 45 consecutive minute-periods to exist, and the flight to have been circling throughout the concatenation of these periods.

(23) Was BA737 circling for 45 minutes?

(24) $\mathit{For}[\mathit{minute}, 45, \mathit{Past}[e^v, \mathit{circling}(\mathit{ba}737)]]$

3 The BOT language

Let us now turn to BOT, the simpler formal language TOP expressions are subsequently translated into. BOT is essentially the traditional first-order predicate logic (FOPL), with some special terms and predicates to refer to time-points and periods. As in TOP, BOT assumes that time is discrete, linear, and bounded.

For simplicity, the same constant and predicate symbols are used as in the corresponding TOP expressions. It is assumed, however, that BOT predicates that correspond to TOP predicates have an additional argument, whose denotations range over the maximal event-time periods where the corresponding TOP predicates hold. For example, (1) could be represented in BOT as (25) (cf. (2)).

(25) $\mathit{empty}(\mathit{tank}5, p^v) \wedge \mathit{subper}(e^v, p^v) \wedge$
$\mathit{subper}(e^v, \mathit{intersect}(\mathit{intersect}([\mathit{beg}, \mathit{end}], 1/1/98), [\mathit{beg}, \mathit{now})))$

(25) requires p^v to denote a maximal period where the tank was empty, and e^v to be a subperiod of both p^v and the intersection of 1/1/98 with the past. e^v corresponds to the event time of (2), and $\mathit{intersect}(\mathit{intersect}([\mathit{beg}, \mathit{end}], 1/1/98), [\mathit{beg}, \mathit{now}))$ emulates TOP's localisation time, initially the entire time-axis, which has been narrowed to cover past points within 1/1/98. (*beg*, *end*, and *now* denote the beginning of time, end of time, and speech time respectively, *intersect* denotes set intersection, and square and round brackets are used to specify the boundaries of periods in the usual manner.) As in TOP, free variables are treated as existentially quantified.

A special BOT predicate symbol *part*, similar to TOP's *Part* operator, allows variables to range over families of periods. In (27), for example, $m1^v$ and $m2^v$ range over minute-periods. *earliest* and *latest* are used to refer to the earliest and latest time-points of a period, *succ* steps forward one-time point, and *eq* requires the denotations of its arguments to be identical. (27) requires a 2-minute long e^v period to exist, and e^v to fall within the past and be a subperiod of a maximal period p^v where tank 5 was empty. As in (25), *intersect* predicates are used to emulate TOP's localisation time (here, $\mathit{intersect}([\mathit{beg}, \mathit{end}], [\mathit{beg}, \mathit{now}))$). (27) represents (26).

(26) Was tank 5 empty for two minutes?

(27) $\mathit{part}(\mathit{minute}, m1^v) \wedge \mathit{part}(\mathit{minute}, m2^v) \wedge \mathit{eq}(\mathit{earliest}(m1^v), \mathit{earliest}(e^v)) \wedge$
$\mathit{eq}(\mathit{succ}(\mathit{latest}(m1^v)), \mathit{earliest}(m2^v)) \wedge \mathit{eq}(\mathit{latest}(m2^v), \mathit{latest}(e^v)) \wedge$
$\mathit{empty}(\mathit{tank}5, p^v) \wedge \mathit{subper}(e^v, p^v) \wedge \mathit{subper}(e^v, \mathit{intersect}([\mathit{beg}, \mathit{end}], [\mathit{beg}, \mathit{now})))$

The semantics of BOT is much simpler than TOP, though BOT formulae tend to be much longer, and hence difficult to grasp, than the corresponding TOP ones. The syntax and semantics of BOT are defined formally in Appendix B.

4 Translating from TOP to BOT

TOP formulae are translated systematically into BOT using a set of rewrite rules of the form:

$$\mathit{trans}(\phi_1, \varepsilon, \lambda) = \phi_2$$

where ϕ_1 is a TOP formula, ϕ_2 is a BOT formula (possibly containing recursive invocations of other translation rules), and ε, λ are BOT expressions representing TOP's event and location times respectively. There are base (non-recursive) translation rules for atomic TOP formulae, and recursive translation rules for conjunctions and each one of TOP's operators. For example, the translation rule for TOP's *At* operator is (28).

(28) $\mathit{trans}(\mathit{At}[\tau, \phi], \varepsilon, \lambda) = \mathit{period}(\tau) \land \mathit{trans}(\phi, \varepsilon, \mathit{intersect}(\lambda, \tau))$

The translation rules essentially express in terms of BOT constructs the semantics of the corresponding TOP constructs. (28), for example, narrows the localisation time to the intersection of its original value with the denotation of τ, which must be a period, mirroring the semantics of TOP's *At* operator (see Appendix A). ϕ is then translated into BOT using the new value of the localisation time.

When translating from TOP to BOT, λ is initially set to $[\mathit{beg}, \mathit{end}]$, which corresponds to the initial value of TOP's localisation time. ε is set to a new variable, a variable that has not been used in any other expression. This reflects the fact that TOP's event time is initially allowed to be any period (see the definition of denotation w.r.t. M, st in Appendix A). For example, to compute the BOT translation of (1), one would invoke (28) as in (29), where et^v is a new variable that stands for the event time.

(29) $\mathit{trans}(\mathit{At}[1/1/98, \mathit{Past}[e^v, \mathit{empty}(\mathit{tank}5)]], et^v, [\mathit{beg}, \mathit{end}]) =$
$\mathit{period}(1/1/98) \land \mathit{trans}(\mathit{Past}[e^v, \mathit{empty}(\mathit{tank}5)], et^v, \mathit{intersect}([\mathit{beg}, \mathit{end}], 1/1/98)$

The translation rules for TOP's *Past* operator and predicates are shown in (30) and (31) respectively. The rule for *Past* narrows the localisation time to the past, and requires β to point to the event time. (The β argument of TOP's *Past* operator is useful in time-asking questions, which are not covered in this paper.) The rule for predicates requires the event time to be a subperiod of both the localisation time and of a maximal period where the predicate holds (here β is a new variable). These are, again, in accordance with TOP's semantics. The reader is reminded that BOT predicates that correspond to predicates in the TOP formula have an additional argument (β in (31)), which ranges over the maximal event-time periods where the corresponding TOP predicate holds.

(30) $\mathit{trans}(\mathit{Past}[\beta, \phi], \varepsilon, \lambda) = \mathit{eq}(\beta, \varepsilon) \land \mathit{trans}(\phi, \varepsilon, \mathit{intersect}(\lambda, [\mathit{beg}, \mathit{now})))$

(31) $\mathit{trans}(\pi(\tau_1, \ldots, \tau_n), \varepsilon, \lambda) = \mathit{subper}(\varepsilon, \lambda) \land \pi(\tau_1, \ldots, \tau_n, \beta) \land \mathit{subper}(\varepsilon, \beta)$

Using (30) and (31), the right-hand side of (29) becomes (32). The right-hand side of (32) is the final result of the translation, which is essentially the same as the hand-crafted (25). (The additional et^v variable and *period* predicate, do not contribute significantly in this case, but they are needed to prove the correctness of the automatic translation.)

(32) $\mathit{period}(1/1/98) \land \mathit{eq}(e^v, et^v) \ \land$
$\mathit{trans}(\mathit{empty}(\mathit{tank}5), et^v, \mathit{intersect}(\mathit{intersect}([\mathit{beg}, \mathit{end}], 1/1/98), [\mathit{beg}, \mathit{now}))) =$
$\mathit{period}(1/1/98) \land \mathit{eq}(e^v, et^v) \ \land$
$\mathit{subper}(et^v, \mathit{intersect}(\mathit{intersect}([\mathit{beg}, \mathit{end}], 1/1/98), [\mathit{beg}, \mathit{now}))) \ \land$
$\mathit{empty}(\mathit{tank}5, p^v) \land \mathit{subper}(et^v, p^v)$

As explained in section 2, TOP's *Culm* operator requires the event time to be the period from the point where the situation described by *Culm*'s argument first started to the point where it last stopped, and the situation to reach its climax at the end of the event time. To be able to translate TOP formulae containing *Culm* operators, we assume two mappings η_1 and η_2 from predicate functors to new (unused elsewhere) predicate functors. If $\pi(\tau_1, \ldots, \tau_n)$ is a predicate in a TOP formula, $\eta_1(\pi)(\tau_1, \ldots, \tau_n)$ is a BOT predicate intended to be true if the situation of $\pi(\tau_1, \ldots, \tau_n)$ reached its climax at the point where it last stopped. $\eta_2(\pi)(\tau_1, \ldots, \tau_n, \varepsilon)$ is another BOT predicate, where ε denotes the period from the first to the last point where the situation of $\pi(\tau_1, \ldots, \tau_n)$ was ongoing. The translation rule for *Culm* is (33).

(33) $\mathit{trans}(\mathit{Culm}[\pi(\tau_1, \ldots, \tau_n)], \varepsilon, \lambda) = \mathit{subper}(\varepsilon, \lambda) \land \eta_1(\pi)(\tau_1, \ldots, \tau_n) \land \eta_2(\pi)(\tau_1, \ldots, \tau_n, \varepsilon)$

Using (28), (30), (33), and assuming $\eta_1(\mathit{building}) = \mathit{cmp\_building}$ and $\eta_2(\mathit{building}) = \mathit{max\_building}$, the TOP formula of (10) (which represents (9)) is translated into BOT as in (34).

(34) $\mathit{trans}(\mathit{At}[1997, \mathit{Past}[e^v, \mathit{Culm}[\mathit{building}(\mathit{housecorp}, \mathit{bridge}2)]]], et^v, [\mathit{beg}, \mathit{end}]) =$
$\mathit{period}(1997) \ \land$
$\mathit{trans}(\mathit{Past}[e^v, \mathit{Culm}[\mathit{building}(\mathit{housecorp}, \mathit{bridge}2)]], et^v, \mathit{intersect}([\mathit{beg}, \mathit{end}], 1997)) =$
$\mathit{period}(1997) \land \mathit{eq}(e^v, et^v) \ \land$
$\mathit{trans}(\ \mathit{Culm}[\mathit{building}(\mathit{housecorp}, \mathit{bridge}2)], et^v,$
$\quad \mathit{intersect}(\mathit{intersect}([\mathit{beg}, \mathit{end}], 1997), [\mathit{beg}, \mathit{now}))) =$
$\mathit{period}(1997) \land \mathit{eq}(e^v, et^v) \ \land$
$\mathit{subper}(et^v, \mathit{intersect}(\mathit{intersect}([\mathit{beg}, \mathit{end}], 1997), [\mathit{beg}, \mathit{now}))) \ \land$
$\mathit{cmp\_building}(\mathit{housecorp}, \mathit{bridge}2) \land \mathit{max\_building}(\mathit{housecorp}, \mathit{bridge}2)$

Like (10), (34) requires the period from the point where the building first started to the point where it last stopped to fall completely within the past and within 1997. Furthermore, the building must have reached its completion at the end of that period.

A complete list of the TOP to BOT translation rules can be found in Appendix C. Although beyond the scope of this paper, a formal proof of the correctness of the TOP to BOT translation rules could be produced by showing that the denotations of the source TOP expressions are identical to the denotations of the resulting BOT expressions by induction on the syntactic complexity of the TOP expressions. The same strategy was used in [3] to prove formally the correctness of the TOP-to-TSQL2 translation rules (see [4] for a summary of the proof).

5 Conclusions

This paper has shown how an existing mapping from English to a complex temporal meaning representation formalism (TOP) can be coupled with a mapping from that formalism to a simpler one (BOT). The simpler formalism is very close to traditional first-order predicate logic, making it possible to exploit existing techniques to interface to time-sensitive applications other than TSQL2 databases, while maintaining the existing linguistic front-end.

Acknowledgements

Part of the work reported here was carried out at the Division of Informatics of the University of Edinburgh, with support from the Greek State Scholarships foundation, under the supervision of Graeme Ritchie and Peter Thanisch.

References

[1] J.F. Allen. Maintaining Knowledge about Temporal Intervals. *Communications of the ACM*, 26(11):832–843, 1983.

[2] J.F. Allen. Towards a General Theory of Action and Time. *Artificial Intelligence*, 23:123–154, 1984.

[3] I. Androutsopoulos. *A Principled Framework for Constructing Natural Language Interfaces to Temporal Databases.* PhD thesis, Department of Artificial Intelligence, University of Edinburgh, 1996.

[4] I. Androutsopoulos, G.D. Ritchie, and P. Thanisch. Time, Tense and Aspect in Natural Language Database Interfaces. *Natural Language Engineering*, 4(3):229–276, 1998.

[5] S. Ceri, G. Gottlob, and G. Wiederhold. Efficient Database Access from Prolog. *IEEE Transactions on Software Engineering*, 15(2):153–163, 1989.

[6] J. Clifford. *Formal Semantics and Pragmatics for Natural Language Querying.* Cambridge Tracts in Theoretical Computer Science, Cambridge University Press, 1990.

[7] B. Comrie. *Aspect.* Cambridge University Press, 1976.

[8] B. Comrie. *Tense.* Cambridge University Press, 1985.

[9] R.S. Crouch and S.G. Pulman. Time and Modality in a Natural Language Interface to a Planning System. *Artificial Intelligence*, 63:265–304, 1993.

[10] S. De, S. Pan, and A.B. Whinston. Temporal Semantics and Natural Language Processing in a Decision Support System. *Information Systems*, 12(1):29–47, 1987.

[11] C. Draxler. *Accessing Relational and Higher Databases through Database Set Predicates in Logic Programming Languages.* PhD thesis, University of Zurich, 1992.

[12] D.M. Gabbay, I. Hodkinson, and M. Reynolds. *Temporal Logic: Mathematical Foundations and Computational Aspects.* Oxford University Press, 1994.

[13] H. Kamp and U. Reyle. *From Discourse to Logic.* Kluer Academic Publishers, 1993.

[14] R. Kowalski and M. Sergot. A Logic-based Calculus of Events. *New Generation Computing*, 4:67–95, 1986.

[15] R. Lucas. *Database Applications Using Prolog.* Halsted Press, 1988.

[16] J. McCarthy and P.J. Hayes. Some Philosophical Problems from the Standpoint of Artificial Intelligence. In *Machine Intelligence 4*, pages 463–502. Edinburgh University Press, 1969.

[17] D. McDermott. A Temporal Logic for Reasoning about Processes and Plans. *Cognitive Science*, 6:101–155, 1982.

[18] M. Moens and M. Steedman. Temporal Ontology and Temporal Reference. *Computational Linguistics*, 14(2):15–28, 1988.

[19] T. Parsons. *Events in the Semantics of English: A Study in Subatomic Semantics.* MIT Press, 1990.

[20] C. Pollard and I.A. Sag. *Information-Based Syntax and Semantics – Volume 1, Fundamentals.* Center for the Study of Language and Information, Stanford, 1987.

[21] C. Pollard and I.A. Sag. *Head-Driven Phrase Structure Grammar.* University of Chicago Press and Center for the Study of Language and Information, Stanford., 1994.

[22] A. Prior. *Past, Present and Future.* Oxford University Press, 1967.

[23] H. Reichenbach. *Elements of Symbolic Logic.* Collier-Macmillan, London, 1947.

[24] R.T. Snodgrass, editor. *The TSQL2 Temporal Query Language.* Kluwer Academic Publishers, 1995.

[25] R.T. Snodgrass, M. Boehlen, C.S. Jensen, and A. Steiner. Transitioning Temporal Support in TSQL2 to SQL3. In O. Etzion and S. Sripada, editors, *Temporal Databases: Research and Practice*, pages 150 – 194. Springer-Verlag, Berlin, 1998.

[26] A. Tansel, J. Clifford, S.K. Gadia, S. Jajodia, A. Segev, and R.T. Snodgrass. *Temporal Databases – Theory, Design, and Implementation.* Benjamin/Cummings, California, 1993.

[27] V.J. Tsotras and A. Kumar. Temporal Database Bibliography Update. *ACM SIGMOD Record*, 25(1), 1996.

[28] J.F.A.K. van Benthem. *The Logic of Time.* D. Reidel Publishing Company, Dordrecht, Holland, 1983.

[29] Z. Vendler. Verbs and Times. In *Linguistics in Philosophy*, chapter 4, pages 97–121. Cornell University Press, Ithaca, NY, 1967.

Appendix

A Definition of the TOP language

This section defines the syntax and semantics of the subset of TOP that was introduced in this paper. See [3] and [4] for a full definition of TOP.

A.1 Syntax of TOP

The syntax of TOP is defined below using BNF. Angle brackets are used to group BNF elements. "*" denotes zero or more repetitions. "+" denotes one or more repetitions. Terminal symbols are in lower case, possibly with an initial capital. Non-terminals are in upper case. The distinguished symbol is *YNFORMS*.

$$\begin{aligned}
\mathit{YNFORMS} \rightarrow\ & \mathit{AFORMS} \mid \mathit{YNFORMS} \land \mathit{YNFORMS} \\
& \mid \mathit{Pres}[\mathit{YNFORMS}] \mid \mathit{Past}[\mathit{VARS}, \mathit{YNFORMS}] \mid \mathit{Perf}[\mathit{VARS}, \mathit{YNFORMS}] \\
& \mid \mathit{Culm}[\mathit{LITERAL}] \mid \mathit{At}[\mathit{TERMS}, \mathit{YNFORMS}] \\
& \mid \mathit{Before}[\mathit{TERMS}, \mathit{YNFORMS}] \mid \mathit{After}[\mathit{TERMS}, \mathit{YNFORMS}] \\
& \mid \mathit{Ntense}[\mathit{VARS}, \mathit{YNFORMS}] \mid \mathit{Ntense}[\mathit{now}, \mathit{YNFORMS}] \\
& \mid \mathit{For}[\mathit{CPARTS}, \mathit{VQTY}, \mathit{YNFORMS}] \mid \mathit{Fills}[\mathit{YNFORMS}] \\
\mathit{AFORMS} \rightarrow\ & \mathit{LITERAL} \mid \mathit{Part}[\mathit{PARTS}, \mathit{VARS}] \\
\mathit{LITERAL} \rightarrow\ & \mathit{PFUNS}(\{\mathit{TERMS},\}^{*}\, \mathit{TERMS}) \\
\mathit{TERMS} \rightarrow\ & \mathit{CONS} \mid \mathit{VARS} \\
\mathit{PARTS} \rightarrow\ & \mathit{CPARTS} \mid \mathit{GPARTS} \\
\mathit{VQTY} \rightarrow\ & 1 \mid 2 \mid 3 \mid \ldots
\end{aligned}$$

PFUNS, *CPARTS*, *GPARTS*, *CONS*, and *VARS* are disjoint open classes of terminal symbols.

A.2 Semantics of TOP

Temporal ontology

A *point structure* $\langle PTS, \prec\rangle$ is assumed, where PTS is the set of time-points, and $\prec$ is a binary, transitive, irreflexive relation over $PTS \times PTS$. Time is assumed to be discrete, bounded, and linear [12] [28]. t_{first} and t_{last} are the earliest and latest time-points respectively. $prev(t)$ and $next(t)$ are used to refer to the immediately previous and following time-points of a $t \in PTS$. For $S \subseteq PTS$, $minpt(S)$ and $maxpt(S)$ denote the earliest and latest time-points in S.

A *period* p over $\langle PTS, \prec\rangle$ is a non-empty subset of PTS. Periods are convex, i.e. if $t_1, t_2 \in p$, $t_3 \in PTS$, and $t_1 \prec t_3 \prec t_2$, then $t_3 \in p$. $PERIODS$ is the set of all periods over $\langle PTS, \prec\rangle$. p_1 is a *subperiod* of p_2 (written $p_1 \sqsubseteq p_2$), iff $p_1, p_2 \in PERIODS$ and $p_1 \subseteq p_2$. p_1 is a proper *subperiod* of p_2 (written $p_1 \sqsubset p_2$), iff $p_1, p_2 \in PERIODS$ and $p_1 \subset p_2$. The usual notational conventions apply when specifying the boundaries of periods; e.g. $(t_1, t_2]$ is an abbreviation for $\{t \in PTS \mid t_1 \prec t \preceq t_2\}$.

If S is a set of periods, then $mxlpers(S)$ is the set of *maximal periods* of S. $mxlpers(S) \equiv \{p \in S \mid \text{for no } p' \in S \text{ is it true that } p \sqsubset p'\}$.

TOP model

A TOP model M is an ordered 7-tuple:

$$M = \langle \langle PTS, \prec \rangle, OBJS, f_{cons}, f_{pfuns}, f_{culms}, f_{gparts}, f_{cparts} \rangle$$

where $\langle PTS, \prec \rangle$ is the point structure, $PERIODS \subseteq OBJS$, and f_{cons}, f_{pfuns}, f_{culms}, f_{gparts}, and f_{cparts} are as specified below:

$OBJS$ is a set containing all the objects in the modelled world that can be denoted by TOP terms, and f_{cons} is a function $CONS \mapsto OBJS$. Intuitively, f_{cons} maps each constant to the object it denotes.

f_{pfuns} maps each $\pi \in PFUNS$ to a function $(OBJS)^n \mapsto pow(PERIODS)$. It is assumed that each predicate symbol $\pi \in PFUNS$ is used with a particular arity (number of arguments) n. $pow(S)$ denotes the powerset (set of all subsets) of S. For every $\pi \in PFUNS$ and every $\langle o_1, o_2, \ldots, o_n \rangle \in (OBJS)^n$, it must be true that:

$$\text{if } p_1, p_2 \in f_{pfuns}(\pi)(o_1, o_2, \ldots, o_n) \text{ and } p_1 \cup p_2 \in PERIODS, \text{ then } p_1 = p_2$$

Intuitively, f_{pfuns} shows the maximal periods where the situation represented by $\pi(\tau_1, \ldots, \tau_n)$ holds.

f_{culms} is a function that maps each $\pi \in PFUNS$ to a function $(OBJS)^n \mapsto \{T, F\}$. Intuitively, f_{culms} shows whether or not a situation reaches a climax at the latest time-point where it is ongoing.

f_{gparts} is a function that maps each element of $GPARTS$ to a *gappy partitioning*. A gappy partitioning is a subset S of $PERIODS$, such that for every $p_1, p_2 \in S$, $p_1 \cap p_2 = \emptyset$, and $\bigcup_{p \in S} p \neq PTS$. f_{cparts} is a function that maps each element of $CPARTS$ to a *complete partitioning*. A complete partitioning is a subset S of $PERIODS$, such that for every $p_1, p_2 \in S$, $p_1 \cap p_2 = \emptyset$, and $\bigcup_{p \in S} p = PTS$.

Variable assignment

A variable assignment w.r.t. a TOP model M is a function $g : VARS \mapsto OBJS$. G_M, or simply G, is the set of all possible variable assignments w.r.t. M.

TOP denotation w.r.t. M, st, et, lt, g

Non-terminal symbols of the TOP BNF are used here as names of sets that contain expressions which can be analysed syntactically as the corresponding non-terminals.

An *index of evaluation* is an ordered 3-tuple $\langle st, et, lt \rangle$, such that $st \in PTS$, $et \in PERIODS$, and $lt \in PERIODS \cup \{\emptyset\}$.

The *denotation* of a TOP expression ξ w.r.t. a model M, an index of evaluation $\langle st, et, lt \rangle$, and a variable assignment g, is written $\|\xi\|^{M,st,et,lt,g}$ or simply $\|\xi\|^{st,et,lt,g}$. When the denotation of ξ does not depend on st, et, and lt, we may write $\|\xi\|^{M,g}$ or simply $\|\xi\|^g$.

- If $\kappa \in CONS$, then $\|\kappa\|^g = f_{cons}(\kappa)$.
- If $\beta \in VARS$, then $\|\beta\|^g = g(\beta)$.
- If $\phi \in YNFORMS$, then $\|\phi\|^{st,et,lt,g} \in \{T, F\}$.

- If $\pi(\tau_1, \tau_2, \ldots, \tau_n) \in \mathit{LITERAL}$, then $\|\pi(\tau_1, \tau_2, \ldots, \tau_n)\|^{st,et,lt,g} = T$ iff $et \sqsubseteq lt$ and for some $p_{mxl} \in f_{pfuns}(\pi)(\|\tau_1\|^g, \|\tau_2\|^g, \ldots, \|\tau_n\|^g)$, $et \sqsubseteq p_{mxl}$.
- If $\phi_1, \phi_2 \in \mathit{YNFORMS}$, then $\|\phi_1 \wedge \phi_2\|^{st,et,lt,g} = T$ iff $\|\phi_1\|^{st,et,lt,g} = \|\phi_2\|^{st,et,lt,g} = T$.
- $\|\mathit{Part}[\sigma, \beta]\|^g = T$ iff $g(\beta) \in f(\sigma)$ (where $f = f_{cparts}$ if $\sigma \in \mathit{CPARTS}$, and $f = f_{gparts}$ if $\sigma \in \mathit{GPARTS}$).
- $\|\mathit{Pres}[\phi]\|^{st,et,lt,g} = T$, iff $st \in et$ and $\|\phi\|^{st,et,lt,g} = T$.
- $\|\mathit{Past}[\beta, \phi]\|^{st,et,lt,g} = T$, iff $g(\beta) = et$ and $\|\phi\|^{st,et,lt \cap [t_{first}, st),g} = T$.
- $\|\mathit{Culm}[\pi(\tau_1, \ldots, \tau_n)]\|^{st,et,lt,g} = T$, iff $et \sqsubseteq lt$, $f_{culms}(\pi)(\|\tau_1\|^g, \ldots, \|\tau_n\|^g) = T$, $S \neq \emptyset$, and $et = [\mathit{minpt}(S), \mathit{maxpt}(S)]$, where:

$$S = \bigcup_{p \in f_{pfuns}(\pi)(\|\tau_1\|^g, \ldots, \|\tau_n\|^g)} p$$

- $\|\mathit{At}[\tau, \phi]\|^{st,et,lt,g} = T$, iff $\|\tau\|^g \in \mathit{PERIODS}$ and $\|\phi\|^{st,et,lt \cap \|\tau\|^g,g} = T$.
- $\|\mathit{Before}[\tau, \phi]\|^{st,et,lt,g} = T$, iff $\|\tau\|^g \in \mathit{PERIODS}$ and $\|\phi\|^{st,et,lt \cap [t_{first}, \mathit{minpt}(\|\tau\|^g)),g} = T$.
- $\|\mathit{After}[\tau, \phi]\|^{st,et,lt,g} = T$, iff $\|\tau\|^g \in \mathit{PERIODS}$ and $\|\phi\|^{st,et,lt \cap (\mathit{maxpt}(\|\tau\|^g), t_{last}],g} = T$.
- $\|\mathit{Fills}[\phi]\|^{st,et,lt,g} = T$, iff $et = lt$ and $\|\phi\|^{st,et,lt,g} = T$.
- $\|\mathit{Ntense}[\beta, \phi]\|^{st,et,lt,g} = T$, iff for some $et' \in \mathit{PERIODS}$, $g(\beta) = et'$ and $\|\phi\|^{st,et',\mathit{PTS},g} = T$.
- $\|\mathit{Ntense}[\mathit{now}^*, \phi]\|^{st,et,lt,g} = T$, iff $\|\phi\|^{st,\{st\},\mathit{PTS},g} = T$.
- $\|\mathit{For}[\sigma_c, \nu_{qty}, \phi]\|^{st,et,lt,g} = T$, iff $\|\phi\|^{st,et,lt,g} = T$, and for some $p_1, p_2, \ldots, p_{\nu_{qty}} \in f_{cparts}(\sigma_c)$, it is true that $\mathit{minpt}(p_1) = \mathit{minpt}(et)$, $\mathit{next}(\mathit{maxpt}(p_1)) = \mathit{minpt}(p_2)$, $\mathit{next}(\mathit{maxpt}(p_2)) = \mathit{minpt}(p_3)$, $\ldots$, $\mathit{next}(\mathit{maxpt}(p_{\nu_{qty}-1})) = \mathit{minpt}(p_{\nu_{qty}})$, and $\mathit{maxpt}(p_{\nu_{qty}}) = \mathit{maxpt}(et)$.
- $\|\mathit{Perf}[\beta, \phi]\|^{st,et,lt,g} = T$, iff $et \sqsubseteq lt$, and for some $et' \in \mathit{PERIODS}$, it is true that $g(\beta) = et'$, $\mathit{maxpt}(et') \prec \mathit{minpt}(et)$, and $\|\phi\|^{st,et',\mathit{PTS},g} = T$.

TOP denotation w.r.t. M, st

The denotation of ϕ w.r.t. M, st, written $\|\phi\|^{M,st}$ or simply $\|\phi\|^{st}$, is defined only for $\phi \in \mathit{YNFORMS}$:

- If $\phi \in \mathit{YNFORMS}$, then $\|\phi\|^{st} =$
  - T, if for some $g \in G$ and $et \in \mathit{PERIODS}$, $\|\phi\|^{st,et,\mathit{PTS},g} = T$,
  - F, otherwise

B Definition of the BOT language

B.1 Syntax of BOT

The syntax of BOT is defined using BNF, with the same conventions as in the definition of TOP. The distinguished symbol is $\mathit{YNFORMS}^B$.

$$
\begin{aligned}
\mathit{YNFORMS}^B &\rightarrow \mathit{AFORMS}^B \;\mid\; \mathit{YNFORMS}^B \wedge \mathit{YNFORMS}^B \\
\mathit{AFORMS}^B &\rightarrow \mathit{LITERAL}^B \;\mid\; \mathit{subper}(\mathit{PEREX}, \mathit{PEREX}) \;\mid\; \mathit{eq}(\mathit{TERMS}^B, \mathit{TERMS}^B) \\
&\quad \mid\; \mathit{period}(\mathit{TERMS}^B) \;\mid \mathit{part}(\mathit{PARTS}, \mathit{TERMS}^B) \\
\mathit{LITERAL}^B &\rightarrow \mathit{PFUNS}(\{\mathit{TERMS}^B,\}^* \mathit{TERMS}^B) \\
\mathit{TERMS}^B &\rightarrow \mathit{CONS} \;\mid\; \mathit{VARS} \;\mid\; \mathit{PEREX} \;\mid\; \mathit{PTEX} \\
\mathit{PARTS} &\rightarrow \mathit{CPARTS} \;\mid\; \mathit{GPARTS} \\
\mathit{PTEX} &\rightarrow \mathit{beg} \;\mid\; \mathit{now} \;\mid\; \mathit{end} \;\mid\; \mathit{earliest}(\mathit{PEREX}) \;\mid\; \mathit{latest}(\mathit{PEREX}) \\
&\quad \mid \mathit{succ}(\mathit{PTEX}) \;\mid\; \mathit{prec}(\mathit{PTEX}, \mathit{PTEX}) \\
\mathit{PEREX} &\rightarrow [\mathit{PTEX}, \mathit{PTEX}] \;\mid\; [\mathit{PTEX}, \mathit{PTEX}) \;\mid (\mathit{PTEX}, \mathit{PTEX}] \\
&\quad \mid\; (\mathit{PTEX}, \mathit{PTEX}) \;\mid \mathit{intersect}(\mathit{PEREX}, \mathit{PEREX})
\end{aligned}
$$

PFUNS, CPARTS, GPARTS, CONS, and VARS are disjoint open classes of terminal symbols that do not contain any of the other BOT terminal symbols. The same symbols for PFUNS, PARTS, CPARTS, GPARTS, CONS, VARS are used as in the definition of TOP, because these classes are the same in both languages.

B.2 Semantics of BOT

BOT assumes the same temporal ontology as TOP.

BOT model

A BOT model M is an ordered 6-tuple:

$$M^B = \langle \langle \mathit{PTS}, \prec \rangle, \mathit{OBJS}, f_{cons}, f^B_{pfuns}, f_{gparts}, f_{cparts} \rangle$$

where $\langle \mathit{PTS}, \prec \rangle$ is the point structure, and OBJS, f_{cons}, f_{gparts}, and f_{cparts} is are as in TOP. f^B_{pfuns} is as a function that maps every $\pi \in \mathit{PFUNS}$ to a function $\mathit{OBJS}^n \mapsto \{T, F\}$, where n is the arity of π.

Variable assignment

A variable assignment for BOT is a function $g : \mathit{VARS} \mapsto \mathit{OBJS}$, as in TOP. G has the same meaning as in TOP.

BOT denotation w.r.t. M, st, g

The denotation of a BOT expression ξ w.r.t. a BOT model M^B, a speech time $st \in \mathit{PTS}$, and a $g \in G$, written $\|\xi\|^{M^B, st, g}$ or simply $\|\xi\|^{st,g}$, is defined as follows:

- If $\kappa \in \mathit{CONS}$, then $\|\kappa\|^{st,g} = f_{cons}(\kappa)$.

- $\|beg\|^{st,g} = t_{first}$, $\|now\|^{st,g} = st$, $\|end\|^{st,g} = t_{last}$.
- $\|earliest(\tau)\|^{st,g} = minpt(\|\tau\|^{st,g})$. $\|latest(\tau)\|^{st,g} = maxpt(\|\tau\|^{st,g})$.
- $\|[\xi_1, \xi_2)\|^{st,g} = \{t \in PTS \mid \|\xi_1\|^{st,g} \preceq t \prec \|\xi_2\|^{st,g}\}$. The denotations of $[\xi_1, \xi_2]$, $(\xi_1, \xi_2]$, and (ξ_1, ξ_2) are defined similarly.
- $\|intersect(\xi_1, \xi_2)\|^{st,g} = \|\xi_1\|^{st,g} \cap \|\xi_2\|^{st,g}$.
- $\|succ(\xi)\|^{st,g} = next(\|\xi\|^{st,g})$.
- $\|prec(\xi_1, \xi_2)\|^{st,g}$ is T if $\|\xi_1\|^{st,g} \prec \|\xi_2\|^{st,g}$ and F otherwise.
- If $\beta \in VARS$, then $\|\beta\|^{st,g} = g(\beta)$.
- If $\phi \in YNFORMS^B$, then $\|\phi\|^{st,g} \in \{T, F\}$.
- If $\pi(\tau_1, \tau_2, \ldots, \tau_n) \in LITERAL^B$, then $\|\pi(\tau_1, \tau_2, \ldots, \tau_n)\|^{st,g} =$ $f^B_{pfuns}(\pi)(\|\tau_1\|^{st,g}, \|\tau_2\|^{st,g}, \ldots, \|\tau_n\|^{st,g})$.
- If $\phi_1, \phi_2 \in YNFORMS^B$, then $\|\phi_1 \land \phi_2\|^{st,g} = T$ iff $\|\phi_1\|^{st,g} = T$ and $\|\phi_2\|^{st,g} = T$.
- $\|part(\sigma, \beta)\|^{st,g} = T$ iff $g(\beta) \in f(\sigma)$ (where $f = f_{cparts}$ if $\sigma \in CPARTS$, and $f = f_{gparts}$ if $\sigma \in GPARTS$).
- $\|eq(\tau_1, \tau 2)\|^{st,g} = T$ iff $\|\tau_1\|^{st,g} = \|\tau_2\|^{st,g}$.
- $\|subper(\xi_1, \xi_2)\|^{st,g} = T$ iff $\|\xi_1\|^{st,g} \sqsubseteq \|\xi_2\|^{st,g}$.
- $\|period(\tau)\|^{st,g} = T$ iff $\|\tau\|^{st,g} \in PERIODS$.

BOT denotation w.r.t. M, st

The denotation of ϕ w.r.t. M^B, st, written $\|\phi\|^{M^B,st}$ or simply $\|\phi\|^{st}$, is defined only for $\phi \in YNFORMS^B$:

- If $\phi \in YNFORMS^B$, then $\|\phi\|^{st} =$
  - T, if for some $g \in G$, $\|\phi\|^{st,g} = T$,
  - F, otherwise

C TOP to BOT translation rules

- If $\pi \in PFUNS$ and $\tau_1, \ldots, \tau_n \in TERMS$, then:
  $trans(\pi(\tau_1, \ldots, \tau_n), \varepsilon, \lambda) = subper(\varepsilon, \lambda) \land \pi(\tau_1, \ldots, \tau_n, \beta) \land subper(\varepsilon, \beta)$,
  where β is a new variable.
- $trans(\phi_1 \land \phi_2, \varepsilon, \lambda) = trans(\phi_1, \varepsilon, \lambda) \land trans(\phi_2, \varepsilon, \lambda)$.
- $trans(Part[\sigma, \beta], \varepsilon, \lambda) = part(\sigma, \beta)$.
- $trans(Pres[\phi], \varepsilon, \lambda) = subper([now, now], \varepsilon) \land trans(\phi, \varepsilon, \lambda)$.
- $trans(Past[\beta, \phi], \varepsilon, \lambda) = eq(\beta, \varepsilon) \land trans(\phi, \varepsilon, intersect(\lambda, [beg, now)))$.

- $trans(Culm[\pi(\tau_1, \ldots, \tau_n)], \varepsilon, \lambda) = subper(\varepsilon, \lambda) \wedge \eta_1(\pi)(\tau_1, \ldots, \tau_n) \wedge \eta_2(\pi)(\tau_1, \ldots, \tau_n, \varepsilon)$, where η_1, η_2 are as in section 4.
- $trans(At[\tau, \phi], \varepsilon, \lambda) = period(\tau) \wedge trans(\phi, \varepsilon, intersect(\lambda, \tau))$.
- $trans(Before[\tau, \phi], \varepsilon, \lambda) = period(\tau) \wedge trans(\phi, \varepsilon, intersect(\lambda, [beg, minpt(\tau))))$.
- $trans(After[\tau, \phi], \varepsilon, \lambda) = period(\tau) \wedge trans(\phi, \varepsilon, intersect(\lambda, (maxpt(\tau), end]))$.
- $trans(Fills[\phi], \varepsilon, \lambda) = eq(\varepsilon, \lambda) \wedge trans(\phi, \varepsilon, \lambda)$.
- $trans(Ntense[\beta, \phi], \varepsilon, \lambda) = period(\beta) \wedge trans(\phi, \beta, [beg, end])$.
- $trans(Ntense[now, \phi], \varepsilon, \lambda) = trans(\phi, [now, now], [beg, end])$.
- $trans(For[\sigma_c, \nu_{qty}, \phi], \varepsilon, \lambda) =$
  $part(\sigma_c, \beta_1) \wedge part(\sigma_c, \beta_2) \wedge \cdots \wedge part(\sigma_c, \beta_{\nu_{qty}}) \wedge$
  $eq(earliest(\beta_1), earliest(\varepsilon)) \;\wedge\; eq(succ(latest(\beta_1)), earliest(\beta_2)) \wedge$
  $eq(succ(latest(\beta_2)), earliest(\beta_3)) \;\wedge\; \cdots \wedge\; eq(succ(latest(\beta_{\nu_{qty}-1})), earliest(\beta_{\nu_{qty}}) \wedge$
  $eq(latest(\beta_{\nu_{qty}}), latest(\varepsilon)) \;\wedge\; trans(\phi, \varepsilon, \lambda)$, where $\beta_1, \beta_2, \ldots, \beta_{\nu_{qty}}$ are new variables.
- $trans(Perf[\beta, \phi], \varepsilon, \lambda) =$
  $subper(\varepsilon, \lambda) \wedge period(\beta) \wedge prec(latest(\beta), earliest(\varepsilon)) \wedge trans(\phi, \beta, [beg, end])$.