Chapter 2
Temporal Modeling and Temporal Reasoning

Overview

In this chapter, the reader is guided through the basic notions of time and temporal information and is presented with some important, general approaches to represent and reason about temporal information. Simple medical examples are used to help the reader to understand the advantages and limitations of the different approaches.

Structure of the Chapter

After a brief introduction, the basic concepts related to the representation of temporal information are presented. The main features of time domains, time primitives, and temporal entities are introduced in Section 2.2. Section 2.3 continues the discussion by presenting some widely acknowledged, general approaches to temporal reasoning and outlining the requirements for temporal reasoning in medicine, pointing out the limitations of some of the considered general approaches. Section 2.4 introduces temporal constraints, a relevant topic both for temporal reasoning and for temporal knowledge-bases and databases.

Keywords

Time domains, instants, intervals, time metrics, linear time, branching time, circular time, granularity, indeterminacy, interval algebra, event calculus, situation calculus, temporal reasoning, temporal representation, temporal entities, temporal constraints, probabilistic reasoning, deterministic reasoning.

C. Combi et al., *Temporal Information Systems in Medicine*,

DOI 10.1007/978-1-4419-6543-1_2,

2.1 Introduction

A common focus of temporal reasoning, temporal abstraction of clinical data, and modeling and managing clinical data, is the definition or the adoption of a set of basic concepts that enable a description of a time-oriented clinical world in a sound and unambiguous way. Several suggestions have emerged from generic fields of computer science, such as artificial intelligence, or the knowledge and data management areas [10, 8, 381, 399, 387, 289]. Within medical informatics, this effort has progressed from an ad-hoc definition of concepts supporting a particular application to the adoption and the proposal of more generic definitions, supporting different clinical applications [140, 212, 197, 196, 235, 111, 86, 359, 214]. For example, the emphasis in the pioneering work of Fagan on the interpretation of real-time quantitative data in the intensive-care domain is on the application-dependent problems, related to the support of a module that suggests the optimal ventilator therapy at a given time [140], while the work described in [214] uses a generic temporal ontology and a general, comprehensive, model of diagnostic reasoning.

2.2 Modeling Temporal Concepts

Time-related representation requirements for medical applications are many and varied because time manifests in different ways in expressions of medical knowledge and patient information. There are two issues here: how to model time per se and how to model time-varying situations or occurrences. In other words, we have here to consider both how to model the concept of time and how to model entities having a temporal dimension [270, 289].

2.2.1 Modeling Time

In general we could say that modeling time as a dense or discrete number line may not provide the appropriate abstraction for medical applications. The modeling of time for the management of, or the reasoning about time-oriented clinical data, requires several basic choices to be made, depending on the needs of the domain.

2.2.1.1 Time Domain

The time domain consists of the set of primitive time entities used to represent the concept of time. It allows one to define and interpret all the other time-related concepts. According to [184], a *time domain* is a pair $(T;\leq)$, where T is a non-empty set of time instants and $\leq$ is a total order on T. It has been extensively debated whether the real time is both either bounded or unbounded and either dense (e.g., isomorphic to real or rational numbers) or discrete (e.g., isomorphic to natural numbers). A time domain is bounded if it contains upper and/or lower bounds with respect to its order relationship. A time domain is dense if it is an infinite set and for all $t', t'' \in T$ with $t' < t''$, there exists $t''' \in T$ such that $t' < t''' < t''$. On the opposite, a time (unbounded) domain is discrete if every element has both an immediate successor and an immediate predecessor. For example, a widely used approach in temporal databases is the use of a basic *timeline*, i.e. a point structure with precedence relation, which is a total order without right and left endpoints. The basic timeline is (partially) partitioned in non decomposable consecutive time intervals, called *chronons* [387, 78]. On the other hand, Allen's intervals are defined on a continuous timeline [10].

2.2.1.2 Instants and Intervals

Usually both the (primitive) concepts of *time point* (or *instant*) and *time interval* have been used to represent time [235, 359, 86, 214, 387]. These concepts are usually related to instantaneous events (e.g. myocardial infarction), or to situations lasting for a span of time (e.g. drug therapy). In defining basic time entities, time points (i.e., instants) are often adopted. Intervals are then represented by their upper and lower temporal bounds (start and end time points). In practice, most systems employed in medical informatics applications have used a time point based approach, similar to McDermott's *points* [259], rather than use time intervals as the basic time primitives, as proposed by Allen [8]. Several variations exist. *Nonconvex intervals* are intervals formed from a union of convex intervals, and might contain gaps (see Figure 2.1). Such intervals are first-class objects that seem natural for representing processes or tasks that occur repeatedly over time. Ladkin defined a taxonomy of relations between nonconvex intervals [232] and a set of operators over such intervals [231], as well as a set of standard and extended time units that can exploit the nonconvex representation in an elegant manner to denote intervals such as "Mondays." [231]. Additional work on models and languages for nonconvex intervals has been done by Morris and Al Khatib [273], who call such intervals *N-intervals*. In the temporal database community non-convex intervals are usually named *temporal elements* [184]. It is interesting to point out that in the database community, complex time structures are sometimes introduced to model in a compact way all the time dimensions related to a fact. For example, in [184] a temporal element is defined as a finite union of n-dimensional time intervals, assuming that a model is able to represent n different (and orthogonal) time dimensions (the main temporal dimensions for temporal databases are introduced and discussed in the next section).

2.2.1.3 Linear, Branching, and Circular Times

Different properties can be associated with a time axis composed by instants. Usually, both in general and clinically-oriented databases, time is *linear*, since the set

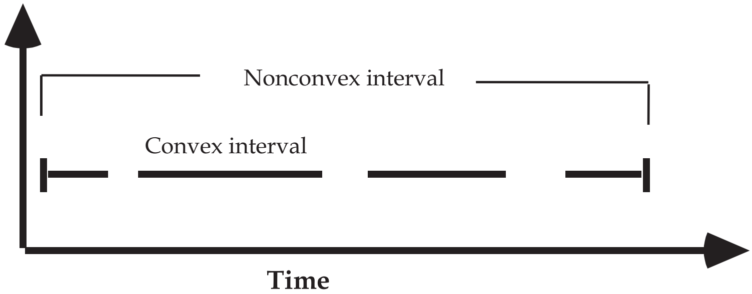


Fig. 2.1 A nonconvex interval. The nonconvex interval comprises several convex intervals.

of time points is completely ordered [111, 86, 214]. However, for the tasks of diagnosis, projection, or forecasting (such as prediction of a clinical evolution over time), a *branching* time might be necessary. In this case, only a partial ordering is defined for times. Such a representation has been found to be useful for pharmacoeconomics, and has been implemented using an object-oriented temporal model, as demonstrated in [158]. *Circular* (or periodic) time is needed when we have to describe times related to recurrent events, such as “administration of regular insulin every morning”. In this case no ordering relations are defined for times.

2.2.1.4 Relative and Absolute Times

The position on the time axis of an interval or of an instant can be given as an absolute position, such as the calendric time, when mapped to the time axis used (e.g.: “on November, 3 1996”) [426, 197, 105, 85, 111, 359]. This is a common approach adopted by data models underlying temporal databases. However, it is also common to reason with relative time references: “the day after” or “sometimes before that moment”.

Relevant to the topic of relative times are several proposals that employ implicit [212, 214, 170] or explicit [348] temporal contexts, which support the representation of relative or context-sensitive temporal clinical information or knowledge. Another concept relevant to this topic is that of *time metrics*: absolute times are generally associated to a metric, being its position given as a *distance* from a given time origin. When a metric is defined for the time domain, relative times can be given quantitatively: “three days after birth”.

2.2.1.5 Modeling Temporal Relationships

In modeling temporal relationships, Allen's interval algebra [8] has been widely used in medical informatics [235, 111, 348, 347]. Section 2.3.2.4 describes in some detail Allen's interval algebra. Extensions to Allen's basic thirteen interval relationships have also been proposed [86]. Temporal relationships include two main types: qualitative (interval I1 before interval I2) and quantitative (interval I1 two hours before interval I2). Several general formalisms and approaches [8, 119, 253, 270] have been effectively adopted for satisfying the various needs encountered while modeling temporal relationships. Moreover, temporal relationships can also be classified according to the entities involved: interval/interval, interval/point, point/interval and point/point [270].

2.2.1.6 Modeling Granularities

The *granularity* of a given temporal information is the level of abstraction at which information is expressed. Different units of measure allow one to represent different granularities. One of the first proposals formally dealing with time granularity was made by Clifford [78], in which he provides a particularly clean view of a structure for temporal domains. Using a set-theoretic construction, Clifford defines a simple but powerful structure of time units. He assumes for every temporal domain a *chronon*. By the repeated operation of *constructed intervallic partitioning* - intuitively equivalent to segmentation of the time line into mutually exclusive and exhaustive intervals (say, constructing 12 months from 365 days) - Clifford defines a temporal universe, which is a hierarchy of time levels and units. He also defines clearly the semantics of the operations possible on time domains in the *temporal universe*. It is interesting to note that, unlike Ladkin's construction of discrete time units [231], Clifford's construction does not leave room for the concept of weeks as a time unit, since *weeks* can overlap months and years, violating the constructed intervallic partition properties. A widely accepted definition of temporal granularity, proposed by Bettini et al [28], has been used both for knowledge representation and for temporal data modelling [28, 184, 240, 282, 241]. According to their framework, a granularity is a mapping G from an index set (e.g., integers) to the powerset of the time domain such that:

1. if $i < j$ and $G(i)$ and $G(j)$ are non-empty, then all elements of $G(i)$ are less than all elements of $G(j)$, and
2. if $i < k < j$ and $G(i)$ and $G(j)$ are non-empty, then $G(k)$ is non-empty.

Any $G(i)$ is called a *granule*. The first condition states that granules in a granularity do not overlap and that their index order is the same as their time domain order. The second condition states that the subset of the index set that maps to non-empty subsets of the time domain is contiguous. It is worth noting that the set of granules is always discrete, no matter whether the time domain is discrete or dense. Besides an index, there may be a textual representation of a granule, as in the case, for example, of "June 1999", which refers to the granule composed by the time points contained in that month. Several relationships have been introduced for granularities [27, 160]; for example, according to Bettini et al [27] a granularity G_1 is finer than another granularity G_2 if for each i, there exists j such that $G_1(i) \subseteq G_2(j)$.

2.2.1.7 Modeling indeterminacy

Indeterminacy is often present in temporal information and is related to incomplete knowledge of when the considered fact happened. A frequent need, especially in clinical domains, is the explicit expression of uncertainty regarding how long a proposition was true. In particular, we might not know precisely when the proposition became true and when it ceased to be true, although we might know that it was true during a particular time interval. Sometimes, the problem arises because the time units involved have different *granularities*: the Hb level may sometimes be dated with an accuracy level of hours (e.g., "Wednesday at 5 P.M., October 23, 2002"), but may sometimes be given for only a certain day ("Wednesday, October 23, 2002"). Sometimes, the problem arises due to the naturally occurring incomplete information in clinical settings: The patient complains of a backache starting "sometime during 2001". There is often a need to represent such vagueness. As an example, Console and Torasso [102, 103, 104] present a model of time intervals that represents such partial knowledge explicitly. The model was proposed in order to represent causal models for diagnostic reasoning. The authors define a *variable interval* as a time interval I composed of three consecutive convex intervals. The first interval is $begin(I)$, the second is called $body(I)$, and the third is called $end(I)$. Operations on convex intervals can be extended to variable intervals. We can now model uncertainty about the time of the start or end of the actual interval, when these times are defined vaguely, since the begin and end intervals of a variable interval represent uncertainty about the start and stop times of the real interval; the body is the only interval during which the proposition represented by the variable interval was certainly true.

2.2.2 *Modeling Temporal Entities*

Let us now consider the more abstract task of modeling temporal entities, i.e., those concepts/things of the real world which must be represented also for their temporal aspects. A rich model providing a number of interrelated basic temporal entities, given at different abstraction levels and with multiple granularities, is often required when dealing with medical temporal information. Many representation issues arise with respect to temporal entities, as detailed below.

2.2.2.1 Defining Temporal Entities

A question that has been investigated in some depth in the literature is: What are the basic (medical) concepts that have temporal dimension? How are they interrelated? In general, we distinguish two different approaches in modeling temporal entities in medical applications: addition of a temporal dimension to existing objects, or creation of model-specific, time-oriented entities. The first approach, originating from research into databases, uses simple, “atomic” temporal entities [111, 86]. This approach is similar to the one underlying the temporal extensions proposed for relational and object-oriented data models: a temporal dimension is added at the tuple/object level or at the attribute/method level [111, 83], as we will see in detail in Chapter 3. The second approach, originating mostly from the area of artificial intelligence, focuses on modeling different temporal features of complex, task-specific entities.

For example, Allen introduces *events*, *properties* and *processes*, to represent different kinds of proposition holding on some intervals (as discussed in Section 2.3.2.4), while McDermott distinguishes between *facts* and *events*, as discussed in Section 2.3.2.3.

Let us now consider some proposals coming from medical applications. Here, several types of compound (abstract) entities are introduced, based on temporal entities that are stored at the database level. For example, in the HyperLipid system [335], patient visits were modeled as instant-based objects called *events*, while administration of drugs was modeled as *therapy* objects whose attributes included a time interval. *Phases* of therapy (inspired by the clinical algorithm modeled by the system) were then introduced to model groups of heterogeneous data, related to both visits and therapies. Events, therapies and phases were connected through a network.

Kahn and colleagues in [197] introduced formally the concept of a Temporal Network (TNET) and later extended it by the Extended TNET, or ETNET model [196]. In both models, a T-node (or an ET-node) models task-specific temporal data, such as a chemotherapy cycle, at different levels of abstraction. Each T-node is associated with a time interval during which the information represented by the T-node’s data is true for a given patient.

In the M-HTP system for monitoring heart-transplant patients [235], clinical facts related to a patient are structured in a temporal network (TN) inspired by Kahn’s TNET model [196]. Through this network, a physician can obtain different temporal views of the patient’s clinical history. Each node of the TN represents an *event* (a visit) or a *significant episode* in the patient’s clinical record. An event is time-point based; its temporal location can be specified by an absolute date or by the temporal distance relative to the transplantation event. An episode holds during an interval, during which a predefined property (evaluated by reasoning about several events) holds.

Keravnou and Washbrook introduce *findings*, *features*, and events to distinguish various types of instantaneous and interval-based information (patient-specific or general) [212].

2.2.2.2 Associating Entities to Instants and Intervals

We observe two main approaches in defining *occurrences* of temporal entities, i.e., in associating time with temporal entities. The first approach deals both with instant-related entities and with interval-related entities [235]. The second approach associates clinical entities only with a certain type of time concept, usually an interval, dealing in a homogeneous way also with intervals degenerating to be a single instant [359, 86, 111]. A further distinction exists between the basic time primitives, usually instants (time points), and the time entities that can be associated with clinical concepts [359, 86, 111].

Shoham's approach, for example, is based on the adoption of a set of time points as primitives; predicates, however, such as values of clinical parameters, can only be interpreted over *time intervals*, which are defined as ordered pairs of time stamps (including instants, which are zero-length intervals).

Depending on the underlying properties for time, the actual occurrences of temporal entities can be specified in several different ways:

- *Absolute and relative temporal occurrences*: the existence of some occurrence can be expressed in absolute terms, relative to some fixed time point, by specifying its initiation and termination (e.g.: "Tachycardia on November 3, 1996 from 6:30 to 6:45 p.m."). This is a common approach adopted by data models underlying temporal databases. Similarly, it can be expressed relative to other occurrences, either by qualitative relationships ("angina after a long walk" or "several episodes of headache during puberty") or by quantitative relationships (angina two hours before headache). Incorporation of purely relative time-oriented, interval-based information (especially disjunctions, such as "the patient had vomited before or during the diarrhea episode") within a standard temporal database is still a difficult task.
- *Absolute and relative vagueness, duration, and incompleteness*: an occurrence is associated with absolute vagueness if its initiation and/or termination cannot be precisely specified in a given temporal context; precision is relative to the particular temporal context. Absolute vagueness (called also *indeterminacy*) may be expressed in terms of quantitative constraints on the initiation, termination, or extent of the occurrence, e.g. the earliest possible and latest possible time for its initiation or termination, or the minimum and maximum for its duration: "an atrial fibrillation episode occurred on December 14th, 1995 between 14:30 and 14:45 and lasted for three-four minutes". An occurrence is associated with relative vagueness if its temporal relation with other occurrences is not precisely known but can only be expressed as a disjunction of primitive relations. Incompleteness in the specification of occurrences is thus a common phenomenon.
- *Point and interval occurrences*: An occurrence may be considered a point occurrence in some temporal context if its duration is less than the time unit, if any, associated with the particular temporal context. A point occurrence may be treated as an instantaneous and hence as a non-decomposable occurrence in the given temporal context. Thus an occurrence may be considered an interval occurrence

in some temporal context if its duration is at least equal to the time unit associated with the particular temporal context. Care needs to be taken in associating these concepts to clinical entities, such as symptoms, therapies, and pathologies: a myocardial infarction, for example, could be considered an instantaneous event, within the overall clinical history of the patient, or an interval-based occurrence, if observed during an ICU staying.

2.2.2.3 Semantic Relations between Temporal Entities

Other, more complex, features of temporal entities and of their occurrences need to be suitably considered.

- *Compound occurrences*: repeated instantiations of some type of occurrence, usually, but not necessarily, in a regular fashion, may need to be collectively represented as a periodic occurrence. An abstract periodic occurrence consists of the basic temporal entity and of the "algorithm" governing the repetition. A specific periodic occurrence is the collation of the relevant, individual, occurrences. A temporal trend, or simply trend, is an important kind of interval occurrence. A trend describes a change, the direction of change, and the rate of change that takes place in the given interval of time. An example of a trend could be "increasing blood pressure". A trend is usually derived from a collection of occurrences at a lower level. A temporal pattern, or simply pattern, is a compound occurrence, consisting of a number of simpler occurrences (and their relations). There are different kinds of patterns. A sequence of meeting trends is a commonly used kind of pattern. A periodic occurrence is another example of pattern. A set of relative occurrences, or a set of causally related occurrences, could form patterns. A compound occurrence can in fact be expressed at multiple levels of abstraction. Abstraction and refinement are therefore important structural relations between occurrences. Through refinement an occurrence can be decomposed into component occurrences and through abstraction component occurrences can be contained into a compound occurrence.
- *Contexts, causality and other temporal constraints*: a context represents a state of affairs that, when interpreted (logically) over a time interval, can change the meaning of one or more facts which hold within the context time interval. Causality is a central relation between occurrences. Changes are explained through causal relations. Time is intrinsically related to causality. The temporal principle underlying causality is that an effect cannot precede its cause. Causally unrelated occurrences can also be temporally constrained, as already mentioned. For example, a periodic occurrence could be governed by the constraint that the distance between successive occurrences should be 4 hours.

2.3 Temporal Reasoning

The ability to reason about time and temporal relations is fundamental to almost any intelligent entity that needs to make decisions. The real world includes not only static descriptions, but also dynamic processes. It is difficult to represent the concept of taking an action, let alone a series of actions, and the concept of the consequences of taking a series of actions, without explicitly or implicitly introducing the notion of *time*. This inherent requirement also applies to computer programs that attempt to reason about the world. In the area of natural-language processing, it is impossible to understand stories without the concept of time and its various nuances (e.g., "by the time you get home, I would be gone for 3 hours"). Planning actions for robots requires reasoning about the *temporal order* of the actions and about the *length of time* it will take to perform the actions. Determining the cause of a certain state of affairs implies considering temporal precedence, or, at least, temporal equivalence. Scheduling tasks in a production line, that aim to minimize total production time, require reasoning about *serial* and *concurrent* actions and about time *intervals*. Describing typical patterns in a baby's psychomotor development requires using notions of absolute and relative time, such as "walking typically *starts* when the baby is about *12 months old*, and is *preceded by* standing." Thus, clinical domains pose no exception to the fundamental necessity of reasoning about time.

Temporal reasoning has been used in medical domains as part of a wide variety of generic tasks [60], such as diagnosis (or, in general, abstraction and interpretation), monitoring, projection, forecasting, and planning (as discussed in chapters 6 and 7). These tasks are often interdependent. Projection is the task of computing the likely consequences of a set of conditions or actions, usually given as a set of cause-effect relations. *Projection* is particularly relevant to the *planning task* (e.g., when we need to decide how the patient's state will be after we administer to the patient a certain drug with known side effects). *Forecasting* involves predicting particular future values for various parameters given a vector of time-stamped past and present measured values, such as anticipating changes in future hemoglobin-level values, given the values up to and including the present. *Planning* consists of producing a sequence of actions for a care provider, given an initial state of the patient and a goal state, or set of states, such that that sequence achieves one of the goal patient states. Possible actions are usually operators with predefined certain or probabilistic effects on the environment. The actions might require a set of enabling *preconditions* to be possible or effective. Achieving the goal state, as well as achieving some of the preconditions, might depend on the correct *projection* of the actions up to a point, to determine whether preconditions hold when required. *Interpretation* involves abstraction of a set of time-oriented patient data, either to an intermediate level of meaningful temporal patterns, as is common in the *temporal-abstraction* task or in the *monitoring* task, or to the level of a definite diagnosis or set of diagnoses that explain a set of findings and symptoms, as is common in the *diagnosis* task. Interpretation, unlike forecasting and projection, involves reasoning about only past and present data and not about the future.

From the methodological point of view, one general criterion that can be used when classifying temporal-reasoning research is whether it uses a deterministic or a probabilistic approach [206].

2.3.1 Temporal Reasoning Requirements

Before introducing some relevant generic models for temporal reasoning, let us consider some important (generic) functionalities, a medical temporal reasoning system should include:

- *Mapping the existence of occurrences across temporal contexts*, if multiple temporal contexts are supported and more than one such context is meaningful to some occurrence.
- *Determining bounds for entity occurrences*. The initiation and termination points of absolute existences are usually expressed in (qualitative) terms which need to be translated into upper and lower bounds for the actual points within the relevant temporal context.
- *Consistency detection and clipping of uncertainty*. If the inferences drawn from a collection of occurrences are to be valid the occurrences must be mutually consistent. Inconsistency arises when there are overlapping occurrences that assert mutually exclusive propositions. The inconsistency can be resolved if the boundaries of the implicated occurrences can be moved so that the overlapping is eliminated. In fact the identification of such clashes usually results in narrowing the bounds for the initiation/termination of the relevant occurrences. More generally, inconsistency arises when the (disjunctive) temporal constraints relating a given set of occurrences cannot be mutually satisfied. A conflict is detected when all the possible temporal relationships between a pair of temporal entities are refuted. Temporal constraint propagation, minimization of disjunctive constraints (i.e. reducing the uncertainty), detection and resolution of conflicts are necessary functionalities, as in many other non-medical applications.
- *Deriving new occurrences from other occurrences*. There are different types of derivation. A predominant type is *temporal-data abstraction*, which is described separately in Chapter 5. Other types include decomposition derivations (the potential components of compound occurrences are inferred), causal derivations (potential antecedent occurrences, consequent occurrences, or causal links between occurrences are derived), etc.
- *Deriving temporal relations between occurrences*. Often the temporal relations that hold between occurrences are significant for the given problem solving. Thus if the temporal relation between a pair of occurrences is not explicitly given, it would need to be inferred.
- *Deriving the truth status of queried occurrences*. This functionality brings together many of the other functionalities. A (hypothesized) occurrence, of any degree of complexity, e.g. periodic, trend, compound, etc, is queried against a set

of occurrences (and temporal contexts) that are assumed to be true. The queried occurrence is derived as true (it can be logically deduced from the assumed occurrences), false (it is counter-indicated by the assumed occurrences), or unknown (possibly true or possibly false).

- *Deriving the state of the world at a particular time*. The previous functionality starts with a specific set of assumed occurrences and a specific queried occurrence. It is considered a necessary functionality because often problem solvers seek to establish specific information. Alternatively though, in an investigative/explorative mode, the problem solver may need to be informed about what is considered to be true at some specific time. The query may be expressed relative to another specific point in time which defaults to now, e.g. at time point t, what was/is/will be believed to be true during some specified period p? This functionality may be used to compose the set of assumed occurrences for queries of the previous type.

2.3.2 Ontologies and Models for Temporal Reasoning

In this section, we present briefly major approaches to temporal reasoning in philosophy and in computer science (in particular, in the AI area). We have organized these approaches roughly chronologically.

2.3.2.1 Tense Logics

It is useful to look at the basis for some of the early work in temporal reasoning. We know that Aristotle was interested in the meaning of the truth value of future propositions [330]. The stoic logician Diodorus Chronus, who lived circa 300 B.C., extended Aristotle's inquiries by constructing what is known as *the master argument*. It can be reconstructed in modern terms as follows [330]:

1. Everything that is past and true is necessary (i.e., what is past and true is necessarily true thereafter).
2. The impossible does not follow the possible (i.e., what was once possible does not become impossible).

From these two assumptions, Diodorus concluded that nothing is possible that neither is true nor will be true, and that therefore every (present) possibility must be realized at a present or future time. The master argument leads to *logical determinism*, the central tenet of which is that what is necessary at any time must be necessary at all earlier times. This conclusion fits well indeed within the stoic paradigm.

The representation of the master argument in temporal terms inspired modern work in temporal reasoning. In particular, in a landmark paper [321] and in subsequent work [322, 323], Prior attempted to reconstruct the master argument using a modern approach. This attempt led to what is known as *tense logic* - a logic of past and future. In Prior's terms,

Fp: it will be the case that p.

Pp: it was the case that p.

Gp: it will always be the case that p (i.e., $\neg F \neg p$).

Hp: it was always the case that p (i.e., $\neg P \neg p$).

Prior's tense logic is thus in essence a *modal-logic* approach (an extension of the first-order logic (FOL) with special operators on logical formulae [136]) to reasoning about time. This modal-logic approach has been called a *tenser* approach [149], as opposed to a *detenser*, or an FOL, approach. As an example, in the *tenser* view, the sentence $F(\exists x)f(x)$ is *not* equivalent to the sentence $(\exists x)Ff(x)$; in other words, if in the future there will be some x that will have a property f, it does not follow that there is such an x now that will have that property in the future. In the *detenser* view, this distinction does not make sense, since both expressions are equivalent when translated into FOL formulae. This difference occurs because, in FOL, objects exist timelessly, time being just another dimension; in tenser approaches, NOW is a point of time in a separate class. However, FOL can serve as a *model theory* for the modal approach [149]. Thus, we can assign precise meanings to sentences such as Fp by a FOL formalism.

An interesting point in the use of time and tenses in natural language was brought out by Anscombe's investigation into the meanings of *before* and *after* [14]. An example is the following: from "The infection was present *after* the fever ended," it does not follow that the fever ended *before* the infection was present. Thus, *before* and *after* are not strict converses. Note that, however, from "The infection started *after* the fever started," we can indeed conclude that the fever started *before* the infection started. Thus, *before* and *after* are converses when they link instantaneous *events*.

2.3.2.2 Kahn and Gorry's Time Specialist

Kahn and Gorry [195] built a general temporal-utilities system, the *time specialist*, which was intended not for temporal *reasoning*, but rather for temporal *maintenance* of relations between time-stamped propositions. However, the various methods they used to represent relations between temporal entities are instructive, and the approach is useful for understanding some of the work in medical domains. The time specialist is a domain-independent module that is knowledgeable specifically about maintaining temporal relations. This module isolates the temporal-reasoning element of a computer system in any domain, but is not a temporal *logic*. Its specialty lies in organizing time-stamped bits of knowledge. A novel aspect of Kahn and Gorry's approach was the use of three different organization schemes; the decision of which one to use was controlled by the user:

1. Organizing by *dates* on a date line (e.g., "January 17 1972")

2. Organizing by special *reference events*, such as *birth* and *now* (e.g., “2 years after birth”)
3. Organizing by *before* and *after* chains, for an event sequence (e.g., “the fever appeared after the rash”).

By using a *fetcher* module, the time specialist was able to answer questions about the data that it maintained. The time specialist also maintained the *consistency* of the database as data were entered, asking the user for additional input if it detected an inconsistency. Kahn and Gorry made no claims about *understanding* temporally oriented sentences; the input was translated by the user to a Lisp expression. Neither did they claim any particular semantic classification of the type of propositions maintained by the time specialist. Rather, the time specialist presents an example of an early attempt to extract the time element from natural-language propositions, and to deal with that time element using a special, task-specific module.

2.3.2.3 Approaches Based on States, Events, and Changes

Some of the approaches taken in AI and general computer science involve a roundabout way of representing time: Time is represented implicitly by the fact that there was some change in the world (i.e., a transition from one state to another), or that there was some mediator of that change.

The Situation Calculus and Hayes’ Histories

The *situation calculus* was proposed by McCarthy [257, 258] to describe *actions* and their effects on the world. The idea is that the world is a set of *states*, or *situations*. Actions and events are functions that map states to states. Thus, that the result of performing the CARE_PROVIDING action in a situation with a suffering patient is a situation where the patient is treated is represented as

$$\forall s True(s, SUFFERING\_PATIENT) \Longrightarrow \\ True(Result(CARE\_PROVIDING, s), TREATED\_PATIENT)$$

Although the situation calculus has been used explicitly or implicitly for many tasks, especially in planning, it is not adequate for many reasons. For instance, concurrent actions are impossible to describe, as are actions with duration (note that CARE_PROVIDING brings about an immediate result) or continuous processes. There are also other problems that are more general, and are not specific to the situation calculus [377].

Hayes, aware of these limitations, introduced the notion of histories in his “Second Naive Physics Manifesto” [174]. A *history* is an ontological entity that incorporates both space and time. An object in a situation, or O@S, is that situation’s intersection with that object’s history [174]. Permanent places are unbounded temporally but restricted spatially. Situations are unbounded spatially and are bounded in time by the events surrounding them. Most objects are in between these two extremes. Events are instantaneous; episodes usually have duration. Thus, we can describe the history of an object over time. Forbus [143] has extended the notion of histories within his qualitative process theory.

Kowalski and Sergot's Event Calculus

Kowalski and Sergot proposed in [228] the Event Calculus (EC), a theory of time and change. EC is an interesting framework, because it is general, well founded and deeply formally studied, and it has also been applied to temporal reasoning in medical domains [212, 235, 206].

From a description of events that occur in the real world and properties they initiate or terminate, EC derives the validity intervals over which properties hold. The notions of event, property, time point, and time interval are the primitives of the formalism: *events* happen at *time points* and initiate and/or terminate *time intervals* over which *properties* hold. Initiated properties are assumed to persist until the occurrence of an event that interrupts them (*default persistence*). An event occurrence, associating the event to the time point at which it occurred, is represented by the $happens(event, timePoint)$ clause. The relations between events and properties are defined by means of *initiates* and *terminates* clauses, such as:

$$initiates(ev1, prop, t) \Longleftarrow happens(ev1, t) \wedge holds(prop1, t) \wedge ... \wedge holds(propN, t)$$

$$terminates(ev2, prop, t) \Longleftarrow happens(ev2, t) \wedge holds(prop1, t) \wedge ... \wedge holds(propN, t)$$

The above *initiates* (*terminates*) clause states that each event of type $ev1$ ($ev2$) initiates (terminates) a period of time during which the property *prop* holds, provided that N (possibly zero) given preconditions hold at instant t. The EC model of time and change is concerned with deriving the maximal validity intervals (MVIs) over which properties hold: a validity interval must not contain any interrupting event for the property; a *maximal* validity interval (MVI) is a validity interval which is not a subset of any other validity interval for the property. The clause $mholds\_for(p, [S, E])$ returns the MVIs for a given property p: each MVI is given by a pair $[S, E]$, where S (Start) and E (End) are the lower and upper endpoints of the interval.

Chittaro and Montanari distinguished two alternative ways of interpreting *initiates* clauses in the derivation of MVIs [73, 72]. In the first one (*weak* interpretation), only terminating events are considered as interrupting events, and an initiating event e for property p initiates an MVI, provided that p has not been already initiated by a previous event in such a way that p already holds at the occurrence time of e [73, 72]. For example, both Fig. 2.2b and 2.2d contain two consecutive weakly initiating events (denoted as wI) for the same property, and thus the derived MVI is initiated by the first of the two events. The alternative interpretation (*strong* interpretation) considers also initiating events as interrupting events: therefore, an initiating event e for property p initiates an MVI, provided that there is no subsequent initiating event for p such that p is not terminated between the two events. For example,

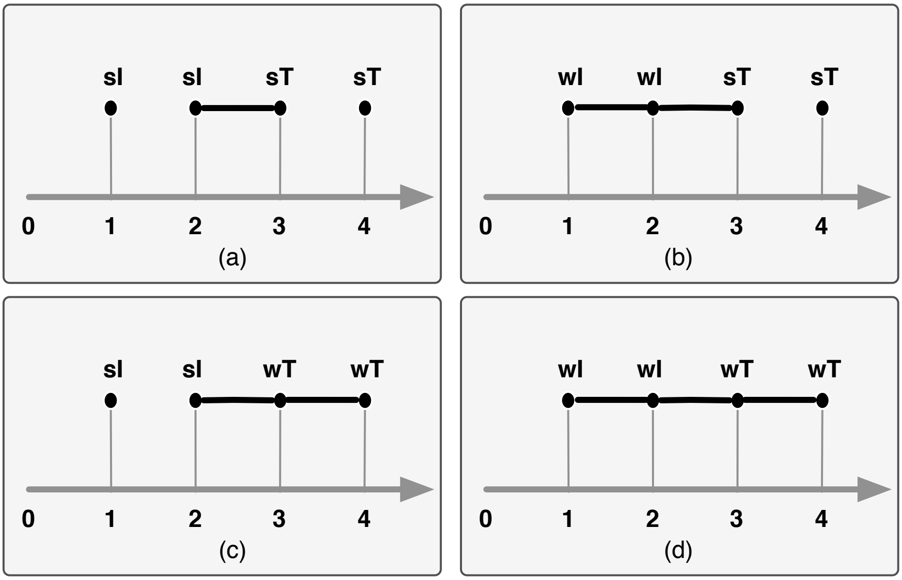


Fig. 2.2 Examples of weakly and strongly initiating and terminating events.

both Fig. 2.2a and 2.2c contain two consecutive strongly initiating events (denoted as *sI*) for the same property, and thus the derived MVI is initiated by the second of the two events. The weak and strong interpretation for *terminating* events give *symmetrical* results: Figures 2.2a and 2.2b show two consecutive strongly terminating events (denoted as *sT*), and the derived MVI terminated by the first of the two events; while Fig. 2.2c and 2.2d show two consecutive weakly terminating events (denoted as *wT*), and the derived MVI terminated by the second of the two events.

As clearly shown by Fig. 2.2, different choices of interpretation for initiating and terminating events may change the derived MVIs. In general, this choice depends on the specific property that needs to be

modeled, as we will also see in the following examples. In particular, weak and strong *initiates* relations can be used to support the so-called temporal *aggregation* and *omission* [73], respectively. For example, consider the problem of monitoring patients who receive a partial mechanical respiratory assistance [235]. A basic requirement of the patient monitoring task is the ability of aggregating similar observed situations. It indeed often happens that data acquired with consecutive samplings do not cause a transition in the classification of the patient ventilatory state. In this case, temporal aggregation requires that the subsequent samples do not clip the validity interval for the patient state. Such a functionality can be easily supported by the EC, provided that a weak interpretation of *initiates* is assumed. Temporal omission is useful when dealing with incomplete sequences of events [73]. As a simple example, consider a patient in a ICU receiving a continuous ECG monitoring, which can be interrupted by patient movements, specific treatments and examinations, and so on; the patient can be connected or disconnected to the device. The situation can be described by means of the property *ECGmonitor*(*Connection*), where the value of *Connection* can be *connected* or *disconnected*, and two events: *connect* (resp. *disconnect*), that changes the status of the connection from *disconnected* to *connected* (resp. from *connected* to *disconnected*). While two *connect* (resp. *disconnect*) events cannot occur consecutively in the real world without a *disconnect* (resp. *connect*) event in between, it might happen that an incomplete sequence consisting of two consecutive *connect* events $e1$, $e2$, followed by a *disconnect* event $e3$, is recorded in the database. In such a case, a strong interpretation of *initiates* allows the EC to recognize that a missing *disconnect* event must have occurred between $e1$ and $e2$. However, since it is not possible to temporally locate such an event, the validity of the property *ECGmonitor*(*connected*) is derived

only between $e2$ and $e3$, and $e1$ is considered as a *pending* initiating event.

2.3.2.4 Allen's Interval-Based Temporal Logic and Related Extensions

Allen [8] has proposed a framework for temporal reasoning, the *interval-based temporal logic*. The only ontological temporal primitives in Allen's logic are *intervals*. Intervals are also the temporal unit over which we can interpret *propositions*. There are no instantaneous events-events are degenerate intervals. Allen's motivation was to express natural-language sentences and to represent plans. Allen has defined 13 basic binary relations between time intervals, six of which are inverses of the other six: BEFORE, AFTER, OVERLAPS, OVERLAPPED, STARTS, STARTED BY, FINISHES, FINISHED BY, DURING, CONTAINS, MEETS, MET BY, EQUAL TO (see Figure 2.3).

It turns out that all of the thirteen relations can be expressed using only a single one, MEETS; for example, A BEFORE B can be expressed as
$\exists C(A \text{ MEETS } C \wedge C \text{ MEETS } B)$ [9].

Incomplete temporal information common in natural-language is captured intuitively enough by a disjunction of several of these relations (e.g., T_1 <starts, finishes, during> T_2 denotes the fact that interval T_1 is contained somewhere in interval T_2, but is not equal to it). In this respect, Allen's logic resembles the event calculus.

Allen defined three types of propositions that might hold over an interval:

1. Properties hold over every subinterval of an interval. Thus, the meaning of Holds(p, T) is that property p holds over interval T. For instance, "John had fever during last night."
2. Events hold only over a whole interval and not over any subinterval of it. Thus, Occur(e, T) denotes that event e occurred at time T. For instance, "John broke his leg on Saturday at 6 P.M."
3. Processes hold over some subintervals of the interval in which they occur. Thus, Occurring(p, T) denotes that the process p is occurring during time T. For instance, "John had atrial fibrillation during the last month."

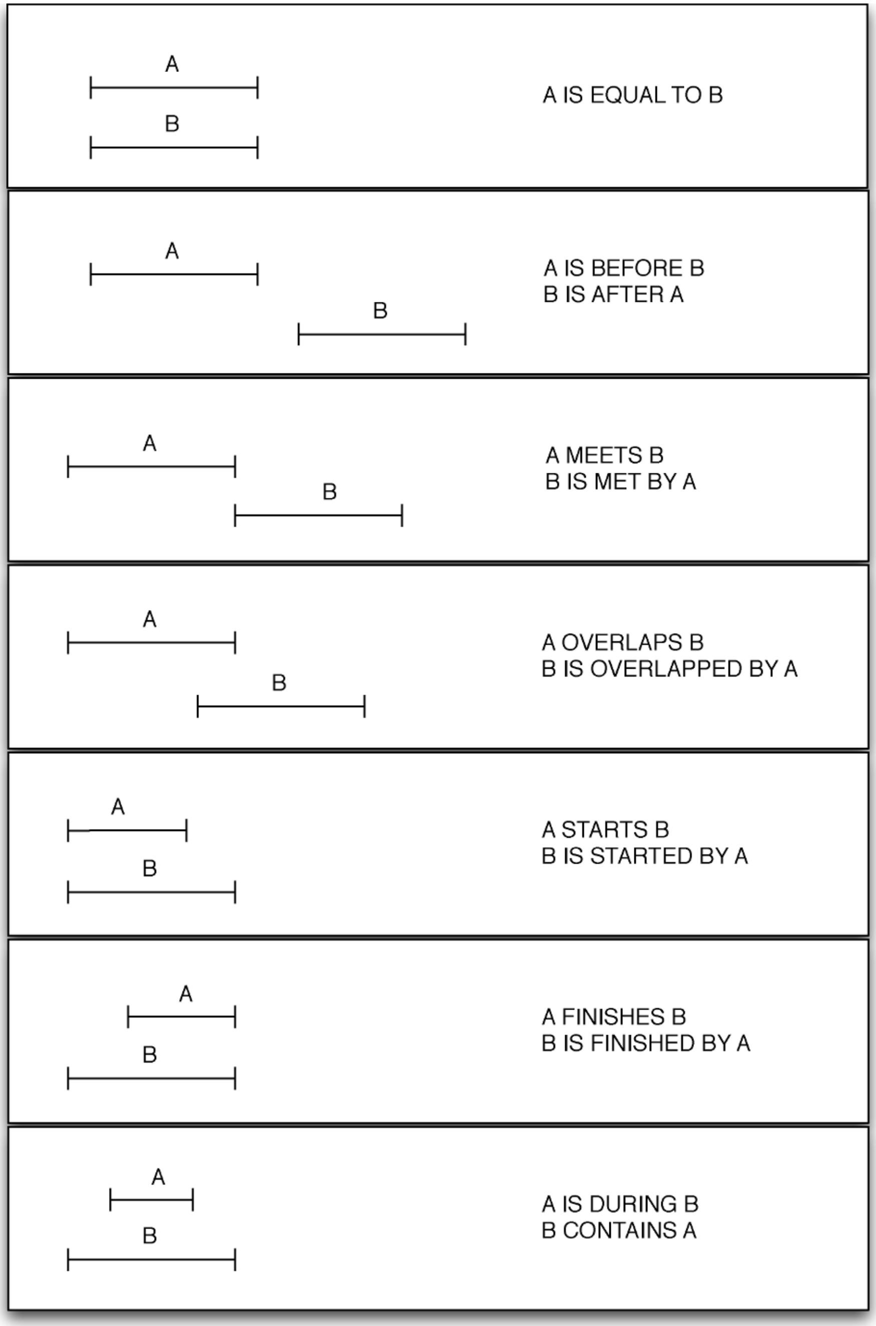


Fig. 2.3 The 13 possible relations, defined by Allen [8], between temporal intervals. Note that six of the relations have inverses, and that the EQUAL relation is its own inverse.

Allen's logic does not allow branching time into the past or the future (unlike, for instance, McDermott's logic).

Allen also constructed a transitivity table that defines the conjunction of any two relations, and proposed a *sound* (i.e., produces only correct conclusions) but *incomplete* (i.e., does not produce all correct conclusions) algorithm that propagates efficiently ($O(n^3)$) and correctly the results of applying the transitivity relations [10].

Unfortunately, the complexity of answering either the question of *completeness* for a set of Allen's relations (finding *all* feasible relations between *all* given pairs of events), or the question of *consistency* (determining whether a given set of relations is consistent) is NP-complete [416, 417]. Thus, in our current state of knowledge, for practical purposes, settling such issues is intractable. However, more work [413] has suggested that limited versions of Allen's relations - in particular, *simple interval algebra (SIA)* networks - can capture most of the required representations in medical and other areas, while maintaining computational tractability. SIA networks are based on a subset of Allen's relations that can be defined by conjunctions of equalities and inequalities between endpoints of the two intervals participating in the relation, but disallowing the ≠ (NOT EQUAL TO) relation [413].

Additional extensions to Allen's interval-based logic include Ladkin's inclusion of *nonconvex intervals* [231, 232].

2.3.2.5 McDermott's Point-Based Temporal Logic

McDermott [259] suggested a point-based temporal logic. The main goal of McDermott's logic was to model causality and continuous change, and to support planning.

McDermott's temporal primitives are *points*, unlike Allen's intervals. Time is continuous: The time line is the set of real numbers. Instantaneous snapshots of the universe are called states. States have an order-preserving *date* function to time instants. Propositions can be interpreted either over states or over intervals (ordered pairs of states), depending on their type. There are two types of propositions, facts and events. Facts are interpreted over points, and their semantics borrow from the situation calculus. The proposition (On Patient1 Bed2) represents the set of states where Patient1 is on Bed2. Facts are of the form (T s p), in McDermott's Lisp-like notation, meaning that p is true in s, where s is a state and p is a proposition, and $s \in p$. An event e is the set of intervals over which the event exactly happens: (Occ s_1 s_2 e) means that event e occurred between the states s_1 and s_2 - that is, over the interval $[s_1\ s_2]$ - where $[s_1\ s_2] \in e$. McDermott's *external* characterization of events by actually *identifying* events as sets of intervals has been criticized (e.g., [149]). Such a characterization seems to define events in a rather superficial way (i.e., by temporal spans) that might even be computationally intractable for certain types of events, instead of relying on their *internal* characterization.

McDermott's states are partially ordered and branching into the future, but are totally ordered for the past (unlike Allen's intervals, which are not allowed to branch into either the past or the future). This branching intends to capture the notion of a known past, but an indeterminate future. Each maximal linear path in such a branching tree of states is a chronicle. A chronicle is thus a complete possible history of the universe, extending to the indefinite past and future; it is a totally ordered set of states extending infinitely in time [259].

2.3.2.6 Shoham's Temporal Logic

Shoham [376], in an influential paper, attempted to clean up the semantics of both Allen's and McDermott's temporal logics by presenting a third temporal logic. Shoham pointed out that the predicate-calculus semantics of McDermott's logic, like those of Allen's, are not clear. Furthermore, both Allen's "properties, seem at times either too restrictive or too general. Finally, Allen's avoidance of time points as primitives leads to unnecessary complications [376].

Shoham therefore presented a temporal logic in which the time primitives are *points*, and propositions are interpreted over time *intervals*. Time points are represented as zero-length intervals, $< t, t >$. Shoham used *reified* first-order-logic propositions, namely propositions that are represented as individual concepts that can have, for instance, a temporal duration. Thus, TRUE(t_1, t_2, p) denotes that proposition p was true during the interval $< t_1, t_2 >$. Therefore, the temporal and propositional elements are explicit. Shoham notes that the simple first-order-logic approach of using time as just another argument (e.g., ON(Patient1, Bed2, t1, t2)), does not grant time any special status. He notes also that the modal-logic approach of not mentioning time at all, but of, rather, changing the interpretation of the world's model at different times (rather like the tense logics discussed in Section 2.3.2.1), is subsumed by reified first-order logic [376, 377, 172]. Shoham provided clear semantics for both the propositional and the first-order-logic cases, using his reified first-order temporal logic. Furthermore, he pointed out that there is no need to distinguish among particular types of propositions, such as by distinguishing *facts* from *events*: Instead, he defined several relations that can exist between the truth value of a proposition over an interval and the truth value of the proposition over other intervals. For instance, a proposition type is downward-hereditary if, whenever it holds over an interval, it holds over all that interval's subintervals, possibly excluding its end points [376] (e.g., "Sam stayed in the hospital for less than 1 week"). A proposition is upward-hereditary if, whenever it holds for all proper subintervals of some nonpoint interval, except possibly at that interval's end points, it holds over the nonpoint interval itself (e.g., "John received an infusion of insulin at the rate of 2 units per hour"). A proposition type is gestalt if it never holds over two intervals, one of which properly contains the other (e.g., the interval over which the proposition "the patient was in a coma for exactly 2 weeks" is true cannot contain any subinterval over which that proposition is also true). A proposition type is concatenable if, whenever it holds over two consecutive intervals, it holds also over their union (e.g., when the proposition "the patient had high blood pressure" is true over some interval as well as over another interval that the first interval meets, then that proposition is true over the interval representing the union of the two intervals). A proposition is solid if it never holds over two properly overlapping intervals (e.g., "the patient received a *full* course of the current chemotherapy protocol, *from start to end*," cannot hold over two different, but overlapping intervals). Other proposition types exist, and can be refined to the level of interval-point relations.

Shoham observed that Allen's and McDermott's *events* correspond to *gestalt* propositions, to solid ones, or to both, whereas Allen's *properties* are both *upward-hereditary* and *downward-hereditary* [376]. This observation immediately explains various theories that can be proved about Allen's properties, and suggests a more expressive, flexible categorization of proposition types for particular needs.

2.3.2.7 Projection, Forecasting, and Modeling the Persistence Uncertainty

The probabilistic approach is typically associated with the tasks of interpretation or forecasting of time-stamped clinical data whose values are affected by different sources of uncertainty [331, 247].

Dean and Kanazawa [118] proposed a model of probabilistic temporal reasoning about propositions that *decay* over time. The main idea in their theory is to model explicitly the probability of a proposition P being true at time t, $\mathrm{P}(< P, t >)$, given the probability of $< P, t\text{-}\Delta >$. The assumption is that there are events of type E_p that can cause proposition p to be true, and events of type $E_{\neg p}$ that can cause it to be false. Thus, we can define a *survivor function* for $\mathrm{P}(< P, t >)$ given $< P, t\text{-}\Delta >$, such as an exponential decay function.

Dean and Kanazawa's main intention was to solve the *projection problem*, in particular in the context of the *planning* task. They therefore provide a method for computing a belief function (denoting a belief in the consequences) for the projection problem, given a set of causal rules, a set of survivor functions, enabling events, and disabling events [118]. In a later work, Kanazawa [201] presented a logic of time and probability, $\mathcal{L}_{cp}$. The logic allows three types of entities: domain objects, time, and probability. Kanazawa stored the propositions asserted in this logic over intervals in what he called a *time network*, which maintained probabilistic dependencies among various facts, such as the time of arrival of a person at a place, or the range of time over which it is true that the person stayed in one place [201]. The time network was used to answer queries about probabilities of facts and events over time.

Dagum, Galper, and Horvitz [110, 109] present a method intended specifically for the *forecasting* task. They combine the methodology of static *belief networks* [299] with that of classical probabilistic time-series analysis [424]. Thus, they create a *dynamic network model* (DNM) that represents not only probabilistic dependencies between parameter x and parameter y at the same time t, but also $\mathrm{P}(x_t|y_{t-k})$, namely the probability distribution for the values of x given the value of y at an *earlier* time. Given a series of time-stamped values, the conditional probabilities in the DNM are modified continuously to fit the data. The DNM model was tried successfully on a test database of sleep-apnea cases to predict several patient parameters, such as heart rate and blood pressure [107].

2.4 Three Well-Known General Theories of Time and the Medical Domain

As already discussed, three well-known general theories of time, that are justifiably credited for the sparking of widespread interest in time representation and temporal reasoning in the AI community are Allen's interval-based temporal logic [8], Kowalski and Sergot's Event Calculus (EC) [228] and Dean and McDermott's Time Map Manager (TMM) [117]. None of these general theories of time was developed with the purpose of supporting knowledge-based problem solving, let alone medical problem solving. Hence it comes as no surprise that in their basic form, none of these adequately supports the identified requirements for medical temporal reasoning discussed above (see Table 2.1). As a matter of fact various extensions of Allen's logic and the event calculus have been applied to medical problems with lesser or greater success; some of these approaches are mentioned in the sequel. Such attempts resulted in revealing the rather limited expressivity of these theories with respect to medical problems. Their widespread adoption is in fact attributed to their relative simplicity. However, their lack of structuredness both with respect to a model of time as well as a model of occurrences, but more importantly their very limited support for the critical process of temporal data abstraction, renders their applicability in the context of medical problems at large, non viable. Below we quote some of the criticisms of the EC that was expressed by Chittaro and Dojat [70] in their attempt to apply this general theory of time to patient monitoring. In the EC a change in a property is the effect of an event. In real-life a symptom may be self-limiting where no event is required to terminate its existence. The designers went around this problem by introducing so-called "ghost" events. Another limitation encountered was that only instantaneous causality could be expressed. So delayed effects or effects of a limited persistence could not be expressed. The limited support for temporal data abstraction, the lack of multiple granularities as well as the lack of any vagueness in the expression of event occurrences, are also pointed out as issues of concern regarding the expressivity of the EC with respect to the realities of medical problems.

To illustrate further the points of criticism raised, we try to represent some medical knowledge in terms of these general theories. The medical knowledge in question describes (in a simplified form) the skeletal dysplasia (SpondyloEpiphyseal Dysplasia Congenital: SEDC), where a skeletal dysplasia is a generalized abnormality of the skeleton. This knowledge is given below:

> "SEDC *presents from birth* and can be lethal. It *persists throughout the lifetime* of the patient. People suffering from SEDC exhibit the following: short stature, due to short limbs, *from birth*; mild platyspondyly *from birth*; absence of the ossification of knee epiphyses *at birth*; bilateral severe coxa-vara *from birth*, *worsening with age*; scoliosis, *worsening with age*; wide triradiate cartilage *up to about the age of 11 years*; pear-shaped vertebral-bodies *under the age of 15 years*; variable-size vertebral-bodies *up to the age of 1 year*; and *retarded ossification* of the cervical spine, epiphyses, and pubic bones."

The text given in italic font refers to time, directly or indirectly.

Table 2.1 Evaluation of General Theories of Time Against Medical Temporal Requirements. key: X - does not support; (V) - supports partly; V - supports

	Allen's Time-Interval Algebra	*Kowalski & Sergot's Event Calculus*	*Dean & McDermott's Time-Token Manager*
multiple conceptual temporal contexts	X	X	X
multiple granularities	X	X	X
absolute existences	X	V	V
relative existences	V	X	(V)
absolute vagueness	X	X	V
relative vagueness	V	X	X
duration	X	V	V
point existences	X	V	V
interval existences	V	V	V
periodic occurrences	X	X	X
temporal trends	X	X	X
temporal patterns	(V)	X	(V)
structural relations (temporal composition)	X	X	X
temporal causality	(V)	(V)	(V)

The temporal primitive of Allen's interval-based logic is the time interval, and eight basic relations (plus the inverses for seven of these) are defined between time intervals. The other primitives of the logic are properties (static entities), processes and events (dynamic entities), which are respectively associated with predicates holds, occurring and occur as already discussed:

$$holds(p,t) \Longleftrightarrow (\forall t' \, in(t',t) \Longrightarrow holds(p,t'))$$
$$occurring(p,t) \Longrightarrow \exists t' \, in(t',t) \wedge occurring(p,t')$$
$$occur(e,t) \wedge in(t',t) \Longrightarrow \neg occur(e,t')$$

The logic covers two forms of causality, event and agentive causality. Allen's logic is a relative theory of time, where time is structured as a dense time line. In order to represent the SEDC knowledge in terms of Allen's logic we need to decide which of the entities correspond to events, which to properties, and which to processes. The relevant generic events are easily identifiable. These are: *birth(P)*, *age1yr(P)*, *age11yrs(P)*, *age15yrs(P)* and *death(P)* which mark the birth, the becoming of 1 year of age, etc of some patient *P*. Deciding whether to model SEDC and its manifestations as properties or processes is not immediately apparent. In the following representation the distinction into processes and properties is decided on a rather ad hoc basis:

$$occurring(SEDC(P),I) \Longrightarrow occur(birth(P),B) \wedge occur(age1yr(P),O) \wedge$$
$$occur(age11yrs(P),E) \wedge occur(age15yrs(P),F) \wedge occur(death(P),D) \wedge$$
$$started\text{-}by(I,B) \wedge finished\text{-}by(I,D) \wedge holds(stature(P,short),I) \wedge$$
$$holds(ossification(P,knee\text{-}epiphyses,absent),B) \wedge$$
$$occurring(coxa\text{-}vara(P,bilateral\text{-}severe,worsening),I) \wedge$$
$$occurring(scoliosis(P,worsening),I) \wedge$$
$$holds(triradiate\text{-}cartilage(P,wide),W) \wedge started\text{-}by(W,B) \wedge finished\text{-}by(W,E) \wedge$$
$$holds(vertebral\text{-}bodies(P,pear\text{-}shaped),F') \wedge started\text{-}by(F',B) \wedge before(F',F) \wedge$$
$$holds(vertebral\text{-}bodies(P,variable\text{-}size),V) \wedge started\text{-}by(V,B) \wedge finished\text{-}by(V,O) \wedge$$

$occurring(ossification(P, cervical\text{-}spine, poor), I) \wedge$
$occurring(ossification(P, epiphyses, retarded), I) \wedge$
$occurring(ossification(P, pubic\text{-}bones, retarded), I)$

In this formalization, a relative representation has been "forced" on absolute occurrences. The specified events are not consequences of the occurrence of SEDC; their role is to demarcate the relevant intervals. For this (disorder) representation to be viable, the implication should either be temporally screened against the particular patient in order to remove future or non-applicable consequences, or simply such happenings should be assumed to be true by default. A particular limitation of any relative theory of time is inability to adequately model the derivation of temporal trends, or the derivation of delays or prematurity with respect to the unfolding of some process, since the notion of temporal distance which is inherently relevant to both types of derivation is foreign to such theories of time. A statement about a trend, delay, prematurity, etc is a kind of summary statement for a collection of happenings over a period of time. Another limitation of relative theories of time is inability to model absolute vagueness. In the above representation the widening of the triradiate cartilage is expected to hold exactly up to the occurrence of the event "becoming 11 years of age" and also it is not possible to delineate a margin for the termination of the property "pear-shaped vertebral bodies"; instead its termination is expressed in a relative way by saying that this happens before the event "becoming 15 years of age" happens, which does not capture the intuitive meaning of the given manifestation.

The temporal primitive of Kowalski and Sergot's EC is the event. Events are instantaneous happenings which initiate and terminate periods over which properties hold. A property does not hold at the time of the event that initiates it, but does hold at the time of the event that terminates it. Default persistence of properties is modeled through negation-as-failure. Causality is not directly modeled, although a rather restricted notion of causality is implied, e.g. an event happening at time t causes the initiation of some property at time (t+1) and/or causes the termination of some (other) property at time t. The calculus can be applied both under a dense or a discrete model of time. The EC representation of the SEDC knowledge consists of a number of clauses like the following:

$$initiates(birth(P), ossification(P, knee\text{-}epiphyses, absent), t) \Longleftarrow happens(birth(P), t) \wedge holds(SEDC(P), t)$$

$$initiates(birth(P), stature(P, short), t) \Longleftarrow happens(birth(P), t) \wedge holds(SEDC(P), t)$$

$$terminates(death(P), stature(P, short), t) \Longleftarrow happens(death(P), t) \wedge holds(SEDC(P), t)$$

$$initiates(birth(P), coxa\text{-}vara(P, bilateral\text{-}severe, worsening), t) \Longleftarrow happens(birth(P)) \wedge holds(SEDC(P), t)$$

$$terminates(age15yrs(P), vertebral\text{-}bodies(P, pear\text{-}shaped), t) \Longleftarrow happens(age15yrs(P), t) \wedge holds(SEDC(P), t)$$

Many of the criticisms discussed above with respect to Allen's logic apply to the EC as well. Properties in the EC are analogous to Allen's properties. They are essentially ‘static’ entities. Evolving situations such as temporal trends, or retardation in the execution of some process, or more generally continuous change, cannot be adequately modeled within pure EC. For example, the above clause concerning coxa-vara talks about some worsening being initiated, and also, based on the various axioms of the EC, it can be inferred that the worsening holds at every instant of time. What is initiated is “bilateral severe coxa-vara” while the worsening of this condition is a kind of meta-level inference on the continuous progression of this condition. Furthermore, absolute vagueness is not addressed, and as with Allen’s logic, the SEDC knowledge is not represented as an integral entity but as a sparse collection of ‘independent’ happenings.

The temporal primitive of Dean and McDermott’s TMM is the point (instant). The other temporal entity is the time-token that is defined to be an interval together with a (fact or event) type. A time-token is a static entity. It cannot be structurally analyzed and it cannot be involved in causal interactions. A collection of time-tokens forms a time map. This is a graph in which nodes denote instants of time associated with the beginning and ending of events and arcs describe relations between pairs of instants. This ontology can be applied both under a dense or a discrete model of time. Below we represent part of the SEDC knowledge as a time map. The granularity used is years and the reference point (denoted as *ref*) is birth. The first argument of the time-token predicate is the (fact or event) type and the second is the interval. Predicate elt expresses margins (bounds) for the beginnings and endings of intervals, with respect to *ref*.

(*time-token*(*SEDC*, *present*)*I*)
(*time-token*(*coxa-vara*, *bilateral-severe*)*C*)
(*time-token*(*coxa-vara*, *worsening*)*C′*)
(*time-token*(*ossification*, *epiphyses*, *retarded*)*E*)
(*time-token*(*triradiate-cartilage*, *wide*)*W*)
(*time-token*(*vertebral-bodies*, *pear-shaped*)*V*)
....
(*elt*(*distance*(*begin C*) **ref**)0, 0)
(*elt*(*distance*(*end C*) **ref**), **pos-inf** **pos-inf**)
(*elt*(*distance*(*begin C′*) **ref**)?, ?)
(*elt*(*distance*(*end C′*) **ref**)?, ?)
(*elt*(*distance*(*begin W*) **ref**)0, 0)
(*elt*(*distance*(*end W*) **ref**)10, 11)
(*elt*(*distance*(*begin V*) **ref**)0, 0)
(*elt*(*distance*(*end V*) **ref**)?, 14)
(*elt*(*distance*(*begin E*) **ref**)?, ?)
(*elt*(*distance*(*end E*) **ref**)?, ?)
....

Again the SEDC process per se and its manifestations are represented as independent occurrences. The expression of absolute temporal vagueness is supported (see instances of predicate elt above), but no mechanism for translating qualitative expressions of vagueness into the relevant bounds based on temporal semantics of properties is provided. In the above representation “up to about the age of 11 years” is translated, in an ad hoc way, to the margin (10 11) while for “under the age of 15 years” it is not easy to see what the earliest termination ought to be. The points raised above regarding the representation of trends, process retardations, etc., apply here as well. Again this is because the types associated with the tokens capture either instantaneous events, or static, downward hereditary, properties.

Thus, the important reasoning process of temporal data abstraction is not supported by any of the three general theories of time considered.

2.5 Temporal Constraints

In the AI community there has been substantial interest in networks of constraints, in particular in arc and path consistency algorithms, over the past 30 years and many authors have contributed to the development of the relevant ideas (see for example [119, 251, 269, 271]). Generally, the work on consistency algorithms focuses on computational matters and not so much on the constraints themselves. As such,

the constraints used are often of a relatively simple form, such as ranges for the temporal distances concerned. In medical tasks such as clinical diagnosis, one needs to address more complex forms of temporal constraints. For example, simple ranges capture uncertainty but in a rather categorical or discrete way. A simple range cannot model the fuzziness that often arise in clinical domains, such as when different ranges with varying degrees of typicality are required. Furthermore, clinical temporal constraints could be of mixed types and more importantly they could involve different granularities.

In this section we define an abstract structure for the representation of temporal constraints. In Chapter 6 we discuss particular instantiations of this structure of relevance to clinical diagnosis. The abstract structure, referred to as an *Abstract Temporal Graph*, or ATG for short, on one hand places the different types of constraints within the same, and thus unifying, framework and on the other hand enables the analysis and differentiation of the various types of constraints. By viewing temporal constraints in an abstract and more holistic way, it is possible to adopt, and appropriately adapt, well known constraint consistency algorithms from the general literature. Such algorithms can be further refined for particular instantiations of the abstract structure.

The problems we wish to address in this section are the following:

1. Checking the consistency of a set of constraints.
2. Deciding the satisfiability of some constraint with respect to a set of constraints that are assumed to be mutually consistent.

The first problem concerns the validation of the temporal consistency of a body of knowledge (e.g. the knowledge comprising the model of some disorder), or a set of data (e.g. the data on some patient). The second problem concerns the evaluation of the temporal consistency of some hypothesis against the evidence.

Abstract Temporal Graph

An Abstract Temporal Graph (ATG) is a directed graph whose nodes represent temporal entities (events, occurrences, etc.) and its arcs are labeled with the possible temporal constraints between the given pairs of nodes.

Let C be the domain of binary temporal constraints. The elements of C are mutually exclusive; only one of these can give the relationship that actually holds between the existence of one temporal entity and that of another temporal entity. C is either a finite or an infinite set. At the general level of discussion we consider the elements of C to be abstract entities processed by the following access functions:

1. $id : C \times C \to \{true, false\}$
   Function *id* returns *true* only if its arguments are identical.
2. $inverse : C \to C$
   Every element in C has an inverse that is also an element of C; hence the domain of constraints is considered symmetrical. For example the inverse of *before* is *after* and of 1 *day* (meaning 1 day before) is -1 *day* (meaning 1 day after). Function *inverse* returns the inverse of its argument. Furthermore, $inverse(inverse(c)) = c$.
3. $transit : C \times C \to 2^C$
   The arguments of function *transit* refer to three (normally distinct) temporal entities, say n_i, n_j and n_k; the first argument, c_{ik}, represents the constraint from n_i to n_k and the second, c_{kj}, the constraint from n_k to n_j. The function returns the disjunctive constraint from n_i to n_j, i.e. the transitivity of the given (atomic) constraints.

In addition, special constant *self_ref* denotes the element of C that gives the constraint of any temporal entity with itself, e.g. *equal* or 0 *days*, etc. Any self-referencing arc in an ATG would have as its label the set {*self_ref*}. Furthermore, $inverse(self\_ref) = self\_ref$.

A more formal definition of an ATG is now given.

Definition 1 — An *Abstract Temporal Graph* (ATG) is a graph consisting of a finite set of nodes, $n_1, n_2, ...n_m$, denoting temporal entities (of the same type), and a finite set of directed arcs. A directed arc from n_i to n_j is labeled with a set of temporal constraints, $tc_{ij} \subseteq C$, denoting a disjunctive constraint from n_i to n_j. An ATG has access functions *match* and *propagate* for processing disjunctive constraints.

1. $match : 2^C \times 2^C \to 2^C$
   Function *match* returns the common elements between the two disjunctive constraints, in other words their intersection. If the function returns the empty set, denoting a complete mismatch between the argument constraints (none of the arguments is the empty set), a conflict is signalled.

```
match(C_i, C_j)
   R ← {}
   for c_i ∈ C_i do
      for c_j ∈ C_j do
```

```
        if id(c_i, c_j) then R ← R ∪ {c_j}
    return R
```

2. *propagate* : $2^C \times 2^C \rightarrow 2^C$
   Function *propagate* returns the transitivity of its argument constraints that respectively represent the pairwise (disjunctive) constraints between three (normally distinct) temporal entities.

```
propagate(C_i, C_j)
    R ← {}
    for c_i ∈ C_i do
        for c_j ∈ C_j do
            R ← R ∪ transit(c_i, c_j)
    return R
```

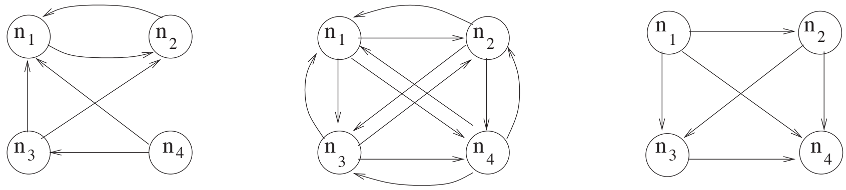
If in an ATG, label tc_{ij} has only one element, there is no uncertainty (from the perspective of C) as to the relationship from n_i to n_j. If however $tc_{ij} = C$, there is complete ignorance of this relationship. Unconnected nodes in an ATG can always be connected via arcs with labels set to C.

There are two extreme cases. One is when for every pair of nodes, n_i and n_j, tc_{ij} is a singleton. This is the case where there is complete temporal knowledge and no uncertainty whatsoever (always with respect to C). This is what we strive to reach, as in reality there is only one possible scenario. The other extreme case is when for every pair of nodes, n_i and n_j, $tc_{ij} = C$. This is the case of complete ignorance regarding temporal information, i.e. everything is unknown or not given and hence everything is completely temporally unconstrained. In this case there is no point in connecting the nodes and hence the ATG degenerates into a set of unconnected nodes.

Definition 2 — A *fully connected ATG* is an ATG for which every pair of nodes n_i and n_j such that $i \neq j$ is connected in both directions and each connection is labeled (possibly with the
entire set of constraints, C).

Any ATG can be easily converted to a fully connected ATG, simply by adding the relevant arcs and labeling them with C. However, inverse arcs are redundant since their labels can be obtained directly from their counterparts. Hence inverse arcs may be deleted; which arc is actually deleted from each pair of counter arcs could be decided on the basis of some ordering of the nodes.

Definition 3 — An *ordered ATG* is an ATG whose nodes $n_1, n_2, \ldots., n_m$ form a topological ordering and for every pair of nodes n_i and n_j such that $i < j$ (i.e. n_i precedes n_j in the topological ordering), there is a labeled connection from n_i to n_j. Pairs of nodes, n_i, n_j, such that $i \geq j$ are not connected. The $(m-1)$ arcs connecting nodes, that are consecutive under the topological ordering, i.e. the arcs from n_i to n_{i+1}, for $i = 1, \ldots, m-1$, are referred to as *basic arcs* because each of them represents the sole path between the given pairs of nodes.



(a) General ATG (b) Fully Connected ATG (c) Ordered ATG

Fig. 2.4 Abstract Temporal Graphs

If one of the nodes in an ATG denotes *now*, presumably this node will either be the first or the last in the chosen topological ordering of the nodes, so that either it points to all other nodes or it is pointed by all other nodes.

An ATG can be converted to an ordered ATG, by adding any missing connections in the specified direction and deleting any connections in the opposite direction. Figure 2.4 shows a general ATG and its fully connected and ordered versions, where the ordering of the nodes is n_1, n_2, n_3, and n_4. The arc labels are omitted in the figure. When an arc from n_i to n_j is added, its label is set to tc_{ji}^{-1} (see below), provided there is an arc from n_j to n_i, otherwise the label is set to C. No self-referencing arcs are included.

A fully connected ATG can be reduced to its ordered version by deleting the inverse arcs according to the chosen ordering of its nodes. If m is the number of nodes, the fully connected ATG has $m(m-1)$ arcs (since self-referencing arcs are excluded) while the ordered ATG has half this number of nodes. Alternatively we can say that in the ordered ATG, $(m-1)$ arcs emanate from n_1, $(m-2)$ from n_2, ..., and 1 from n_{m-1}, giving a total of $m(m-1)/2$ arcs. Hence in the ordered ATG, the first node, n_1, points to $n_2, ..., n_m$, and no node points to it, an intermediate node, n_i, is pointed at by $n_1, ..., n_{i-1}$ and points to $n_{i+1}, ..., n_m$ for $i = 2, ..., m-1$ and the last node, n_m, is pointed at by every other node and points to none.

If in some ATG, a pair of nodes, n_i, n_j, is connected in both directions, the labels of the opposite arcs, tc_{ij} and tc_{ji}, must be consistent. Consistency can be interpreted in a strict sense as $match(tc_{ij}, tc_{ji}^{-1}) = tc_{ij}$, or in a more liberal sense as $match(tc_{ij}, tc_{ji}^{-1}) \neq \{\}$, where $tc^{-1} = \{inverse(c) \mid c \in tc\}$ gives the inverse of tc in C; in particular $C^{-1} = C$. The strict interpretation requires complete match, while the liberal one is satisfied with a match on just one constraint and thus a conflict is raised only if there is no match. In the following algorithms we use the liberal interpretation[1].

[1] An alternative interpretation is to use the union of the labels, i.e., $tc_{ij} \leftarrow tc_{ij} \cup tc_{ji}^{-1}$. This interpretation extends the possibilities.

Definition 4 — A *minimal ATG* is an ordered ATG whose arcs have labels that cannot be further reduced.

If in a minimal ATG, all the arcs involving some node n_i are labeled by the entire set of constraints, C, n_i is temporally unconstrained with respect to all other nodes and as such it represents an unconnected component. If all nodes are temporally unconstrained, the ATG degenerates to the extreme case of complete temporal ignorance mentioned above. Likewise if some arc in a minimal ATG has label {}, a conflict should be raised since all possible temporal relations between the given pair of temporal entities have been refuted.

Starting from a general ATG, the goal is to turn it into its minimal form by propagation and matching of constraints. Below we examine some algorithms for this task.

The first algorithm, *minimize_full*, extends the general ATG to its fully connected version (including self-referencing arcs as well, the presence of which simplifies the expression of the algorithm), then performs the propagation and matching of constraints and finally deletes the inverse and self-referencing arcs to obtain the ordered ATG, with minimal constraints. The algorithm is an adaptation of the well known Floyd-Warshall algorithm [119].

```
minimize_full
    (* add connections *)
    for i = 1 to m do
        for j = 1 to m do
            if i = j
            then add a self-referencing arc at node n_i
                with label set to {self_ref}
            else if there is an arc from node n_i to node n_j
                then do nothing
            else if there is an arc from node n_j to node n_i
                then add an arc from n_i to n_j with label set to tc_ji^-1
            else add an arc from node n_i to node n_j with label set to C
    (* propagate constraints *)
    repeat
        for k = 1 to m do
            for i = 1 to m do
                for j = 1 to m do
                    (* critical step *)
                    tc_ij ← match(tc_ij, propagate(tc_ik, tc_kj))
                    if tc_ij = {} a conflict is raised
    until no arc label is reduced (* i.e. nothing changes *)
    (* remove self-referencing and inverse arcs *)
    for i = 1 to m do
        for j = i to m do
            if i = j remove self-referencing arc at node n_i
            else do
```

$tc_{ij} \leftarrow match(tc_{ij}, tc_{ji}^{-1})$
if $tc_{ij} = \{\}$ a conflict is raised
else delete arc from n_j to n_i

The critical step of the algorithm is taken to be the combined application of functions *propagate* and *match* with respect to triples of nodes, namely $match(tc_{ij}, propagate(tc_{ik}, tc_{kj}))$, which is executed m^3 times, m being the number of nodes, at each round of the repeat cycle. The number of executions of the critical step can be reduced if the minimization of constraints is done, not with respect to the fully connected ATG, but with respect to the ordered ATG. Performing the minimization on the fully connected ATG, in some sense defeats the purpose of having the ordering which is not just to reduce the space complexity by halving the number of arcs, but also to reduce the speed of execution of the minimization process.

The second algorithm, *minimize_ordered*, first converts the general ATG to its ordered version and then does the minimization by propagating and matching constraints with respect to triples of distinct nodes. In this algorithm the critical step is executed $(m^3 - 3m^2 + 2m)/6$ times, at every round of the repeat cycle. Thus although both algorithms have the same complexity, namely $O(rm^3)$, where r is the number of times the repeat cycle is executed, in real terms the critical step will be executed substantially fewer times in the second algorithm, assuming that m is of the order of tens rather than hundreds.[2] However, there is a point of caution. In the fully connected ATG, there are at least two distinct paths between every pair of nodes and the propagation is bidirectional. In the ordered ATG the propagation is unidirectional and thus the labels of the basic arcs (recall that a basic arc represents the sole path between the given pair of nodes — see Definition 3) stay invariant under this propagation. Basic arcs influence (except the last in sequence), but are not influenced by the propagation. Thus if a basic arc has label C prior to the execution of the minimization process, it will continue to have this label at the end of it. In other words nothing more is learned about that arc. Thus the results of the two algorithms are not equivalent,[3] since the first does a complete minimization but not necessarily the second, unless the labels of the basic arcs are given in minimal form to start with.

minimize_ordered
let $n_1, n_2, ..., n_m$ be the specified ordering of the nodes
(* covert to ordered form *)
for $i = 1$ to $(m-1)$ do
    for $j = i+1$ to m do
        if nodes n_i and n_j are unconnected
            then add an arc from n_i to n_j with label set to C
        else if there is only an arc from n_i to n_j

[2] This assumption is not unrealistic, especially when disorders are modeled separately in terms of their own temporal graphs.

[3] In [119] two constraint networks are equivalent if they give the same solution set, where a solution set is the set of all feasible scenarios. If the arc labels are minimal, every temporal relationship included should participate in at least one feasible scenario.

```
            then do nothing
        else if there is only an arc from n_j to n_i
            then add an arc from n_i to n_j with label set to tc_ji^-1
                and delete the arc from n_j to n_i
        else do (* there is a bidirectional connection *)
            tc_ij ← match(tc_ij, tc_ji^-1)
            if tc_ij = {} a conflict is raised
            else delete arc from n_j to n_i
(* propagate constraints *)
repeat
    (* repeat for every intermediate node in the ordering *)
    for k = 2 to (m − 1) do
        (* repeat for every incoming arc to node n_k *)
        for i = 1 to (k − 1) do
            (* repeat for every outgoing arc from node n_k *)
            for j = (k + 1) to m do
                (* critical step *)
                tc_ij ← match(tc_ij, propagate(tc_ik, tc_kj))
until no arc label is reduced (* i.e. nothing changes *)
```

Let us now return to the problems given at the beginning of the section.

Checking the Consistency of a Set of Constraints

The solution of the first problem, checking the consistency of a set of constraints is given by algorithm minimize_full or minimize_ordered. If during the execution of these algorithms, function *match* returns an empty set denoting a complete mismatch, a conflict is raised. Complete mismatch means that the disjunctive constraint relating two temporal entities, obtained via some route in the ATG, is in complete disagreement with the constraint, for the same pair of entities, obtained via another route in the ATG. In other words all possible temporal relations between the two entities have been refuted. The minimization algorithms detect the presence of some inconsistency but do not say which of the (original) constraints are responsible for it.[4]

We can sketch an algorithm for determining the possible causes of the inconsistency, as follows. Its aim is to identify minimal subsets of arc labels, each of which when omitted results in the resolution of the conflict. Omission of a label means that the particular arc label is replaced with C, which says that "If this is an erroneous label causing a conflict, it should be replaced with the label of complete ignorance.". First the algorithm assumes that a single label is the cause. Each label that is not equal to C is omitted in turn, and every time the minimization algorithm is ran to see if the omission has erased the conflict. If no single label is responsible for the

[4] There could be multiple possibilities as to the cause of the inconsistency; the identification of the actual cause amongst them may require external means.

conflict, the next step is to omit pairs of labels (not equal to C) in turn, and so forth. This is a complex algorithm. If k is the number of labels, originally not equal to C, in the worst case all these labels will be mutually inconsistent, meaning that the minimization algorithm will be ran $2^k - 1$ times. Obviously the use of *minimize_ordered* instead of *minimize_full* reduces the speed of execution, not only because this algorithm executes the critical step fewer times, but also because k would be expected to be half that for the fully connected ATG.

Deciding the Satisfiability of Some Constraint

The solution of the second problem, deciding the satisfiability of some constraint with respect to a set of constraints that are assumed to be mutually consistent, is as follows. It is based on the assumption that the queried constraint and the ATG representing the set of constraints are of the same form, i.e. the temporal entities are of the same type and the domain of constraints, C, is the same. In most applications of this problem, the queried constraint would describe a datum about the patient and the ATG would represent the model of some disorder.

Let n_i and n_j be the temporal entities implicated in the queried constraint, and let qc be the (disjunctive) constraint itself (from n_i to n_j). The temporal entities, n_i and n_j, could respectively denote the start and end of some symptom, or the starts of two distinct symptoms, etc. We distinguish the following cases:

1. Both n_i and n_j appear as nodes in the ATG.
2. Only one or none of these temporal entities appears as a node in the ATG.

In the first case, the solution is given as follows:

convert the ATG to minimal form
if there is an arc from n_i to n_j in the ATG
then if $match(tc_{ij}, qc) \neq \{\}$
    then the queried constraint, qc, is satisfied
    else it is not satisfied
else if $match(tc_{ji}, qc^{-1}) \neq \{\}$
    then the queried constraint, qc, is satisfied
    else it is not satisfied

In the second case, the liberal approach is to say that the queried constraint is satisfied by default (especially if the sentence denoted by the queried constraint expresses normality), and the strict approach is to say that it is not satisfied (except again when the queried constraint expresses normality). Which approach is taken would depend on whether satisfiability is interpreted as absence of explicit conflict, or as explicit consistency.

Summary

In this chapter we have overviewed the fundamental notions of temporal modeling and temporal reasoning. First, we discussed the modeling of basic temporal concepts, starting with the modeling of time and moving to the modeling of temporal entities, i.e., objects/facts having some temporal dimension. A model of time should take into consideration the following aspects: the time domain, the representation of instants and intervals, the structure of time (linear, branching, or circular), the representation of absolute and relative times and the representation of relations. Temporal granularity and indeterminacy are also important aspects when modeling time. Analogous considerations apply to the modeling of temporal entities and of their occurrences; different kinds of temporal entities were discussed as well as their association with the time domain and their semantics. Then, we discussed temporal reasoning starting with listing the required functionalities for medical temporal reasoning and moving to the presentation of the most influential ontologies and models for temporal reasoning (tense logics, time specialist, situation calculus, event calculus, interval-based and point-based temporal logics). In order to demonstrate the multiple aspects of medical temporal reasoning, a simple medical example was analyzed against three well-known general theories of time. We ended the chapter by discussing temporal constraints through an abstract representation, i.e. the abstract temporal graph. The main purpose of this chapter was to acquaint the reader with the necessary basic background with respect to temporal modeling and temporal reasoning, to facilitate the coverage of the remaining chapters of this book.

Bibliographic Notes

Apart from the specific references that are mentioned in this chapter, additional discussion on temporal logic is given in Rescher and Urquhart's excellent early work in temporal logic [330]. The AI perspective has been summarized well by Shoham [376, 377]. An overview of temporal logics in the various areas of computer science, and of their applications, was compiled by Galton [149]. Van Benthem's comprehensive book [414] presents an excellent view of different ontologies of time and their logical implications.

Problems

2.1. Using Prior's notation, we can write the two following predicates:

$$\mathrm{P}(\exists\, \textit{patient}\ \mathrm{diagnosis}(\textit{patient}, \mathrm{tuberculosis}))$$

$$\exists\, \textit{patient}\ \mathrm{P}(\mathrm{diagnosis}(\textit{patient}, \mathrm{tuberculosis}))$$

1. Explain in your own words exactly what each expression means, and why is their meaning different in Prior's logic. Create two convincing examples to demonstrate that there might be two mutually exclusive interpretations.
2. Explain why the distinction between the two expressions is meaningless in standard First Order Logic (predicate calculus).
3. Write, using Tense Logic notation, an expression that means "Patient Jones will have had the operation." Use expressions such as "procedure(Jones, Operation)".

2.2. Explain how, using only the MEETS relation between temporal intervals, and one or more interval variables and existential quantifiers, we can define the following relations:

1. The relation A DURING B
2. The relation A OVERLAPS B
3. What is the computational advantage of using 13 different temporal relations as opposed to only one relation? Think about applications such as theorem proving, planning, temporal queries, storage and retrieval of data.

2.3. Create a set of situation calculus axioms to express the following facts:

1. The effect of the action of entering the hospital room by a patient, when the patient is at the door, is that the patient is inside the hospital room. Use expressions such as At(Door, Patient), Enter(Patient, HospitalRoom), Within(HospitalRoom, Patient).
2. Explain the semantics of the predicates and/or functions you are using in terms of sets of states.
3. Can we express in the situation calculus the fact that the patient is looking around while moving from one room to another one? How, or Why?

2.4. What is the representational advantage of a reified logic, in which temporal arguments are explicitly separated from the rest of the predicate?

2.5. Consider the following 4 propositions:
i) "Mark had a complete removal of the appendix on January 15 1988 between 6 to 9 PM."
ii) "Joe has earned 3000 Euros during February 1999."
iii) "Mary had mild anemia during March to May 1997."
iv) "Peter was occasionally using insulin shots during July and August 1995."

1. Indicate in a small table what are the temporal-proposition properties of each proposition with respect to the properties *downward-hereditary* (dh), *upward-hereditary* (uh), *gestalt* (g), *concatenable* (c), and *solid* (s) as defined by Shoham.
2. What is the ontological type of each proposition according to Allen?
3. What is the ontological type of each proposition according to McDermott?

2.6. Study in more detail the classical approaches to temporal reasoning (Allen's time-interval logic, Event Calculus, etc.) together with Shoham's criticism. Analyze similarities and differences between these approaches and examine their appropriateness with respect to some medical domain and task you are familiar with.

2.7. Implement the notion of an ATG as an abstract data type, together with the discussed minimization algorithms. Apply your code to some example temporal graphs (see also Chapter 6).