Temporal Representation and Reasoning in Artificial Intelligence: Issues and Approaches

Luca Chittaro and Angelo Montanari
Dipartimento di Matematica e Informatica, Università di Udine, 33100 Udine, Italy
E-mail: {chittaro,montana}@dimi.uniud.it

Time is one of the most relevant topics in AI. It plays a major role in several areas, ranging from logical foundations to applications of knowledge-based systems. In this paper, we survey a wide range of research in temporal representation and reasoning, without committing ourselves to the point of view of any specific application. The organization of the paper follows the commonly recognized division of the field in two main subfields: reasoning about actions and change, and reasoning about temporal constraints. We give an overview of the basic issues, approaches, and results in these two areas, and outline relevant recent developments. Furthermore, we briefly analyze the major emerging trends in temporal representation and reasoning as well as the relationships with other well-established areas, such as temporal databases and logic programming.

1 Introduction

Time is one of the most relevant topics in AI. It plays a major role in several areas, ranging from logical foundations to applications of knowledge-based systems [51]. This ubiquity of temporal aspects in AI often leads researchers to specialize and focus on the role of time in particular contexts or applications, overlooking meaningful connections among different domains. In this survey, we deal with temporal representation and reasoning in isolation, without committing to the point of view of any specific application. Since, to the best of our knowledge, there are no surveys covering the whole spectrum of temporal representation and reasoning research, we aim at filling that gap. The organization of the paper follows the commonly recognized division of the field in two main subfields: reasoning about actions and change, and reasoning about temporal constraints. We give an overview of the basic issues, approaches, and results in these two areas, and outline some recent developments. Our goal is to highlight the main problems and the most significant contributions, limiting technicalities as much as possible. Therefore, our work is complementary with respect to more specialized surveys confined to specific theoretical issues, such as temporal logic [25, 26], temporal constraints [99, 211], nonmonotonic temporal reasoning [11], circumscription [152], the frame problem [181, 217], or to specific applications, such as temporal deductive databases [16], medical informatics [56], reasoning about actions and plans [145], and executable temporal logics [82]. Finally, as in any survey, there are some limitations to this paper. In particular, we do not consider counterfactual reasoning (a widely studied topic in philosophy and linguistics, which received some attention also in computer science, e.g., [96]), the interplay between temporal and spatial reasoning [102], and representation and reasoning techniques for periodic events [144, 187].

The paper is organized as follows. In Section 2, we consider the problem of reasoning about actions and change. After introducing the basic issues (frame, ramification, and qualification problems), we present and contrast the Situation Calculus and the Event Calculus, which are commonly recognized as the most influential formalisms. The section highlights also the major research trends such as concurrent actions and continuous change. In Section 3, we focus on the problem of reasoning about temporal constraints. We first characterize temporal constraint problems as a special case of general constraint satisfaction problems; then, we provide a comparative classification of the main formalisms with respect to their expressiveness and computational complexity. In Section 4, we briefly analyze the major emerging trends in temporal representation and reasoning as well as the relationships between this and other well-established areas such as temporal databases and logic programming.

2 Reasoning about actions and change

The field of reasoning about actions and change studies the evolution of (a portion of) the world as the result of the occurrence of a set of actions and/or events (in the following, we will use the terms action and event interchangeably). The basic reasoning mechanisms are concerned with predicting the effects of the occurrence of events as well as explaining a given situation in terms of possible causes. This kind of temporal reasoning is commonly referred to as *temporal projection*. There is a broad consensus within the AI community that temporal projection is crucial for many applications, such as planning and plan validation [31, 75, 108, 134, 239], diagnosis [131, 166], cognitive robotics [201, 218, 219], deductive databases [50, 136], agent modeling [74], qualitative physics [83, 32], monitoring [47]. A formalization of the notion of temporal projection and a systematic analysis of its computational complexity is provided in [184].

Some authors distinguish between forward and backward temporal projections with respect to an arbitrary time point [173]. *Forward temporal projection* concerns the prediction of the future and can be seen as the process of inferring consequences of specific actions on the basis of what is currently believed (using general domain knowledge). *Backward temporal projection* concerns the explanation of the past (postdiction) and can be seen as the process of inferring explanations of specific situations on the basis of what is currently believed. Explanations may consist of new links among already known facts as well as of new facts. More generally, we can say that *forward temporal projection* supports inferences of the form: "if X is true at time t_1, then Y is true at time t_2, with $t_1 \leq t_2$", while *backward temporal projection* supports inferences of the form: "if X is true at time t_2, then Y is true at time t_1, with $t_1 \leq t_2$".

In the following, we first introduce the basic issues of reasoning about actions and change; then, we present the main ontologies and formalisms proposed in the literature; finally, we discuss some of the major research topics.

2.1 Basic issues

Temporal projection is based on cause-effect relations defined among temporal entities of the application domain[1]. Cause-effect relations provide a model of change that relates the validity of effects to the validity of their causes. Forward projection leads from causes to their effects, while backward projection leads from effects to their possible causes. Forward and backward projections are usually performed by deduction and abduction, respectively [214], but alternative approaches are possible. As an example, Console et al. [57] exploit deduction to reason from effects to causes under a completeness assumption. The definition of the cause-and-effect relations of a given domain actually suffers from two main drawbacks that strongly weaken the confidence in the conclusions drawn using temporal projection: the ramification and qualification problems [11]. Both of them present a representational and a computational side.

The *ramification problem* concerns the specification of the effects of a given event, i.e., it is the problem of deciding which domain properties are affected by the event. An event can indeed result in a large number of very complex effects, and it can be very difficult to explicitly list and compute all of them. As an example [11], if one has driven a car from A to B, we have to be able to conclude that not only the car, but also all its parts (tyres, engine,...) are at B. Instead of extensively describing all the effects of an event, one could list only some of them (primary effects), and derive all the remaining effects (secondary effects) from the primary ones by exploiting suitable domain constraints. In the previous example, the primary effect of the driving event is to change the location of the car; the change in location for its parts could be derived using the part-whole relationships for the car, and a general axiom specifying that moving a compound object implies moving all its subparts. The identification of the primary effects of an event and the definition of the inference rules that allow one to compute the secondary effects are the basic issues underlying the ramification problem. Another relevant aspect is the specification of domain constraints. As the number of constraints can become very large, characterizing the exact ramification effects might become impractical.

The *qualification problem* is about the specification of the conditions under which an event actually produces the expected effects [97, 162]. If they get too large, they cannot be of practical use; at the same time, if part of them is omitted, it is possible to obtain wrong predictions about effects. To summarize, the qualification problem is concerned with how to predict effects in a sound way without taking into account too many causes.

A problem which is strongly related to ramification and qualification is that of determining the properties (normally the near-totality of the considered ones) which remain unaffected by an event (frame problem). The *frame problem* is precisely the problem of determining what stays the same about the world as time passes and actions are performed, without having to explicitly state all the things that stay the same [181]. For

[1] In the following, we will restrict ourselves to the artificial intelligence perspective on the problem of modeling causal relations, e.g. [148, 223, 228]. The investigation of the relationships between logic and causality is a classical topic in philosophical logic. A general consensus on the exact nature and form of these relationships is still lacking. As an example, the use of material implication to model causal relations has been questioned by several authors. As an alternative to material implication, some authors propose to introduce a specific meta-predicate $CAUSES$ to model causal relations, and to explicitly constrain the temporal ordering between causes and effects. This last approach has been followed by several AI researchers, e.g., [3, 128].

instance, given a situation where blocks x and y are clear, the effect of an action of stacking block x on block y is that of x being on y. To solve the frame problem, we would need to state also that block x remains clear as well as that the stack action does not change the colors and the sizes of blocks (assuming that colors and sizes of blocks are relevant to the considered application), and so on. As a general rule, we have to formalize a sort of *law of inertia* according to which relevant properties which are not affected by the action persist in their state. Even if the frame problem was originally formulated by McCarthy and Hayes in the context of the Situation Calculus [164], its universal nature was almost immediately recognized. We will discuss it further, together with the proposed solutions, in the section on temporal ontologies and formalisms.

The ramification, qualification, and frame problems are clearly not independent. If we were able to solve the qualification and ramification problems by completely specifying the preconditions of an event and its effects, we could conclude for sure that all the properties which are not affected by the event remain unchanged. Notice that the opposite is not true (unless we restrict ourselves to the propositional setting), because changes must always be explicitly specified: it is not sufficient to know what changes; we need to specify also how it changes. To cope with these problems, many approaches do not commit themselves to a causal interpretation of domain laws. They introduce weaker relations between events and properties that allow one to conclude that a property holds at a given point (interval) only if there is no evidence to the contrary. Conclusions derived in this way are clearly *defeasible*, that is, they can be withdrawn if the addition of new information makes them inconsistent. This requires the addition of nonmonotonic features to temporal reasoning systems, e.g., the replacement of classical negation with negation as failure. Such an extension, however, is not trivial at all [11]. As shown by Hanks and McDermott [111], the naive addition of nonmonotonic features to temporal reasoning systems leads to a serious difficulty, exemplified by the so-called Yale shooting problem, which we will discuss in detail in the next section.

In [225], Shoham and McDermott point out another problem (*extended prediction problem*) that weaken our confidence in the prediction of the effects of an event. The extended prediction problem concerns the temporal extension of effects. Even if we assume that a complete specification of causes in the cause-and-effect relations is available, prediction is completely safe only in the case of instantaneous effects, while extended predictions are defeasible, due to possible interferences among events. As an example, the occurrence of an inhibiting event during the elapsing time between the occurrence of a cause and its expected effects can prevent the latter from happening. In general, the shorter the temporal extension of effects, the more reliable is their prediction. At the same time, shortening the temporal extension of effects imposes the need to iterate prediction many times when the knowledge of the effects over a given time period is required. Again, there is a trade-off between efficiency and reliability in dealing with persistence of effects[2].

[2]The above presentation of the extended prediction problem differs from the original one proposed in [225], which was harshly criticized by Rayner [198], who questions Shoham and Mc Dermott's claim that Newtonian mechanics cannot be exploited to build a computational mechanism for extended prediction. In particular, he shows how the motivating example proposed by Shoham and Mc Dermott can be formulated and solved by using classical logic together with differential calculus.

Although less frequently discussed in the literature, it is worth mentioning that some additional meaningful problems of general nature have been identified, such as the divided instant problem [25, 27] and the intermingling problem [89].

2.2 Temporal ontologies and formalisms

The Situation Calculus (SC) [164] and the Event Calculus (EC) [138] are the most widely adopted and influential formalisms for reasoning about actions and change. They are the two conceptual paradigms to which almost all the various approaches proposed in the literature are related. They differ from each other in their ontologies, in the set of basic relations they represent, and in their treatment of persistence [216]. In the following, we first discuss the one basic choice every formalism is faced with, that is, reification. Subsequently, we present SC and EC in isolation, highlighting both their basic and advanced features; then, we compare them to point out major similarities and differences. Finally, a brief presentation of other relevant formalisms is included for the sake of completeness.

The reification option. A common technique exploited in many temporal representation formalisms to increase expressive power without abandoning a first-order logical framework is *reification* [3, 63, 138, 165, 221, 224]. Many problems in temporal reasoning require dealing with propositions whose truth values change as a function of time. Instead of adding one or more temporal arguments to any time-varying predicate, reified formalisms treat propositions (of the original object language) as terms of a meta-language, and relate them to time through suitable truth meta-predicates. As an example, in Allen's reified theory of actions and time [3], the fact that block A is on block B during the time interval I is expressed by the formula $HOLDS(on(A,B),I)$ instead of the formula $on(A,B,I)$. Besides assigning a special status to time, temporal reification has the advantage of allowing quantification over propositions. This feature makes it possible to express assertions like "effects never precede their causes", that cannot be captured by non-reified first-order formalisms. Given a meta-predicate $CAUSES$ taking two propositional terms as arguments, the above condition can be expressed in a reified formalism as follows [8]:

$$\forall p_2, t_2(\exists p_1(CAUSES(p_1,p_2)) \wedge HOLDS(p_2,t_2) \rightarrow$$
$$\exists p_3, t_1(CAUSES(p_3,p_2) \wedge HOLDS(p_3,t_1) \wedge t_1 < t_2)),$$

where p_1, p_2, and p_3 are propositional variables, and t_1 and t_2 are temporal variables. The formula can be interpreted as follows: if p_2 holds over t_2 and it is not "spontaneous" (that is, whenever p_2 holds, there must be some p_1 that caused it), then there must be some time t_1 preceding t_2 at which one of its causes, p_3, took place.

The major disadvantage of reified formalisms is that they require the addition of a *sort structure* to distinguish between terms that denote real objects of the domain (terms of the original object language) and terms that denote propositional objects (propositions of the original object language). Such a structure prevents the formation of meaningless terms, e.g., terms obtained by applying functions of the original object language to propositional terms. An interesting example of sorted reified formalism is provided by Lifschitz in [148]. Things become worse if one allows the reification of non-atomic propositional formulae (this happens, for instance, in Allen's reified theory of actions and time [3]):

boolean connectives become functions on (possibly complex) propositional objects, whose properties must be explicitly specified by means of suitable equality axioms [8, 223].

In [88], Galton raises a number of philosophical and technical objections to reification, and he proposes a procedure for unreifying reified temporal theories. Then, he shows that part of the lost expressive power can be recovered in a first-order framework by replacing reified (event and state) types by reified (event and state) *tokens*. While event and state types are universal concepts, event and state tokens respectively denote individual event occurrences and maximal stretches of a given state of affairs, and they can be naturally related to their occurrence/validity times. Galton's approach has been further developed by Vila and Reichgelt in [242]. They propose a token reified logic that associates meaningful names with tokens, thus allowing one to express statements about all the tokens that satisfy certain conditions.

The Situation Calculus. The basic concepts of the model of change of the Situation Calculus (SC) are those of situation and action. According to McCarthy and Hayes [164], a *situation* s is the complete state of the (modelled portion of the) world at a given instant of time[3]. SC deals with hypothetical situations that an agent can bring about from a given situation. Hypothetical situations consist of the set of all facts which are true at the considered instant of time. If the world were amenable to a complete description and complete knowledge about its initial state were available, SC would boil down to a logical counterpart of STRIPS [80]. However, since the world is typically too large for a complete description, it is not possible to actually provide the full set of facts, but only a meaningful subset of it. In SC, partial information about situations is given by means of fluents. A *fluent* is a function whose domain is the set of situations Sit. McCarthy and Hayes distinguish between propositional fluents, whose range is the set $\{true, false\}$, and situational fluents, whose range is Sit. In most cases, (propositional) fluents are values of functions. As an example, $raining(x)$ is a propositional fluent such that $raining(x)(s)$ (or, equivalently, $raining(x, s)$, if rewritten as a two variables function) evaluates to true if and only if it is raining at place x in situation s [164]. The effects of actions on situations are described by means of the situational fluent $result(p, a, s)$, whose value is the situation that results when the agent p executes the action a in the situation s.[4] For instance, a situation in which a safe sf is open can be the result of the execution of an action $open(sf, k)$ of opening the safe sf with a key k performed by a person p in a situation s where p can act on the safe sf and has the key k that fits it [164]:

$$open(sf, result(p, open(sf, k), s)),$$

[3]This notion of situation has later been reinterpreted in a number of different ways. In particular, in [24, 110, 196] it has been interpreted as the (hypothetical) period of time between two actions during which the world does not change its state. Another interpretation has been proposed by Reiter, who defines situations as possible *world histories*, simply consisting of sequences of actions [201]. For instance, $result(putDown(a), result(walk(l), result(pickup(a), s_0)))$, where s_0 is a constant denoting the initial situation, is a situation denoting the world history consisting of the sequence of actions $[pickup(a), walk(l), putDown(a)]$.

[4]The first argument of *result* disappeared in subsequent versions of SC, where information about the agent (if any) is embedded into the representation of the performed action. In the following, we will provide examples of both representations (with and without an explicit agent argument).

where the innermost occurrence of *open* denotes the action of opening and the outermost one denotes a fluent stating that the safe sf is open in the resulting situation.

The combined use of propositional and situational fluents makes it possible to express certain laws of ability, as well as certain laws of causality, as suitable axiom schemata. An example of ability law for the fluent described above is the following:

$$\forall p, k, sf, s\ (has(p,k,s) \wedge fits(k,sf) \wedge at(p,sf,s) \rightarrow open(sf, result(p, open(sf,k), s))).$$

In order to model the evolution of a given domain, besides describing the effects of actions on the values of the affected fluents, it is necessary to state that such actions do not change the values of the remaining fluents. As an example, we need to be able to express the fact that the action of rearranging the furniture of a house does not affect the color of the house [225]. Since its original formulation by McCarthy and Hayes in [164], the problem of specifying which fluents are not affected by the occurrence of a given action (called the *frame problem*, as we have already seen in the previous section), has been largely debated within the temporal reasoning community. The simple solution of adding a condition (*frame axiom*) stating that the action a does not change the value of the fluent f, for each action a and each fluent f which is not affected by a, cannot be pursued. The problem is that one needs too many frame axioms to express all the invariance conditions associated with the various actions: in principle, given n actions and m fluents, one might have to specify $n \cdot m$ frame axioms. In [164], McCarthy and Hayes outline different approaches to the frame problem. The most rudimentary one defines the notion of *frame* as a set of relevant fluents, and describes the effects of an action by explicitly specifying the subset of fluents whose values are changed by the action and thus implicitly assuming that the values of all other fluents remain unchanged. Such an approach has been later refined into a simple theory of abnormality. This refinement, which was only sketched in [164], is detailed in [161, 162]. The basic claim is that the occurrence of an event in a given situation does not affect the truth value of most fluents. As a consequence, one can assume "by default" that for any given fluent f, action a, and situation s,

$$f(s) \rightarrow f(result(a,s)).$$

To formalize this statement, McCarthy extends SC with a special predicate AB (*abnormality*) such that for any fluent f, action a, and situation s, $AB(f,a,s)$ holds if and only if the occurrence of action a in situation s causes fluent f to change its truth value. As an example [110], we can express the condition that the occurrence of an action $FallAsleep(p)$ in a situation s makes the fluent $awake(p)$ false, for any person p and situation s, as follows:

$$\forall p, s\ AB(awake(p), FallAsleep(p), s).$$

In many practical cases, we need to restrict abnormality to the set of situations satisfying certain conditions. As an example, it is abnormal that a person remains alive when a loaded gun is shot at him, but it is absolutely reasonable if the gun is unloaded. Such a "conditional abnormality" can be expressed by means of the following sentence:

$$\forall s(loaded(s) \rightarrow AB(alive, shoot, s)),$$

where, for the sake of simplicity, we omit both the *gun* argument of *loaded* and *shoot*, and the *person* argument of *alive*.

McCarthy's solution exploits the predicate AB to state that all *normal* fluents do not change their values when *normal* events occur. In order to be able to quantify over propositional fluents and actions within a first-order formalism, in the following we introduce a reified version of SC where the truth of the fluent f in the situation s is expressed by the formula $HOLDS(f, s)$. The persistence of normal fluents under the occurrence of normal events can be codified by means of the following formula:

$$\forall f, a, s(HOLDS(f, s) \wedge \neg AB(f, a, s) \rightarrow HOLDS(f, result(a, s))).$$

In this way, the frame problem is reduced to the problem of deriving $\neg AB(f, a, s)$ in any situation s, when the fluent f and the action a are normal. McCarthy proposes to minimize the abnormal temporal individuals, that is, the triplets satisfying the predicate AB, by circumscribing the predicate AB with $HOLDS$ varied. Intuitively, asserting that the extent of AB is as small as possible makes it possible to use the law of inertia above as often as possible to determine the value of fluents in $result(f, a)$[5]. Unfortunately, in [110, 111] Hanks and McDermott demonstrate that minimization of abnormalities does not meet the goal, that is, it is not sufficient to guarantee that a normal fluent does not change its truth value when moving from a situation to the next. They prove their claim using the Yale shooting problem. In the following, we provide a simplified description of such a problem, borrowed from Baker and Shoham [11], that clearly illustrates its essential features. Let us consider a scenario involving two propositional fluents, *loaded* and *alive*, that respectively concern the state of a gun and of a person, and two events, *wait* and *shoot*. Appropriate commonsense knowledge states that if a loaded gun is shot at a person, then the person dies:

$$\forall s(HOLDS(loaded, s) \rightarrow AB(alive, shoot, s)).$$

Suppose that in the initial situation s_0 the gun is *loaded* and the person is *alive*:

$$HOLDS(loaded, s_0) \wedge HOLDS(alive, s_0),$$

and then the events of *wait* and *shoot* occur in succession. Since there are no laws constraining the effects of *wait*, one would expect that it would not affect the state of the gun, that is, that the gun would remain *loaded* after its occurrence. Unfortunately, circumscribing over AB produces two different minimal models. In the first one (the expected one), *wait* does not change the state of the gun so that *shoot* produces the death of the person. Such a model contains one abnormality during the shooting event. In the second (unexpected) one, *wait* unloads the gun, and thus the person survives after *shoot*. This model also contains only one abnormality during the waiting event, while there are no abnormalities during the shooting one (it is not abnormal that a person remains alive when an unloaded gun is shot at him). The discovery of the Yale shooting problem, revealing the deficiencies of the solution to the frame problem originally proposed by McCarthy, originated several research efforts.

A solution to the Yale shooting problem in a logic programming framework, which successfully exploits negation−as−failure under stable model semantics [91], has been suggested in [76, 77]. As pointed out by Evans [77], by substituting negation−as−failure for

[5] A systematic account of the work done on circumscription is outside the scope of this survey. We refer the interested reader to Lifschitz's overview of such a topic [152].

classical negation in the standard formulation of the Yale shooting problem, one simultaneously solves the frame problem—all triples $(fluent, action, situation)$ are taken to be false where they cannot be proven to be true—and the multiple model problem–under the stable model semantics, only the intended model is selected, taking it as abnormal to be alive after shooting a loaded gun, but not abnormal to remain loaded after a period of waiting[6].
A simple solution to the frame problem in the SC framework, which benefits from ideas of Pednault [193], Haas [106], Davis [61], and Schubert [209], has been proposed by Reiter [199, 201]. On the one hand, Pednault provides a systematic, and easy to implement, method for generating all the frame axioms, but does not provide a compact representation of the resulting axioms. On the other hand, the proposals by Haas, Schubert, and Davis provide a parsimonious representation of frame axioms, by means of suitable *explanation closure axioms* [201], without explicitly providing any systematic method for generating them. Reiter's proposal combines the two solutions, thus providing both an effective method to generate frame axioms and a compact representation of them. However, it must be noted that these solutions apply only to deterministic actions without ramifications. In Section 2.4, we will briefly report on a number of other solutions to the frame problem.

As for the problem of building large-scale formalizations of real-world domains in SC, a number of shortcomings has been pointed by Hayes [113]. In that paper, Hayes lists a set of desiderata (*naive physics manifesto*) for a formalization of our knowledge of the everyday physical world. With reference to the treatment of time and change, he stresses some drawbacks of a representation based on the notion of situation, as defined in SC. For example, to describe a set of courses of events (narratives), one has to update the (global) state of the world whenever something happens in any specific narrative, even if the narratives are completely independent from each other. To overcome this limitation, he introduces the notion of *history*, which is defined as a piece of space-time with natural boundaries, both temporal and spatial. As an example [113], the event of putting four blocks together in a square is the beginning of the history of a platform, and the end of the history is when and where they are separated from one another. Additional drawbacks of the original SC have been recently highlighted by Hayes [114]. Firstly, SC is based on an overly simplistic view of the world where domain evolution has to be modeled only by means of static states and functions between them. Secondly, SC reduces change to actions, forcing the modeler to associate fictitious actions with some relevant changes, e.g., changes caused by natural phenomena. Thirdly, the way actions are described involves asserting conditions on the past and inferring conclusions about the future, and reasoning from the present to the past would require giving necessary and sufficient conditions for every action description. Several researchers have worked on the above-mentioned limitations, and, in the process, have proposed new versions of SC adding new relations, new axioms, or altering the original ones. It is thus important to keep in mind that although recent versions of SC try to preserve the basic ideas of the original calculus we have presented above, they may actually deviate significantly from it.

[6] Since applying negation−as−failure may lead to unintuitive results, e.g., as a consequence of its indiscriminate use, we may have that everything is false at the initial situation, a fine (non-trivial) tune of its application is needed.

Recent versions of SC have extended the formalism in several directions. In [196], Pinto and Reiter extend SC with the capability of dealing with information about the actual course of events; this subject has been further developed by McCarthy [163], Miller and Shanahan [168], and Shanahan [216]. Lin and Shoham [157], Alferes et al. [1], and Pinto [194, 195] deal with the problem of generalizing SC to the concurrent actions setting. Lin [158] provides a formal treatment of nondeterministic actions. Scherl and Levesque [208] provide a SC account of sensing actions; then, Bacchus, Halpern, and Levesque [5] extend it to cope with noisy sensors; finally, Levesque [146] provides a SC solution to the problem of planning for agents that can sense their environments. Shapiro, Lespérance, and Levesque [220] model agent goals and rational actions in SC. A uniform SC framework for concurrency, natural actions, and continuous time to model complex dynamic domains and agent behaviors has been proposed by Pinto [194], Ternovskaia [233], and Reiter [200].

The Event Calculus. The Event Calculus (EC) was originally introduced by Kowalski and Sergot as a logic programming framework for representing and reasoning about events and their effects [138]. An axiomatization of EC in classical logic, using circumscription as a method for default reasoning to solve the frame and related problems, is given in [217] and summarized in [169] (where alternative axiomatizations of EC in classical logic are described). The basic concepts of the model of time and change of EC are those of event and property. *Properties* are fluents which hold during time periods initiated and terminated by *event* occurrences. The predicate *initiates(e,p)* is used to state that event e initiates property p. For example, the event *hire(x,y)* of hiring a person x as y *initiates* a period of time during which the property "x has rank y" holds. According to the original EC proposal [138], this fact is represented as $HOLDS(after(hire(x,y), rank(x,y)))$, where $after(event, property)$ denotes a time period, started by *event* and whose end is undefined, over which *property* holds. Properties are assumed to persist until an event occurs that interrupts them (*future persistence*). In a similar way, *terminates(e,p)* means that the event e terminates the property p. As an example, the event *promote(x,y,z)* of promoting x from rank y to rank z *terminates* a period of time during which the property "x has rank y" holds. This fact is represented as $HOLDS(before(promote(x,y,z), rank(x,y)))$, where $before(event, property)$ denotes a time period, ended by *event* and whose beginning is undefined, over which *property* holds. Since the original EC formulation treats future and past symmetrically, the occurrence of a terminating event allows one to conclude that the property holds before that event (*past persistence*), possibly up to an initiating event. From a description of events which occur in the real world and of properties that they initiate or terminate, EC derives that a given property p holds over a finite time period whenever it is known that p holds after an event e and before an event e', e precedes e', and no event occurring between e and e' affects p. In such a case, the periods $after(e,p)$ and $before(e',p)$ coincide. In the above example, assuming that no other events affecting x's rank occur between $hire(x,y)$ and $promote(x,y,z)$, it can be concluded that $rank(x,y)$ holds between them. Initiation and termination of properties can be made *context-dependent* by requiring that some preconditions hold at the event occurrence time, e.g. [50].

Most of the subsequent literature on EC simplifies the original formulation of the calculus, without any loss of generality, using time points as the basic temporal elements, and replacing time periods with pairs of time points (*intervals*). Along this line, many variants of EC have been proposed by different authors [50, 65, 75, 136, 214]. In [202], Sadri and Kowalski propose a version of EC that captures the common features of all these variants. In the following, we present a logic programming formulation of EC which adheres to this unifying perspective. The notions of event, property, time point and interval are the primitives of the formalism: *events* happen at *time points* and initiate and/or terminate *intervals* over which *properties* hold. A real line of time points is assumed, and events happen at time points belonging to it. An event occurrence can be represented by associating the event with the time point at which it occurred by means of the *happens(event,timePoint)* clause. The relation between events and properties can be defined by means of *initiates* and *terminates* clauses of the following form:

$$\begin{aligned}&initiates(event, property, T) \leftarrow\\&\quad happens(event, T) \wedge holds(prop1, T) \wedge \ldots \wedge holds(propN, T).\end{aligned}$$

$$\begin{aligned}&terminates(event, property, T) \leftarrow\\&\quad happens(event, T) \wedge holds(prop1, T) \wedge \ldots \wedge holds(propN, T).\end{aligned}$$

This *initiates* (*terminates*) clause states that each event of type *event* initiates (terminates) a period of time during which *property* holds, provided that N (possibly zero) given preconditions hold at instant T. The model of time and change of EC is concerned with determining the time points at which properties hold. For example, consider Figure 1a with one initiating event (marked with i) and two subsequent terminating events (*t1* and *t2*) for a property p: p holds between 1 and 3, and it cannot hold between 1 and 4, because *t1* would be an interrupting event which would clip the interval. The model of time and change of EC can be defined by means of the following axioms:

$$\begin{aligned}&holdsAt(P, T) \leftarrow\\&\quad initiates(Ei, P, Start) \wedge Start < T \wedge \neg clipped(Start, P, T).\end{aligned}$$

$$\begin{aligned}&clipped(T1, P, T3) \leftarrow\\&\quad terminates(Et, P, T2) \wedge T1 \leq T2 < T3.\end{aligned}$$

The *holdsAt* axiom states that a property P holds at a time T if event Ei initiates P and occurs before T, provided that there is no known interruption in between[7]. The negation in front of the *clipped* predicate is indeed interpreted using negation-as-failure. The *clipped* predicate states that a given property P is interrupted during the time-interval *[T1,T3]* if there is an event Et that happens between them and terminates P. This axiom provides the so-called *weak interpretation* of the *initiates* predicate [50]: an initiating event initiates a validity interval for a property P, provided that the property does not already hold at the time point of its occurrence. For example, this interpretation is the one applied in Figure 1b, where two initiating events (*i1* and *i2*) are followed by a terminating event (t) and the validity interval for the property is [1,4]. As a more general case, consider an ordered sequence of events $e_1, \ldots, e_n$, each one initiating the same property p, followed by an event e_{n+1} terminating p. According to a weak interpretation, EC derives the validity of p between the outermost initiating event e_1 and the terminating event e_{n+1}.

[7]Note that the *holdsAt* axiom conventionally assumes that a property is not valid at the starting point of its validity interval, while it is valid at its ending point. For example, in Figure 1a, p is valid for any instant greater than 1, and less than or equal to 3.

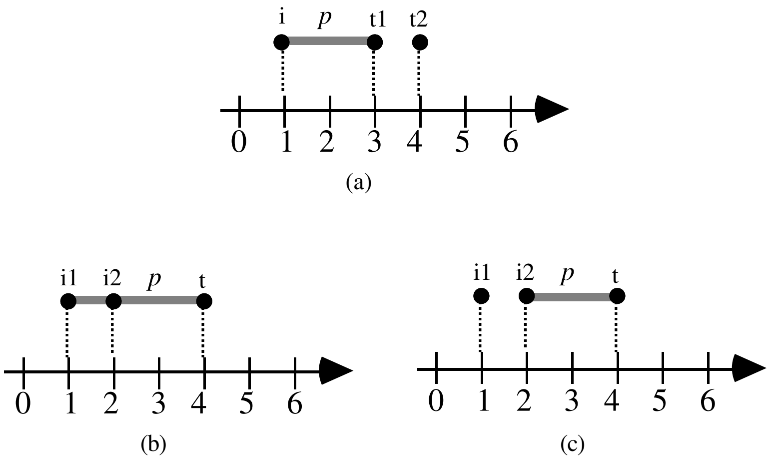

Figure 1: Examples of (a) validity interval, (b) weak and (c) strong interpretation.

The weak interpretation is needed to aggregate homogeneous states. Consider the problem of monitoring patients who receive a given therapeutic assistance [47]. In this context, it often happens that data acquired with two consecutive samplings do not cause a transition in the classification of the patient state. Adopting the weak interpretation, the second data acquisition does not clip the validity interval of the condition of the patient initiated by the first. The weak interpretation is more frequently found in the literature and also proved more useful in our experience with medical applications [44, 45, 46, 47].

An alternative interpretation, called *strong interpretation*, can also be adopted (this is the interpretation originally adopted by Kowalski and Sergot [138]). According to it, a given property P is clipped at some point T during the time-interval *[Start,End]* if there is an event E which *initiates* or *terminates* P occurring at time T. Applying it to the same three events of Figure 1b, we would thus obtain the validity interval depicted in Figure 1c. As a further example, consider again an ordered sequence of initiating events $e_1, \ldots, e_n$, followed by the terminating event e_{n+1}. According to the strong interpretation, EC would derive the validity of p only between the innermost initiating event e_n and e_{n+1}. To support a *strong interpretation* the definition of *clipped* must be slightly extended so that also initiating events can clip the validity of property P.

The basic version of EC guarantees the correctness of derived validity intervals only if complete knowledge about the set of occurred events and their temporal order is available; otherwise, it may produce incorrect results. In the following, we distinguish three types of incompleteness which can (also jointly) affect a given domain. More precisely, we consider the possibility of having incomplete information about which events happened (*missing events*), incomplete information about the ordering of events (*partially ordered events*), and indeterminacy in the time of occurrence of some events (*temporal indeterminacy)*.

To give an example of an incomplete sequence of events, we consider a switch that can take two different positions (on and off). Its behavior can be described by means of two events: one that changes the position from off to on, the other from on to off. While two 'turn on' events cannot occur consecutively in the real world, it may happen that an incomplete sequence consisting of two consecutive 'turn on' events, followed by a 'turn off' event, is recorded in the database. To deal with this kind of problems, Kowalski and Sergot [138] propose to introduce suitable rules to derive the existence of implicit end points. For example, in the switch case, they would conclude that there must be a terminating event for the switch being in the on position, whose occurrence time lies somewhere in between the occurrence times of the two 'turn on' events. The strong interpretation [50] is another technique that can be used to deal with incomplete sequences of events. The strong interpretation of *initiates* allows EC to recognize that a missing 'turn off' event must have occurred between the two 'turn on' events. However, since it is not possible to temporally locate such an event, we conclude only that the switch is certainly on between the second 'turn on' event and the 'turn off' event, and the first 'turn on' event is considered as a *pending* initiating event.

We now deal with the case where there is only partial information about the set of occurred events and their occurrence times [41, 62, 171, 184]. For example, consider a set of three events, where the available ordering information allows one to conclude that event *e1* has occurred before *e2*, event *e3* has occurred before *e2*, and nothing can be derived about the ordering between *e1* and *e3*: *e1* can precede *e3*, *e3* can precede *e1*, or they can occur simultaneously. Different techniques for reasoning about partially ordered events can be devised, producing different kind of outputs. A first distinction can be given by the traditional modal operators of possibility and necessity. Two variants of EC, namely the skeptical EC and the credulous EC, that respectively derive which properties are necessarily and possibly true in the presence of partial event ordering, are presented in [34, 53]. Dean and Boddy [62] studied the task of deriving which facts must be or may be true over certain time intervals (in our context, which maximal validity intervals necessarily or possibly hold) in the presence of partially ordered events, focusing on the computational complexity of the task and showing that it is intractable in the general case. To tackle this problem, they propose to develop polynomial algorithms to compute subsets of the set of necessary facts, or supersets of the set of possible facts. Another possibility is obviously that of imposing restrictions on the possible input, in order to derive the full sets of necessary and possible validity intervals in polynomial time. A systematic analysis of the problem of reasoning about partially ordered events in EC has been started in [53]. This resulted in the definition of a uniform framework encompassing a hierarchy of modal event calculi of different expressiveness and complexity [34, 35, 36, 38, 39, 40, 84, 85].

Temporal indeterminacy refers to an incomplete knowledge about the precise instant at which an event happened. For example, we may know that an event happened "between 10 and 13". To the best of our knowledge, the only approach to introduce indeterminacy in EC is described in [43, 44]. In that approach, indeterminacy in locating an event occurrence on the time axis is modeled by an interval (*interval of indeterminacy*), consisting of all the time points possibly coinciding with the considered, indeterminate event occurrence. The definition of validity interval changes accordingly to the more general concept of event. Since a property is initiated and terminated by events, the occurrence time of these events can now be indeterminate, and a validity interval has the form $\langle Start, End \rangle$ where *Start* (*End*) denotes either a specific instant, when it is known, or the *minimal* interval over which the initiation (termination) of the property is necessarily located in the case of indeterminacy. An interesting point here is that *Start* and *End* can also be non-convex intervals: by combining the different intervals of indeterminacy associated with initiating and terminating events in order to derive necessary validity intervals, the derived conclusions can be, for example, that a property necessarily holds between 9 and 11, and its initiating instant is necessarily located between 2 and 3 or between 8 and 9. Since temporal indeterminacy may result in partially ordered sequences of events, Boddy and Dean's general results discussed above apply also to reasoning about events with temporal indeterminacy.

On the relations between Situation Calculus and Event Calculus. The concepts of *property* and *event* can be viewed as the EC counterparts of the concepts of *fluent* and *action* of SC. However, SC relates fluents to *situations*, while EC relates properties to *time points* belonging to the real time line. Since situations are hypothetical, SC supports temporal reasoning about possible sequences of actions. On the contrary, time points are actual and EC can only be used to reason on actual sequences of events (narratives). The two main differences between EC and SC are thus that (i) EC introduces a real line of time points, and properties may or may not hold over each point, and (ii) EC has no explicit notion of situation: events happen at time points, and initiate and/or terminate intervals over which *specific* properties hold.

Another important difference between EC and SC concerns the representation of time itself. SC adopts a *branching time* representation, where each branch is a discrete sequence of situations, and change is an instantaneous transition between two situations. EC is based on *linear time*: there exists a unique timeline, and the period between any two events is a segment of time points where each time point exists independently. As pointed out in [24], the linear time structure of EC has the advantage of facilitating representation of continuous change with respect to SC, and the disadvantage of making hypothetical reasoning more difficult. More specifically, the difficulty lies in the representation of hypothetical (as well as counterfactual) statements of the form: "if A would have happened, then B would hold", which can be represented correctly in SC's branching time. In [23], van Belleghem et al. point out these difficulties by using a simple variant of the Yale Shooting problem. They consider a scenario where there are a living turkey and a loaded gun; then a spin event with a non-deterministic effect (it may or may not unload the gun) and a shoot event occur. Given the additional information that if the spin event were followed by waiting instead of shooting, the gun would still be loaded, the problem is to determine whether the turkey is alive after the shooting event. The conclusion has to be that the turkey is dead, because the additional information allows us to conclude that the gun was loaded during the shooting, which results in the turkey being dead. In order to solve this problem, information about two different sequences of events (the actual one and a hypothetical one) has to be combined. They show that this is not possible within plain EC, because only one sequence of events can be maintained, and the problem can only be solved by adding a meta-reasoning level (which is not a solution within the formalism). On the contrary, in the branching time structure of SC, all hypothetical sequences of events can be maintained, and information from different branches can be combined within the formalism.

In [137], Kowalski and Sadri have shown that, by adding appropriate axioms, SC and EC imply one another, and thus they can be considered formally equivalent. This does not mean that they are *practically* equivalent, in the sense that the use of EC or SC is equally natural and flexible for any problem. For example, since Kowalski and Sadri had to resort to induction to derive SC from EC, while there was no such a need for the derivation of EC from SC, they conjecture that EC might be more expressive.

Although for many years SC and EC have been presented as alternative approaches, it is now recognized that they simply focus on different aspects and intuitions, and are suited to solve different problems. This awareness is prompting attempts to unify the two approaches. For example, Pinto and Reiter show how to introduce a real timeline into SC [196]; Provetti [197] and Kowalski and Sadri [137] show how EC can inherit SC features useful for hypothetical reasoning; van Belleghem et al. [24] combine both calculi with the aim of retaining all their advantages under a common framework.

Other relevant formalisms. Even though most formalisms for reasoning about actions and change can be brought back to either SC or EC, there are a few significant exceptions that deserve to be discussed. In the following, we briefly describe Sandewall's *features and fluents* formalism [206], Hölldobler and Thielscher's *fluent calculus* [117, 238], the language $\mathcal{A}$, introduced by Gelfond and Lifschitz in [92], and Kakas and Miller's language $\mathcal{E}$ [122].

In [206], Sandewall proposes a systematic approach to the representation of knowledge about dynamic systems, called *features and fluents* framework, which aims at formally characterizing the range of applicability of the different nonmonotonic logics of actions proposed in the literature. Its distinctive characteristic is that of making explicit the ontological and epistemological assumptions about the world, the actions, and the agent's knowledge about them. Ontological assumptions may include the validity of the law of inertia; the possibility that an action produces alternative effects, depending on the state in which it occurs; the admissibility of concurrent actions; the possibility of representing dependencies among the effects of an action. Epistemological assumptions mainly concern the correctness and completeness of the available knowledge. The features and fluents framework associates with each nonmonotonic logic the set of its ontological and epistemological assumptions. This association makes it possible to characterize the class of reasoning problems for which any given logic obtains exactly the intended conclusions. The basic features and fluents framework has been later extended with high-level features to represent complex scenarios. As an example, the problem of enriching the basic framework, that only deals with actions that occur sequentially, with concurrent actions has been addressed by Yi in [248] and by Gustafsson and Karlsson in [105]. Yi provides an extension of Sandewall's framework to take into account concurrent actions with duration. However, he restricts himself to the case of actions that do not change each other's course (non-conflicting actions). Gustafsson and Karlsson propose a logic of actions and change, called *TAL-C*, that deals with a number of phenomena related to concurrency, including interferences between the effects of concurrent actions, bounds on concurrency, conflicting, synergetic, and cumulative effects of concurrent actions.

In the following, we present the basic elements of the features and fluents approach; a more detailed description can be found in [205, 206]. The *features and fluents* formalism represents the evolution of a dynamic system in terms of the interactions between an agent and the relevant portion of the world. These interactions are modeled as a *game* in which the agent and the world alternate in such a way that the world executes the actions that are commanded by the agent. The game is formalized by means of the notion of *finite development*. A finite development is a quintuple $\langle \mathcal{B}, M, R, \mathcal{A}, \mathcal{C} \rangle$, where $\mathcal{B}$, $\langle M, R \rangle$, and $\langle \mathcal{A}, \mathcal{C} \rangle$ respectively represent the current temporal extension of the game, the corresponding temporal evolution of the object domain, and the set of actions (possibly partially) performed up to the current time. More precisely, $\mathcal{B}$ is a finite prefix of the set of non-negative integers $\{0, 1, \ldots, n\}$, representing the time points at which the agent and the world alternate, whose largest element n is interpreted as the current time; M is a function that assigns a value to temporal and object constants; R is a sequence of $n+1$ states $s_0, s_1, \ldots, s_n$ describing the history of the world up to the current time n; $\mathcal{A}$ and $\mathcal{C}$ respectively denote the sets of accomplished and ongoing actions. Each accomplished action is represented by a triplet $\langle s, E, t\rangle$, where E is an action, s $(< t)$ is its starting time and t $(\leq n)$ is its ending time, while each ongoing event is represented by a pair $\langle s, E\rangle$, where E is an action and s $(\leq n)$ is its starting time. The effects of executing an action are described by means of the notion of *trajectory semantics*. It is defined by introducing two functions, *Infl* and *Trajs*, such that for any action E and any state r, $Infl(E, r)$ represents the set of features which are affected by the occurrence of E in r and $Trajs(E, r)$ is the set of possible trajectories for E initiated in r. Given an accomplished action $\langle s, E, t\rangle$, with $t = s + k$, a trajectory for E initiated in r is a finite sequence of partial states $r_1^{'}, r_2^{'}, \ldots, r_k^{'}$ that describes the effects of the action execution by assigning a value (only) to the features belonging to $Infl(E, r)$ at each time $s + i$, with $1 \leq i \leq k$. In the features and fluents framework, any specific theory of actions and change Υ is a tuple $\langle \mathcal{O}, \Pi, SCD, OBS\rangle$. $\mathcal{O}$ is an object domain and Π, which provides a logical counterpart of the function $Trajs$, is a set of formulae describing the effects of actions. As an example [248], the fact that the execution of the action of opening the window over the time periods $[s, t]$ has the effect of making the window open at time $[t]$ can be expressed as follows:

$$\forall s, t([s, t]OpenWindow \rightarrow [t]Window = Open).$$

SCD consists of a set of action statements, each one specifying a scheduled action and its execution time, e.g., $[s_1, t_1]OpenWindow$, and of a set of temporal constraints, e.g., $t_1 < s_2$. The effects, denoted by $\Pi(SCD)$, of actually executing the scheduled actions can be determined by pairing the universal conditions specified in Π with the action statements belonging to SCD. As an example, the execution of the action $[s_1, t_1]OpenWindow$ has the effect $[t_1]OpenWindow$ which can be derived by first substituting the constant symbols s_1 and t_1 for the variable symbols s and t in the formula $\forall s, t([s, t]OpenWindow \rightarrow [t]Window = Open)$, and then exploiting the action statement $[s_1, t_1]OpenWindow$. Finally, OBS is a set of observation statements.
The set of intended models of a theory Υ is built up as follows. We first select a world whose behavior is captured by Π, an agent, an initial state of the world, and an initial mapping for temporal and object constants. Then, we identify the set $mod(\Upsilon)$ consisting of those finite developments $\langle \mathcal{B}, M, R, \mathcal{A}, \mathcal{C}\rangle$ that correspond to the games between the agent and the world in which all the scheduled actions have been executed (complete games). Complete games are formally characterized as those games for which there exist a bijection between $\mathcal{A}$ and SCD. The set of intended models consists of all the pairs $\langle M, R\rangle$ such that $\langle \mathcal{B}, M, R, \mathcal{A}, \mathcal{C}\rangle$ belongs to $mod(\Upsilon)$.

The *Fluent Calculus* (FC for short) [117, 238] is based on the idea that the facts that characterize a situation (i.e., the fluents that hold in it) can be represented as *resources*, which are consumed and produced by actions. An action can consume resources whenever the conditions of the action are to be satisfied in the situation, while resources can be produced as effects of the action.

The model of change of FC is closely related to that of SC. Facts about a world are modeled as positive or negative *fluents*. Conflicts in the world, e.g., the impossibility for an individual to be both alive and dead at the same time, are dealt with by means of *domain constraints*. For instance, the following domain constraint solves the alive/dead conflict: $alive \wedge dead \leftrightarrow alive \wedge \neg alive \leftrightarrow \bot$. World *states* are defined as maximal sets of positive and negative fluents (fluent literals), which are consistent with the given domain constraints. A *dynamic system* consists of a set of fluents $\mathcal{F}$, a set of actions $\mathcal{A}$, a set of domain constraints $\mathcal{D}$, and a (partial) state transition function $\Phi : \mathcal{A} \times \mathcal{C} \mapsto \mathcal{C}$, where $\mathcal{C}$ is the set of states. The system is said to be deterministic if and only if Φ is deterministic. Each triplet $(a, z, z') \in \Phi$ (according to a relational reading of Φ) is represented by the *action description* $\langle c, a, e\rangle$, where $c \subseteq z$ and $z' = (z \setminus c) \cup e$. The sets c and e respectively denote the conditions consumed by a and the effects it produces. As a general rule, an action description $\langle c, a, e\rangle$ can be applied to a state z if and only if $c \subseteq z$, and its application yields the successor state $z' = (z \setminus c) \cup e$. If there are two or more action descriptions which are applicable to a given state s, the most specific one is chosen, where an action description $\langle c_1, a, e_1\rangle$ is more specific than a description $\langle c_2, a, e_2\rangle$ if and only if $c_2 \subseteq c_1$.

If the world were a deterministic dynamic system fully accessible to an agent, computing complete action descriptions and applying them accordingly, starting from a given initial state, would be sufficient. Unfortunately, as it is impossible to completely describe the world at a particular time or to completely specify an action, an agent has generally only partial knowledge about the current state as well as about the available actions. To deal with partially specified states, FC (as SC) introduces the notion of *situation* as a consistent subset s of a state z. Hence, any given situation s implicitly represents all states z' such that $s \subseteq z'$. As a situation is a set of fluents, action descriptions can be applied to situations as well. However, it may happen that the situation s', which can be reached by applying the action a to the current situation s, represents a set of states which cannot be reached by applying a to the states represented by s. Similarly, conditions and effects of an action description may be only subsets of the actual conditions and effects. As a consequence, it may happen that the application of a partially specified action description to a (consistent) situation may lead to an inconsistent scenario. In [117], Hölldobler analyzes and exemplifies these problems, and shows how FC deals with (part of) them. In particular, he discusses the consistency problem that arises when domain constraints enforce dependencies which are not reflected in action descriptions. He proposes a solution based on the automatic derivation of *causal relationships* from the given domain constraints to be used to restore consistency [234].

FC has been formalized as an equational logic program, where fluent literals are represented as resources and, consequently, situations are represented as multisets of resources, while action descriptions are represented by facts (ground atoms) [116, 118, 120, 235]. More formally [117], fluent literals are reified and represented by constants. The empty multiset is denoted by the constant $\emptyset$, while non-empty multisets are obtained by using a suitable binary function symbol $\circ$, which is required to be associative, commutative, and to admit the constant $\emptyset$ as a unit element. The resulting equational theory AC1 (Associative, Commutative, and with a unit element) is built into the unification mechanism of equational logic programming. FC represents a situation $s = \{f_1, \ldots, f_n\}$ by a term $\tau_s = f_1 \circ \ldots \circ f_n$, where every f_i, with $i = 1, \ldots, n$, is a fluent, and an action description $\langle c, a, e\rangle$ by a fact $action(\tau_c, a, \tau_e)$. The problem of establishing whether or not an action description $\langle c, a, e\rangle$ can be applied to a situation s (that is, to check whether $c \subseteq s$) can be easily reduced to the AC1-unification problem. Furthermore, the problem of verifying whether $\langle c, a, e\rangle$ is not the most specific action description for A that can be applied to s can be codified by a simple equational logic program. Finally, the fact that an initial situation s can be turned into a goal situation g by applying a sequence of actions $a_1, \ldots, a_n$ is captured in FC by means of a suitable ternary predicate $causes(\tau_s, [a_1, \ldots, a_n], \tau_g)$ defined as an equational logic program.

Thanks to the representation of fluent literals as resources, FC does not need specific additional axioms to compute the set of fluents which are not changed by actions, because it solves the frame problem by mapping it onto the unification problem of the underlying equational theory AC1 [117]. Furthermore, FC deals with the ramification problem by a suitable postprocessing: once an action a has been applied to a situation s, causal relationships are applied until a consistent set of resources is obtained [234, 237]. The qualification problem in FC is discussed in [236].

In [92], Gelfond and Lifschitz introduce the declarative language $\mathcal{A}$ for describing the effects of actions, and provide it with a semantics based on the notion of finite automata. Its ontology is quite similar to that of SC. Indeed, the alphabet of $\mathcal{A}$ consists of two disjoint sets of symbols: fluent and action names. To model a domain, $\mathcal{A}$ provides two types of expressions: v- and e-propositions. A *v-proposition* is an expression of the form: f **after** $a_1, \ldots, a_m$, where f is a fluent name possibly prefixed by $\neg$ (fluent literal) and $a_1, \ldots, a_m$, with $m \geq 0$ are action names. Whenever $m = 0$, v-propositions take the form: **initially** f, and state that f holds in the initial situation. An *e-proposition* is an expression of the form: a **causes** f **if** $p_1, \ldots, p_n$, where a is an action name, and $f, p_1, \ldots, p_n$, with $n \geq 0$, are fluent literals. The literals $p_1, \ldots, p_n$ are called preconditions. As observed by Kakas and Miller [122], the primary motivation underlying the proposal of $\mathcal{A}$ was to provide a language whose syntax and semantics are simple enough to be regarded as uncontroversial and intuitive, and to exploit it to characterize advantages and limitations of other formalisms by defining general translation procedures from $\mathcal{A}$ domains into them. As an example, Kartha uses translations from $\mathcal{A}$ to prove the equivalence of three well-known SC-based formalizations with respect to a relevant class of domains [123]. The limited expressive power of basic $\mathcal{A}$ has been improved in several directions, e.g., [13, 14, 119, 125]. In particular, in [13] Baral and Gelfond propose an extension of $\mathcal{A}$ that deals with concurrent actions by removing the assumptions that (i) every action is executable in any situation and (ii) only one action can be performed at a time.

In [122], Kakas and Miller propose a new declarative language, called $\mathcal{E}$, for reasoning about actions that occur at different times, and about the properties that hold or do not hold as a consequence of these occurrences. Such a language shares general motivations and goals with $\mathcal{A}$; however, unlike $\mathcal{A}$, it assumes a narrative-based ontology that closely resembles that of EC. Syntax and semantics of $\mathcal{E}$ are similar to those of $\mathcal{A}$. In particular, its model-theoretic semantics is based on a simple characterization of persistence that allows one to easily extend the expressiveness of $\mathcal{A}$ in a modular way. The relationships between $\mathcal{E}$ and $\mathcal{A}$, as well as between $\mathcal{E}$ and $\mathcal{L}_0$ (a narrative extension of $\mathcal{A}$ proposed by Baral et al. [14]), are analyzed in detail by Kakas and Miller in [122].

2.3 Major extensions

In order to model complex real world domains, formalisms for reasoning about actions and change need to be extended in various directions. In this section, we briefly discuss two of the major extensions studied in the literature: continuous change, and concurrent actions.

Discrete and continuous change. While formalisms for reasoning about actions and change typically focus on discrete fluents (e.g. a light that can be either on or off, a block X that can be either on a block Y or not,...), it is easy to find examples of fluents for which a representation of continuous change can be required in real applications (e.g., the position of a moving object, the level of a tank,...). This kind of change is traditionally represented with equations, which unfortunately do not fit well into formalisms like SC or EC. For example, the continuous change of the level of a tank cannot be simply modeled by an event in EC or an action in SC. The event (action) can at most start a discrete fluent which states that the level is rising or falling, but all the changes in value which occur during that fluent need to be represented by a proper equation. An integration of this kind has been proposed for EC by Shanahan [215], who introduces in EC the notion of *trajectory* of a continuously changing quantity through a space of values. A trajectory describes the functional relationship between the value of a quantity and the time it elapsed since it started to change. A trajectory is attached to a discrete fluent (e.g., moving, rising, falling,...) by means of proper axioms, which have the purpose to ensure that the trajectory is valid only when the associated fluent holds. A similar solution has been provided in SC by means of the notion of non-boolean fluents [200], that is, fluents which are evaluated on domains other than $\{true, false\}$.

It is worth noting that while the temporal representation and reasoning community was concentrating mainly on a logical representation of discrete change, the qualitative physics community was focusing mainly on an algebraic representation of continuous change. The concern for continuous change in qualitative physics is motivated by the tasks it considers, especially qualitative simulation, i.e. the construction of a set of possible behaviors consistent with a model of a dynamic system represented by qualitative versions of differential equations (see e.g. [133, 139]). While the algebraic approach is good at handling continuous change, it is not suited to represent even simple relations among things, such as the fact that "a block is red" or that "John gave the book to Mary" [215]. This suggests that an integration of the two approaches can be beneficial to both communities. Indeed, researchers in the temporal representation and reasoning community are working at introducing algebraic expressions into logic-based temporal representation and reasoning approaches (e.g. [203, 215]), and researchers in the qualitative reasoning area are exploiting ideas from reasoning about actions and change inside their frameworks. For example, the Qualitative Process Theory proposed by Forbus [83] allows giving a specification of the situation (in particular, the relations among individuals) for which it is appropriate to use a given qualitative differential equation. With QPC (Qualitative Process Compiler), Farquhar [79] implements and extends Qualitative Process Theory. In QPC, a physical situation is described by a set of *model fragments*, each of which captures some aspects of the domain, by providing the appropriate knowledge of both algebraic and logical nature. A model fragment definition specifies a set of participants (e.g., a pot and a gas ring), the relations which must hold in order to instantiate the fragment (e.g., the pot must be on the ring), and the consequences (e.g., there is an heat flow between the ring and the pot, expressed by a given equation). A specific system or situation is given to the simulator by listing the entities involved and their relationships, together with some initial conditions. Starting from this specification, the system is able to instantiate the proper model fragments and simulate the system. In case the predicted behavior extends across the boundary of a fragment applicability, a new model of the resulting situation is dynamically constructed by considering other fragments. Since QPC is built on top of the well-known qualitative simulator QSIM [139], it inherits the capability of building a space of possible behaviors when the available knowledge does not allow to predict a single outcome.

Concurrent actions and processes. Most formalisms for reasoning about actions and change make the simplifying assumptions that (i) only one action can be performed at a time[8] and (ii) actions are instantaneous. When we remove the first assumption, it must be taken into account that concurrent actions may interact. Interactions may lead to both synergistic effects (their combined effect is more than the sum of their individual effects) and interferences (their individual effects may be partially or totally cancelled). For example [4], if one agent lifts one end of a piano, while another agent lifts the other end, then the entire piano is lifted off the floor; on the contrary, if one agent pushes a door to open it, while the other is pushing on the door the other way, the two actions cancel each other out. To handle these situations, many formalisms have been provided with the possibility of giving explicit axioms to model interaction among concurrent actions, e.g. [12, 13, 94, 105, 157, 248]. If we also remove the assumption that actions are instantaneous, occurrences of actions may overlap partially [168]. In such a case, the length of the partial overlap may influence the outcome of the interacting actions.

Modeling real-world domains may require representing complex patterns of actions, which, besides sequentiality and concurrency, involve additional relations among actions, such as the exclusive occurrence of just one action in a given set, the occurrence of at least one action in a given set, the iteration of occurrences of the same action or of a pattern of actions. The notion of *process* has been introduced to define compound actions in terms of patterns of simpler actions. Simpler actions can be grouped together into one process by specifying relations which have to hold among them. As an example, the action of dialing an eight-digit number can be defined as a set of eight basic actions in strict sequence relation. Foundational work on process modeling, which inspired research in many computer science areas, including knowledge representation, has been done by Hoare [115] and Milner [170]. In temporal reasoning, these approaches has been followed, for example, by Belegrinos and Georgeff [22], and by Chittaro and Montanari [49].

2.4 Entailment methods

As pointed out in Section 2.2, the Yale shooting problem definitely revealed the limitations of the solutions originally proposed to the frame problem, such as Reiter's default logic and McCarthy's circumscription, and originated a great deal of research mainly on the identification of suitable selection criteria able to rule out unexpected models[9]. A

[8] Some formalisms actually allow two or more actions to be performed at the same time, but they assume that simultaneous actions do not influence each other.

[9] Another direction consists in making circumscription "local" to a set of axioms by which we *define* a given relationship. In [153], Lifschitz presents a SC theory of actions which is proved correct by applying circumscription to subtheories of it. Although a theory embedding several circumscriptive subtheories may result rather puzzling, Lifschitz shows that in several cases such theories admit a first-order equivalent.

general discussion of the frame problem and a survey of several of the trends in solutions to the frame problem are provided by Morgenstern in [181][10], while a systematic mathematical investigation of the large variety of solutions to the frame problem proposed in the literature is performed by Shanahan in [217]. In the following, we will briefly review some of the proposed solutions, limiting ourselves to the most significant ones.

An attempt to repair the deficiencies of basic default logic is done by Hanks and McDermott in [111]. They add the requirement that a sentence ϕ can be inferred from a default logic if and only if ϕ belongs to every extension of that theory. As noticed by Evans [77], however, such a requirement only expresses a form of logical soundness, and it is generally too strong. Hanks and McDermott's approach has been later refined by Morris [182] and Gelfond [90] with the introduction of nonnormal default rules. Unfortunately, this solution is not fully satisfactory, since it does not handle temporal explanation and ramification problems.

Another set of solutions, usually grouped together under the expression *chronological minimization*[11], adopts a selection criterion according to which the preferred models are those in which abnormalities occur as late in time as possible. Such a criterion amounts to assigning a higher priority to the minimization of abnormalities that happen at earlier situations than to the minimization of abnormalities that happen at later situations. It is not difficult to check that chronological minimization actually solves the Yale shooting problem. Nevertheless, it is not a completely satisfactory solution. As basic circumscription, it is indeed unable to deal with temporal prediction problems involving *surprises* [126] as well as temporal explanation problems [11]. Furthermore, there are scenarios for which chronological minimization behaves even worse than basic circumscription (e.g., see Baker's murder mystery example [9]).

A third set of solutions provide domain description languages with an explicit notion of causality, and modify the minimization criterion to distinguish between abnormalities for which there is a cause and abnormalities which are not supported (*causal minimization*). This is the approach followed by Haugh [112] and by Lifschitz [148, 150]. They also allow constraining the effects of an action in a given situation to the validity of the set of its preconditions. Then, both causes and preconditions are circumscribed with $HOLDS$ varied, the circumscription of preconditions being an attempt to solve the qualification problem. The major limitation of causal minimization is that it does not allow any of the effects of an action to be implicitly described by means of domain constraints [11], thus ignoring the ramification problem.

In [10], Baker proposes a different minimization method based on a change in the circumscription policy: when the predicate AB is circumscribed, the function $result$, instead of the predicate $HOLDS$, is allowed to vary. The idea is that since the predicate AB has a situation argument, it is important for the meaning of situations to be held constant across the various models to be compared. Since, unlike chronological minimization, this

[10] In that paper, Morgenstern correctly points out that one of the problems with the frame problem is the fact that everybody seems to define it a little differently.

[11] The logic of chronological minimization (later generalized to the logic of *chronological ignorance*) has been originally proposed in [222, 223]. A similar solution to the Yale shooting problem is proposed by Kautz in [126], using the *logic of persistence*, and by Lifschitz in [149], using *pointwise circumscription*.

approach does not exploit any notion of temporal priority, it handles both forward and backward temporal projections. Furthermore, it can be shown that, unlike causal minimization, it handles also ramification in a natural manner. In [124], Kartha shows that Baker's solution does not extend to nondeterministic actions, because of the interferences of the observation axioms with minimizations. Kartha suggests a small modification of Baker's solution that avoids these shortcomings.

Finally, in [204], Sandewall proposes a refinement of chronological minimization, called *filter preferential entailment*, that allows dealing with a number of temporal reasoning problems, including temporal explanation. First, he strengthens the selection criterion by stating that a model m_i is preferred to a model m_j if there exists a time point t such that the abnormalities of m_i at t are a subset of the abnormalities of m_j at t, and m_i and m_j agree on the properties that hold at all time points before t. As noticed by Baker and Shoham [11], this last constraint is more restrictive than the one imposed by chronological minimization which only requires that m_i and m_j agree on the abnormalities at all time points before t. Then, Sandewall introduces a distinction between axioms specifying the behavior of the domain (causal axioms) and axioms specifying which properties hold at the various times (observation axioms), and restricts the application of revised chronological minimization to causal axioms only. Finally, he further reduces the resulting set of models by filtering out those models which disagree with the observations. The exclusion of observations from minimization, so that they can be used to rule out unwanted models, but not to include new ones, has also been formalized by Lifschitz [151] in terms of the *restricted monotonicity* property, and by Crawford and Etherington [60] in the context of avoiding interferences between minimizations solving the frame and the qualification problem.

In his subsequent work, Sandewall explores several variants of the original filter preferential entailment. As a result, in [206] he formally defines a variety of logics of *preferential* entailment and uses his features and fluents framework (cf. Section 2.2) for assessing the correctness of any given logic with respect to a certain class of reasoning problems. The base logic *PMON* is proved correct for the class of action scenarios encompassing nondeterministic actions, incomplete specification of states, timing of actions, and observations at arbitrary states in a scenario. In [67, 68], Doherty and Lukaszewicz generalize *PMON* and translate it into an order-sorted first-order logic with a circumscription axiom that captures the *PMON* definition of preferential entailment. A number of different extensions and/or generalizations to the original *PMON* have been proposed by several authors. The resulting family of logics is commonly identified as the *TAL* (*Temporal Action Logic*) family. As an example, Gustafsson and Doherty propose the logic *TAL-RC* that solves the ramification problem for a broad class of action scenarios by adding a specialization of fluent dependency constraints, called *causal constraints* [104]. The minimization policy adopted by *TAL-RC* has been later borrowed by other extensions, such as Gustafsson and Karlsson's *TAL-C* [105] (cf. Section 2.2).

3 Reasoning about temporal constraints

Mechanisms for reasoning about temporal constraints are concerned with the handling of relations among temporal entities (points, intervals, etc.) on the basis of the properties of the underlying temporal domain (ordering, discreteness, metric, etc.), and make no assumptions about the sorts of facts that can be said to hold over time. Temporal relations can thus be interpreted as constraints on temporal entities.

Representing temporal constraint problems. A generic *constraint satisfaction problem* can be represented by a finite set of *variables* (or *nodes*), their associated *domains*, and a set of *constraints* on these variables. The domain of a variable is the set over which the variable takes its value. Each element of the domain is called a *label*. The association between a variable v and a label l is denoted by the pair (v, l). Constraints can be partitioned into unary, binary, ternary, ..., n-ary constraints over (v, l) pairs. Unary constraints are expressed by a relation R_1 such that $R_1(v, l)$ means that l is an admissible label for variable v, binary constraints are expressed by means of a relation R_2 such that $R_2(v, l, z, m)$ means that one can consistently assign the labels l and m respectively to variables v and z, and so on. The basic constraint satisfaction problem consists of finding one/all assignments of labels to variables which satisfy the given constraints [160, 172].
In a *qualitative temporal constraint problem*, variables take their values over the set of possible *relations* among temporal entities and the constraints are those explicitly imposed upon a variable by restricting its set of possible values (unary constraints) as well as those implicitly imposed upon pairs, triplets, ..., n-tuples of variables by the basic properties of relations (n-ary constraints, with $n \geq 2$). As an example, in Allen's interval-based temporal reasoning framework [2], any variable represents the relation between a pair of intervals I_1 and I_2. All variables take their value over a finite domain of 13 possible relations $S = \{before, meets, overlaps, \ldots\}$. Explicit unary constraints are used to restrict the domain of possible values of variables to proper subsets of S, while implicit ternary constraints enforce that any "triangle" of intervals must be consistently labeled[12]. Since there is only one variable for each pair of intervals I_1 and I_2 (i.e., there is no additional variable to represent the pair of intervals I_2 and I_1), binary constraints are not needed to impose that the set of relations associated with the pair of intervals I_1 and I_2 must be the inverses of those associated with the pair I_2 and I_1. A different, but equivalent, formulation of Allen's temporal reasoning framework can be given, where variables correspond to the intervals and take their values over the infinite domain $\{(a, b) : a < b \wedge a, b \in \mathcal{Q}\}$ (or the domain $\{(a, b) : a < b \wedge a, b \in \mathcal{Z}\}$), and the constraints are binary. Both formulations have been used in the literature. When doing backtracking search (cf. the paragraph on Allen's Interval Algebra), the formulation with finite domains is used.
In a *quantitative temporal constraint problem*, variables take their values over the set of *temporal entities* and the constraints are those explicitly imposed upon a variable by restricting its set of possible values (unary constraints) as well as those constraining the admissible values of pairs of variables (binary constraints). As an example, in Dechter, Meiri, and Pearl's point-based formalism [64], any variable represents a time point over a continuous domain. Unary constraints are used to restrict the set of admissible values of each variable to a finite set of intervals, while binary constraints are used to constrain

[12] It must be pointed out that enforcing 3-consistency does not suffice to guarantee global consistency. In [2], Allen provides the example of a network which appears consistent by considering any three nodes, but has no consistent overall labeling. In general, given a constraint satisfaction problem with n variables, k-consistency, with $k < n$, does not imply global consistency. The relationships between local and global consistency in temporal constraint networks have been systematically analyzed in [135].

the distance between pairs of time points.

Solving temporal constraint problems. The two fundamental and thoroughly studied problems in the area of reasoning about temporal constraints are (i) deciding whether or not there exists an instantiation of the variables that satisfies the given set of constraints, and (ii) determining, for each pair of temporal entities, the strongest implied relation between them. In the case of interval-based formalisms, the former problem is often referred to as the *interval satisfiability problem* (ISAT), while the latter has also been called the *minimal labeling problem* (MLP) [17] and the *deductive closure problem* [244]. In [101], an additional distinction between decision and constructive versions of ISAT is highlighted: the decision version outputs only a yes/no answer, while the constructive one outputs also a solution (in the case of a yes answer). Two other related problems are that of finding (a polynomial representation structure for) all consistent instantiations (ASP, *all solutions problem*) and that of enumerating all distinct ways of localizing temporal entities over the temporal axis according to a given instantiation (ARP, *all realizations problem*) [101]. In the case of Allen's formalism, ASP and ARP are equivalent, while, in other formalisms, there may be several realizations which correspond to the same instantiation.

Reasoning about temporal constraints is performed using a number of different algorithms for constraint satisfaction, which build on the seminal work of Montanari [180], Mackworth [160], and Freuder [86, 87]. These algorithms adopt a network representation of constraints by associating intervals (or points) to nodes and labeling arcs with the algebraic elements that denote the temporal relation between the nodes they connect. The problem of computing the minimal representation of a set of temporal constraints or reporting its unsatisfiability is brought back to the problem of computing the minimal consistent labeling of the corresponding network [17, 18, 127, 244]. The complexity of this problem is affected by the expressiveness of the adopted temporal constraint formalism. In particular, the most known formalisms for representing qualitative (Allen's Interval Algebra [2]) and quantitative (Dechter, Meiri, and Pearl's Distance Algebra [64]) relations among temporal entities favored expressiveness over efficiency: in [244], Vilain and Kautz consider the network representation of Allen's Interval Algebra (IA) and show that computing the minimal consistent labeling as well as finding one or all possible models of the network are NP-complete problems; in [64], Dechter et al. show that the same results hold for their Distance Algebra (DA), unless suitable restrictions to constraint types are imposed. Approximations of the exact solutions for IA and DA networks can be obtained in polynomial time using consistency algorithms. They remove local inconsistencies that could never be a part of a global solution, but they do not always compute the minimal label between any two nodes, and thus do not guarantee the consistency of the whole network. One, two, and three-consistency (generally referred to as node, arc, and path consistency, respectively) are the most commonly used algorithms [86, 87].

3.1 Temporal constraint formalisms

Several formalisms for expressing and reasoning about temporal constraints have been proposed in the literature. They can be classified either according to the type of temporal constraints they deal with (qualitative, quantitative, or both) or according to the type of temporal entities they consider (intervals, points, or both). The most relevant formalisms for dealing with qualitative temporal constraints are Allen's Interval Algebra [2], Vilain

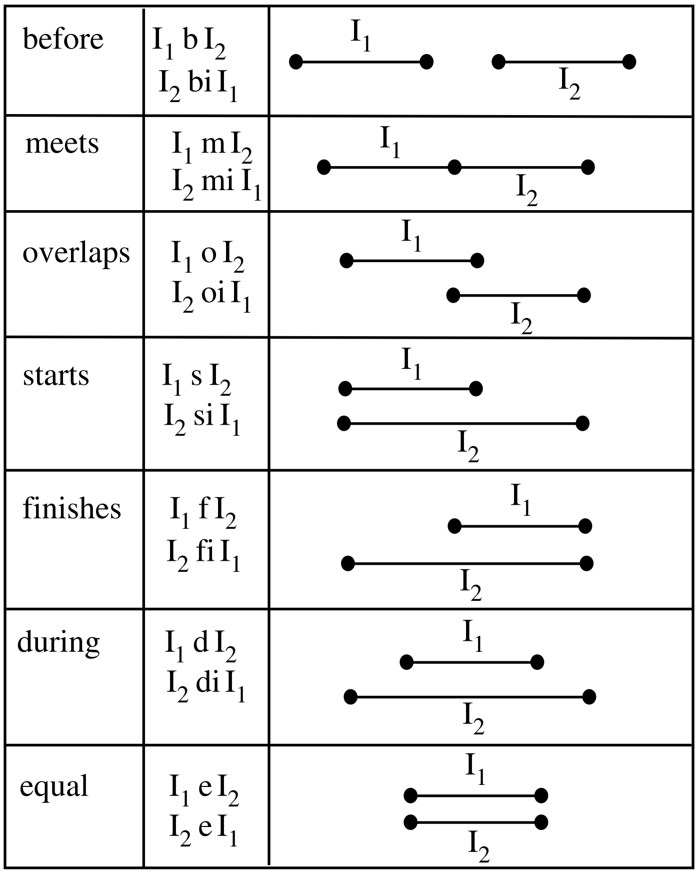

Figure 2: Basic relations of Allen's Interval Algebra.

and Kautz's Point Algebra [244], and Nebel and Bürckert's ORD-Horn Algebra [185]. Quantitative temporal constraints are dealt with by Dechter, Meiri, and Pearl's Distance Algebra [64], which models distances between points as well as durations of intervals. In the following, we will illustrate the whole spectrum of the most significant formalisms proposed in the literature, including the above mentioned ones, and analyze each of them from the point of view of expressiveness and complexity[13].

Allen's Interval Algebra. Allen's *Interval Algebra* (IA) [2] models the relation between any two intervals as a suitable subset of a set of thirteen basic relations (depicted in Figure 3.1), namely, *before, meets, overlaps, starts, during, finishes* (*b, m, o, s, d, f*), together with their *inverses* (*bi, mi, oi, si, di, fi*), and the basic relation *equal* (*e*), which coincides with its inverse.

[13]We do not discuss the formal characterization of these formalisms as relation algebras, as defined by Tarski [231], and use the terms *relation class* and *relation algebra* interchangeably. Such a characterization for Allen's formalism is provided by Ladkin and Maddux in [141].

As an example, the relation $\{b, bi\}$ between intervals I_1 and I_2 expresses the fact that I_1 *is before or after* I_2. In such a way, 2^{13} relations can be specified between any pair of intervals, including the *empty relation* corresponding to the empty set of basic relations. In the case of total lack of knowledge about the actual relation between two intervals, the set of all basic relations is assumed. The unary operation of converse and the binary operations of intersection and composition are defined on the set of relations. The *converse* of a relation is the inverse relation. More precisely, the converse of a basic relation is its inverse, while the converse of a non-basic relation is the set of the inverses of its constituent basic relations. The *intersection* of two relations is the set-theoretic intersection of their constituent basic relations. The *composition* of two relations, r_i between I_1 and I_2, and r_j between I_2 and I_3, is a new relation r_k between I_1 and I_3 induced by r_i and r_j: r_k is the set-theoretic union of the component-wise composition of their constituent basic relations defined by means of a set of transitivity rules. As an example, given three intervals I_1, I_2, and I_3, if $I_1\{o, s\}I_2$ and $I_2\{m\}I_3$, then, by transitivity, one can conclude that $I_1\{b\}I_3$. The whole set of transitivity rules can be found in [2]. From a logical point of view, the set of basic relations associated with any pair of intervals can be interpreted as the logical disjunction of its elements, and the operations of converse, intersection, and composition can be modeled in terms of logical derivations. This logical interpretation is fully detailed in [173].

It is worth noting that IA is not as expressive as full temporal logic, because it allows only a restricted form of disjunction. Using IA, one can model disjunctive information about the relation between two given intervals, but one cannot model disjunctive information about the relations between different pairs of intervals. For instance, IA allows one to assert that $I_1\{b, bi\}I_2$, but not that $(I_1\{b\}I_2 \ or \ I_3\{m\}I_2)$. Even under these limitations, however, Allen's formalism is computationally intractable. In [244], Vilain and Kautz consider the network representation of Allen's IA and prove that ISAT is NP-complete. They also show that ISAT and MLP for IA are equivalent with respect to polynomial Turing-reductions. Such a result is generalized in [185], where Nebel and Bürckert show that ISAT and MLP are equivalent under polynomial Turing-reductions for any subclass of IA that includes all basic relations.

The exact solution of MLP can be computed by using traditional backtracking algorithms for constraint satisfaction (e.g., see [86, 87, 160]). In [142], Ladkin and Reinefeld propose a significant improvement to the standard search methods, using path-consistency to forward prune the search tree and thus reduce backtracking. Approximations of the exact solutions can be obtained in *polynomial time* using consistency algorithms [160]. In [2], Allen proposes to use a path-consistency algorithm for IA, which provides an approximate solution in $\mathcal{O}(n^3)$, where n is the number of nodes. Algorithms for IA which provide better approximations have successively been developed; in particular, we mention Van Beek's algorithm that guarantees 4-consistency in $\mathcal{O}(n^4)$ [17]. Consistency algorithms produce exact solutions if less expressive representation formalisms are adopted. As an example, Valdez-Perez [240] shows that networks which are not labeled with disjunctions of interval relations can be solved exactly in $\mathcal{O}(n^3)$ time using path-consistency.

A significant research effort in the area of temporal constraints has been devoted to identify tractable fragments of IA obtained by restricting the set of allowed relations. Among these fragments, Vilain and Kautz's Point Algebra [244], van Beek's Continuous Endpoint Algebra [17], and Nebel and Bürckert's ORD-Horn Algebra [185] play a major role.

Vilain and Kautz's Point Algebra. Vilain and Kautz's *Point Algebra* (PA) models the relation between any two points as a subset (disjunction) of a set of three basic relations: $<, >$, and $=$. In such a way, it allows the specification of 2^3 relations between two given points, namely, $<, >, =, <=$ ($\leq$ for short), $=>$ ($\geq$ for short), $<>$ ($\neq$ for short), and $<=>$ (total lack of knowledge), plus the empty relation. As in the case of IA, the unary operation of converse and the binary operations of intersection and composition are defined on the set of relations. It is possible to show that PA is as expressive as a proper subset of IA, called "pointizable algebra" or Simple Interval Algebra (SIA) [21], consisting of those interval relations that can be expressed as conjunctions of point relations between interval endpoints (pointizable relations). As an example, it allows the representation of the interval relation $I_1\{b, m, o\}I_2$ as the conjunction of point relations:

$$I_1^- < I_2^- \wedge I_1^+ < I_2^+ \wedge I_1^- < I_1^+ \wedge I_2^- < I_2^+,$$

where I^- and I^+ respectively denote the left and the right endpoint of the interval I and the inequality $I^- < I^+$ express the fact that the left endpoint of I precedes the right endpoint of I. SIA covers approximately 2% of IA [183]. As an example, it does not allow one to model the interval relation $I_1\{b, bi\}I_2$, because a disjunction of point relations is required to represent it:

$$(I_1^+ < I_2^- \wedge I_1^- < I_1^+ \wedge I_2^- < I_2^+) \vee (I_2^+ < I_1^- \wedge I_1^- < I_1^+ \wedge I_2^- < I_2^+).$$

In [244], Vilain and Kautz show that the path-consistency method suffices to decide satisfiability in PA. They also claim that the path-consistency method is complete for MLP too, but van Beek [18] gives a counter-example showing that the claim is wrong. The basic results about the complexity of the path-consistency method and about the reducibility of MLP to ISAT make it possible to conclude that ISAT and MLP for PA can be solved in $\mathcal{O}(n^3)$ time and $\mathcal{O}(n^5)$ time, respectively. Such results have been refined by van Beek [17, 18] and by van Beek and Cohen [21] which propose algorithms for solving ISAT in $\mathcal{O}(n^2)$ time and MLP in $\mathcal{O}(n^4)$ time. In [167], Meiri shows that arc- and path-consistency algorithms suffice to decide satisfiability of PA augmented with single-interval domains in $\mathcal{O}(n^3)$ time.

van Beek's Continuous Endpoint Algebra. A relevant proper subset of PA, called *Continuous Endpoint Algebra* (CEA), has been identified by van Beek [17, 21, 245]. CEA is a proper subset of PA that only models continuous relations between time points. A relation r_i between two points P_1 and P_2 is said to be *continuous* if, for any admissible value x of P_1, from the fact that $x\ r_i\ y_1$ and $x\ r_i\ y_2$, where both y_1 and y_2 are admissible values for P_2 and $y_1 < y_2$, it follows that $x\ r_i\ y$, for any value y of P_2 such that $y_1 < y < y_2$, and vice versa. In other words, given two points P_1 and P_2, the set of admissible values for each of them, considered as a function of the other, is *convex.* CEA covers approximately 1% of IA [183]. It is possible to show that all relations, except the relation $\neq$, between two points are continuous. Then, CEA is not able to model any relation between two intervals whose representation involves the relation $\neq$ between their endpoints (in the subsequent discussion of ORD-Horn, we will provide an alternative equivalent characterization of CEA). For instance, CEA represents the interval relation $I_1\{o, s, d\}I_2$ in the following way:

$$I_2^- < I_1^+ \wedge I_1^+ < I_2^+ \wedge I_1^- < I_1^+ \wedge I_2^- < I_2^+,$$

but it is not able to model the interval relation $I_1\{b, o\}I_2$, that can be represented in PA as follows:

$$I_1^- < I_2^- \wedge I_1^+ < I_2^+ \wedge I_1^+ \neq I_2^- \wedge I_1^- < I_1^+ \wedge I_2^- < I_2^+.$$

In [17, 21, 19, 245], van Beek et al. show that the application of the path-consistency method to CEA not only allows one to detect inconsistency, but also leads to a globally consistent network. Since global consistency implies minimality [87], it follows that the application of the path-consistency method to CEA also solves MLP.

Deville and Van Hentenryck [66] and Meiri [167] show that arc- and path-consistency algorithms allow deciding in polynomial time the consistency of CEA augmented with discrete and multiple-interval domains, respectively.

Nebel and Bürckert's ORD-Horn Algebra. The ORD-Horn subclass of IA (OH) is a strict superset of SIA, and it can be viewed as the borderline between tractability and intractability. OH is indeed the only maximal subset of IA, including all basic relations, for which path-consistency is sufficient for deciding satisfiability [185] and Ladkin and Reinefeld's algorithm is sound and complete [183].

OH is built as follows. Let us consider the subset of PA relations including $\leq$ and $=$ only. Each relation between two intervals I_1 and I_2 can be translated into an equivalent set of clauses over relations of the form $\leq$ and $=$ between their endpoints I_1^-, I_1^+, I_2^-, and I_2^+. An *ORD clause* is obtained by forbidding literals of the form *not* $X \leq Y$ ($X \not\leq Y$ for short). Since every clause of the form $X \not\leq Y \vee C$ can be equivalently rewritten as the conjunction of the ORD clauses $X \neq Y \vee C$ and $Y \leq X \vee C$, it follows that each interval relation can also be translated into an equivalent set of ORD clauses. The OH subclass is obtained by restriction to the set of ORD clauses that contain at most one positive literal (of the form $X = Y$ or $X \leq Y$) and an arbitrary number of negative literals (of the form $X \neq Y$). It is easy to show that PA and CEA are proper subclasses of OH: PA is the subclass of OH including the interval relations whose clausal representation contains only unit clauses; CEA is the subclass of PA including those interval relations such that if their clausal representation contains a unit clause of the form $X \neq Y$, then it contains also a unit clause of the form $X \leq Y$ or $Y \leq X$. OH covers approximately 10% of IA [183]; it does not include the disjointness (*sequentiality*) relations of IA, that is, the relations that constrain two intervals to not overlap (a typical example of disjointness relation is $\{b, bi\}$).

As far as complexity is concerned, the upper bounds of $\mathcal{O}(n^3)$ for deciding satisfiability and of $\mathcal{O}(n^5)$ for solving MLP are given in [185], where it is also shown that ISAT is NP-complete in all minimal algebras which strictly include OH. From a practical point of view, Nebel and Bürckert outline two possible exploitations of OH. On the one hand, it can be used to determine the complexity of reasoning in new subclasses of IA (provided that they include all basic relations) by checking whether or not all their relations fall into OH. On the other hand, faster backtracking algorithms for IA, that decrease the branching factor of a backtrack search, can be obtained by substituting OH for less expressive tractable subclasses, such as PA and CEA. This second application of OH is detailed in [183].

A different perspective on efficiency issues is proposed by Bessière, Isli, and Ligozat in [29]. They look for subclasses of OH which preserve as much as possible its expressiveness, while ensuring that path consistency is sufficient to guarantee global consistency, thus allowing a backtrack-free search for solutions. In [29], they identify two subclasses of OH that satisfy this property, one of which covers more than 60% of OH.

Other maximal tractable subclasses of IA. The practical relevance of ORD-Horn maximality is questioned by some authors. In particular, Drakengren and Jonsson [69, 70] stress that OH is not capable of expressing sequentiality among intervals, which is essential in reasoning about actions. Their research concentrates on studying (maximal) tractable subclasses of IA which, unlike OH, do not contain the basic relations *meets* and its inverse, assuming that these two relations are scarcely important from a cognitive point of view. In [70], they present 21 distinct tractable subclasses of IA (nine of them being maximal). Each algebra contains exactly 3 basic relations, and 4 of these subclasses allow expressing sequentiality by containing the relation $\{b, bi\}$. In [71], Drakengren and Jonsson extend their investigation to extensions of IA with metric temporal information (see also the next paragraph on the integration of qualitative and quantitative temporal constraints), and identify 8 tractable subclasses of IA, proving their maximality. They show that 6 of the algebras contain exactly 5 basic relations each, while 2 of them contain 3 basic relations. Finally, in [72], they propose a general result which provides a partial classification of tractability in IA: any tractable subclass of IA that is not a subset of either OH or one of the algebras presented in [70, 71], cannot contain more than three basic relations, which have to be e (*equal*), r, and the converse of r, where $r \in \{d, o, s, f\}$. Furthermore, there are no remaining tractable subclasses containing the relation $\{b, bi\}$. Thus, any remaining tractable subclass is necessarily of weak expressivity concerning sequentiality. More generally, they advocate the usefulness of identifying *all* maximal tractable subclasses among the possible 2^{8192} subsets of IA, where $8192 = 2^{13}$, because an application specifying a set of intervals could then search for the most appropriate subclass, or report that no such subclass exists.

Empirical analyses of complexity. While most of the past research on complexity of qualitative temporal reasoning was mainly concentrated on worst-case analyses, recent work is focusing on the empirical analysis of algorithms' efficiency. With respect to this, a major subject of discussion is how to find a suitable set of instances for testing the efficiency of algorithms. As an example, van Beek and Manchak's experimental analysis of the use of Ladkin and Reinefeld's search algorithm with OH showed no improvements with respect to the use of PA [20], but Nebel pointed out that these results are due to a lack of hard problem instances in the experimental set [183]. In order to choose an appropriate set of problem instances, Nebel [183] adopts a phase transition approach. He follows Cheeseman et al.'s conjecture [42] that "all NP-complete problems have at least one order parameter and the hard to solve problems are around a critical value (a phase transition) of this order parameter". The critical order parameter identified by Nebel is the average degree of the constraint graph. This parameter is independent of the instance size, and the probability of satisfiability of problem instances has a sudden transition from 1 to 0 around a 9.5 value for the parameter (as we move away from this value, problems are satisfiable or unsatisfiable depending on the direction we are moving in). Instances around this value are then particularly suited for testing algorithms. Using these instances, Nebel concludes that it can pay off for search intensive cases to use OH instead of PA. In fact, he also shows that other design decisions (such as using different kinds of heuristics for processing constraint, or weighted queue schemes) contribute to efficiency much more than simply adopting OH. Since different search methods vary their performance on different instances, Nebel suggests a possible parallel implementation that runs different methods independently up to a time limit. For example, in his experimental activity, he first ran just one carefully implemented method based on OH on 500 instances with a time limit of 1800 seconds, obtaining a result for 85% of them. By running 16 methods in parallel, he was able to obtain a result for 99 % of the instances with a time limit of 20 seconds.

Golumbic and Shamir's Macro-Relation Algebras. In [101], Golumbic and Shamir reconsider the problem of reasoning about temporal constraints from the point of view of graph theory. This allows them to unify the notions of interval algebras in artificial intelligence with those of interval orders and interval graphs in combinatorics. Furthermore, they introduce the concept of macro-relation algebras as suitable partitions of Allen's 13 basic relations into coarser relations, called macro-relations, and discuss in detail three specific examples of such algebras. The simplest one is the macro-relation algebra $\mathcal{A}_3$ that distinguishes 3 different basic relations: $\prec$, which corresponds to Allen's b, $\succ$, corresponding to Allen's bi, and $\cap$, which comprises all the remaining 11 basic relations of IA. The macro-relation algebra $\mathcal{A}_7$ is obtained from $\mathcal{A}_3$ refining the macro-relation $\cap$ into 5 finer macro-relations: $\subset = \{s, f, d\}$, $\subset^{-1} = \{si, fi, di\}$, $\alpha = \{m, o\}$, $\alpha^{-1} = \{mi, oi\}$, and the basic relation e. A third macro-relation algebra ($\mathcal{A}_6$) is obtained from IA by assuming that all interval endpoints are distinct: under such an assumption, 7 of Allen's basic relations can be eliminated, and the resulting algebra comprises only the remaining ones, namely, *b, bi, o, oi, d*, and *di*.

It is possible to show that ISAT, MLP, ASP, and ARP are NP-hard for all three macro-algebras and that ISAT is actually NP-complete for $\mathcal{A}_3$ [101]. This negative result motivates the study of suitable restrictions of the proposed classes of relations. Golumbic and Shamir consider the case of $\mathcal{A}_3$. Since $I_1\{\succ\}I_2$ is equivalent to $I_2\{\prec\}I_1$ and $I_1\{\cap\succ\}I_2$ is equivalent to $I_2\{\prec\cap\}I_1$, $\mathcal{A}_3$ encompasses 5 different relations and thus there exist only $2^5 - 1 = 31$ possible ways of restricting the input domain of a problem in $\mathcal{A}_3$. They show that 21 of them are polynomial and 6 are NP-complete (the remaining 4 cases were later shown to be NP-complete by Webber [246]). In particular, they prove that some of the polynomial cases are linear. This is the case, for instance, of ISAT for the domain $\{\prec, \succ, \cap, \prec\cap, \cap\succ, \prec\cap\succ\}$, which is a subset of PA.

Ligozat's Generalized Interval Calculus. In [156], Ligozat proposes a general uniform framework for dealing with convex and non-convex intervals as well as points. The approach subsumes both Vilain's point-to-interval relations [243] and Ladkin's non-convex interval relations [140]. It is based on the notion of *generalized interval* which is defined as a linearly ordered sequence of distinct points, a sequence of p points being called a p-interval. As an example, a *3*-interval can represent either three points or a point preceding an interval or an interval followed by a point. To distinguish between interval endpoints and single points, it is possible, for instance, to introduce a notion of point type.

Relations between a p-interval and a q-interval are called *(p,q)*-relations. For any pair (p, q), the set of (p, q)-relations is denoted by $\Pi(p, q)$. A (p, q)-relation between $x = (x_1, \ldots, x_p)$ and $y = (y_1, \ldots, y_q)$ is defined as follows: first, the q-interval is used to partition the domain into $2q+1$ zones, namely, the set of points preceding y_1 (zone 0), the point y_1 (zone 1), the set of points between y_1 and y_2 (zone 2), and so on, up to the set of points following y_q (zone $2q$); then, each point of the p-interval is assigned to the zone it belongs to, obtaining a *sequence of p zone numbers* where each odd number occurs at most once. It is easy to show that the sets of basic relations of PA, Vilain's point-to-interval formalism, and IA correspond to $\Pi(1,1)$, $\Pi(1,2)$, and $\Pi(2,2)$, respectively.

Ligozat's Interval Calculus is based on two operations which are defined on the set of (p,q)-relations: transposition and composition. Given a (p,q)-relation between x and y, transposition returns the corresponding (q,p)-relation between y and x. Given the relation between a p-interval x and a q-interval y and the relation between y and an r-interval z, composition returns the set of possible (p,r)-relation between x and z.

Dechter, Meiri and Pearl's Distance Algebra. Unlike the previously presented formalisms which deal only with qualitative temporal relations, Dechter, Meiri and Pearl's *Distance Algebra* (DA) allows representing and processing quantitative temporal information [64]. DA models distances between time points as well as durations of intervals, and also allows constraining the value of dates. DA is a point-based approach, which considers variables (over continuous domains) whose values can be constrained using both unary and binary constraints. Let $X_1,..,X_n$ be a set of variables, each one denoting the value of a distinct time point. A *unary* constraint C_i on X_i restricts the domain of the variable to a finite set of intervals $I_1,..,I_n$ (*domain constraints*). A constraint $C_i = \{I_1,..,I_n\}$ is represented by the disjunction

$$I_1^- \leq X_i \leq I_1^+ \vee ... \vee I_n^- \leq X_i \leq I_n^+,$$

where either or both inequalities of each disjunct may be replaced by strict ones ($<$). A *binary* constraint $C_{i,j}$ on variables X_i and X_j restricts the admissible values for the distance $X_j - X_i$ to a finite set of ranges $R_1,..,R_n$. Each range R_k is an ordered pair $\langle min_k, max_k\rangle$, where min_k and max_k are respectively a lower and an upper bound on the temporal distance separating X_i and X_j. Assuming that bounds belong to the set of admissible values, a constraint $C_{i,j} = \{R_1,..,R_n\}$ is represented by the disjunction

$$min_1 \leq X_j - X_i \leq max_1 \vee ... \vee min_n \leq X_j - X_i \leq max_n.$$

To deal with unary and binary constraints in a uniform way, unary constraints on a variable X_i can be transformed into binary constraints restricting the distance between the point denoted by X_i and a suitable reference point X_0. The usual operations of intersection and composition are defined on the set of quantitative relations.

The intersection of two constraints $C'_{i,j} = \{R'_1,..,R'_n\}$ and $C''_{i,j} = \{R''_1,..,R''_m\}$ on variables X_i and X_j is a constraint

$$C_{i,j} = \{R'_1 \oplus R''_1, R'_1 \oplus R''_2, ..., R'_n \oplus R''_m\},$$

where $R'_i = \langle min'_i, max'_i\rangle$, for $i = 1,..,n$, $R''_j = \langle min''_j, max''_j\rangle$, for $j = 1,..,m$, and $R'_i \oplus R''_j = \langle min_{i,j}, max_{i,j}\rangle$ when $min_{i,j} = max\{min'_i, min''_j\}$, $max_{i,j} = min\{max'_i, max''_j\}$ and $min_{i,j} \leq max_{i,j}$; otherwise, $R'_i \oplus R''_j$ is equal to the empty constraint $\perp$, that can never be satisfied.

The composition of two constraints $C'_{i,j} = \{R'_1,..,R'_n\}$ and $C''_{j,k} = \{R''_1,..,R''_m\}$ on variables X_i and X_j, and X_j and X_k, respectively, is a constraint

$$C_{i,k} = \{R'_1 \otimes R''_1, R'_1 \otimes R''_2, ..., R'_n \otimes R''_m\},$$

where $R'_i = \langle min'_i, max'_i\rangle$, for $i = 1,..,n$, $R''_j = \langle min''_j, max''_j\rangle$, for $j = 1,..,m$, and $R'_i \otimes R''_j = \langle min_{i,j}, max_{i,j}\rangle$, with $min_{i,j} = min'_i + min''_j$ and $max_{i,j} = max'_i + max''_j$.

A meaningful subset of DA is obtained when every constraint $C_{i,j}$ specifies a single range, that is, $C_{i,j}$ consists of one disjunct only. A single-range binary constraint is defined as follows:

$$min_{i,j} \leq X_j - X_i \leq max_{i,j}.$$

Single-range DA can be applied to a wide class of problems named *simple temporal (constraint satisfaction) problems* (STPs for short) [64].
As an example, Dean and McDermott's *time map* [63] allows defining a constraint $C'_{i,j}$ on the distance separating two points X_i and X_j as a conjunction of ranges $R_1, .., R_n$. According to the previous notational conventions, a constraint $C'_{i,j}$ is represented by the following conjunction:

$$min_1 \leq X_j - X_i \leq max_1 \wedge ... \wedge min_n \leq X_j - X_i \leq max_n.$$

If ranges are consistent (i.e., they have a non empty intersection), then they can be replaced by a single range $\langle min, max \rangle$, where min is the maximum of the lower bounds of ranges $R_1, .., R_n$ and max is the minimum of their upper bounds. It is easy to see that such type of constraints are exactly those relevant to STPs.

DA subsumes PA, which can be considered a special case of DA lacking quantitative information. PA can be easily encoded within DA on the basis of a suitable set of equivalences according to which $X_j < X_i$ corresponds to $C_{i,j} = \{(-\infty, 0)\}$, $X_j = X_i$ corresponds to $C_{i,j} = \{[0, 0]\}$, and so on, provided that $-\infty$ and $+\infty$ are special constants whose properties are given by means of a suitable set of qualitative and quantitative axioms, e.g. $\forall x(x < +\infty)$. The following is the complete list of correspondences between PA relations and DA binary constraints:

$$\begin{array}{ll}
X_j < X_i: & C_{i,j} = \{(-\infty, 0)\}; \\
X_j > X_i: & C_{i,j} = \{(0, +\infty)\}; \\
X_j = X_i: & C_{i,j} = \{[0, 0]\}; \\
X_j \leq X_i: & C_{i,j} = \{(-\infty, 0]\}; \\
X_j \geq X_i: & C_{i,j} = \{[0, +\infty)\}; \\
X_j <> X_i: & C_{i,j} = \{(-\infty, 0), (0, +\infty)\}; \\
X_j <=> X_i: & C_{i,j} = \{(-\infty, +\infty)\}.
\end{array}$$

On the other hand, IA cannot be encoded within DA: they have orthogonal expressive power. Properties like disjointedness ($I_1\{b, bi\}I_2$) cannot be represented by a binary constraint, but require the 4-ary constraint already introduced in the discussion of PA limitations.

In [64], Dechter et al. prove that the complexity of deciding consistency in DA (the quantitative temporal constraint satisfaction problem, TCSP for short) is NP-hard, even in the case where no more than two intervals per constraint are allowed. They also show that the STP can be solved in polynomial time, using the well-known Floyd-Warshall's all-pairs-shortest-paths algorithm.

In [210], Schwalb and Dechter extend the work on TCSP by proposing different ways of weakening path consistency which improve significantly on the algorithms proposed in [64]. A problem with enforcing path consistency with full DA is the so-called fragmentation problem, i.e., some intervals may be broken into several smaller subintervals with an increase in the number of intervals per constraint that may result in an exponential blow up. To solve this problem, Schwalb and Dechter propose (different variants of) an algorithm, called *Loose Path Consistency*, based on a simple intersection operator between sets of intervals. The algorithm dynamically reduces TCSP to STP by collapsing sets of intervals/ranges corresponding to given binary constraints into single intervals/ranges. This simplification makes the consistency problem polynomial, but the resulting algorithm may return non-admissible solutions.

Integrating qualitative and quantitative temporal relations. Some formalisms have been proposed to deal with both qualitative and quantitative temporal information in a uniform framework. They typically build on seminal work by Kautz and Ladkin [127], and by Meiri [167].

In [127], Kautz and Ladkin define a simple logical formalism for expressing both kinds of temporal information. They assume a linear model of time, where time points are identified with the rationals under the usual ordering relation $<$ and time intervals are represented as pairs of points $\langle n, m \rangle$, with $n < m$. They also define a subtraction function that returns the rational corresponding to the difference between two time points. They formalize such a model of time using a typed predicate calculus with equality and provide a suitable set of axioms, including arithmetic axioms for subtraction, $<$, $\leq$, and numerals, the axiom $\forall I(I^- < I^+)$ constraining the relation between starting and final endpoints of intervals, and axioms expressing qualitative relations between intervals in terms of qualitative relations between their endpoints. They also show that the set of IA transitivity axioms for intervals are theorems of that language. Their work focuses on two meaningful sublanguages, namely, $\mathcal{L}_A$ and $\mathcal{L}_M$.

$\mathcal{L}_A$ consists of formulae of the form:

$$I_1(r_1)I_2 \vee ... \vee I_1(r_n)I_2,$$

where I_1 and I_2 are intervals and $r_1, .., r_n$ are IA relations.

$\mathcal{L}_M$ consists of formulae of the form:

$$(I_1^F - I_2^G) \leq n \wedge (I_2^G - I_1^F) \leq m$$

or equivalently:

$$-m \leq (I_1^F - I_2^G) \leq n$$

where $F, G \in \{+, -\}$, m and n are numerals (or $-\infty$ / $+\infty$), and $\leq$ may be replaced by $<$.

The full language is clearly much richer than the union of $\mathcal{L}_A$ and $\mathcal{L}_M$. Nevertheless, $\mathcal{L}_A$ and $\mathcal{L}_M$ are as expressive as IA and the subset of DA for STP, respectively, and their union suffices to deal with many relevant temporal problems.

In [167], Meiri defines a formalism combining quantitative and qualitative temporal constraints, together with a model of time capable of dealing with disjunctive constraints, involving both quantitative and qualitative constraints. Such a model consists of a set of variables $\{X_1, .., X_n\}$, which may represent points as well as intervals so that one does not need to commit to a single type of objects, and a set of unary and binary constraints. The domain of point and interval variables are the set of real numbers and the set of ordered pairs of real numbers, respectively. Qualitative constraints are disjunctions of the form:

$$X_1(r_1)X_2 \vee ... \vee X_1(r_n)X_2,$$

where X_1, X_2 are points or intervals, and each one of the r_i's is an admissible relation between X_1 and X_2, belonging to the set of basic interval-interval, point-point, point-interval, and interval-point relations. Intersection and composition of relations are defined as follows. The intersection of two relations is their set-theoretic intersection, as usual, while composition is given by means of six transitivity tables that define all legal combinations of basic relations. Quantitative constraints are borrowed from DA together with the relevant operation of intersection and composition. Meiri also defines a mapping between qualitative and quantitative constraints on pairs of points, that allows him to extend intersection and composition to cases where the operands are constraints of different types. Finally, he defines three *augmented qualitative models* obtained by adding unary domain constraints to PA and CEA. These models differ from each other in the types of domain they support: discrete domains, where each variable may assume only a finite number of values; single-interval domain, where the value of each variable is constrained to belong to a single range as in STP; multiple-interval domains, where each variable can belong to a set of ranges as in full DA (multiple-interval domains subsume both single-interval and discrete ones). The relevant feature of these models is that they extend the expressive power of PA and CEA preserving their computational properties.

3.2 Exploiting temporal constraints in applications

As far as applications of temporal constraints are concerned, most of the literature confines itself to simple toy examples. Although these examples are useful to point out advantages and limitations of one formalism over another, they do not contribute to the evaluation of the actual relevance of reasoning about temporal constraints for real world applications, which remains a scarcely explored subject.

Some applications of reasoning about temporal constraints have been studied in the areas of scheduling and planning. These applications involve up to 10000 temporal constraints [31]. They include transportation planning software environments, where the temporal component is used to apply doctrinal constraints that were previously checked by hand (if at all), and aerospace applications, e.g., processor and communication scheduling for the Flight Management Systems of an airliner and operations planning for a Space Shuttle science module [31]. Temporal constraints have been used also for minimizing assembly line slack time and projecting critical steps in a deployment plan to ensure proper interaction between them in a industrial context [229]. Another recent application of temporal constraints in the transportation domain is illustrated in [143], which describes a distributed advisor system which helps human experts in the management and control of traffic within railway stations and along railway branches. Besides scheduling and planning, temporal constraints have also been exploited for other relevant tasks, including natural language understanding, e.g. [227], molecular biology, e.g. [100], and diagnosis, e.g. [188].

In general, the major shortcomings of theoretical research with respect to the actual requirements of applications are:

- **Scalability problem**. For systems which involve tens of thousands of temporal propositions, showing that an algorithm has a time cost that is a low-order polynomial function of the size of the database does not suffice to show that it will be useful in practice [30]. To overcome these limitations, efficient ad-hoc algorithms have been developed that take into account the specific temporal structure underlying relevant classes of problems (well-behaved instances). As an example, Gerevini and Schubert [95] consider the case of application domains such planning and story understanding where temporal data tend to fall naturally into chain-like structures. They present a set of graph-based algorithms for dealing with pointizable relations and disjointness relations that exploit the internal temporal structure of information and provide better performance in practice compared to those based on constraints. They model temporal relations as edges of a graph, called *timegraph*, whose vertices represent time points. The distinctive feature of timegraphs is their partitioning into sets of linearly ordered time points (chains). Instead of precomputing the deductive closure of the given set of relations, the authors propose to build a timegraph to be used at query time to efficiently support deduction of relations. They report a number of experimental results in the above mentioned domains, showing that such an approach is actually quite efficient.

- **Integration issues**. Temporal constraint reasoners by themselves are generally not sufficient to tackle a whole real problem, and they thus need to be integrated with other reasoning modules. In most of the above mentioned applications [31], the temporal constraint reasoner has been actually extended with other functionalities. Brusoni et al. [33] discuss the issue of integrating temporal constraints into applications, advocating an approach where the constraint reasoner is embedded in a so-called *temporal knowledge server*. The server hides the physical implementation layer of the constraint reasoner and provides a higher level layer devoted to interface to the other modules of the application. In particular, the main requirements for the server are that (i) it must operate as a loosely coupled cooperative agent for use by various applications that need to deal with time, (ii) it must provide a clear, easy-to-use query and update language, and (iii) it must perform query processing efficiently. The last requirement leads them to adopt CEA for the physical implementation layer of the server.

- **Oversimplification**. The most studied temporal constraint problems, such as the satisfiability and the minimal labeling problems, are formulated in such a way that a number of important factors are neglected. Taking into account these factors is not merely a matter of implementation, but it has significant returns of theoretical nature. As an example, in the following we consider in detail the problem of exploiting temporal constraints in planning applications.

In planning, a typical need is that of modifying and extending a partial plan in order to meet pre-specified goals. A partial plan indeed evolves by propagating new constraints or by incrementally adding new tasks (and their effects). To this purpose, traditional techniques for reasoning about temporal constraints, as they stand, are not sufficient. Consider, for example, an activity whose starting point is set by the planner, but whose actual duration (although belonging to a given range of values) cannot be predicted. This is the case of the activity of boarding an airplane: its starting time can be decided by the personnel, while its duration (although belonging to a known range) depends on factors such as the actual number and type of passengers. The planner must be able to handle contingent constraints, such as duration of boarding, in a different way from usual ones, because it cannot freely instantiate them.

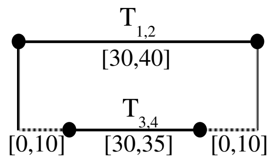

Figure 3: An example with contingent constraints.

This problem has been studied by Vidal and Ghallab [241]. They restrict themselves to situations where a planning system must guarantee the consistency of a constraint network for which the actual values of some variables are univocally determined by the evolving context, but not known in advance. Formally, they distinguish between *free* and *contingent* constraints. *Free* constraints are the usual ones: they can be freely instantiated and their domains can be freely reduced by propagation; *contingent* constraints are constraints whose value is a random variable over their domain, and thus they must not be reduced during the propagation (as this could remove possible occurring values). On the basis of this distinction, they replace the classical notion of consistency in the STP by the notion of *controllability* in the STP with contingent constraints, called STPU (STP under Uncertainty). An STPU is said to be *strongly controllable* if it is possible to find a priori, i.e., before starting the execution of the plan, a solution for it. An STPU is said to be *weakly controllable* when, although it is not possible to determine an a priori solution, it is guaranteed that a solution will be found during plan execution. If an STPU is neither strongly nor weakly controllable, it is not possible to guarantee that the planner will succeed.

Let us consider the case (depicted in Figure 3) of two tasks $T_{1,2}$ and $T_{3,4}$, whose durations are represented by the *contingent* constraints $C_{1,2}$ and $C_{3,4}$, respectively. Suppose that $C_{1,2} = \{[30, 40]\}$, $C_{3,4} = \{[20, 25]\}$, and $T_{3,4}$ must start after or simultaneously and end before or simultaneously with $T_{1,2}$. The execution of a classical STP propagation step would return the constraint $C_{1,3} = \{[0, 20]\}$ for the temporal displacement between the starting point of $T_{1,2}$ and that of $T_{3,4}$. This answer is not satisfactory in the considered planning framework, because for many values satisfying constraint $C_{1,3}$, it may happen that the actual values of contingent constraints violate the planning goals. As an example, if we choose to start $T_{3,4}$ 15 time units after $T_{1,2}$ and we get the values 30 and 25 for the actual durations of $T_{1,2}$ and $T_{3,4}$, respectively, the resulting plan does not satisfy the requirements. To guarantee the controllability of the whole set of constraints for every admissible value of the temporal displacement between the starting points of $T_{1,2}$ and $T_{3,4}$ (strong controllability), we need to restrict the interval $[0, 20]$ to $[0, 5]$, where $5 = 30 - 25$, i.e., the lower bound of the first task minus the upper bound of the second one. This goal is achieved by imposing the free constraint $C_{1,3} = \{[0, 5]\}$. In general, free constraints can be automatically generated before the execution of the plan, thus allowing one to establish a priori whether an STPU problem is strongly controllable or not. An example of *uncontrollable* STPU can be obtained by replacing $C_{3,4} = \{[20, 25]\}$ by $C_{3,4} = \{[30, 35]\}$ in the previous example. When the actual duration of $T_{1,2}$ is equal to 30 and that of $T_{3,4}$ is greater than 30, there is no way of finding a valid plan, neither a priori nor at execution time. As an example of weakly, but not strongly, controllable STPU, consider a situation in which we only require that $T_{1,2}$ meets $T_{3,4}$. In such a case, a decision about when to start $T_{3,4}$ cannot be taken a priori. Nevertheless, the constraint can always be satisfied at execution time by initiating $T_{3,4}$ exactly when $T_{1,2}$ terminates.

4 Further topics in temporal representation and reasoning

Besides the previously discussed evolution of established areas of research, temporal representation and reasoning is both growing new subareas, e.g., time granularity, and expanding itself towards other fields, e.g., temporal databases. This section discusses some of the most significant developments in both directions.

Time granularity and temporal abstraction. A relevant case of incomplete temporal information, which is not dealt with by standard temporal reasoning systems, is that of time information specified at different time granularities. *Time granularity* can be defined as the resolution power of the temporal qualification of a statement. Providing a formalism with the concept of time granularity makes it possible to specify time information with respect to differently-grained temporal domains. This does not only mean that one can use different time units – say, months and days – to represent time quantities in a *unique flat temporal model*, but it involves more difficult semantic issues related to the problem of assigning a proper meaning to the association of facts with the different temporal domains of a *layered temporal model* and of switching from one domain to a coarser/finer one. Mechanisms to handle time granularity consist of default projection rules to map a fact from its original granularity level into another granularity level. Projection from coarser to finer granularities and from finer to coarser ones are performed by means of downward temporal projection and upward temporal projection, respectively. *Upward temporal projection* supports inferences of the form: "if X is true at time t_1 of a given domain, then Y is true at time t_2 of a *coarser* domain", while *downward temporal projection* supports inferences of the form: "if X is true at time t_1 of a given domain, then Y is true at time t_2 of a *finer* domain". A formalization of the notion of time granularity and a thorough discussion of the involved expressibility, axiomatizability, decidability, and executability issues are provided in [54, 59, 177, 175, 178, 179, 176]. An example of an application of reasoning about actions and change with multiple granularities is presented in [44].

Temporal abstraction synthesizes single higher-level concepts from raw time-stamped data. In this way, it allows abstracting from the considered domain only those aspects which are relevant to the actual goal. As an example, in a clinical context where a huge number of patient data is regularly acquired [45], an emerging pattern over a span of time has much more significance than an isolated finding or even a set of findings. According to Shahar [212, 213], the temporal abstraction task can be decomposed into into five parallel subtasks: (i) *temporal-context restriction* creates the relevant contexts for the interpretation of data, crucial for focusing and limiting the scope of the inference; (ii) *vertical temporal inference* computes values of higher-level concepts starting from values of contemporaneous input data or abstractions; (iii) *horizontal temporal inference* computes values of concepts at a given level starting from similar-type propositions that hold over different time intervals at that level; (iv) *temporal interpolation* bridges the gaps between similar-type, but temporally disjoint, propositions to create longer intervals; (v) *temporal pattern matching* creates intervals by matching patterns over disjoint intervals over which propositions of various types hold.

It is worth noting that time granularity and abstraction are *orthogonal* dimensions, that is, changing the level of time granularity does not necessarily require changing the level of abstraction, and vice versa.

Providing a formalism with the notions of time granularity and abstraction increases both its expressive and computational power. With respect to the *expressive power*, time granularity and abstraction increase both the temporal distinctions that a language can make and the distinctions that it can leave unspecified [247]. This means that considering two events as simultaneous or temporally distinct, or two time dependent relations as temporally overlapped or disjoint, depends on the time granularity one refers to. Similarly, we can analyze the evolution of a quantity at the detailed level of sampled measurements as well as at the abstract level of qualitative states, e.g. only distinguishing stable and unstable states, or normal and abnormal states. Temporal abstraction is affected by the context, e.g. the concept of normal/abnormal state of a patient may vary according to the specific patient pathology and morphology [55]. With respect to the *computational power*, reasoning at different levels of time granularity and abstraction allows dealing with incomplete and uncertain knowledge. They also allow tailoring the visibility of the knowledge base and the reasoning process to the needs of the actual task. Secondly, they allow alternating among different levels of time granularity and/or abstraction during the execution of a task in order to solve each incoming problem at a level of time granularity and/or abstraction as coarse as possible. Finally, they also possibly allow solving a problem at a level of time granularity and/or abstraction coarser than the required one. Such a simplification speeds up the reasoning, but implies a decrease in the precision of the solution.

Uncertainty in temporal knowledge. Uncertainty can affect a number of different aspects of reasoning about actions and change. In [132], Kim identifies four different types of uncertainty which are being dealt with by current research:

- *Observation uncertainty*: uncertainty about the values of observations returned by sensors or other devices.
- *Nondeterministic action laws*: effects of actions can be nondeterministic, leading to uncertainty about the consequences of an action.
- *Persistence duration uncertainty*: reasoning correctly about some properties requires to know how long they persist (e.g., we have to know how long a battery remains charged to decide when to replace it or to predict the effects of actions which refer to the battery charge as a precondition), but their period of persistence cannot be stated precisely in many cases.
- *Action time uncertainty*: we may not know precisely when an action occurred (this issue has been discussed also in Section 2.2). This uncertainty can affect both the changing time of properties (e.g., we know that someone is in a given place, but we do not know exactly when he arrived there) or the properties themselves (e.g., we do not know exactly when someone arrived at the station to catch a departing bus, so we cannot conclude that he/she is on that bus).

In order to deal with uncertainty in temporal representation and reasoning, a logic-based formalism which assigns a boolean truth value to the occurrence of an event at a given time point or to the validity of a property over a time interval is clearly inadequate. A suitable formalism should allow: (i) replacing boolean truth values by degrees of certainty, e.g., probabilities; (ii) modeling the indeterminacy in the relations between an event and its possible effects, e.g., an event produces a given effect with a given probability; (iii) taking into account the influence of the context on the effects of an event, e.g., the probability that an event produces a given effect can vary according to contextual conditions, and the probability of the different conditions must be considered.

Moreover, temporal aspects themselves often play a role in determining probabilities. For example, the effects of a fever depend also on how long it lasted, and the probability of a fault in an engineered device is higher when it is new (due to manufacturing defects) or when it is old (due to wear, stress, aging, etc.). Researchers working at integrating time and probability typically extend temporal approaches by introducing mathematical machinery such as Bayesian networks or Markov processes.

Seminal work on probabilistic reasoning about actions has been performed by Pearl [191]. In particular, he proposes a symbolic approach that admits both probabilistic and causal knowledge about a given domain and produces probabilistic statements about the effects of actions and the impact of observations. The calculus admits two types of conditioning operators: ordinary Bayes conditioning $p(y|X = x)$, which represents the observation $X = x$, and causal conditioning, $P(y|do(X = x))$, i.e. the probability of $Y = y$ conditioned on holding X constant (at x) by deliberate action. Given a mixture of such observational and causal sentences, together with the topology of the causal graph, the calculus derives new conditional probabilities of both types, thus enabling one to quantify the effects of actions and observations. A general account of the development of the Bayesian networks area in AI is given in [191, 192].

The approaches which introduce temporal aspects in Bayesian networks typically represent time discretely and create an instance of each time-varying random variable for each point in time [108]. This leads to an increase in the size of the network which can be problematic in the calculation of posterior probabilities (which is an NP-hard problem [58]). This problem is discussed in detail by Ngo et al. [186], who propose to use deterministic information as a context to index the probabilistic information, thus restricting the number of nodes and links to be considered. For example, the representation of a plan as a sequence of single actions gives deterministic knowledge about which actions we intend to execute at each time point. In particular, Ngo et al. illustrate their approach on the problem of reasoning about the effects of medications and interventions on the state of a patient in cardiac arrest.

An example of a possible use of Markov chains is in dealing with the issue of irrelevance [232]: the problem is to identify which information is relevant and which is irrelevant in order to draw a conclusion, and relevance is influenced by time, e.g. if a cat was seen in the living room two weeks ago and we are interested in its actual location, the previous information is not of much help (on the contrary, if the cat has been seen two minutes ago, the relevance greatly increases). The approach proposed by Tawfik and Neufeld models the degeneration of relevance with time by using a Markov chain representation.

Temporal logics. Temporal logic (re-)emerged in our century as a significant research area in philosophy and linguistics, mainly due to Prior's work. Prior's ultimate aim was the development of a calculus that formally captures tenses in natural language and, to this end, he conceived a logical system embodying temporal operators analogous to the operators of modal logic. The use of temporal logic in artificial intelligence dates back to McCarthy and Hayes' seminal work on SC in the late '60s. The basic notion of situation variable in SC is indeed closely related to the interpretation of time instants as world-state propositions characterizing Prior's third grade of tense-logical involvement. Since then, temporal logic is the backbone of reasoning about actions and change. As shown in Section 2, several logic-based formalisms have been proposed and extensively used for representing and reasoning about temporal knowledge. Traditionally, they are roughly partitioned on the basis of the interval-based vs. point-based paradigm they choose. Broad surveys on both these paradigms can be found in [25, 26]. More recently, significant research effort has been devoted to the integration of the two paradigms as well as to the study of the representational complexity of temporal logic. An increasing attention has also been paid to the application of temporal logic to classical tasks such as planning and monitoring.

The complexity of temporal logics is the main focus of [28]. On the basis of the observation that the complexity of temporal phenomena is often a reflection of their formal modeling as much as of their intrinsic nature, van Benthem discusses several logical strategies for remodeling temporal logics so as to lower their complexity. Then, he introduces a two-sorted geometrical temporal model, having points/states and (finite and infinite) paths as primitives, and shows how this model makes it possible to overcome the traditional opposition between point-based and interval-based temporal ontologies. This allows viewing the various temporal logics proposed in the literature as specific theories of states and paths. Finally, he presents two possible ways of hierarchically ordering temporal logics. On the one hand, there exists a natural syntactic hierarchy of modal fragments, starting from Prior's tense logic, running upward to 'Since'/'Until' logics, and beyond. On the other hand, a matching semantic hierarchy of two-sorted point-path bisimulations can be introduced that relates temporal models that are geometrically alike to some degree.

As far as applications to classical reasoning tasks is concerned, the usefulness of temporal logic for planning has been shown by several authors including Drummond [73], Haddawy and Hanks [107], Kabanza et al. [6, 7, 15, 98, 121]. While traditional planning concentrated on primitive actions and instantaneous states, the introduction of a temporal dimension allows reasoning about sequences of actions and/or about the sequences of states they generate. As an example, Bacchus and Kabanza [7] exploit quite simple temporal logics, whose syntax contains the usual temporal modalities and whose models are linear sequences of states. Nevertheless, such logics can be successfully used to specify search control information as well as to specify goals that are temporally extended over time, that is, to constrain the way the agent achieves its final goal. Furthermore, they show how temporal logic can be used to define history-dependent rewards to be assigned to states in order to generate optimal plans. Brajnik and Clancy [32] consider the task of qualitative simulation, i.e. constructing a set of possible behaviors consistent with a model of a dynamical system represented by qualitative versions of differential equations. They extend the well-known qualitative simulator QSIM [139] introducing a propositional linear-time temporal logic, in order to provide temporal constraints on the trajectories of variables as an additional input to the simulator. Then they enforce the validity of these temporal constraints during simulation in order to restrict the set of possible evolutions as opposed to testing the outcomes of the simulation after its completion.

The problem of supporting derivability in temporal logic, i.e., automating temporal logic theorem proving, is a research area on its own. There are essentially two possible approaches to such a problem: the direct approach of finding a proof algorithm for a specific class of temporal logics and the translation of temporal logics into suitable first-order theories to which standard predicate calculus theorem provers are then applied. An up-to-date introduction to the main contributions adopting the first approach can be found in [82], while a survey on translation methods for non-classical logics is given in [189].

Logic Programming. Among the fields which present a significant overlap with research on temporal representation and reasoning, logic programming is probably one of the most important. In particular, it turned out to be a natural setting for the formalization of the theories of actions and change [154]. Kowalski and Sergot's EC was indeed originally developed in a logic programming framework; McCarthy and Hayes' SC can be easily reformulated in logic programming as well [24, 137]. Other subsequent formalisms for describing actions and their effects, such as the languages $\mathcal{A}$ [93] and $\mathcal{E}$ [122], have also been defined in logic programming.

In [154], Lifschitz highlights the major advantages of using logic programming for reasoning about actions and change. First of all, the use of negation as failure leads to a simple solution to the frame problem. As already shown in Section 2.2, unlike the first attempts to solve the frame problem using circumscription, the logic programming formulation does not suffer from the problems pointed out by the Yale Shooting Problem example. Moreover, some of the reasoning tasks related to actions can be solved by simply running the corresponding programs under Prolog, without developing any additional algorithms for nonmonotonic reasoning. Furthermore, research on representing actions by logic programs has demonstrated the potential of several extensions of basic logic programming such as the use of abduction, disjunction, and explicit negation.

A different perspective on the relationships between logic programming and reasoning about actions and change is suggested in [147], where Levesque et al. propose the new logic programming language GOLOG, based on the Situation Calculus, for defining complex agent behavior, and experiment with it in applications such as high-level control of robots and industrial processes, intelligent software agents, discrete event simulation.

Temporal databases. There is an increasing evidence that a number of important problems in AI have a natural counterpart in relational databases, and vice versa. As an example, there exists a close relation between constraint satisfaction problems in AI and the satisfaction of join-dependencies in relational databases [103]. Thanks to this relation, decomposition techniques developed for relational databases can be successfully applied to decompose constraint satisfaction problems. The existence of similar connections between problems in the area of temporal representation and reasoning and the area of temporal databases has been investigated in recent years.

Unlike conventional databases which are designed to store the most recent available data, temporal databases maintain both the current and the past states of the database [190, 226]. Moreover, each state is not a simple snapshot of the world, but a temporally extended description of it, involving past, present, and possibly future data. Since in temporal databases many data change their values over time, it makes sense to provide them with the machinery for performing basic temporal inferences. In particular, some temporal databases give the possibility of managing temporal information not only to retrieve information as stored in the database, but also to automatically derive further data. To this end, temporal query and update languages must take into account that temporal information may be incomplete (events may be only partially ordered), metric and layered (events may be associated with different time granularities and their occurrence times may be known within metric bounds) and defeasible (assumptions about the persistence of certain facts may turn out to be unwarranted as new information is acquired). Further interesting connections between AI and temporal databases concern the problem of concurrency control in transaction processing. As an example, Finger and McBrien [81] propose a *temporal serialization theory* for concurrency control, based on the notion of perceivedly instantaneous interactions, that overcomes the problems of standard serialization theories; such a theory presents interesting connections with reasoning about partially ordered events. In particular, there are many similarities: (i) between the notion of locality in event ordering proposed by Lin and Dean [159] and the collection of a number of operations into a single transaction, and (ii) between the problem of establishing if a given history is temporally serializable and the problem of determining the existence of a completion of a given partial order of events that satisfies certain conditions [40].

Medical informatics. Much work is currently being devoted to the design of temporal problem solving methods in the medical informatics area, resulting in relevant applications. One indication of the significance of research on time-oriented systems in medicine is that the recent level and amount of scientific work in this area motivated two different special issues of the journal *Artificial Intelligence in Medicine* [129, 130], and a special issue of the journal *Computers in Biology and Medicine* [55].

In their survey on temporal representation and reasoning for medical informatics, Combi and Shahar [56] point out its most important applications, which are: (i) management of time-oriented data stored in medical records of ambulatory or hospitalized patients; (ii) prediction of future values of clinical data, given past trends; (iii) abstraction of time-oriented clinical data; (iv) time-oriented knowledge-based decision-support systems, such as systems supporting diagnosis, monitoring, or therapy planning.

As it appears from an analysis of the three special issues mentioned above, the clinical areas where most investigations of time-oriented applications have been carried out are cardiology, oncology, internal medicine, intensive care and anesthesiology. The most investigated tasks are patient management, protocol- and guideline-based therapy, therapy administration and monitoring, and diagnosis.

Among the various temporal representation and reasoning areas, temporal databases is probably the one which is attracting more attention in medical informatics. Two major approaches to the development of clinical temporal databases are followed [56]. The first is based on extending conventional existing clinical databases, by adding a temporal dimension at the tuple/object level or at the attribute/method level. Then, temporal features of clinical data can be retrieved by suitable query languages. The second approach is based on creating task-specific, time-oriented entities, such as events, at the database level. This second approach facilitates the definition of abstraction hierarchies among different concepts, which can satisfy the need for abstracting time-oriented data typical of many medical applications. For example, the general temporal abstraction mechanisms proposed by Shahar [213] have been developed in the context of medical applications, and are being used for abstraction of clinical data in the patient monitoring task.

Even more theoretical areas of temporal representation and reasoning, such as reasoning about actions and change, are currently under study and experimentation in the medical informatics domain. In particular, there have been some recent experiments involving the Event Calculus. For example, Keravnou and Washbrook [131] used EC to represent time-dependent knowledge in the diagnosis of skeletal dysplasias, and Chittaro and Dojat employed it inside a prototype system for the monitoring and control of mechanical ventilation in intensive care [47]. The results of the evaluation of the prototype cite non-monotonic reasoning and basic temporal data base capabilities as the most appreciated aspects of EC, while it also points out that there is a number of additional aspects which are useful in the medical domain but are not yet adequately supported by actions and change theories (e.g., temporal abstraction, temporal granularity, intuitive query languages).

5 Concluding Remarks

With this paper, we aimed at both providing a comprehensive and up-to-date survey on the main areas of temporal representation and reasoning research (temporal constraints and reasoning about actions and change) and at briefly outlining some of the major emerging trends in the field. The reader interested in deepening specific topics is referred to specialized conferences, such as the TIME (International Workshop on Temporal Representation and Reasoning) and ICTL (International Conference on Temporal Logic) series. In recent years, the area of temporal representation and reasoning has also received an increasing recognition in the major general AI conferences and journals.

Temporal constraint research explores an area whose scope is clearly delimited, and benefits from known results coming from the broader, well-established constraint satisfaction community. Its most original contributions are in the identification of languages which are both tractable and sufficiently expressive for modeling meaningful temporal applications. The area is characterized by a general consensus on the main topics and achievements (cf. Section 3.1). The area of reasoning about actions and change is less stabilized, but we have to consider that it has a broader scope and its goals are more ambitious. Since its beginnings in the '60s, research in this area raised a number of interesting new problems (cf. Section 2.1), whose study generated a rich variety of approaches and proposals. As a consequence, there is not yet a general consensus on the solutions to adopt, as witnessed by the debate on the Electronic Newsletter on Reasoning about Actions and Change [207]. As it happened with the area of temporal constraints, reasoning about actions and change could benefit from strengthening connections with research in other fields. In particular, many basic results of philosophical, mathematical, and computer science research on temporal logic should become part of the common background of the community.

Acknowledgements

We would like to thank Peter van Beek, Carlo Combi, Federico Fontana, Martin Golumbic, Alessandro Provetti, and the anonymous referees for their useful comments to earlier versions of this paper. We are also grateful to Iliano Cervesato who proofread the paper.

References

[1] Alferes J., R. Li, and L. Pereira, Concurrent Actions and Changes in the Situation Calculus, Proc. of IBERAMIA-94, 93–104, 1994.

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

[3] Allen J. F., Toward a general theory of action and time, *Artificial Intelligence*, 23:123–154, 1984.

[4] Allen J. F., and G. Ferguson, Actions and Events in Interval Temporal Logic, *Journal of Logic and Computation*, 4(5):531–580, 1994.

[5] Bacchus F., J. Y. Halpern, and H.J. Levesque, Reasoning about noisy sensors in the Situation Calculus, Proc. of 14th International Joint Conference on Artificial Intelligence (IJCAI), Montréal, Québec, Morgan Kaufmann, 1933–1940, 1995.

[6] Bacchus F., and F. Kabanza, Using temporal logic to control search in a forward chaining planner, in *New Directions in AI Planning*, M. Ghallab and A. Milano (eds.), 141–153, IOS Press, 1996.

[7] Bacchus F., and F. Kabanza, Planning for temporally extended goals, in [52], 5–27, 1998.

[8] Bacchus F., J. Tenenberg, and J.A. Koomen, A non-reified temporal logic, *Artificial Intelligence*, 52:87–108, 1991.

[9] Baker A. B., A Simple Solution to the Yale Shooting Problem. Proc. of the 1st International Conference on Principles of Knowledge Representation and Reasoning (KR), Morgan Kaufmann, 11–20, 1989.

[10] Baker A. B., Nonmonotonic reasoning in the framework of situation calculus, *Artificial Intelligence*, 49:5–23, 1991.

[11] Baker A. B., and Y. Shoham, Nonmonotonic Temporal Reasoning, in *Handbook of Logic in Artificial Intelligence and Logic Programming*, D. Gabbay, C. Hogger, and J. Robinson (eds.), Oxford University Press, 1995.

[12] Baral C., and M. Gelfond, Representing Concurrent Actions in Extended Logic Programming, Proc. of 13th International Joint Conference on Artificial Intelligence (IJCAI), Chambéry, France, Morgan Kaufmann, 866–871, 1993.

[13] Baral C., and M. Gelfond, Reasoning about Effects of Concurrent Actions, in [155], 85–117, 1997.

[14] Baral C., M. Gelfond, and A. Provetti, Representing Actions: Laws, Observations and Hypotheses, in [155], 201–243, 1997.

[15] Barbeau M., F. Kabanza, and R. St-Denis, Synthesizing Plant Controllers using Real-Time Goals, Proc. of 14th International Joint Conference on Artificial Intelligence (IJCAI), Montréal, Québec, Morgan Kaufmann, 791–798, 1995.

[16] Baudinet M., J. Chomicki, and P. Wolper, Chapter 13: Temporal Deductive Databases, in [230], 294–320, 1993.

[17] van Beek P., Approximation Algorithms for Temporal Reasoning, Proc. of the 11th International Joint Conference on Artificial Intelligence (IJCAI), Detroit, Mich., Morgan Kaufmann, 1291–1296, 1989.

[18] van Beek P., Reasoning about Qualitative Temporal Information, Proc. of the 8th National Conference of the American Association for Artificial Intelligence (AAAI), AAAI Press/MIT Press, Cambridge, Mass., 728–734, 1990.

[19] van Beek P., Reasoning about Qualitative Temporal Information, *Artificial Intelligence*, 58:297–326, 1992.

[20] van Beek P., and D. W. Manchak, The Design and Experimental Analysis of Algorithms for Temporal Reasoning, *Journal of Artificial Intelligence Research*, 4:1-18, 1996.

[21] van Beek P., and R. Cohen, Exact and Approximate Reasoning about Temporal Relations, *Computational Intelligence*, 6:132-144, 1990.

[22] Belegrinos P., and M. Georgeff, A Model of Events and Processes, Proc. of 12th International Joint Conference on Artificial Intelligence (IJCAI), Sydney, Australia, Morgan Kaufmann, 506–511, 1991.

[23] van Belleghem K., M. Denecker, and D. De Schreye, Combining Situation Calculus and Event Calculus, Proc. of the 12th International Conference on Logic Programming (ICLP), Kanagawa, Japan, MIT Press, 83–97, 1995.

[24] van Belleghem K., M. Denecker, and D. De Schreye, On the relation between Situation Calculus and Event Calculus, in [155], 3–37, 1997.

[25] van Benthem J., *The logic of time (2nd Edition)*, Kluwer Academic Publishers, 1991.

[26] van Benthem J., Temporal Logic, in: *Handbook of Logic in Artificial Intelligence and Logic Programming, Vol. 4*, D. Gabbay, C. Hogger, and J. Robinson (eds.), Oxford University Press, 241-350, 1995.

[27] van Benthem J., Points on Time, *Electronic News Journal on Reasoning about Actions and Change*, Vol. 2, http://www.ida.liu.se/ext/etai/rac/notes/1998/03/, 1998.

[28] van Benthem J., Temporal Patterns and Modal Structure, *The Logic Journal of IGPL*, 7(1), 1999.

[29] Bessière C., A. Isli, and G. Ligozat, Global consistency in interval algebra networks: tractable subclasses, Proc. of the 12th European Conference on Artificial Intelligence (ECAI), Budapest, Hungary, John Wiley & Sons, 3–7, 1996.

[30] Boddy M., AAAI-92 Workshop Report: Implementing Temporal Reasoning, *SIGART Bulletin*, 4(3):15–16, 1993.

[31] Boddy M., Temporal Reasoning for Planning and Scheduling, *SIGART Bulletin*, 4(3):17–20, 1993.

[32] Brajnik G., and D. Clancy, Focusing Qualitative Simulation Using Temporal Logic: Theoretical Foundations, in [52], 59–86, 1998.

[33] Brusoni V., L. Console, P. Terenziani, and B. Pernici, Later: Managing Temporal Information Efficiently, *IEEE Expert*, 12(4):56–64, 1997.

[34] Cervesato, I., L. Chittaro, and A. Montanari, A Modal Calculus of Partially Ordered Events in a Logic Programming Framework, Proc. of the 12th International Conference on Logic Programming (ICLP), Kanagawa, Japan, MIT Press, 299–313, 1995.

[35] Cervesato I., M. Franceschet, and A. Montanari, A Hierarchy of Modal Event Calculi: Expressiveness and Complexity, Proc. of the 2nd International Conference on Temporal Logic (ICTL), Manchester, UK, Kluwer Academic Press, 75–89, in press, 1999.

[36] Cervesato I., M. Franceschet, and A. Montanari, Modal Event Calculi with Preconditions, Proc. of the 4th International Workshop on Temporal Representation and Reasoning (TIME), Daytona Beach, FL, IEEE Computer Society Press, 38–45, 1997.

[37] Cervesato I., M. Franceschet, and A. Montanari, Event Calculi with Explicit Quantifiers, Proc. of the 5th International Workshop on Temporal Representation and Reasoning (TIME), Sanibel Island, FL, IEEE Computer Society Press, 81–88, 1998.

[38] Cervesato I., M. Franceschet, and A. Montanari, The Complexity of Model Checking in Modal Event Calculi with Quantifiers, *Electronic Transactions on Artificial Intelligence* (http://www.ep.liu.se/ej/etai/1998/001/), 2(1-2):1-23, 1998. (Extended and revised version of Cervesato I., M. Franceschet, and A. Montanari, The Complexity of Model Checking in Modal Event Calculi with Quantifiers, Proc. of the 6th International Conference on Principles of Knowledge Representation and Reasoning (KR), Trento, Italy, Morgan Kaufmann, 368–379, 1998.)

[39] Cervesato I., M. Franceschet, and A. Montanari, A Guided Tour Through some Extensions of the Event Calculus, accepted for publication in *Computational Intelligence*, August 1999.

[40] Cervesato I., and A. Montanari, A General Modal Framework for the Event Calculus and its Skeptical and Credulous Variants, *Journal of Logic Programming*, 38(2):111–164, 1999. (Extended and revised version of Montanari A., L. Chittaro, and I. Cervesato, A general

modal framework for the event calculus and its skeptical and credulous variants, Proc. of the 12th European Conference on Artificial Intelligence (ECAI), Budapest, Hungary, John Wiley & Sons, 33–37 , 1996.)

[41] Chapman D., Planning for conjunctive goals, *Artificial Intelligence*, 32:333–377, 1987.

[42] Cheeseman, P., B. Kanefsky, and W.M. Taylor, Where the really hard problems are, Proc. of 12th International Joint Conference on Artificial Intelligence (IJCAI), Sydney, Australia, Morgan Kaufmann, 331–337, 1991.

[43] Chittaro L., and C. Combi, Temporal Indeterminacy in Deductive Databases: an Approach based on the Event Calculus, Proc. of 2nd International Workshop on Active, Real-time and Temporal Database Systems (ARTDB), LNAI 1553, Springer, 212–227, 1998.

[44] Chittaro L, and C. Combi, Reasoning about Events with Imprecise Location and Multiple Granularities, Proc. of the 10th International Conference on Database and Expert Systems Applications (DEXA), Florence, Italy, Springer, 1006–1017, 1999.

[45] Chittaro L, and C. Combi, Abstraction on clinical data sequences: an object-oriented data model and a query language based on the Event Calculus, *Artificial Intelligence in Medicine Journal*, in press, 1999.

[46] Chittaro L., C. Combi, E. Cervesato, A. Cervesato, F. Antonini-Canterin, G.L. Nicolosi, and D. Zanuttini, Specifying and representing temporal abstractions of clinical data by a query language based on the Event Calculus, Proc. of the 24th Annual Conference on Computers in Cardiology, IEEE Press, 1997.

[47] Chittaro L., and M. Dojat, Using a General Theory of Time and Change for Patient Monitoring: Experiment and Evaluation, in [55], 435–452, 1997.

[48] Chittaro L., Hamilton H., Goodwin S., and Montanari A. (eds.) *Proc. TIME-96: 3rd International Workshop on Temporal Representation and Reasoning*, IEEE Computer Society Press, 1996.

[49] Chittaro L., and A. Montanari, Experimenting a Temporal Logic for Executable Specification in an Engineering Domain, in Rzevski G., J. Pastor, and R.A. Adey (eds.), *Applications of Artificial Intelligence in Engineering VIII*, Elsevier Applied Science, 185–202, 1993.

[50] Chittaro L., and A. Montanari, Efficient Temporal Reasoning in the Cached Event Calculus, *Computational Intelligence*, 12(3):359–382, 1996.

[51] Chittaro L., and A. Montanari, Trends in temporal representation and reasoning, *The Knowledge Engineering Review*, 11(3):281–288, 1996.

[52] Chittaro L., and A. Montanari (eds.), Special issue on Temporal Representation and Reasoning, *Annals of Mathematics and Artificial Intelligence*, 22(1-2), 1998.

[53] Chittaro L., A. Montanari, and A. Provetti, Skeptical and Credulous Event Calculi for Supporting Modal Queries, Proc. of the 11th European Conference on Artificial Intelligence (ECAI), Amsterdam, The Netherlands, John Wiley & Sons, 361–365, 1994.

[54] Ciapessoni E., E. Corsetti, A. Montanari, and P. San Pietro, Embedding Time Granularity in a Logical Specification Language for Synchronous Real-Time Systems, *Science of Computer Programming*, 20(1-2):141–171, 1993.

[55] Combi C., and Y. Shahar (eds.), Special Issue on Time-oriented Systems in Medicine, *Computers in Biology and Medicine*, 27(5), 1997.

[56] Combi C., and Y. Shahar, Temporal Reasoning and Temporal Data Maintenance in Medicine: Issues and Challenges, in [55], 353–368, 1997.

[57] Console L., D. Theseider Dupré, and P. Torasso, On the relationship between abduction and deduction, *Journal of Logic and Computation*, 1(5):661-690, 1991.

[58] Cooper G.F., The computational complexity of probabilistic inference using Bayesian belief networks, *Artificial Intelligence* 42, 393–405, 1990.

[59] Corsetti E., A. Montanari, and E. Ratto, Dealing with Different Time Granularities in Formal Specifications of Real-Time Systems, *The Journal of Real-Time Systems*, 3(2):191–215, 1991.

[60] Crawford J., and D. W. Etherington, Formalizing Reasoning about Change: a Qualitative Reasoning Approach, Proc. of the 10th National Conference of the American Association for Artificial Intelligence (AAAI), AAAI Press/MIT Press, 577-583, 1992.
[61] Davis E., *Representation of Commonsense Knowledge*, Morgan Kaufmann, 1990.
[62] Dean T., and M. Boddy. Reasoning about partially ordered events, *Artificial Intelligence*, 36:375–399, 1988.
[63] Dean T., and D. V. Mc Dermott, Temporal Data Base Management, *Artificial Intelligence*, 32:1–55, 1987.
[64] Dechter R., I. Meiri, and J. Pearl, Temporal constraint networks, *Artificial Intelligence*, 49:61–95, 1991.
[65] Denecker M., L. Missiaen, and M. Bruynooghe, Temporal Reasoning with Abductive Event Calculus, Proc. of the 10th European Conference on Artificial Intelligence (ECAI), Vienna, Austria, John Wiley & Sons, 1992.
[66] Deville Y., and P. Van Hentenryck, An efficient arc consistency algorithm for a class of CSP problems, Proc. of 12th International Joint Conference on Artificial Intelligence (IJCAI), Sydney, Australia, Morgan Kaufmann, 325–330, 1991.
[67] Doherty P., Reasoning about Action and Change Using Occlusion, Proc. of the 11th European Conference on Artificial Intelligence, Amsterdam (ECAI), The Netherlands, John Wiley & Sons, 401–405, 1994.
[68] Doherty P., and W. Lukaszewicz, Circumscribing Features and Fluents, Proc. of the 1st International Conference on Temporal Logics (ICTL), LNAI 827, Bonn, Germany, Springer-Verlag, 82–100, 1994.
[69] Drakengren T., and P. Jonsson. Maximal Tractable Subclasses of Allen's Interval Algebra: Preliminary Report, Proc. of the 13th Conference of the American Association for Artificial Intelligence (AAAI), Portland, OR, AAAI Press/MIT Press, 389–394, 1996.
[70] Drakengren T., and P. Jonsson. Twenty-One Large Tractable Subclasses of Allen's Algebra, *Artificial Intelligence*, 93:297–319, 1997.
[71] Drakengren T., and P. Jonsson. Eight Maximal Tractable Subclasses of Allen's Algebra with Metric Time, *Journal of Artificial Intelligence Research*, 7:25–45, 1997.
[72] Drakengren T., and P. Jonsson. Towards a complete classification of tractability in Allen's Algebra, Proc. of 15th International Joint Conference on Artificial Intelligence (IJCAI), Nagoya, Japan, Morgan Kaufmann, 1466–1471, 1997.
[73] Drummond M., Situated Control Rules, Proc. of the 1st International Conference on Principles of Knowledge Representation and Reasoning (KR), Morgan Kaufmann, 103–113, 1989.
[74] Errico B., and L. C. Aiello. Agents in the Situation Calculus: an Application to User Modeling, in D. Gabbay and H. J. Ohlbach (eds.), *Practical Reasoning*, LNAI 1085, Springer-Verlag, 126-140, 1996.
[75] Eshghi K., Abductive Planning with Event Calculus, Proc. of the 5th International Conference and Symposium on Logic Programming (ICSLP), Seattle, WA, MIT Press, 562–579, 1988.
[76] Eshghi K., and R. Kowalski, Abduction through deduction, Technical Report, Department of Computing, Imperial College, University of London, 1988.
[77] Evans C., Negation-as-failure as an an approach to the Hanks and McDermott problem, Proc. of the 2nd International Symposium on Artificial Intelligence, 1989.
[78] Evans C., The Macro-Event Calculus: Representing Temporal Granularity, Proc. PRICAI-90, 1990.
[79] Farquhar A., A Qualitative physics compiler, Proc. of the 12th Conference of the American Association for Artificial Intelligence (AAAI), AAAI Press/MIT Press, 1168–1174, 1994.
[80] Fikes R., and N. Nilsson, STRIPS: a new approach to the application of theorem proving to problem solving, *Artificial Intelligence*, 2:189–208, 1971.
[81] Finger M., P. McBrien, Concurrency control for perceivedly instantaneous transactions in

valid-time databases, Proc. of the 4th International Workshop on Temporal Representation and Reasoning (TIME), Daytona Beach, FL, IEEE Computer Society Press, 112–118, 1997.
[82] Fisher M. An Introduction to Executable Temporal Logics, *The Knowledge Engineering Review*, 11(1), 1996.
[83] Forbus K., Qualitative Process Theory, Artificial Intelligence 24:85–168, 1984.
[84] Franceschet M., and A. Montanari, A graph-theoretic approach to efficiently reasoning about partially ordered events in the event calculus, Proc. of the 6th International Workshop on Temporal Representation and Reasoning (TIME), Orlando, FL, IEEE Computer Society Press, 55–66, 1999.
[85] Franceschet M., and A. Montanari, Pairing transitive closure and reduction to efficiently reason about partially ordered events, Proc. of the 3rd International Workshop on Non-monotonic Reasoning, Action and Change (NRAC), 79–86, 1999.
[86] Freuder E. C. Synthesizing Constraint Expressions, *Communications of the ACM*, 21(11):958–966, 1978.
[87] Freuder E. C. A Sufficient Condition for Backtrack-Free Search, *Journal of the ACM*, 29(1):24–32, 1982.
[88] Galton A., Reified temporal theories and how to unreify them, Proc. of 12th International Joint Conference on Artificial Intelligence (IJCAI), Sydney, Australia, Morgan Kaufmann, 1177–1182, 1991.
[89] Galton A., An Investigation of "Non-Intermingling" Principles in Temporal Logic, *Journal of Logic and Computation*, 6(2):271–294, 1996.
[90] Gelfond M., Autoepistemic logic and formalization of common-sense reasoning, Proc. of the 2nd International Workshop on Non-Monotonic Reasoning, LNAI 346, Springer-Verlag, 1989.
[91] Gelfond M., and V. Lifschitz, The stable model semantics for logic programming, Proc. of the 5th International Conference and Symposium on Logic Programming (ICSLP), Seattle, MIT Press, 1988.
[92] Gelfond M., and V. Lifschitz, Representing Actions in Extended Logic Programs, Proc. of the Joint International Conference and Symposium on Logic Programming (ICSLP), MIT Press, 559–573, 1992.
[93] Gelfond M., and V. Lifschitz, Representing action and change by logic programs, *Journal of Logic Programming*, 17(2–4):301–322, 1993.
[94] Gelfond M., V. Lifschitz, and A. Rabinov, What are the Limitations of the Situation Calculus?, in Boyer R. (ed.), *Automated Reasoning: Essays in Honor of Woody Bledsoe*, Kluwer Academic, 1991.
[95] Gerevini A., and L. Schubert, Efficient algorithms for qualitative reasoning about time, *Artificial Intelligence*, 74:207–248, 1995.
[96] Ginsberg M. L., Counterfactuals, *Artificial Intelligence*, 30(1):35–79, 1986.
[97] Ginsburg M., and D. E. Smith, Reasoning about action II: the qualification problem, *Artificial Intelligence*, 35(3):311–342, 1988.
[98] Godefroid P., and F. Kabanza, An Efficient Reactive Planner for Synthesizing Reactive Plans, Proc. of the 9th National Conference of the American Association for Artificial Intelligence (AAAI), AAAI Press/MIT Press, 640–645, 1991.
[99] Golumbic, M.C., Reasoning about Time, in *Mathematical Aspects of Artificial Intelligence*, Hoffman F. (ed.), American Math. Society, Proc. of Symposia in Applied Mathematics, 55:19–53, 1998.
[100] Golumbic, M.C., Kaplan, H., and R. Shamir, On the complexity of DNA physical mapping, *Advances in Applied Mathematics*, 15:251–261, 1994.
[101] Golumbic, M.C., and R. Shamir, Complexity and algorithms for reasoning about time: a graph-theoretic approach, *Journal of ACM*, 40(5):1108–1133, 1993.
[102] Guesgen H.W., Anger F. D., Ligozat G., and R.V. Rodriguez (eds.), Special issue on Spatial

and Temporal Reasoning, *Constraints*, 3(2-3), 1998.
[103] Gyssens M., P. Jeavons, and D. Cohen, Decomposing constraint satisfaction problems using database techniques, *Artificial Intelligence*, 66(1):57–89, 1994.
[104] Gustafsson J., and P. Doherty, Embracing Occlusion in Specifying the Indirect Effects of Actions, Proc. of the 5th International Conference on Principles of Knowledge Representation and Reasoning (KR), Morgan Kaufmann, 87–98, 1996.
[105] Gustafsson J., and L. Karlsson, Reasoning about actions in a multi-agent environment, *Linköping Electronic Articles in Computer and Information Science*, http://www.ep.liu.se/ea/cis/1997/014, 1997.
[106] Haas A., The case for domain-specific frame axioms, in *The frame problem in artificial intelligence*, F. Brown (ed.), 343–348, Morgan Kaufmann, 1987.
[107] Haddawy P., and S. Hanks, Utility Models for Goal-Directed Decision-Theoretic Planners, Technical Report 93-06-04, University of Washington, 1993.
[108] Haddawy P., Representing plans under uncertainty: a logic of time, change, and action, *Lecture Notes in Artificial Intelligence* 770, Springer-Verlag, Berlin, 1994.
[109] Halpern J.Y., and Y. Shoham, A propositional modal logic of time intervals, *Journal of the ACM*, 38(4):935–962, 1991.
[110] Hanks S., and D. McDermott, Default Reasoning, Nonmonotonic Logics, and the Frame Problem. Proc. of the 5th National Conference on Artificial Intelligence (AAAI), Philadelphia, PA, AAAI Press, 328–333, 1986.
[111] Hanks S., and D. McDermott, Nonmonotonic logics and temporal projection, *Artificial Intelligence*, 33:379–412, 1987.
[112] Haugh B. A., Simple Causal Minimizations for Temporal Persistence and Projection, Proc. of the 6th National Conference of the American Association for Artificial Intelligence (AAAI), AAAI Press, 218–223, 1987.
[113] Hayes P. J., The Second Naive Physics Manifesto, in *Formal Theories of the Commonsense World*, J. R. Hobbs and R. C. Moore (eds.), 1–36, Ablex, 1985.
[114] Hayes P. J., Contribution to the panel debate on theory evaluation, *Electronic Newsletter on Reasoning about Actions and Change*, Issue 97009, http://www.ida.liu.se/ext/etai/actions/njl/97009, 1997.
[115] Hoare C. A. R., *Communicating Sequential Processes*, Prentice Hall, 1985.
[116] Hölldobler S., *Foundations of Equational Logic Programming*, LNAI 353, Springer-Verlag, 1989.
[117] Hölldobler S., Situations, Actions, and Causality in the Fluent Calculus, Technical Report WV-97-01, Knowledge Representation and Reasoning Group, AI Institute, Dept. of Computer Science, Dresden University of Technology, 1997.
[118] Hölldobler S., and J. Schneeberger, A new deductive approach to planning, *New Generation Computing*, 8:225–244, 1990.
[119] Hölldobler S., and M. Thielscher, Actions and Specificity, Proc. of the 10th International Conference on Logic Programming (ICLP), MIT Press, 164–180, 1993.
[120] Hölldobler S., and M. Thielscher, Computing change and specificity with equational logic programs, *Annals of Mathematics and Artificial Intelligence*, 14:99–133, 1995.
[121] Kabanza F., Synthesis of Reactive Plans for Multi-Path Environments, Proc. of the 8th National Conference of the American Association for Artificial Intelligence (AAAI), 164–169, 1990.
[122] Kakas A., and R. Miller, A simple declarative language for describing narratives with actions, in [155], 157–200, 1997.
[123] Kartha G.N., Soundness and Completeness Theorems for Three Formalizations of Actions, Proc. of 13th International Joint Conference on Artificial Intelligence (IJCAI), Chambéry, France, Morgan Kaufmann, 724–729, 1993.
[124] Kartha G.N., Two Counterexamples Related To Baker's Approach to the Frame Problem,

Artificial Intelligence, 69:379–392, 1994.
[125] Kartha G.N., and V. Lifschitz, Actions with Indirect Effects (preliminary report), Proc. of the 4th International Conference on Principles of Knowledge Representation and Reasoning (KR), Morgan Kaufmann, 341–350, 1994.
[126] Kautz, H., The Logic of Persistence. Proc. of the 5th National Conference on Artificial Intelligence (AAAI), Philadelphia, PA, AAAI Press, 401–405, 1986.
[127] Kautz, H., and P. Ladkin, Integrating Metric and Qualitative Temporal Reasoning, Proc. of the 9th National Conference of the American Association for Artificial Intelligence (AAAI), Anaheim, CA, AAAI Press/MIT Press, 241–246, 1991.
[128] Kayser D., and A. Moktari, Time in a Causal Theory, in [52], 117–138, 1998.
[129] Keravnou E. T. (ed.) Special Issue on Temporal Reasoning in Medicine, *Artificial Intelligence in Medicine*, 3(6), 1991.
[130] Keravnou E. T. (ed.) Special Issue on Temporal Reasoning in Medicine, *Artificial Intelligence in Medicine*, 8(3), 1996.
[131] Keravnou E. T., and J. Washbrook, A temporal framework used in the diagnosis of skeletal dysplasias, *Artificial Intelligence in Medicine*, 2:239–265, 1990.
[132] Kim H., *Prediction and Postdiction under Uncertainty*, PhD thesis, Department of Computer and Information Science, Linköping University, Sweden, 1995.
[133] de Kleer J., and Brown J.S., A Qualitative Physics based on Confluences, *Artificial Intelligence*, 24:7–84, 1984.
[134] Koehler J., Correct Modification of Complex Plans. Proc. of the 11th European Conference on Artificial Intelligence (ECAI), Amsterdam, Netherlands, John Wiley & Sons, 605–609, 1994.
[135] Koubarakis M., From local to global consistency in temporal constraint networks, *Theoretical Computer Science*, 173:89–112, 1997.
[136] Kowalski R., Database Updates in the Event Calculus, *Journal of Logic Programming*, 12:121–146, 1992.
[137] Kowalski R., and F. Sadri, Reconciling Event Calculus with Situation Calculus, in [155], 39–58, 1997.
[138] Kowalski R., and M. Sergot, A Logic-based Calculus of Events, *New Generation Computing*, 4:67–95, 1986.
[139] Kuipers B., Qualitative Simulation, *Artificial Intelligence*, 29:289–388, 1986.
[140] Ladkin P., Time representation: a taxonomy of interval relations, Proc. of the 5th National Conference of the American Association for Artificial Intelligence (AAAI), AAAI Press, Philadelphia, PA, 360–366, 1986.
[141] Ladkin P., and R. Maddux, The algebra of constraint satisfaction problems and temporal reasoning, Technical Report, Kestrel Institute, Palo Alto, CA, 1988.
[142] Ladkin P., and Reinefeld, Effective Solution of Qualitative Interval Constraint Problems, *Artificial Intelligence*, 57:105–124, 1992.
[143] Lamma E., P. Mello, and M. Milano, A Distributed Constraint-Based Scheduler for Railway Traffic, *Artificial Intelligence in Engineering*, 11(2):91-105, 1997.
[144] Leban B., D. McDonald, and D. Foster, A representation for collections of temporal intervals, Proc. of the 5th National Conference of the American Association for Artificial Intelligence (AAAI), AAAI Press, Philadelphia, PA, 367–371, 1986.
[145] Lee H., J. Tannock, and J.S. Williams, Logic-based reasoning about actions and plans in artificial intelligence, *The Knowledge Engineering Review*, 8(2):91-120, 1993.
[146] Levesque H.J., What is planning in the presence of sensing?, Proc. of the 13th National Conference of the American Association for Artificial Intelligence (AAAI), Portland, OR, AAAI Press/MIT Press, 1996.
[147] Levesque H.J., R. Reiter, Y. Lespérance, F. Lin, and R. Scherl, GOLOG: a Logic Programming Language for Dynamic Domains, in [155], 59–83, 1997.

[148] Lifschitz V., A theory of actions, Proc. of the 10th International Joint Conference on Artificial Intelligence (IJCAI), Milano, Italy, Morgan Kaufmann, 1987.
[149] Lifschitz V., Pointwise Circumscription. In Ginsberg M. (ed.), *Readings in Nonmonotonic Reasoning*, Morgan Kaufmann, 1987.
[150] Lifschitz V., Formal theories of action, Proc. of the 1987 Workshop on The Frame Problem in Artificial Intelligence, Morgan Kaufmann, 1987.
[151] Lifschitz V., Restricted monotonicity, Proc. of the 13th International Joint Conference on Artificial Intelligence (IJCAI), Chambéry, France, Morgan Kaufmann, 432–437, 1993.
[152] Lifschitz V., Circumscription, in *Handbook of Logic in Artificial Intelligence and Logic Programming*, Volume 3, D. Gabbay, C. Hogger, and J. Robinsons (eds.), Oxford University Press, 1993.
[153] Lifschitz V., Nested abnormality theories, *Artificial Intelligence*, 74(2): 351-365, 1995.
[154] Lifschitz V., Guest Editor's Introduction: Reasoning about Action and Change, in [155], 1–2, 1997.
[155] Lifschitz V. (ed.), Special issue on Reasoning about Action and Change, *Journal of Logic Programming*, 31(1-3), 1997.
[156] Ligozat, G., On generalized interval calculi, Proc. of the 9th National Conference of the American Association for Artificial Intelligence (AAAI), AAAI Press/MIT Press, 234–240, 1991.
[157] Lin F., and Y. Shoham, Concurrent Actions in the Situation Calculus, Proc. of the 10th National Conference of the American Association for Artificial Intelligence (AAAI), AAAI Press/MIT Press, 590-695, 1992.
[158] Lin F., Specifying the Effects of Indeterminate Actions, Proc. of the 13th National Conference of the American Association for Artificial Intelligence (AAAI), Portland, OR, AAAI Press/MIT Press, 1996.
[159] Lin S., and T. Dean, Localized temporal reasoning using subgoals and abstract events, *Computational Intelligence* 12: 423-449, 1996.
[160] Mackworth A. K., Consistency in Networks of Relations, *Artificial Intelligence*, 8:99–118, 1977.
[161] McCarthy J., Circumscription - A Form of Non-Monotonic Reasoning, *Artificial Intelligence*, 13:27–29, 1980.
[162] McCarthy J., Applications of circumscription to formalizing common-sense knowledge, *Artificial Intelligence*, 28:89–116, 1986.
[163] McCarthy J., Situation Calculus with Concurrent Events and Narrative, http://www-formal.stanford.edu/gmc/narrative.html, 1995.
[164] McCarthy J., and P.J. Hayes, Some philosophical problems from the standpoint of Artificial Intelligence, *Machine Intelligence 4*, Meltzer B. and D. Michie (eds.), Edinburgh University Press, 463–502, 1969.
[165] McDermott D., A temporal logic for reasoning about processes and plans, *Cognitive Science*, 6:101–155, 1982.
[166] McIlraith S. A. Representing Actions and State Constraints in Model-Based Diagnosis, Proc. of the 14th National Conference of the American Association for Artificial Intelligence (AAAI), Providence, RI, AAAI Press/MIT Press, 43-39, 1997.
[167] Meiri I., Combining qualitative and quantitative constraints in temporal reasoning, *Artificial Intelligence*, 87(1-2):343–385, 1996.
[168] Miller R., and M. Shanahan, Narratives in the Situation Calculus, *Journal of Logic and Computation*, 4(5):513–530, 1994.
[169] Miller R., and M. Shanahan, The Event Calculus in Classical Logic – Alternative Axiomatisations, *Linköping Electronic Articles in Computer and Information Science* (http://www.ep.liu.se/ea/cis/1999/016/), 4(16), 1999.
[170] Milner R., *Communication and Concurrency*, Prentice Hall International Series in Com-

puter Science, 1989.
[171] Moffat D., and G. Ritchie, Modal queries about partially-ordered plans, *Journal of Experimental and Theoretical Artificial Intelligence*, 2:341–368, 1990.
[172] Mohr R., and T. Henderson, Arc and path consistency revisited, *Artificial Intelligence*, 28:225–233, 1986.
[173] Montanari A., and B. Pernici, Chapter 21: Temporal Reasoning, in [230], 534–562, 1993.
[174] Montanari A., *Metric and Layered Temporal Logic for Time Granularity*, ILLC Dissertation Series, 1996-02, Institute for Logic, Language and Computation, University of Amsterdam, 1996.
[175] Montanari A., A. Peron, and A. Policriti, Decidable Theories of ω-Layered Metric Temporal Structures, Proc. of ICTL'97: 2nd International Conference on Temporal Logic, H. Barringer, M.Fisher, D. Gabbay and G. Gough (Eds.), 75-89, 1997.
[176] Montanari A., A. Peron, and A. Policriti, Theories of ω-Layered Metric Temporal Structures: Expressiveness and Decidability, *The Logic Journal of IGPL*, 7(1), 1999.
[177] Montanari A., and A. Policriti, Decidability results for metric and layered temporal logics, *Notre Dame Journal of Formal Logic*, 37:260-282, 1996.
[178] Montanari A., and A. Policriti, Executing Metric Temporal Logic, Proc. of the IJCAI'97 Workshop on Programming in Temporal and Non Classical Logics, C. Brzoska and M. Fisher (Eds.), 1997.
[179] Montanari A., and M. de Rijke, Two-Sorted Metric Temporal Logic, *Theoretical Computer Science*, 183:187-214, 1997.
[180] Montanari U., Networks of Constraints: Fundamental Properties and Applications to Picture Processing, *Information Sciences*, 7:95–132, 1974.
[181] Mongenstern L., The Problem with Solutions to the Frame Problem, in *The Robot's Dilemma Revisited: The Frame Problem in AI*, K. Ford and Z. Pylyshyn (eds.), Ablex, Norwood, 1996.
[182] Morris P.H., The anomalous extension problem in default reasoning, *Artificial Intelligence*, 35:383–399, 1988.
[183] Nebel B., Solving Hard Qualitative Temporal Reasoning Problems: Evaluating the Efficiency of Using the ORD-Horn Class, Proc. of the 12th European Conference on Artificial Intelligence (ECAI), Budapest, Hungary, John Wiley & Sons, 38–42, 1996.
[184] Nebel B., and C Bäckström, On the computational complexity of temporal projection, planning, and plan validation, *Artificial Intelligence*, 66:125–160, 1994.
[185] Nebel B., and H.J. Bürckert, Reasoning About Temporal Relations: a Maximal Tractable Subclass of Allen's Interval Algebra, *Journal of the ACM* 42(1):43–66, 1995.
[186] Ngo L., P. Haddawy, Krieger R.A., and Helwig J., Efficient temporal probabilistic reasoning via context-sensitive model construction, in [55], 453–476, 1997.
[187] Niezette M., and J. Stevenne, An efficient symbolic representation of periodic time, Proc. of the 1st International Conference on Information and Knowledge Management (CIKM), 1992.
[188] Nökel K., *Temporally distributed symptoms in technical diagnosis*, Springer-Verlag, Berlin, 1991.
[189] Ohlbach H.J., Translation methods for non-classical logics: a survey, *The Logic Journal of the IGPL*, 1:69–89, 1991.
[190] Ozsoyoglu G, Snodgrass RT. Temporal and real-time databases: a Survey. *IEEE Transactions on Knowledge and Data Engineering*, 7(4):513–532, 1995.
[191] Pearl J., *Probabilistic Reasoning in Intelligent Systems (2nd Edition)*, Morgan Kaufmann, 1992.
[192] Pearl J., Belief networks revisited, *Artificial Intelligence*, 59:49–56, 1993.
[193] Pednault E., ADL: exploring the middle ground between STRIPS and the Situation Calculus, Proc. of 1st International Conference on Knowledge Representation and Reasoning

(KR), Morgan Kaufmann, 324–332, 1989.
[194] Pinto J., Temporal Reasoning in the Situation Calculus, PhD Thesis, University of Toronto, Department of Computer Science, 1994.
[195] Pinto J., Concurrent Actions and Interacting Effects, Proc. of the 6th International Conference on Knowledge Representation and Reasoning (KR), Trento, Italy, Morgan Kaufmann, 292–303, 1998.
[196] Pinto J., and R. Reiter, Temporal Reasoning in Logic Programming: A Case for the Situation Calculus, Proc. of the 10th International Conference on Logic Programming (ICPL), MIT Press, 203–221, 1993.
[197] Provetti A., Hypothetical reasoning about actions: From Situation Calculus to Event Calculus, *Computational Intelligence*, 12(3):478–498, 1996.
[198] Rayner M. On the applicability of nonmonotonic logic to formal reasoning in continuous time, *Artificial Intelligence*, 49:345–360, 1991.
[199] Reiter R., The frame problem in the Situation Calculus: a simple solution (sometimes) and completeness result for goal regression, in *Artificial Intelligence and Mathematical Theory of Computation: papers in honor of John McCarthy*, V. Lifschitz (ed.), 359–380, Academic Press, 1991.
[200] Reiter R., Natural Actions, Concurrency and Continuous Time in the Situation Calculus, Proc. of the 5th International Conference on Knowledge Representation and Reasoning (KR), 2–13, Morgan Kaufmann 1996.
[201] Reiter R., *KNOWLEDGE IN ACTION: Logical Foundations for Describing and Implementing Dynamical Systems*, Book Draft, Version of January 5, 1999.
[202] Sadri F., and R. Kowalski, Variants of the Event Calculus, Proc. of the 12th International Conference on Logic Programming (ICLP), Kanagawa, Japan, MIT Press, 67–81, 1995.
[203] Sandewall E., Combining logic and differential equations for describing real-world systems. Proc. of the 1st International Conference on Principles of Knowledge Representation and Reasoning (KR), Morgan Kaufmann, 412–420, 1989.
[204] Sandewall E., Filter Preferential Entailment for the Logic of Action in Almost Continuous Worlds. Proc. of 11th International Joint Conference on Artificial Intelligence (IJCAI), Detroit, Michigan, Morgan Kaufmann, 1989.
[205] Sandewall E., The range of applicability of nonmonotonic logics for the inertia problem. Proc. of 13th International Joint Conference on Artificial Intelligence (IJCAI), Chambéry, France, Morgan Kaufmann, 1993.
[206] Sandewall E., *Features and Fluents. A Systematic Approach to the Representation of Knowledge about Dynamical Systems*, Oxford University Press, 1994.
[207] Sandewall E. (ed.), *Electronic Newsletter on Reasoning about Actions and Change*, http://www.ida.liu.se/ext/etai/actions/njl/, 1998.
[208] Scherl R., H.J. Levesque, The Frame Problem and Knowledge Producing Actions, Proc. of the 11th National Conference of the American Association for Artificial Intelligence (AAAI), Washington, DC, AAAI Press/MIT Press, 689–695, 1993.
[209] Schubert L., Monotonic solution of the frame problem in the Situation Calculus: an efficient method for worlds with fully specified actions, in *Knowledge Representation and Defeasible Reasoning*, H. Kyberg, R. Loui, and G. Karlson (eds.), Kluwer Academic Press, 1990.
[210] Schwalb E., and R. Dechter, Processing Disjunctions of Temporal Constraints, in [48], 30–35, 1996.
[211] Schwalb E., and L. Vila, Temporal constraints: a survey, in [102], 129–150, 1998.
[212] Shahar Y., A framework for knowledge-based temporal abstraction, *Artificial Intelligence*, 90(1-2):79–133, 1997.
[213] Shahar Y., Dynamic temporal interpretation contexts for temporal abstraction, in [52], 159–192, 1998.
[214] Shanahan M., Prediction is deduction but explanation is abduction, Proc. of the 11th

International Joint Conference on Artificial Intelligence (IJCAI), Morgan Kaufmann, 1055–1060, 1989.

[215] Shanahan M., Representing continuous change in the Event Calculus, Proc. of the 9th European Conference on Artificial Intelligence (ECAI), Stockholm, Sweden, John Wiley & Sons, 1990.

[216] Shanahan M., A Circumscriptive Calculus of Events, *Artificial Intelligence*, 77:249–284, 1995.

[217] Shanahan M., *Solving the Frame Problem: a Mathematical Investigation of the Common Sense Law of Inertia*, MIT Press, 1997.

[218] Shanahan M., Robotics and the common sense informatic situation, Proc. of the 12th European Conference on Artificial Intelligence (ECAI), Budapest, Hungary, John Wiley & Sons, 684–688, 1996.

[219] Shanahan M., Noise and the common sense informatic situation for a mobile robot, Proc. of the 13th National Conference of the American Association for Artificial Intelligence (AAAI), Portland, OR, AAAI Press/MIT Press, 1098-1103, 1996.

[220] Shapiro S., Y. Lespérance, and H.J. Levesque, Goals and Rational Action in the Situation Calculus - A Preliminary Report, Working Notes of the AAAI Fall Symposium on Rational Agency: Concepts, Theories, Models and Applications, Cambridge, MA, 1995.

[221] Shoham Y., Reified temporal logics: semantical and ontological considerations, Proc. of the 7th European Conference on Artificial Intelligence (ECAI), Brighton, UK, 390–397, 1986.

[222] Shoham Y., Chronological Ignorance: Experiments in Nonmonotonic Temporal Reasoning, *Artificial Intelligence*, 36:279–331, 1988.

[223] Shoham Y., *Reasoning about change: time and causation from the standpoint of Artificial Intelligence*, MIT Press, 1988.

[224] Shoham Y., Nonmonotonic reasoning and causation, *Cognitive Science*, 14:213–252, 1990.

[225] Shoham, Y., and D. McDermott, Problems in Formal Temporal Reasoning, *Artificial Intelligence*, 36:49–61, 1988.

[226] Snodgrass R. (ed), *The TSQL2 temporal query language*, Kluwer Academic Publishers, Boston, MA, 1995.

[227] Song F., and R. Cohen, The interpretation of temporal relations in narrative, Proc. of the 7th National Conference of the American Association for Artificial Intelligence (AAAI), Saint Paul, MI, AAAI Press, 745-750, 1988.

[228] Stein L. A., and L. Morgenstern, Motivated Action Theory, *Artificial Intelligence*, 71:1–42, 1994.

[229] Stillman J., R. Arthur, and A Deitsch, Tachyon: A constraint-based temporal reasoning model and its implementation, *SIGART Bulletin*, 4(3):T1–T4, 1993.

[230] Tansel A., J. Clifford, S. Gadia, S. Jajodia, A. Segev, and R. Snodgrass, *Temporal Databases: Theory, Design and Implementation*, Tansel et al. (eds.), Benjamin/Cummings Publishers, 1993.

[231] Tarski A., On the calculus of relations, *Journal of Symbolic Logic*, 6:73–89, 1941.

[232] Tawfik A. Y, and E. M. Neufeld, Irrelevance in Uncertain Temporal Reasoning, in [48], 182–187, 1996.

[233] Ternovskaia E., Interval Situation Calculus, Proc. of ECAI-94 Workshop on Logic and Change, 153–164, Amsterdam, NL, 1994.

[234] Thielscher M., Computing ramifications by postprocessing, Proc. of 14th International Joint Conference on Artificial Intelligence (IJCAI), Montréal, Québec, Morgan Kaufmann, 1994–2000, 1995.

[235] Thielscher M., On the completeness of SLDENF-resolution, *Journal of Automated Reasoning*, 17(2):199-214, 1996.

[236] Thielscher M., Causality and the qualification problem, Proc. of the 5th International Conference on Principles of Knowledge Representation and Reasoning (KR), Morgan Kaufmann,

1996.

[237] Thielscher M., Ramification and causality, *Artificial Intelligence*, 89(1-2):317–364, 1997.

[238] Thielscher M., Introduction to the Fluent Calculus, *Linköping Electronic Articles in Computer and Information Science*, Vol. 3(1998): nr 14, http://www.ep.liu.se/ea/cis/1998/014, 1998.

[239] Tsang, E. P. K. Elements in Temporal Reasoning in Planning. Proc. of the 8th European Conference on Artificial Intelligence (ECAI), München, Pitman Publishing, 1988.

[240] R. E. Valdes-Perez, The Satisfiability of Temporal Constraint Networks, Proc. of the 6th National Conference of the American Association for Artificial Intelligence (AAAI), Seattle, Wash., AAAI Press, 256-260, 1987.

[241] Vidal T., and M. Ghallab, Dealing with Uncertain Durations in Temporal Constraint Networks dedicated to Planning, Proc. of the 12th European Conference of Artificial Intelligence (ECAI), Budapest, Hungary, J. Wiley & Sons, 48–52, 1996.

[242] Vila L., and H. Reichgelt, The token reification approach to temporal reasoning, *Artificial Intelligence*, 83:59–74, 1996.

[243] Vilain M.B., A system for reasoning about time, Proc. of the 2nd National Conference of the American Association for Artificial Intelligence (AAAI), AAAI Press, Pittsburgh, PA, 197–201, 1982.

[244] Vilain M.B., and H. Kautz, Constraint propagation algorithms for temporal reasoning, Proc. of the 5th National Conference of the American Association for Artificial Intelligence (AAAI), Philadelphia, PA, AAAI Press, 377-382, 1986.

[245] Vilain M.B., H. Kautz, and P. van Beek, Constraint propagation algorithms for temporal reasoning: a revised report. In *Readings in Qualitative about Physical Systems*, D. S. Weld and J. de Kleer (eds.), Morgan Kaufmann, San Mateo, CA, 373–381, 1989.

[246] Webber A. B., Proof of the interval satisfiability conjecture, *Annals of Mathematics and Artificial Intelligence*, 15:231–238, 1995.

[247] Woods W. A., What's important about knowledge representation?, *IEEE Computer*, 22–29, 1983.

[248] Yi C., Reasoning about concurrent actions with features and fluents, in [48], 6–13, 1996.