

Representation and Inference with Consistent Temporal Propositions

Cristina Ribeiro

António Porto

Departamento de Informática
Universidade Nova de Lisboa
2825 Monte da Caparica, Portugal
e-mail: {mcr, ap}@fct.unl.pt
phone: 351-1-2953270

Abstract. Reasoning with temporal information is essential in Artificial Intelligence. We consider a knowledge base where the internal representation language deals with temporally qualified propositions and constraints on the ordering of time points. As temporal information is typically partial, a representation including constraints on the order of temporal objects is particularly suited. Temporal statements associate maximal intervals to basic atemporal propositions. Queries posed to a knowledge base which includes facts and rules use deduction to explore its consequences and abduction to generate consistent hypotheses. The inference system relies on a set of constraint primitives providing temporal consistency both for points and for intervals. The relevant aspects of the temporal framework are the underlying propositional language, the abductive derivation procedure and the facility of built-in constraint handling. The abductive procedure in the inference system provides the strategy for completing partial information in order to produce informative answers. The integration of constraint solving with abduction allows a query oriented generation of answers where the enforcement of constraints is efficiently performed.

1 Introduction

In most approaches in AI the representation of temporal objects is basically constituted by a proposition and an associated temporal extent [1, 6, 10]. This corresponds to adopting a very simple temporal ontology. The internal language considered here is based on this ontology, incorporating a feature that is also treated as basic: the intervals for the same proposition are maximal, and therefore disjoint. Briefly, there are two reasons for such an assumption: first, it is intuitively correct to view a proposition in time as a set of disjoint intervals; second, from a practical point of view a knowledge base where only intervals of maximal extent stand is more compact and so are the answers to queries.

The language of *maximal intervals* will be referred to in the sequel as *MI* and a description can be found in [8]. The propositional (plus time) version considers temporal constants, variables, Skolem constants and functions and proposition symbols. The temporal relations $\{=, >, <, \leq, \geq\}$ are used to build constraint

formulas, and we call predicative formulas those associating a proposition symbol with a pair of points. Clauses are formed with an atomic formula in the head and a conjunction of formulas in the body. One interesting consequence of dealing with maximal intervals is that the negative of a proposition p can be identified with its temporal complement denoted \bar{p} . The domain for propositions is closed for the complement operation. In the semantics, a temporal proposition is interpreted as a set of disjoint time periods.

To illustrate the language and the inference scheme, we will be using the classic Yale Turkey Shoot (YTS) example. The scenario description is as follows [4]. Initially, there is a turkey which is alive and an unloaded gun. Then loading and shooting take place. It is known that if a loaded gun is shot, alive ceases to hold. The formalization in *MI* can be stated as

- (F1) $alive[0, \alpha]$.
- (F2) $load[1, 1]$.
- (F3) $shoot[3, 3]$.
- (R1) $loaded[L, f(L)] \leftarrow load[L, L]$.
- (R2) $Y \leq S \leftarrow shoot[S, S], loaded[A, B], A \leq S, S \leq B,$
 $alive[X, Y], X \leq S$.
- (C1) $\alpha \geq 0$.
- (C2) $f(L) \geq L$.

Note that the notions of event and property implicit in the problem description are both modelled with temporal propositions. To deal with maximal intervals, Skolem constants are taken for the unspecified endpoints and constraints between them are explicitly stated. Atoms in the rules are either predicative atoms, where a proposition is associated with a time interval, or constraints relating two time points.

In this formalization of the scenario, R1 and R2 serve to illustrate the two classes of rules in the language. In R1 the head of the rule is a predicative atom, while in R2 it is a constraint atom. Predicative and constraint atoms are treated differently in the inference, and so are the two kinds of rules. No negation connective is considered for predicative literals, while constraint literals intrinsically have their negative correspondents due to the semantics of the temporal relations. This leads to some ambiguity in the expression of rules such as R2: the constraint in the head may alternatively be placed in the body in negated form, and the complement of some other constraint in the body may be moved to the head. To avoid it, the format of this kind of rule is normalized to

$$\leftarrow P, C$$

where P and C are respectively the predicative and constraint part of the body. Rules of this kind are treated in the inference as integrity constraints. They are not used deductively in resolution, but rather explored in a forward manner to enforce consistency of the configuration of the knowledge base obtained during the derivation.

2 Query answering

In a knowledge base (KB) having rules as well as facts we want to achieve a query-oriented answering system incorporating an efficient manipulation of constraints. In a logic-based approach, resolution can be used to work from the query as a goal using the facts and rules of the KB. In a language with constraints, this general approach has the drawback of treating them as regular predicates that have to be axiomatized; as they are central in the system, inference becomes quite inefficient. On the other hand, constraint based approaches to temporal reasoning [2] deal efficiently with constraints but are not query oriented and only consider facts for the temporal relations. In *MI*, clauses involve predicative and constraint literals, expressing relations between propositions at different times. To have an inference system driven by queries and also deal with constraints in an efficient manner, our strategy is to maintain a constraint machine based on state-of-the-art constraint satisfaction algorithms working under the control of a resolution machine.

It is important for the definition of an inference system to specify the kind of queries considered and what is expected as an answer. According to the logic-based approach adopted queries have no special language and are expressed as conjunctions of predicative and constraint atoms. With such queries it is possible to set up a particular configuration of intervals for a set of propositions, imposing constraints on their endpoints, and then test for its consistency with the KB. In the simplest form, a query is just a tentative retrieval of an interval for a proposition. Possible queries in our example are

Q1- *alive*[X , 3]

Q2- *alive*[X , Y], $Y > 5$

Q3- *loaded*[X , Y], $Y < 3$

Q1 asks about an ending of alive in point 3, Q2 about the possibility of an alive extending past time point 5 and Q3 about the existence of an interval for loaded terminating before time point 3.

As we are considering a proposition-based language, only intervals for propositions and constraints on their endpoints appear in the answers. The emphasis of the inference is on consistency: in the absence of complete knowledge, the answer provides information on scenarios consistent with the query. All informative scenarios appear as possible answers. Constraints appearing in the answers involve KB constants and functions and query variables; a predicative formula appearing as part of an answer indicates that the query can be satisfied by assuming it and following the consequences through the KB rules. This kind of reasoning has to do with completing partial information and is commonly designated as abduction. Our point of view is not, however, that of nonmonotonic logic, since we do not regard abductions as conclusions, but as conditions to the success of the query. This duality has already been pointed out, namely in [3]. Given a KB K and a query Q the answer to Q is constituted by a set of atoms A including substitutions and constraints for the query variables and the KB entities as well as abduced predicative atoms. Query variables are regarded as

existential entities just as the Skolem constants and functions in the KB. The answer comprises atoms which are abduced in the derivation process and also those which are consequences of such abductions through the KB, i.e. necessary conditions of the abductions.

Declaratively, the definition of answer follows the general one proposed in [5] for abductive logic programming.

Definition 1 Answer. A conjunction of atoms A is an *answer* to a query Q to a knowledge base K if and only if it is consistent, i.e.

$$K \cup A \not\approx \perp \quad (1)$$

and it is sufficient, i.e.

$$K \cup A \approx Q. \quad (2)$$

The properties of answers are expressed in terms of entailment and consistency in the temporal context. Entailment in the temporal context is noted \approx and has implicit in it the axioms for the maximality of intervals and the consistency of the point relations. Inconsistency of an answer with the KB is obviously excluded, and it is also required that the answer is a consequence of the knowledge base extended with the abductions.

This notion of answer, as opposed to those considered in systems of temporal constraints, has a goal-directed nature. In [2], for example, the reasoning tasks are enumerated as: finding all feasible times for a KB entity, finding the possible relations among two entities and generating scenarios consistent with the existing information. These tasks are also important here, as they have to do with checking consistency of the temporal constraints and reducing them to a minimal form. We also consider the possibility of abducing predicative formulas (creating hypothetical intervals for propositions) and constraints (excluding models of the KB such that the corresponding relations do not hold). This means that we can reason not only with the set of intervals and time points in the KB, but also with new query variables arbitrarily constrained. In this framework, a query may also be viewed as a tentative update, and the abductions as sufficient conditions for the KB to entail it.

Answers are nondeterministic, there is in general more than one way to satisfy a query. For example, if the query reduces to a predicative goal, as Q1 above, we obtain a solution by unifying it with an existing interval in the KB, and another by creating a new interval (abduction) that is made disjoint from all others for the same proposition. As an example, two possible answers to query Q1 are

$$A_1 = \{X = 0, \alpha = 3\}$$

$$A_2 = \{X > \alpha, \text{alive}[X, 3]\}.$$

The approach to temporal reasoning using abduction differs from the non-monotonic approaches for which the YSP has been proposed in several aspects. The most fundamental is that abduction is used to generate hypotheses, and the existence of different hypotheses leads to different answers to the same query.

The nonmonotonic approaches are oriented towards some minimality criterion, preferring some models of the information over others. The YSP intended to illustrate the failure of nonmonotonic formalisms to identify the intuitive or right model of the problem. The abductive approach proposed here does not exclude any model of the knowledge for which a consistent scenario can be generated. A scenario where the turkey is not alive after the shooting is possible, but also the one where the gun is previously unloaded and the turkey still alive after the shooting. Judging the plausibility of each scenario is not considered part of the inference process.

3 Inference system

The inference manipulates representations for the temporal knowledge at two levels: the facts and rules of the KB and a constraint network (CN). The network is used as an alternative representation for the factual knowledge in the KB. The two kind of atomic formulas, constraints and predicative atoms, correspond respectively to the assertion of a relation between two time points and the statement about two time points being endpoints of an interval for a selected proposition. The constraint network represents points and intervals by nodes. A constraint formula is incorporated as an arc between two point nodes and a predicative formula by two arcs between an interval node and a pair of point nodes. Interval nodes are labelled by propositions and some additional structures are provided to account for metric information, such as interval endpoints associated to specific points on the time line or constraints involving such constants.

In the constraint network, the interval nodes are partitioned by proposition to facilitate the enforcement of interval maximality. The network can be regarded as a central network with the relations between point nodes, plus several mutually disjoint subnetworks of interval nodes, connected to the central network by 'begin' and 'end' arcs. Constraint propagation takes place in each subnetwork, and modifications are propagated between them.

The information in the KB rules can be partly captured in the network by unfolding them with available facts to produce new factual consequences. The unfolded rules are kept as a set of "reduced rules" which serve to complement the information in the constraint network. Note that integrity constraints are also subject to this reduction, and may also be used to obtain new consequences of the KB. Figure 1 shows the constraint network obtained from the YSP description in *MI*.

The derivation of a query is described using inference rules for the deductive and abductive procedure. The operations on constraints appear as calls to constraint solver functions. The derivation is started by taking the query as the top goal and proceeds using resolution until the resolvent is empty. The constraint network and the set of rules are used and updated as required along the derivation.

A goal can be solved by reduction with an existing fact, or abduced when it is not a consequence of the KB. An abduced goal must be consistent with the KB

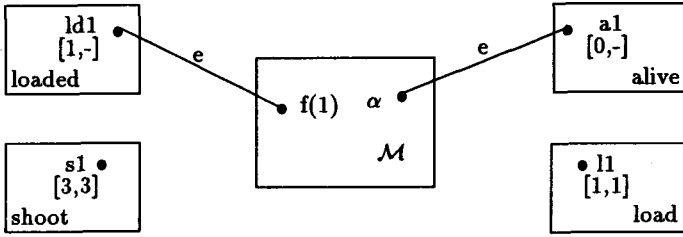


Fig 1: Constraint network for the YSP.

and any previous abductions. The abduction of a constraint goal is equivalent to a choice in the possible completion of the temporal order, excluding models of the KB where the constraint does not hold. Abducing a predicative goal corresponds to creating an interval for the proposition that is not supported by existing knowledge—does not unify with and is disjoint from any existing intervals for the same proposition.

The consistency of the KB and abductions is enforced at two levels, a local one involving the constraint network and a global one requiring consistency of rules and integrity constraints. Local consistency is enforced whenever abduction occurs while global consistency is only tested when a locally consistent answer is computed. The inference rules that follow define the abductive procedure and only deal with local consistency enforcement. They make use of several constraint manipulation primitives to be executed on the constraint network in order to extract information and perform updates.

3.1 The abductive procedure

Inference is described by a set of rules each relating a goal with a set of subgoals, with explicit conditions under which it is applicable. The rules show the evolution of the constraint network in the course of inference. K designates the KB and $N-R$ its network and rule set components. It is assumed that during a derivation the successive contexts are always available in the $N-R$ form and that this form is updated according to the abductions made. Tuples for the abductive relation are expressed as

$$K \sim g \langle A \mid K' \rangle.$$

In the derivation of g , the KB is extended from an initial state K to a final one K' with the abduction of a set of atoms A .

Empty goal

$$\overline{K \sim \emptyset \langle \emptyset \mid K \rangle} \quad (3)$$

Conjunctive goal

$$\frac{K \vdash g(A_1 | K') \quad K' \vdash G(A_2 | K'')}{K \vdash (g, G)(A_1, A_2 | K'')} \quad (4)$$

Rules 3 and 4 are the structural rules that express the ramification of derivations in the conjunctions and establish termination of a branch when the empty goal is obtained. Abductions are propagated from each branch of a conjunction to the next by updating the KB, so that the initial scenario for the derivation of a goal is the final scenario of the previous one. The abductions for the conjunction are collected from the abductions obtained for each of the conjuncts.

Atomic constraint goal - network retrieval

$$\frac{N \vdash c}{K \vdash c(\emptyset | K)} \quad \{c \in C\} \quad (5)$$

Atomic predicative goal - network retrieval

$$\frac{N \vdash p[X', Y'] \quad N-R \vdash (X = X', Y = Y')(A | K')}{N-R \vdash p[X, Y](A | K')} \quad (6)$$

Rule 5 shows that a constraint is derived when it is a consequence of the constraint network. Rule 6 is the reduction of a predicative goal by unification with an interval in the constraint network. The derivation of the two equality constraints corresponding to the unification is required to solve the goal.

Atomic goal - direct abduction

$$\frac{K \ll g(A | K')}{K \vdash g(g, A | K')} \quad \left\{ \begin{array}{l} N \not\vdash g \\ g \in \mathcal{PUC} \end{array} \right\} \quad (7)$$

Rule 7 accounts for the possibility of direct abduction of the atom to be derived. It is required that the goal is not a consequence of the constraint network. If this was the case the abduction would be unnecessary. The consistency of the abduction with the KB is required, and the call to \ll includes this test. \ll adds to the abductions the new consequences A which are produced while enforcing consistency of the abduced goal with K . K' is the expanded KB after incorporation of the abductions.

Atomic predicative goal - reduction through a rule

$$\frac{\begin{array}{l} K \vdash (X = X', Y = Y')(A_1 | K') \\ N' \gg p[X, Y](A_2) \\ K' \ll A_2(A_3 | K'') \\ K'' \vdash b(A_4 | K''') \end{array}}{K \vdash p[X, Y](A_1, A_2, A_3, A_4 | K''')} \quad \left\{ \begin{array}{l} K = N-R \\ K' = N'-R' \\ \text{vrt}(p[X', Y'] \leftarrow b) \in R \end{array} \right\} \quad (8)$$

Besides the unification with one of the intervals in the network and the direct abduction possible with previous rules, a predicative goal can be derived using the KB rules, as considered in 8. $urt(h \leftarrow B)$ denotes a variant of a rule, with fresh variables to avoid variable conflicts. The endpoints of the interval in the goal are unified with those for the interval in the head of the rule, with abductions A_1 and resulting scenario K' . As this inference rule should provide an interval not already in the network, the network primitive \gg is used to obtain the set of constraints A_2 forcing $[X, Y]$ to be disjoint from other intervals in the network. These are incorporated in K' producing K'' . The body of the KB rule is derived in K'' , resulting in the final scenario K''' . This inference rule allows, in the derivation of a goal, the abduction of "upstream" goals. By making abductions in the body of a rule whose head is the goal to be derived, the goal becomes a logical consequence of the extended knowledge base.

The disjointness condition imposed by \gg makes rule 8 exclusive from rule 7. The resulting set of abductions A_4 is not empty, otherwise $p[X, Y]$ would already be a consequence of the KB and belong to its expansion.

The abductive derivation relation \vdash relies on the abductive incorporation relation \ll . Operationally, \ll is invoked for a knowledge base K and a set of abductions to be incorporated A and succeeds whenever the abductions and the KB are locally consistent. The consistency check produces new information which is returned in two forms: an updated knowledge base K' and a set of new facts A' . For details on the definition of \ll see [7].

3.2 Constraint satisfaction and update

Besides the derivation relation, three auxiliary relations are mentioned in the inference rules. The most primitive notion required from the constraint network is logical consequence. The relation

$$N \vdash g \quad (9)$$

expresses the fact that g is a logical consequence of the constraint network, and therefore of the knowledge base. g may be any ground predicative or constraint fact. This primitive is viewed operationally as a retrieval operator on the constraint network. For constraint goals, it succeeds if the binary relation between the two time points in g is subsumed by the relation holding in the network. For predicative goals, it succeeds if g is a fact for the corresponding proposition.

The second network relation involves a predicative fact and relates it with the constraints that make the corresponding interval disjoint from all other intervals in the network. The notation is

$$N \gg p[X, Y](A) \quad (10)$$

The conditions required to make $[X, Y]$ disjoint from other intervals for p in N become available in A .

The previous relations are used to explore the information in the constraint network. Another necessary manipulation has to do with the update of the network with new facts. This is expressed with the network incorporation relation \ll .

$$K \ll A(A' | K') \quad (11)$$

The incorporation of the set of facts A in knowledge base K succeeds if the facts are consistent with the information in K . The outcome of the incorporation is the updated knowledge base K' and a set of new consequences A' which does not include A . K' comprises an updated constraint network and an updated set of reduced rules.

The actual mechanisms for exploring consequence and consistency in the constraint network and for assimilating new facts into the network are described in [7]. When the constraint network is updated as a consequence of the abductions, the modifications are propagated until the tightest possible constraint holds between each pair of nodes. The account of new constraints that appear as a consequence of the abductions is informative of the scenario induced in the KB by the abductions. Such new constraints also appear in the answer.

3.3 Global consistency

The abductive procedure produces a solution which is consistent with respect to the constraint network, but which may still present inconsistencies with the rules and integrity constraints in the KB. Such inconsistencies are not detectable from the network, and the enforcement of global consistency requires the generation of a *scenario* for the rules in the KB. The need for scenario generation arises from the existence of disjunctive statements in the integrity constraints, for which a model is not guaranteed by the network. In [7] we show that it is possible to enforce global consistency restricting the scenario generation to a subset of the reduced rules obtained during the derivation, the residual rules where no universal variables remain.

4 Conclusions

We propose a framework for temporal reasoning based on a language of maximal intervals. Temporal statements are clauses whose atoms are either predicative, associating an interval to a proposition, or constraints between time points. The representation language provides existential entities in the form of Skolem constants and functions to account for underspecified times. Constraints on time points are also a form of expressing partial knowledge on the relations holding between time points.

Formulating a knowledge base in the temporal language, it is possible to pose queries for which answers are computed by an inference system. Answers are defined as abductive: new relations holding between times and new predicative facts can be assumed as hypotheses in order to complete the derivation

of an answer. Consistency of the hypotheses with the KB is required and its enforcement is a central task for the inference system.

To achieve a query-oriented behaviour and efficiency in the consistency enforcement, the factual knowledge in the KB is kept separately in a constraint network and the KB rules are processed forward with existing facts to produce implicit factual consequences. The network is used for retrieving temporal relations and facts for propositions and the updates required when abduction is performed give rise to constraint propagation. To allow an easy enforcement of interval maximality, the network of facts is partitioned into subnetworks of interval nodes, one for each proposition. They are linked to a central subnetwork for point nodes. The inference we describe is sound with respect to the semantics of the language and the definition of abductive answers [7].

The inference system as presented is non-deterministic, answers corresponding to possible models of the KB plus the axioms of maximality. It is possible to have specializations where models are not equally treated, and preferences are established. One such possibility is the exclusion of answers that are obtained creating intervals that are not supported on KB evidence; in this case, only the temporal order predicates are abducible.

Having an internal representation that only deals with maximal intervals does not require all knowledge to be viewed in this form. We have proposed an external language to deal with assertions on nonmaximal intervals, considering them as partial information on maximal ones; a systematic translation is presented in [9]. This means that the interface with the KB does not require the reduction to maximal intervals as the external expressions may refer to nonmaximal intervals. Transitions are stated explicitly and give information on endpoints of maximal intervals. The external language statements resort to the simple notions of periods of time where a proposition holds and points in time where a proposition "starts" and "ends". These are expressed with maximal assertions in *MI* by creating existential entities for the interval endpoints which are not specified.

The mechanism devised for consistency of points and maximal intervals, switching between point and interval domains, can be generalized in several ways. First, the intervals to be kept disjoint need not be intervals for the same proposition. If two propositions are known to be incompatible, an overall condition of disjointness may be applied to the union of the two sets of intervals. This extension has no impact on the inference, but the (external) language has to be able to express such constraints. Note that disjointness of intervals is expressible in *MI*, but does not benefit from a constraint-level implementation except within the intervals for the same proposition. For arbitrary propositions an integrity constraint is required, such as

$$\leftarrow p[X, Y], q[A, B], A < Y, B \geq X$$

to enforce that intervals for p and q are disjoint.

References

1. James Allen. Maintaining Knowledge About Temporal Intervals. *Communications of the ACM*, 26(11):832–843, 1983.
2. R. Dechter, I. Meiri, and J. Pearl. Temporal constraint networks. *Artificial Intelligence*, 49(1), 1991.
3. K. Eshghi and R. A. Kowalski. Abduction compared with negation by failure. In *Proceedings of the 6th International Conference on Logic Programming*, 1989.
4. Steve Hanks and Drew McDermott. Default reasoning, nonmonotonic logics and the frame problem. In *Proceedings of the 5th National Conference on Artificial Intelligence*, pages 328–333. AAAI, 1986.
5. A. C. Kakas, R. A. Kowalski, and F. Toni. Abductive logic programming. *Journal of Logic and Computation*, 2(6):719–770, 1992.
6. Drew McDermott. A Temporal Logic for Reasoning About Processes and Plans. *Cognitive Science*, (6):101–155, 1982.
7. Cristina Ribeiro. *Representation and Inference of Temporal Knowledge*. PhD thesis, FCT-UNL, 1993.
8. Cristina Ribeiro and António Porto. Maximal intervals, an approach to temporal reasoning. In P. Barahona, L. Moniz Pereira, and A. Porto, editors, *EPIA 91—5th Portuguese Conference on Artificial Intelligence*. Springer Verlag, Lecture Notes on Artificial Intelligence 541, 1991.
9. Cristina Ribeiro and António Porto. Internal and external languages in a temporal representation scheme. *Compulog-Net Workshop on Knowledge-Based Systems*, Athens, 1993.
10. Yoav Shoham. *Reasoning about Change*. The MIT Press, 1987.