

human reasoning, and hence which theoretical framework will prevail.

Further Reading

Braine MDS and O'Brien DP (1998) *Mental Logic*. Mahwah, NJ: Lawrence Erlbaum.

Evans JBT and Over DE (1996) *Rationality and Reasoning*. Hove, UK: Psychology Press.

Fiddick L, Cosmides L and Tooby J (2000) No interpretation without representation: the role of domain-specific representations and inferences in the Wason selection task. *Cognition* 77: 1–79.

Johnson-Laird PN and Byrne RMJ (1991) *Deduction*. Hove, UK: Lawrence Erlbaum.

Manktelow KI (1999) *Reasoning and Thinking*. Hove, UK: Psychology Press.

Oaksford M and Chater N (1998) *Rationality in an Uncertain World*. Hove, UK: Psychology Press.

Rips L (1994) *The Psychology of Proof*. Cambridge, MA: MIT Press.

Slovan SA (1996) The empirical case for two systems of reasoning. *Psychological Bulletin* 119: 3–22.

Sperber D, Cara F and Girotto V (1995) Relevance theory explains the selection task. *Cognition* 57: 31–95.

Reasoning about Time

Intermediate article

Paolo Terenziani, Univ. Piemonte Orientale 'Amedeo Avogadro', Italy

CONTENTS

Introduction

Temporal primitives and the structure of time

General-purpose (logical) approaches

Constraint-based approaches

Further issues

Summary

Reasoning about time is a fundamental task in many 'intelligent' activities. From the standpoint of artificial intelligence, it includes representing and making inferences about the dynamic phenomena of the world, change, persistence and temporal constraints between activities.

INTRODUCTION

The human way of perceiving and understanding the world incorporates at a deep level the notion of time. Time seems to be a primitive entity to which objects in the world are related. In particular, the notions of action, causality and change are intrinsically related to the notion of time. Events occur in a given state of the world and cause changes in that state. Only in a static world with no changes would there be no notion of time. Thus, time plays a fundamental role in any model of reality, and reasoning about time is essential in most intelligent activities. For instance, dealing with time is important in many different areas of computer science, including process simulation, databases, protocol, guideline and workflow management, and many areas of artificial intelligence (AI), such as planning, scheduling, diagnosis, expert systems,

natural language understanding and knowledge representation. Therefore, many researchers in AI have attempted to define suitable formalisms to represent time-related phenomena and to reason with them.

TEMPORAL PRIMITIVES AND THE STRUCTURE OF TIME

The definition of the basic primitives (i.e., the ontological analysis) of time is a matter of debate within the AI community. First of all, one has to choose between time points and time intervals as the basic temporal entities. There are intuitive and technical arguments in favour of both choices. Time points are taken as the basic primitives, for example, in the situation calculus (McCarthy and Hayes, 1969). Other authors have suggested that time intervals are more intuitive than the abstract mathematical notion of points, and have pointed out some technical problems arising from the use of time points (Allen, 1983).

Next, one has to specify a suitable structure for the temporal domain. For instance, time may be regarded as a discrete collection of temporal elements or, on the other hand, as a dense collection (so

that between any two temporal elements there is always a third). Moreover, time may be unbounded (infinite in one or both directions) or bounded. Some ordering relation (temporal precedence) must be defined on the set of temporal elements. This may be linear (i.e., the temporal structure is totally ordered) or branching (for example, each branch may represent a possible future evolution of the given world). Circular time is also used in some applications. (See Van Benthem (1983) for a detailed analysis of these alternatives.)

Naturally, different approaches have been proposed and different choices made, depending on the specific phenomena and tasks at hand. However, one can distinguish two broad streams in research about time within the AI community, henceforth called ‘general-purpose’ and ‘constraint-based’ approaches.

GENERAL-PURPOSE (LOGICAL) APPROACHES

General-purpose approaches mainly focus on the definition of a formalism general enough to represent a wide range of temporal phenomena. Such approaches aim at dealing with the dynamic aspects of the world, at describing the internal structure of actions and events occurring in the world, and at modeling how the world changes in response to such actions and events. Although very different formalisms have been devised most of them are first-order or modal logical formalisms.

The Situation Calculus

An important example is the situation calculus (McCarthy and Hayes, 1969). The situation calculus adopts a very natural (and widely used) way of introducing time in first-order logic: functions and predicates are extended with an additional argument representing the time at which they are to be interpreted. Specifically, in the situation calculus, the temporal argument denotes a situation, i. e., ‘the complete state of the universe at an instant of time’ (McCarthy and Hayes, 1969). ‘Propositional fluents’ (i.e. functions from situations to truth-values) are used to describe facts. For instance, a situation s in which a person p is in place x and it is raining in place x can be described as follows:

$$\text{at}(p, x, s) \wedge \text{raining}(x, s) \quad (1)$$

Causation is modeled using the propositional fluent $F(a, s)$ stating that situation s will be followed

by a situation that satisfies the propositional fluent a . For instance, one could use F to assert that if a person is out in the rain he will get wet:

$$\forall x \forall p \forall s \text{ raining}(x, s) \wedge \text{at}(p, x, s) \wedge \text{outside}(p, s) \Rightarrow F(\lambda s' \text{ wet}(p, s'), s) \quad (2)$$

Actions are modeled using the ‘situational fluent’ *result*. $\text{result}(p, a, s)$ denotes the situation that results when the person p carries out the action a in the situation s . For instance, the following formula represents the axiom schema that, if in a situation s a person p has a key k that fits the safe sf , then in the situation resulting from the action of opening sf with the key k , the safe is open.

$$\text{has}(p, k, s) \wedge \text{fits}(k, sf) \wedge \text{at}(p, sf, s) \Rightarrow \text{open}(sf, \text{result}(p, \text{open}(sf, k), s)) \quad (3)$$

Action and Change

Difficult problems arise when dealing with the interplay between actions and changes of the world. Many researchers have realised that one is not able to infer everything about a situation resulting from an action without having a very large number of axioms describing how the action relates to the world and changes it. First, one has to specify all the conditions that a situation s must satisfy in order that an action a may be performed in that situation. The number of such conditions tends to be very large even for simple actions (for example, I can drive a car if I have a car, a driving license, petrol in it... but also I must have hands, I must not be blind and the car must have wheels...). This is the so-called ‘qualification problem’. Secondly, one has to specify everything that changes as a result of the action (if I am driving, I am moving, the the atmosphere is changing because of the exhaust produced by the car...). This is called the ‘ramification problem’. The specification of everything that does not change is called the ‘frame problem’. In classical first-order logic, it is not easy to express notions such as ‘everything that is not explicitly asserted to change remains unchanged’. Dealing with the qualification, ramification and frame problems involves a certain capacity for ‘nonmonotonic’ reasoning. That is, one must be able to make inferences not only from known facts about the world (situation), but also from assumed facts. For instance, if I parked my car in front of my house yesterday, I can assume that the car is still there today, and draw conclusions on the basis of such an assumption (called a ‘persistence assumption’). However, such conclusions are defeasible, in the sense that they may have to be retracted

if further information about the world becomes available (for example, the police telephone to advise me that my car has been stolen). In the AI literature, many different logical frameworks have been devised in order to deal with nonmonotonicity (see the survey in Lee, Tannok and Williams, (1993). In recent years, many approaches have been devised that deal with the above problems by taking advantage of the 'negation by failure' of logic programming. For instance, a formulation of the situation calculus based on logic programming that deals with the frame problem has been recently proposed (Kowalski and Sadri, 1997).

Temporal (Modal) Logic

Another popular way of introducing the notion of time into a logical framework is by means of modal logics dealing with time (usually called 'temporal logics'). Modal logics were originally introduced to study the different 'modes' of truth. For example, a property p may be false in the present world, and yet the assertion 'possibly p ' may be true if there is another possible world where p is true. A temporal logic is a type of modal logic in which each possible world is associated with a temporal element. A temporal precedence relation (akin to the 'accessibility' relation in modal logics) relates temporal elements, and the standard modal operators are reinterpreted over the temporal context. Any temporal logic must provide some temporal operators in order to describe and reason about how the truth-values of assertions vary with time. Different temporal logics have been devised to deal with linear time or branching time; and both propositional and first-order temporal logics have been devised. A simple example of a temporal logic is propositional linear temporal logic (PLTL) (Emerson, 1990). PLTL adopts linear time, and introduces the temporal operators Fp ('sometime p '), Gp ('always p '), Xp ('next time p ') and pUq (' p until q '). Given the current situation s_{now} , the above assertions evaluate to true if and only if the following conditions hold:

- Fp : p is true in at least one of the situations that follow s_{now} .
- Gp : p is true at s_{now} and in all situations following s_{now} .
- Xp : p is true in the situation that immediately follows s_{now} .
- pUq : p is true at s_{now} and in all situations between s_{now} and the first situation (following s_{now}) in which q is true.

A survey of different temporal logics can be found in Emerson, 1990.

CONSTRAINT-BASED APPROACHES

Constraint-based approaches mainly focus on the definition of a representation formalism and of reasoning techniques to deal specifically with temporal constraints between temporal entities (time points or intervals), independently of the events and states associated with them. For instance, given three time intervals I_1 , I_2 and I_3 , if I_1 is before I_2 and I_2 is before I_3 , then one can infer that I_1 is before I_3 , independently of the events that occurred in I_1 , I_2 and I_3 . By restricting the problem in this way, one can obtain better results than general-purpose approaches. For example, with a careful definition of the temporal constraint language, one can define specialized reasoning techniques that make inferences such as the one above in a more efficient way than, say, a standard theorem-prover for first-order logic. There is a trade-off between the expressiveness of the constraint language and the computational complexity of the inference techniques that are possible with it, and this is a central topic of research (see the survey in (Vila, 1994)).

Because dealing with temporal constraints is a fundamental task in so many AI applications, there has been much research aimed at building application-independent and domain-independent temporal constraints managers. These are intended to be *specialized knowledge servers* that represent and reason with temporal constraints, and that cooperate with other software modules in order to solve problems in different applications (Console and Terenziani, 1999). For instance, a temporal constraints manager could cooperate with a planner, in order to check incrementally the temporal consistency of a plan. Specialized temporal constraints managers are very useful from the computational point of view, as they allow programmers to focus on problems specific to their domain and application and to design modular software architectures. Temporal constraints managers can be distinguished on the basis of the types of temporal constraints they deal with.

Representing and reasoning with qualitative temporal constraints

Qualitative temporal constraints concern the relative positions of time points or intervals. J.F. Allen's interval algebra (IA) was an attempt to formalize these. Allen defined the 13 primitive qualitative relations between time intervals shown in Figure 1 (Allen, 1983).

These relations are exhaustive and mutually exclusive, and can be combined to represent ambiguities. For example, the constraint I_1 (BEFORE,

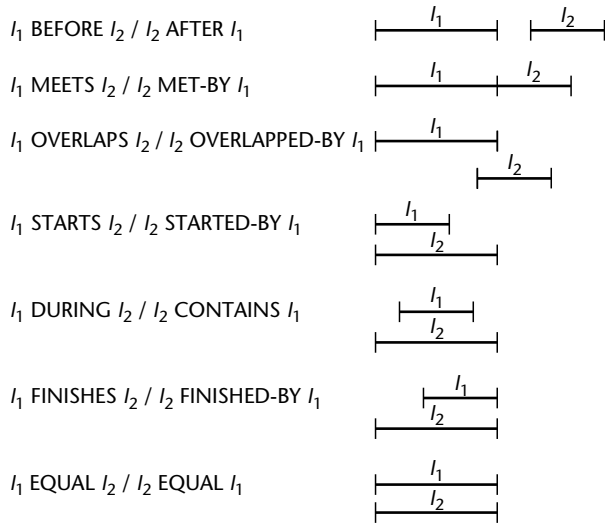


Figure 1. Qualitative relations between intervals in Allen’s Interval Algebra.

CONTAINS,OVERLAPS) I_2 represents the fact that the time interval I_1 is before I_2 , overlaps I_2 or contains I_2 .

Temporal inference is based on two algebraic operations on relations between time intervals: intersection and composition. Given two possibly ambiguous relations R_1 and R_2 between two time intervals I_1 and I_2 , the temporal intersection $R_1 \cap R_2$ is the most constraining relation between I_1 and I_2 . For example, the temporal intersection between

$$R_1 : I_1(\text{BEFORE, CONTAINS, OVERLAPS})I_2 \quad (4)$$

and

$$R_2 : I_1(\text{BEFORE, MEETS, OVERLAPS})I_2 \quad (5)$$

is

$$R_1 \cap R_2 : I_1(\text{BEFORE, OVERLAPS})I_2 \quad (6)$$

Given a relation R_1 between I_1 and I_2 and a relation R_2 between I_2 and I_3 , the composition $R_1 @ R_2$ gives the resulting relation between I_1 and I_3 . For example, the composition of

$$R_1 : I_1(\text{BEFORE})I_2 \quad (7)$$

and

$$R_2 : I_2(\text{BEFORE, CONTAINS, OVERLAPS})I_3 \quad (8)$$

is

$$R_1 @ R_2 : I_1(\text{BEFORE})I_3 \quad (9)$$

In Allen’s approach, temporal inference is performed by a path consistency algorithm that computes the transitive closure of the constraints by

repeatedly applying intersection and composition. Abstracting from many optimizations, such an algorithm can be schematized as follows. Let R_{ij} denote the (possibly ambiguous) relation between I_i and I_j . For all triples of time intervals $\langle I_i, I_k, I_j \rangle$, do $R_{ij} \leftarrow R_{ij} \cap (R_{ik} @ R_{kj})$.

The time taken by Allen’s algorithm increases as the cube of the number of time intervals. However, such an algorithm is not complete for IA (checking the consistency of a set of temporal constraints in IA is NP-hard). Other researchers have tried to design less expressive but more tractable formalisms. For example, the point algebra (PA) is defined similarly to IA, but the temporal elements are time points. Thus, there are only three primitive relations ($<$, $=$, and $>$), and four ambiguous relations ($(<, =)$, $(>, =)$, $(<, >)$, and $(<, =, >)$) between them. In PA, the constraint closure can be computed in polynomial time by algorithms that are both sound and complete. The price to be paid for tractability is a loss of expressive power: not all (ambiguous) relations between time intervals can be mapped onto relations between their endpoints. Another interesting algebra is the continuous point algebra (CPA), which consists of all relations in PA except for the inequality relation ($<$, $>$). Allen’s path consistency algorithm is both correct and complete for such an algebra (see the survey in Vila, 1994).

Representing and reasoning with quantitative temporal constraints

Quantitative temporal constraints involve metric time and include *dates* (‘John arrived on 10 October 1999 at 10.00’), *durations* (‘John worked for 3 hours’) and *delays* (‘John arrived 10 minutes after Mary’). Different approaches have been developed to deal with quantitative temporal constraints, and their complexity depends on the types of constraints taken into account. The simplest case is when temporal information is available in the form of dates that exactly locate points and intervals on the timeline. However, in many cases, metric temporal information is not so precise: one can have approximate dates, durations, and delays:

$$\text{The time interval } I_1 \text{ ended on 10 October 1999, between 10.00 and 10.15.} \quad (10)$$

$$I_1 \text{ lasted between 20 and 30 minutes.} \quad (11)$$

$$I_2 \text{ started between 20 and 40 minutes after the end of } I_1. \quad (12)$$

In such cases, temporal reasoning is important in order to infer new temporal constraints and to detect inconsistencies. For instance, from the three

constraints above one can infer that I_2 started on 10 October 1999, between 10.20 and 10.55, so that the set of constraints above is inconsistent with the constraint

$$I_2 \text{ started at } 10.10. \quad (13)$$

Problems become more complex if disjunctions of temporal constraints are taken into account:

$$I_2 \text{ started } 20\text{--}30 \text{ or } 50\text{--}60 \text{ minutes after} \\ \text{the end of } I_1. \quad (14)$$

Many approaches have been developed to deal with these problems. For instance, the temporal constraint satisfaction problem (TCSP) model (Dechter, Meiri and Pearl, 1991) is based on the primitive notions of time points and distances between time points. A TCSP is a set of constraints of the form

$$P_i([l_1, u_1], \dots, [l_n, u_n])P_j \quad (15)$$

where P_i and P_j denote time points ranging over a dense domain, and l_i and u_i ($l_i \leq u_i$) are real numbers, stating that the temporal distance between P_i and P_j is between l_i and u_i for some i . For instance, if we denote by I^- and I^+ the starting and ending points of a time interval I , the temporal constraint sentence 14 can be represented as:

$$I_1^+([20, 30], [50, 60])I_2^- \quad (16)$$

TCSP can easily be applied to graphs, where nodes represent time points and arcs are labelled with distances. Dechter, Meiri and Pearl developed an optimized path-finding algorithm to perform temporal reasoning with a graph representation. Their algorithm takes exponential time. However, if disjunctions of distances are not allowed (i.e., all constraints are of the form $P_i([l_1, u_1])P_j$), the resulting constraint problem (called the simple temporal problem (STP)) can be solved in cubic time using a standard algorithm for finding the shortest path between each pair of nodes in the graph (such as Floyd Warshall's algorithm, see Papadimitriou and Steiglitz, 1982). These algorithms are both correct and complete for STP. Finally, notice that constraints (10–12) (but not constraint (14)) can be easily mapped onto an STP. In particular, dates can be represented as distances from a given reference time which is unique for the overall set of constraints.

Hybrid approaches

Recently, various integrated approaches have been devised in order to deal with both qualitative and quantitative temporal constraints. Some researchers have proposed combining a standard

specialized manager dealing only with qualitative temporal constraints with another manager dealing only with quantitative temporal constraints, and to let the two managers cooperate by exchanging pieces of information whenever new constraints are inferred. However, it has not been proven that the exchange of information between the two managers ever terminates.

Others have proposed high-level constraint languages in which both qualitative and quantitative temporal constraints can be represented, using reasoning techniques based on a homogeneous constraint framework. For example, in the LaTeR temporal constraint manager (Console and Terenziani, 1999), the high-level language allows one to deal with both time points and time intervals, and to express both quantitative and qualitative temporal constraints. The language has been carefully restricted so that all the constraints can be mapped onto an STP framework. In particular, this means that LaTeR can deal with quantitative temporal constraints such as those in sentences 10–12 (but not sentence 14), with all qualitative constraints between points except the inequality relation (i.e., with CPA), and with all qualitative constraints between time intervals that can be mapped onto CPA. While the user interacts with LaTeR through the high-level language, LaTeR internally translates all constraints into an STP framework, in which it performs temporal reasoning.

Recently (Jonsson and Backstrom, 1998), a homogeneous framework has been proposed, based on linear programming, that deals with all the types of constraints discussed above, and that also allows one to express constraints on the relative durations of events, such as: 'John takes at least 30 minutes more than Fred to drive to work.'

FURTHER ISSUES

L. Vila's survey (Vila, 1994) gives an overview of other issues concerning temporal reasoning techniques.

Terenziani has presented a high-level language to model user-defined periodicity (e.g., 'the first Monday of each month'), and a constraint language which extends IA to consider qualitative relations between periodic events (Terenziani, 1997). Terenziani's approach deals with constraints such as: 'Between 1 January 1999 and 31 December 1999, on the first Monday of each month, Mary went to the post office before going to work.' Temporal reasoning over such constraints can be performed by a path consistency algorithm which extends Allen's one. Such an algorithm is sound and

operates in polynomial time, but is not complete for the high-level constraint language.

SUMMARY

Many logical approaches have been devised by AI researchers to deal with time, action and change, including the situation calculus, modal temporal logics and nonmonotonic logics. Other approaches have focused on the treatment of qualitative or quantitative temporal constraints, or both, in order to provide efficient domain-independent representation and reasoning techniques.

References

- Allen JF (1983) Maintaining knowledge about temporal intervals. *Communications of the ACM* **26**(11): 832–843.
- Console L and Terenziani P (1999) Efficient processing of queries and assertions about qualitative and quantitative temporal constraints. *Computational Intelligence* **15**(4): 442–465.
- Dechter R, Meiri I and Pearl J (1991) Temporal constraint networks. *Artificial Intelligence* **49**: 61–95.
- Emerson AE (1990) Temporal and modal logic. In: Van Leeuwen (ed.) *Handbook of Theoretical Computer Science*, vol. B, pp. 997–1072. Amsterdam: Elsevier.
- Jonsson P and Backstrom C (1998) A unifying approach to temporal constraint reasoning. *Artificial Intelligence* **102**: 143–155.
- Kowalski R and Sadri F (1997) Reconciling the event calculus with the situation calculus. *The Journal of Logic Programming* **31**(1–3): 34–58.
- Lee H, Tannok J and Williams JS (1993) Logic-based reasoning about actions and plans in artificial intelligence. *Knowledge Engineering Review* **8**(2): 91–120.
- McCarthy J and Hayes PJ (1969) Some philosophical problems from the standpoint of AI. *Machine Intelligence* **4**: 463–502.
- Papadimitriou C and Steiglitz K (1982) *Combinatorial Optimization: Algorithms and Complexity*. Englewood Cliffs, NJ: Prentice Hall.
- Terenziani P (1997) Integrating calendar-dates and qualitative temporal constraints in the treatment of periodic events. *IEEE Transactions on Knowledge and Data Engineering* **9**(5): 763–783.
- Van Benthem J (1983) *The Logic of Time*. Dordrecht: Kluwer.
- Vila L (1994) A survey on temporal reasoning in artificial intelligence. *AI Communications* **7**(1): 4–28.

Further Reading

- Allen JF (1991) Time and time again: the many ways to represent time. *Intelligent Systems* **6**(4): 341–355.
- Hajnicz E (1995) *Time structures. Lecture Notes in Artificial Intelligence 1047*. Berlin: Springer.
- Kowalski R and Sergot M (1986) A logic-based calculus of events. *New Generation Computing* **4**: 67–95.
- Lifschitz V (ed.) (1997) *Journal of Logic Programming* **31**(1–3). Special issue on reasoning about action and change.
- Long D (1989) A review of temporal logics. *Knowledge Engineering Review* **4**(2): 141–162.
- Sandewall E (1994) *Features and Fluents: The Representation of Knowledge About Dynamical Systems*. Oxford: Oxford University Press.
- Snodgrass RT (ed.), Ahn I, Ariav G, Batory D et al. (1995) *The Temporal Query Language TSQL2*. Boston, MA: Kluwer.
- Vilain M and Kautz H (1986) Constraint propagation algorithms for temporal reasoning. In: Kehler T, Rosenschein S, Filman R and Patel-Schneider P (eds), *Proceedings of the Fifth National Conference on Artificial Intelligence (AAAI'86)*, pp. 377–382. Philadelphia, PA.
- Vilain M, Kautz H and van Beek P (1990) Constraint propagation algorithms for temporal reasoning: a revised report. In: Weld DS and de Kleer J (eds) *Readings in Qualitative Reasoning About Physical Systems*, pp. 373–381. San Mateo, CA: Morgan Kaufmann.