Pergamon

Comput. Biol. Med. Vol. 27, No. 5, pp. 411–434, 1997

Printed in Great Britain
0010-4825/97 $17.00+0.00

PII: S0010–4825(97)00013–9

REPRESENTING AND DEVELOPING TEMPORALLY ABSTRACTED KNOWLEDGE AS A MEANS TOWARDS FACILITATING TIME MODELING IN MEDICAL DECISION-SUPPORT SYSTEMS

C.F. Aliferis*, G.F. Cooper, M.E. Pollack, B.G. Buchanan and M.M. Wagner

Section on Medical Informatics, Suite 8084, Forbes Tower, 200 Lothrop Street, Pittsburgh, PA 15213-2582, U.S.A.

(*Received* 20 *February* 1997)

Abstract—The utilization of the appropriate level of temporal abstraction is an important aspect of time modeling. We discuss some aspects of the relation of temporal abstraction to important knowledge engineering parameters such as model correctness, ease of model specification, knowledge availability, query completeness, inference tractability, and semantic clarity. We propose that versatile and efficient time-modeling formalisms should encompass ways to represent and reason at more than one level of abstraction, and we discuss such a hybrid formalism.

Although many research efforts have concentrated on the automation of specific temporal abstractions, much research needs to be done in understanding and developing provably optimal abstractions. We provide an initial framework for studying this problem in a manner that is independent of the particular problem domain and knowledge representation, and suggest several research challenges that appear worth pursuing. © 1997 Elsevier Science Ltd.

Medical decision-support systems Temporal reasoning Temporal representation
Temporal abstraction Belief networks

1. INTRODUCTION

A persistent problem in medical artificial intelligence (AI) is the appropriate modeling of time. There are many theoretical and practical difficulties associated with making medical decision-support systems (MDSSs) "aware" of time, and capable of putting patient data, problems and solutions in the appropriate temporal context. In section 2 of this paper we discuss certain pertinent issues in temporal representation and reasoning and provide a common vocabulary of terms on which we build the subsequent discussion. In section 3 we examine the qualitative relationship of the degree of abstraction of a model to crucial engineering properties of the model. We come to the conclusion that a combination of different levels of temporal abstraction offers advantages in terms of devising tractable models that are easy to specify and instantiate with expert knowledge and data.

The importance of incorporating grades of temporal abstraction in a model leads us to consider two important questions. The first one is how this task can be accomplished at a technical level. We discuss as an example a new formalism for time modeling in MDSSs that we developed, which we call "modifiable temporal belief networks" (MTBNs). The second question is how to select the most appropriate levels of temporal abstraction for any given temporal model. We introduce operational definitions of temporal abstraction (TA) optimality and show that under specific conditions it is possible to study the optimality of TAs expressed in one formalism using another formalism. We show that the development of provably optimal TAs is fraught with problems due to the complexity of the task, and propose ways to constrain the search space of possible TAs, as a means of approaching the problem. In the Discussion section we review research efforts in this area, and conclude by listing several open research areas in the representation and development of TAs.

*Author to whom correspondence should be addressed.

We emphasize that this paper does not aim at providing algorithms or precise procedures for constructing MDSSs. Instead we want to shed some light on fundamental issues pertaining to abstraction usage, to explain why abstractions can be useful engineering and analytical tools, and to delineate a broad research agenda in time modeling that we find interesting and of significant potential benefit.

2. FUNDAMENTAL CONCEPTS

2.1. *The time problem*

Temporal reasoning is considered an important component of good clinical decision making. Despite the attention that time attracts in medical curricula and everyday clinical practice, time is not represented and reasoned with in an explicit manner in most MDSSs. There are important historical and technical reasons for this situation, which are discussed in [1]. We will define the time problem as: the difficulty in representing and reasoning about entities in MDSSs when taking time into account. More specifically the problem involves: (a) using time to facilitate problem-solving; and (b) reasoning about time and time-related entities as part of the problem-solving process.

Several different aspects of the problem have been identified as a result of intense research during the last two decades: Kahn [2] identifies trading expressivity for computational tractability, and the unattainability of a single (unifying) formalism for temporal representation and reasoning as the main components of the time problem. Cooper [3] describes the proliferation of model variables that comes along with the temporal expansion of atemporal models and the subsequent inference intractability. He discusses three specific possibilities for coping with the problem, without giving implementation details. While these authors are primarily concerned with MDSSs, logicians and AI theoreticians have described several difficulties of modeling time that apply to any intelligent system, such as: obtaining general logical theories of temporal commonsense reasoning, obtaining general temporal ontologies, lack of formal semantics for major temporal logics, and (allegedly) formalism-independent problems (i.e. the frame problem, the qualification problem, the ramification problem, and the extended prediction problem) [4,5].

2.2. *Explicit vs implicit time modeling*

An important dichotomy in modeling time for MDSS is the one between explicit and implicit temporal representation and reasoning. The majority of MDSSs adopt the implicit modeling method. We define implicit temporal representation and reasoning as the incorporation of temporal associations, patterns, trends, and so on, in propositional statements that are treated by the MDSS inference mechanism in exactly the same manner as atemporal ones. In [1] we analyze the types of temporal reasoning found in Quick Medical Reference (QMR), a well known MDSS designed to help as a diagnostic aid in the domain of general internal medicine. QMR is a characteristic example of a MDSS employing implicit time modeling. Examples of QMR implicit-time propositions include: "history of a disease prior to current admission", "history of a finding prior to current admission", and "history of exposure to a risk factor prior to current admission".

We define explicit temporal representation and reasoning as modeling and subsequent problem-solving that incorporates three elements:

1. A model of time, in which time itself consists of primitive entities that cannot be conceptually decomposed further and have specific properties (they might be points, intervals, "moments", or more abstract entities). Time is represented as continuous or discrete, circular or linear, bounded or not.
2. An association (mapping) of all constructs (objects, concepts, properties, relations, etc.) in the MDSS with the model of time, so that everything occurs in time (i.e. in some temporal context), and we can represent and reason with the temporal aspects of our objects and their relations to infer interesting conclusions.

3. A set of inference procedures that operate on the first two elements, so that we can infer conclusions about problem instances.

As an example of a (very simple) explicit temporal representation and reasoning model, consider the proportional hazards model of Cox [6]. In terms of the three components we described above, the model is composed of: (i) a finite, unidirectional model of time; (ii) a one-to-one association of hazard rates (i.e. typically chances of dying within a time period) with the time points H_t; and (iii) an inference rule (i.e. mathematical function) that returns the hazard rate which applies in each time period, given a constant H_0, the values of n independent parameters (i.e. covariates) Cov_1 to Cov_n, and their corresponding weights (i.e. coefficients) b_1 to b_n:

$$H_t = H_0 \times e^{b_1 \times Cov_1 + \ldots + b_n \times Cov_n}.$$

2.3. *Temporally detailed vs abstracted "views"*

Reasoning about a dynamic process (e.g. a time-evolving disease, a therapeutic protocol and corresponding patient–state measurements, or a homeostatic mechanism involving feedback loops) can be seen from two fundamental extreme viewpoints. The first one is the fully temporally detailed view in which we record and/or reason about domain features (variables) at multiple times, while the second one is the fully abstracted view in which we record and/or reason about domain variables without explicitly taking into consideration their temporal evolution.

As an example, consider Fig. 1, where a disease (peptic ulcer) is causing a particular finding, GI bleeding (*GIB*). We use here the graphical language of belief networks to express probabilistic dependencies and independencies among propositional variables [7]. More specifically, nodes represent the variables, and arcs represent dependencies between each variable *X* and its "parent" variables (i.e. the nodes that have an arc going to *X*). Causal belief networks are special cases of belief networks in which arcs denote causal influences. The figure depicts two versions of the model, corresponding to the two views. In the detailed view (Fig. 1a) we represent and reason about the values of ulcer and GI bleeding at all times in our "time horizon" (i.e. the time-span of interest). In this example, we consider a time horizon of 3 yr, examined at a granularity of 1 yr. We also represent and reason about how the disease and the finding at any time can influence other instances of the disease and finding. In the abstracted view (Fig. 1b), we reason about some of the values of prior *ulcer* only (e.g. whether ulcer was present at a time prior to the first observation of *GIB*). We note that the fully abstracted view also offers an additional perspective, one of reasoning about meta-values obtained after processing the original time-specific values of the variables of interest. In our example we reason about entities that are secondary to (i.e.

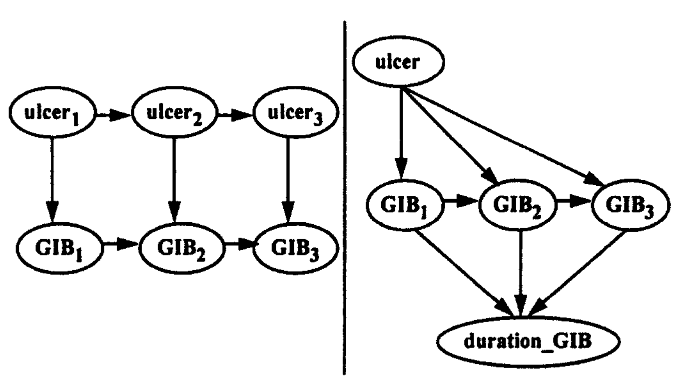


Fig. 1. Example of a temporally detailed view (left Fig. 1a) vs an abstracted view (right Fig. 1b) of the same set of variables. *Ulcer* (prior to time 1) replaces the time-stamped $ulcer_i$ variable, and a temporal abstraction *duration_GIB* (i.e. duration of GI bleeding) is introduced based on values of GIB_i (i.e. time-stamped GI bleeding).

have to be extracted from) the process, as for example the total amount of time that GI bleeding has existed (*duration_GIB*).

2.4. *Operational definitions*

2.4.1. *Model, formalism, query, problem, answer and model worth.* Consider a model M to be a well-defined representation of a problem domain P, expressed in some formal language L. The language L is referred in AI literature as a "knowledge representation" or "formalism" (we prefer the later term, for this paper, since it is more neutral with respect to the semantic interpretation of the model's parameters and other internal constructs). Let V be a finite set of random variables that correspond to the features of the problem domain that we care about (i.e. the essential features of the domain which we include in M). We define a question q and evidence e, to be value assignments to subsets of V, such that e corresponds to the conditioning (available or hypothesized) information we have about the domain, while q to the question for which we wish to obtain an answer given e. We call any pair of the form: $\{e, q\}$ a *query* about P. Let Q_{re} be the set of queries in the domain P that are of interest, and assume that any other query expressible in terms of V is irrelevant.

Let A be a set containing all possible answers to queries within P. We define a problem D to be the set $\{P, V, Q_{re}, A\}$. That is, D corresponds to finding appropriate answers (belonging to A) for all possible queries of interest about a domain. In the example model M of Fig. 1a, a question about the hypothetical domain "ulcer causes GI bleeding" could be: "what is the likelihood of GI bleeding at time 3 being present?". A potential evidence could be "at time 1, ulcer was present". A query then could be "if ulcer was present at time 1, what is the likelihood of GI bleeding being present at time 3?". The potential answers to queries like this one (i.e. set A) could take probability values in [0,1], or in the set {low, medium, high}, and so on. The set V is $\{ulcer_1, ulcer_2, ulcer_3, GIB_1, GIB_2, GIB_3\}$, while the set Q_{re} is $\{ (GIB_3\text{=ON} \mid ulcer_1\text{=ON}) \}$, if we assume no other query is of interest for this example.

We denote the worth of an answer A_i to a query Q_j, by $W(i, j)$. It expresses the value the model user assigns to a specific answer A_i that might be returned by a model as a response to query Q_j (in arbitrary units). We require that the worth for any particular query is finite. For instance, assuming in the ulcer domain that the set A is $\{A_1, A_2, A_3\}$ = {low, medium, high}, then for the first query the worth of the answer "low" will be expressed as $W(1,1)$, which could—for example—have the value "0.5". The complete set of worth assignments will comprise the worth function $W(.)$ for the problem. In the ulcer example the full worth function could be: $\{(W(1,1),.5), (W(2,1),.7), (W(3,1), 1)\}$.

There are several observations we need to make regarding our formalisation. We adopt two assumptions: (a) all evidence is given in the system, and the final answer is obtained in one step; and (b) there is a specific context of use of the system, to which the worth assignments apply, and this context is sufficient to evaluate the system (i.e. every concern outside this context will not change our worth assignments). From a practical perspective, these two assumptions are cumbersome, but for the purposes of a theoretical discussion such as this one, and since the context of use is arbitrarily large, we are justified in using them. A consequence of assumption (b) above, is that we accept that the costs of acquisition, maintenance, and computation are all included in the worth model.

Another observation is that worth is not assigned to specific variable values, but to answers to queries. More importantly our model of decision problem does not indicate how worths are going to be used by a decision maker to select the best answer or model. A second component is required for this purpose, that of a *decision criterion* that selects among alternative query answers or among alternative models. Our definition of TA optimality (section 4.1) is such a criterion. From a decision–theoretic perspective, our worths could conceivably model *expected* utilities, since they could encapsulate multiple utilities for uncertain outcomes or actions. For example, the worth to a diagnostic MDSS's answer to a particular query could be the expected utility for the actions that would be taken by the system user given that answer in the system's domain. This expectation could be further weighted by the expectation that the system will be given this query by the user. Of course a decision–theoretic principle (i.e. maximum expected utility optimization) should be in place in the form of a rule choosing the best model, for our decision model to simulate decision–theoretic reasoning. These expected utility assignments can be viewed either as relative to a baseline expected utility that corresponds to asking no query and receiving no answer, or (with the introduction of minor enhancements to our theoretical formulation having to do with null queries and null answers) as an absolute utility.

Formalizing problems this way can facilitate our reasoning about a wide variety of medical scenarios, such as diagnosis, prognosis, prevention, therapeutic planning, and policy formulation. The answers to queries can be arbitrary, such as probability distributions over possible diagnoses, rank-ordered lists of diseases, desired interventions, and so on. The framework is general enough to capture diverse types of medical reasoning goals such as determining what is wrong with the patient, construction of explanations, suggesting appropriate courses of therapeutic actions, among others. Thus, this approach is flexible enough to capture the decision–analytic framework, is not by design constrained to a decision–theoretic view of the world (i.e. utilities and decision criteria satisfying the axioms of decision theory), and at the same time is simple enough to facilitate the theoretical analysis that follows.

2.4.2 *Time-stamped vs non-stamped variables.* A variable that is associated with specific time entities (e.g. points, intervals) and whose instances are tagged by the corresponding time entities will be defined as a time-stamped variable. Variables $ulcer_1$, and GIB_3 of Fig. 1a are examples of time-stamped variables. A *non-time-stamped* variable is defined to be a variable for which we have no time tag ("stamp"). In other words it is a variable for which we do not know or care to know its exact temporal location. In the example of Fig. 1b, variables *ulcer* and *duration_GIB* are non-stamped variables. Non-stamped variables serve the following important goals:

1. *Model parsimony*: we do not have to replicate variables unnecessarily.
2. *Tractability*: usually, fewer variables lead to more efficient computational models.
3. *Unavailability of explicit temporal knowledge*: we might not know exactly how ulcer causes GI bleeding at every time point, but we know something about how past ulcer influences the chances of having GI bleeding at times 1 to 3.
4. *Building of useful summarizations and patterns*: in our example, the total duration of GI bleeding might be a determining factor of treatment (e.g. prolonged duration that is non-responsive to medication would be a major indication for surgical treatment of the ulcer).

2.4.3. *Temporal abstraction.* A temporal abstraction of a variable set V is any function of V that maps to variable set V' such that: (a) V' contains equal or less information than V, and (b) V has a temporal semantics relative to some specific temporal model for the domain (i.e. there is a temporal interpretation of the values of V).

A temporal abstraction of a model M is any function that maps M to a model M', such that some of the variables V in M have been replaced in M' by temporal abstractions V' of V.

In this paper we will not take a position as to which notion of information loss is applied, since our analysis is insensitive to either a description-length or probabilistic interpretation of information (i.e. given some certainty C about a value vector V_i of V, there is no corresponding value vector V_i' of V' such that it is possible to infer V_i from V_i' with certainty higher than C). Also, while mathematically any abstraction function is non-distinguishable from a temporal one, the temporal semantics of a temporal abstraction may suggest—based on substantive criteria— a temporally meaningful structure of information loss (i.e. a temporal abstraction pattern). For instance, whether the individual values of a variable X over time are normal or not might be insignificant for a predictive task, but whether the values rise or drop over time might be important. So only the trend-related information is preserved in this abstraction. Although the particular structures of information loss will not be the focus of the discussion in the current paper, we will present a list of possibilities that it opens up with respect to efficient temporal abstraction from data and models, in the discussion section.

We now provide some examples of temporal abstraction function types: V' might have a coarser temporal granularity than V (which we call type "a" abstraction), or smaller temporal range (i.e. horizon) than V (type "b"), or smaller dimensionality (i.e. number of values) than V (type "c"), or V' may substitute values of V with patterns such as trends, repetitive structures, etc. (type "d"). Combinations of these types are possible too. Figures 2a, 2b, and 2c illustrate examples of TA types a, b, and c, respectively. Figure 1b illustrates a TA of type d. For several additional examples of temporal abstraction types employed in

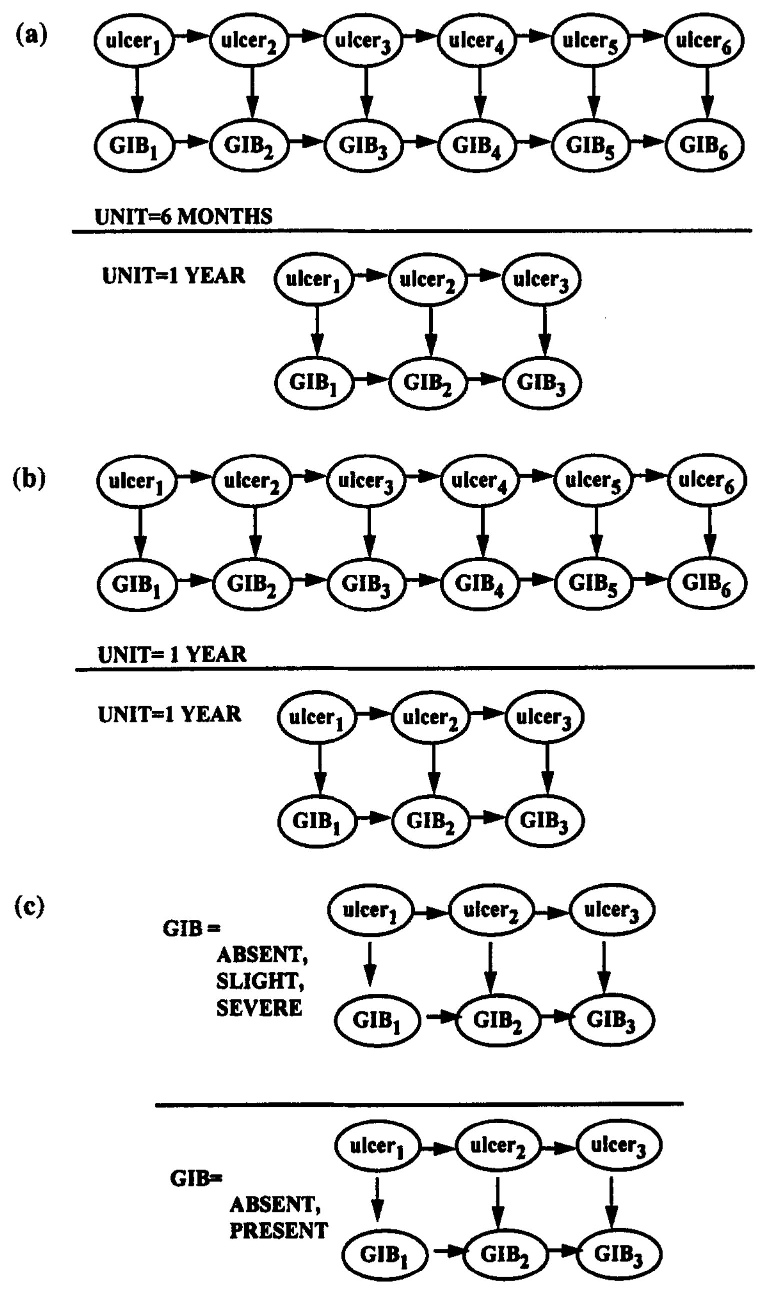


Fig. 2. a. Temporal abstraction type "a". We utilize a coarser granularity (i.e. 1 yr vs 6 months). This reduces the number of indexed variables for the same time horizon. b. Temporal abstraction type "b". We focus on a smaller time horizon, thus dropping several indexed variable instances that fall outside the new horizon. c. Temporal abstraction type "c". We model variable GIB_i at all points in time as taking two instead of three values.

real-life MDSSs, we refer the reader to [1,8].

Having defined the notions of explicit and abstracted representation, we can discuss their relationships. Temporal abstraction (a concept related to the information content of a set of variables recorded over time) does not always imply the lack of temporal explicitness (a concept that has to do with the transparency of the temporal semantics of our model). An explicit temporal model can be abstracted (as in all examples of Fig. 2). The lack of temporal explicitness does not always imply abstraction (we may have no information loss but simply a non-transparent/well-defined model of time). By definition, we will use the terms time-stamped variable and explicit variable interchangeably. Likewise, temporal implicit and non-time-stamped variables are interchangeable.

Empirically, in current MDSSs, implicit variables/models are often highly abstracted. This creates much confusion between the two concepts. We will analyze their relationships more fully in section 3.6.

3. IMPORTANCE OF REPRESENTING TAS

In this section we address the claim that "implicit time modeling leads to MDSS that cannot always provide correct solutions". Our motivation comes from both published reports [9], and initial reactions towards our empirical demonstration that the temporal complexity of real-life cases (and by implication the complexity of the temporal reasoning required) did not induce significant performance penalties in QMR (a pure implicit-time MDSS) relative to cases of smaller temporal complexity [1]. We will provide theoretical and empirical arguments that support the claim that *implicitness does not necessarily imply non-correctness*. Although abstraction implies by definition information loss and thus correctness degradation (assuming the full information is necessary for correctness), we will show that TAs can help in developing MDSS by improving knowledge acquisition and tractability, while maintaining correctness (in circumscribed query contexts). For this purpose we use an example from the domain of endocrinology to present specific trade-offs that result from the use of abstracted variables. The example is simple and serves only for demonstration purposes. The principles revealed, however, have general applicability, regardless of model size and complexity.

The model captures some basic regulatory features of thyroid function. The thyroid gland produces the main thyroid hormone T4 when stimulated by the pituitary hormone TSH. TSH production is suppressed by high levels of T4, and stimulated by the hormone TRH. A functional thyroid adenoma (FTA) is a benign tumor that produces T4 independently of TSH. This in effect disrupts the feedback loop between TSH and T4, resulting in high concentrations of T4 and low TSH. A particular cause for thyroid adenoma is exposure to X-radiation in the past. The abnormality in the T4 regulation can be detected by a TRH stimulation test that involves measuring TSH levels over a period of time after administering TRH to the patient. In a normal person the observed pattern is bell-shaped, while in the presence of FTA, it is flat and closer to 0 [10]. Ideally we would like to formulate queries of the type "Given that the values of TSH at times 1 to n is known, what is the probability of a FTA in the patient?"

All the example models in this section are represented in a time-modeling formalism we call modifiable temporal belief networks (MTBNs), which is a temporal and structural extension of ordinary belief networks. The following constitutes a brief description of MTBNs: MTBNs have three types of variables: ordinary observable variables which are represented by nodes (e.g. node A in Fig. 3), arc variables corresponding to causal associations among variables of any type as represented by arcs (e.g. arc from A to B in Fig. 3), and time-lag variables corresponding to the time delay between cause and effect as represented by squares on arcs (e.g. lag node L in Fig. 3). The dependencies denoted by arcs are all temporal since they have a temporal location and time lags relative to the MTBN model of time. Some arcs have instantaneous effects, while abstract variables have implied locations and time lags (i.e. they are directly or indirectly constrained by their association to time-stamped variables). MTBNs have two forms: a "condensed" form used to define a model (Fig. 4, left), and a "deployed" form, which is the condensed form with variables

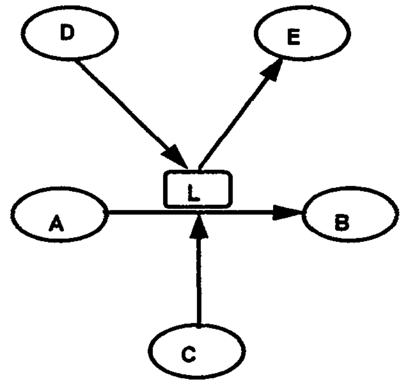
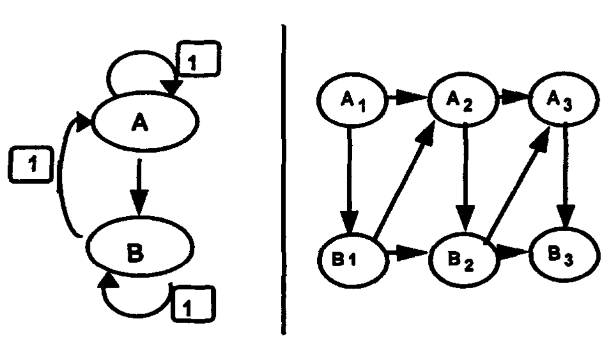


Fig. 3. The three possible types of variables in an MTBN. Ordinary variables (e.g. A, B. These can be indexed or abstracted), arc variables (e.g. A->B. They take one of two values: "activated" or "deactivated"), and time-lag variables (e.g. L. They take non-negative integer values).

replicated over time as needed for inference (Fig. 4, right). The dynamic causal structure and condensed representation permit parsimonious descriptions of models. Compare for instance the condensed MTBN model of Fig. 5 (left) with its BN equivalent (right).

MTBNs allow the coexistence of variables of different temporal detail in the same model. Figure 6 gives an example of such a mixed set of variables: variables A_i, and B_i are modeled at all three time points of interest, variables C and D are recorded only once, variables B_{st}, and B_{end}, indicate abstracted properties of B_i, and B_{dur} is a variable built from B_{st} and B_{end} (and thus is even more abstracted). In the deployed figures of MTBNs, we follow the convention that in time-stamped variables, time increases from left to right, so that leftmost instances of time-stamped variables precede in time the rightmost ones. More details about MTBNs, and other temporal applications and extensions of belief networks, can be found in [11–16]. We now return to modeling the TRH stimulation domain example. A first approach towards modeling this problem is shown in Fig. 7. Here we model all variables as temporally explicit (i.e. indexed). Figure 7 depicts the condensed form of the MTBN model, so it is understood that all variables are replicated and stamped in the deployed form.

Another approach, shown in Fig. 8, involves abstracting X-radiation (variable X), to take values present or absent in the past, and FTA (variable D), to take values present/absent now, and replacing variables *TRH*, *TSH*, and *T4* by an abstraction variable *TRH*



(Condensed form) (Deployed form)

Fig. 4. Condensed (left) and deployed (right) forms of an MTBN. The condensed form is user-specified and facilitates definition, sharing, and presentation. The deployed form is primarily used for inference and is automatically created at inference time. Note that time lags are represented as constants or variables in squares attached to an arc (e.g. B_i causes A_i with a time lag of 1 time unit). The time lags can take values of 0 in which case the corresponding arc denotes instantaneous influence (e.g. A_i causing B_i). Lags can change as the causal process unfolds over time (see text for details).

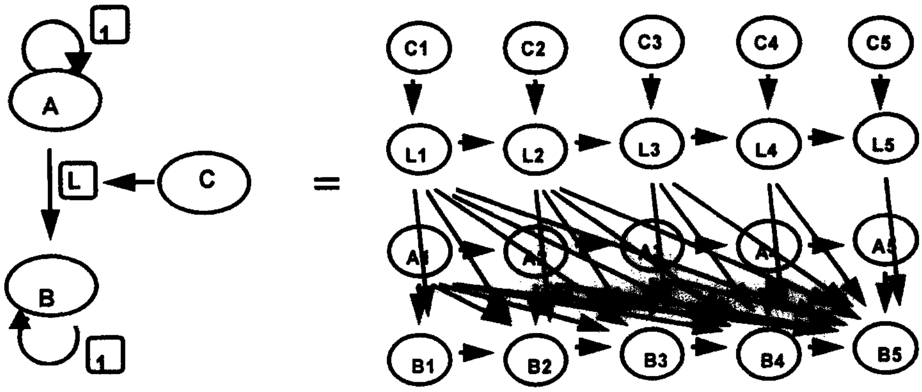


Fig. 5. An example of MTBN representational clarity. The MTBN left is equivalent to the BN right. The MTBN is easier to understand because it uses single variables for repeated entities (arcs, ordinary variables), and also because it can express explicitly and concisely that certain temporal relations are dependent on modeled factors (e.g. the time lag in the causal influence between *A* and *B* is determined by *C*). Such relationships are embedded in a forest of arcs in a standard temporal BN.

STIMULATION RESPONSE PATTERN (taking as values: normal, or abnormal). No copies of these variables are needed since each abstracted variable is designed to encode a limited amount of the information of its time-stamped counterparts. Figure 8 illustrates the method used in most current MDSSs (implicit/abstrcated time modeling). We next compare these two models in order to illustrate qualitative relationships of TAs with several engineering and epistemological parameters. We keep in mind that in our example (as in most real-life MDSSs) the concepts of implicit/abstracted can be used practically interchangeably (for more explanations see section 3.6), as can the concepts of explicit/detailed.

3.1. *TAs and correctness*

Assume that an oracle makes a worth assignment (i.e. characterizes) the possible answers to every relevant query as "correct" or "incorrect" (i.e. "true" or "false"). "Correctness" is a measure of the ability of an algorithm that implements the model to output the correct answer for all queries of interest. In our discussion we will define correctness as the percentage of correct answers over all relevant queries. Correctness is a special class of worth function that facilitates discussing the question of whether implicit modeling entails

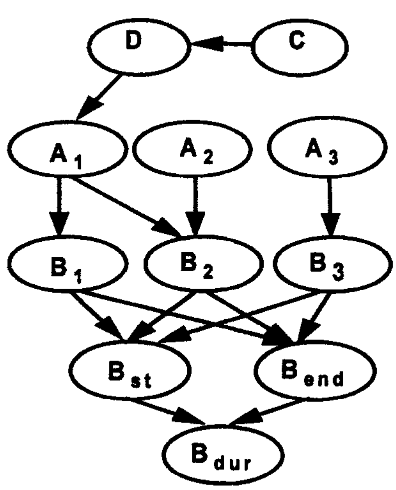


Fig. 6. Different levels of temporal detail in the same MTBN model. Variables A_i and B_i are modeled at each of three time points. Variables *C* and *D* have unspecified temporal locations and are only constrained to precede their effects. Variables B_{st}, B_{end}, B_{dur} are abstractions built on the basis of the values of B_i. They are similarly constrained to occur after their causes (i.e. parent variables).

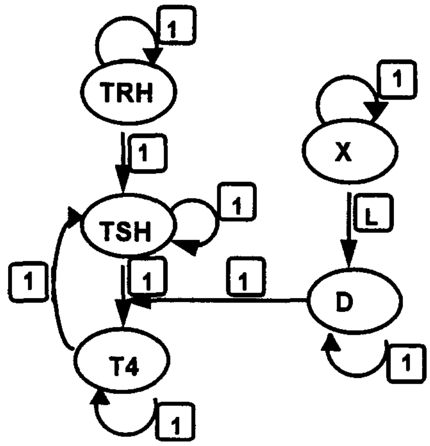


Fig. 7. Fully explicit model for TRH example (see text).

decrease in the accuracy of the MDSS. Correctness overlooks the costs associated to building and maintaining an MDSS by focusing only on the validity of its output. First we note that we can trivially derive (e.g. by using belief networks instead of MTBNs) an implicit model that would contain exactly the same information (and thus give the same answers) as the explicit model of Fig. 7. Therefore implicitness does not necessarily decrease correctness. A more important observation however is that both the model of Fig. 7 and the one of Fig. 8 are equally capable of providing correct answers to queries expressible in the corresponding models. To verify this claim consider that the abstracted model can be derived from the explicit model by marginalization (after new variables were introduced by means of abstraction). This example illustrates that temporal abstraction need not directly affect the correctness of the model's answers. Is there an indirect relationship between abstraction and correctness? The answer is yes and the following section explains why.

3.2. *TAs and query completeness*

While we were able to obtain correct answers using temporally detailed and abstract models, the same is not true when it comes to the number of queries that can be answered to by the two models ("query completeness" of the models). In our discussion we will define query completeness as the percentage of queries that can be answered by the model relative to all relevant queries. The detailed model can answer an astronomical number of queries (most of which are of no clinical interest), whereas the abstracted model can answer only a few queries (possibly excluding many queries of clinical importance). In our example, the explicit model can answer the (useful) query: "given that the values of TSH

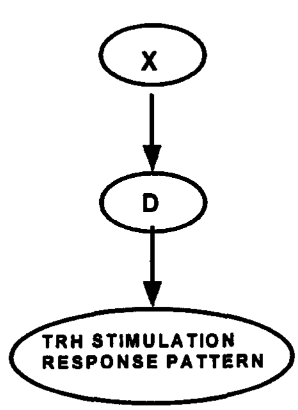


Fig. 8. Fully implicit model for TRH example (see text).

on 6 June 1994 at 3:03, 3:08, 3:13, 3:18, 3:23 were low, medium, high, medium, low, what is the probability of a FTA in the patient on 6 June 1994 at 3:00?" By the same token it can answer the (useless) query: "given that the value of TSH on 3 March 1981 at 5:20, was medium, what is the probability of a FTA in the patient on 31 December 2001 at 12:00?". The abstracted model on the other hand can answer (useful) queries like: "Given that we have an abnormal TRH stimulation test pattern now, what is the probability that the patient has FTA now?", but cannot answer (potentially useful) queries like: "what is the probability that the patient was exposed to X-radiation prior to 10 years ago, given that we have an abnormal TRH stimulation test pattern now?".

Temporal abstraction reduces the query completeness (of the abstracted model compared to the original one) unless we decide to heuristically map queries that are not expressible in the abstracted variable set to queries that are. In this case we introduce the possibility for erroneous answers (since by mapping a non-expressible query to an expressible one we make the potentially flawed assumption that the evidence or question of the former is equivalent to the evidence or question of the latter) and thus affect correctness indirectly. But if we restrict the use of the model to the set of queries that are expressible in the abstracted model, no loss in correctness need occur. We note that a guaranteed smooth transition from the fully explicit/detailed model to the fully implicit/abstracted one is not a property shared by all knowledge representation and reasoning formalisms. An example of this is various *ad hoc* uncertainty calculi employed in MDSSs.

Figure 9 summarizes the relationship between TA and correctness when query completeness is maximized. We note that explicit queries need to be mapped to abstract queries in order for us to derive a measure of correctness of the abstract system. Although there might be an initial region (a) for which correctness is maintained at optimal levels, as TA increases, correctness in general will decrease. The exact form of the curve will be different among alternative problem domains, and regions a and c will be present in some domains only.

3.3. *TAs and model usage ease*

Both models of Figs 7 and 8 require that a mechanism external to the model will examine raw time-stamped data and will assign appropriate values to the variables in the model. The explicit model by virtue of using time-stamped variables, can facilitate automatic input of the necessary information and thus be used in non-supervised fashion (e.g. embedded in a reminding application), relative to the (heavily abstracted and thus) implicit model. This is because the implicit model requires an additional elaborate abstraction layer between the data and its inputs. This abstraction requirement is so severe that in most cases it implies

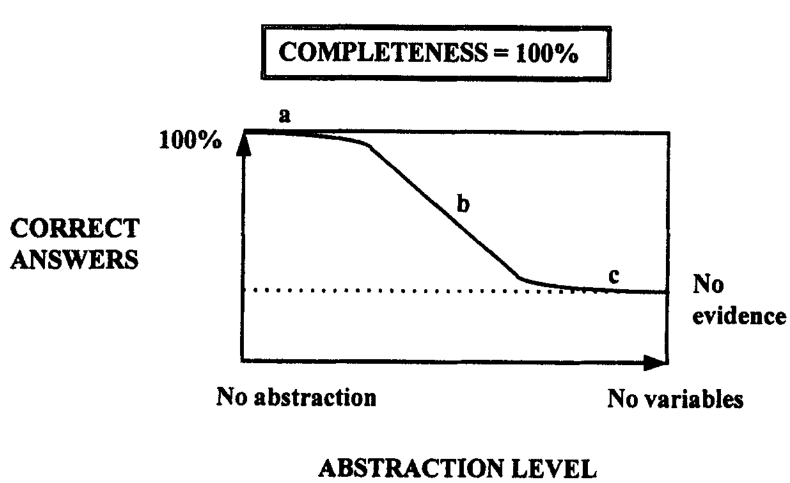


Fig. 9. Generalized relationship between abstraction and correctness when query completeness is required to be maximal.

that implicit/abstracted models will need a *human abstractor* to create the input to the MDSS.

3.4. *TAs and knowledge/data availability*

The availability of domain knowledge and data can be one of the most important determinants of the success or failure of a MDSS. Whether temporally explicit or implicit knowledge will be predominantly available for a specific problem domain is strictly dependent on the domain. For example, detailed epidemiological knowledge about coronary heart disease may allow the development of Cox proportional hazards models for predicting mortality at any given time (explicit model), but lack of corresponding knowledge for HIV-infected populations in non-industrial regions of the world may necessitate the adoption of an implicit model linking time of symptomatic disease after the infection to patient features.

3.5. *TAs and tractability*

Generally, abstracted models involve far fewer variables than explicit ones, and thus are expected to be more tractable computationally. In our example, the explicit model with a uniform temporal granularity of 1 min, and a time-span of 15 yr contains an astronomical number of variables and is completely intractable. In contrast, the abstracted model contains three variables and is clearly tractable. It might be the case that we can exploit domain independencies and construct temporally explicit, but still tractable models. The feasibility of such an approach will always be dependent on the domain. For example, we cannot simplify a model for which all variable instances are dependent on the variable instances of the past, using a hierarchical model of temporal dependence (i.e. with variables forming a tree with each level corresponding to different granularity levels). This is the case, for example, in the hypothetical model of Fig. 1a. In domains where such simplifications apply, the structure of temporal association will facilitate tractability.

3.6. *TAs and temporal semantic clarity*

The more we abstract a model, especially at the individual variable level, the more temporal clarity we lose. Eventually, the temporal semantics are encapsulated in the variables values and thus lose their special representational status. This is a major reason why "abstract" and "implicit" are so highly correlated (and confused). In our endocrinology example, variable D_i of the explicit model indicates the values of D at time point i (note that only the condensed forms of MTBN models are shown, but it is implied that all non-abstract variables, such as D in Fig. 6, will be indexed in the deployed form). When we abstract to the implicit model, the value of D means "value of D *now*". There is no indication as to the temporal meaning of this proposition other than this descriptive label. As far as the formalism is concerned, the value of D could as well be the value at time t (for an arbitrary t). This arrangement (no special semantic status to time) has been the target for criticism of logicians towards non-temporal logics [4,5]. It is important to note that the argument applies to all formalisms, however. We also note that although implicit models necessarily do not provide temporal semantics, there are cases where temporally explicit models are inconsistent, as can be determined based on their temporal semantics. Figure 10 illustrates this using an example of a causal model that is invalid (assuming standard notions of causation as applied to medicine). In particular, the causal arc to variable A2 from A3 imply that A is causing itself in the past. If we reverse the direction of time, then A1 is causing itself (A2) in the past. So there is no time directionality that preserves the notion of asymmetry in causation.

3.7. *Hybrid-implicitness/abstraction formalisms for time modeling*

The above comparisons indicate specific tradeoffs between explicit and abstracted time modeling. Ideally we would like to have the benefits of both while avoiding their individual

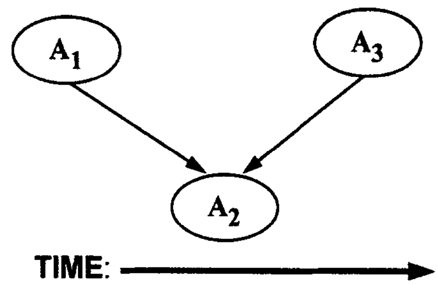


Fig. 10. A simple temporal model that is obviously causally inconsistent (A_3 causes itself back in time). Despite that, any BN implementation will allow such a definition (and inference), whereas in the MTBN model this error will be detected. The MTBN formalism enforces temporal causal consistency because it supports explicit temporal semantics. Much more subtle errors of similar nature can arise when we have mixtures of time-stamped and abstracted variables.

shortcomings. *Hybrid-abstraction formalisms* are formalisms that allow us to have part of the model in time-explicit/less-abstracted and part of it in time-implicit/more-abstracted form while maintaining consistency and semantic clarity. As an example, consider the model in Fig. 11. Here the model remains the same, except for the replacement of the indexed variables X_i and D_i with their abstracted counterparts X and D, and the recording of the rest of the variables only for a few time points corresponding to the latest TRH-stimulation test. The model immediately becomes tractable and still it does not need an external pattern recognizer to assess the abnormality of the pattern of values of TSH after TRH stimulation.

4. IMPORTANCE OF DEVELOPING OPTIMAL ABSTRACTIONS

In the previous sections we provided arguments in favor of the need to utilize both explicit and abstracted views of the modeled processes in our models to achieve satisfactory efficiency and query completeness and to facilitate knowledge/data acquisition. In general, there is an infinite number of temporal abstractions which we can consider. The following example demonstrates that indeed this is the case: assume that the presence of a disease D at a particular time point (D_i) is determined by the values of a finding F at the same time point (F_i) as well as at a previous time point (F_{i-1}). D takes values in the set {present, absent}, whereas F_i, F_{i-1} in [0,100]. Also assume the probability of D_i being present is given by: $p(D_i) = c + b_1 \times F_{i-1} + b_2 \times F_i$, where c, b_1, b_2 are appropriate constants. Assume the queries of interest (i.e. Q_{re}) are "what is the probability of D_i being present given that $F_i = x$ and $F_{i-1} = y$?" for various values x, y. Finally let the worth function for this problem be 1 if the probability of D_i is correct, and 0 otherwise.

Consider the following abstraction TA_1 over F_{i-1}: $TA_1 = F_{i-1}$ if $F_{i-1} > z$, or 1 otherwise

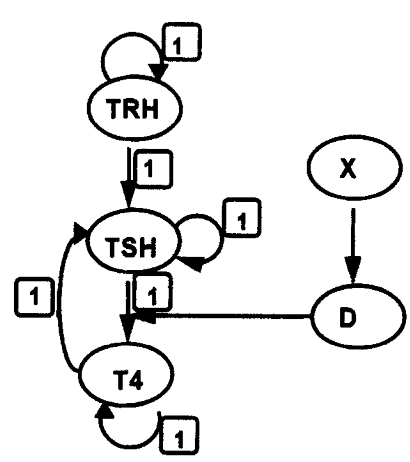


Fig. 11. A hybrid-abstraction model for the TRH example (see text).

Table 1. Probability of disease D_i being present, given joint values of findings F_i, F_{i-1} in hypothetical medical domain

Joint instantiation #	F_{i-1}	F_i	$p(D_i\text{=present} \mid F_i, F_{i-1})$
1	H	H	0.8
2	H	M	0.4
3	H	L	0.8
4	M	H	0.8
5	M	M	0.4
6	M	L	0.8
7	L	H	0.5
8	L	M	0.01
9	L	L	0.5

(with z a real variable taking values in [0,100]). If z is a parameter set by the developer of the model, TA_1 is a *deterministic abstraction.* If z is a random variable with a particular probability distribution, TA_1 is a *nondeterministic abstraction.* Clearly in both cases since z can take any one of an infinite number of values, there is an infinite number of instances of TA_1. Also TA_1 is indeed a TA, since for all F_{i-1} that are not equal to 1, and are not greater than z, they are mapped to 1, thus (in general) losing information about F_{i-1}. We will introduce TA optimality using the same example modified to allow for easier computations. Assume that in the modified example, F_{i-1}, F_i are ternary taking values in: $\{L, M, H\}$ (i.e. low, medium, high). Also assume that the probability distribution of D_i given joint values of F_i, F_{i-1} is given by Table 1. Suppose further that in this example the queries of interest (query context) are $Q_{re}=\{(p(D_i \mid F_{i-1}= H, F_i=H), (p(D_i \mid F_{i-1}=H, F_i=M)\}$. Abstracting the model so that F_{i-1} is completely eliminated still yields the same worth for the queries of interest (since in both cases the value of F_{i-1} was the same, thus it did not help in discriminating further the conditional probability of D_i, than does just knowing F_i). But if we abstract F_i by merging the values H and M, then assuming equal priors for instantiations #1,2 shown in Table 1, we do not get answers of worth 1 to either one of the two queries of interest, since the answer to both queries is 0.6 given the new evidence that does distinguish between the values H and M for F_i. The correct answers have probabilities 0.8 and 0.4, however. Since this abstraction derives incorrect probabilities (relative to the gold standard of Table 1), our assigned worth function will lead to the abstraction being sub-optimal.

4.1. *TA optimality*

Now we introduce by example our working definition of TA optimality. Consider the example of Table 1 with Q_{re} being the set of all possible queries given by that table. A model M is called contextually temporally optimal (CTA-optimal) relative to Q_{re} if and only if the following holds: there exists no model M' that contains the same or less abstracted variables than M, such that for some queries M' gives an answer with a higher worth than any of the corresponding answers given by M. If, moreover, there is no model M'' that is more abstracted than M, such that M'' has the same worth for all answers as M, then M will be called baseline CTA-optimal (BCTA-optimal) relative to Q_{re}. In our example, a model M_1 that replaces values H and M of F_{i-1} with a new value HM can be constructed so that it will give the same answers as the original model, and thus would be CTA-optimal. Similarly M_1' that replaces values H and M of F_{i-1} with a new value HM, and values H and L of F_i with HL, can be constructed so that it will give the same answers as the original model. There is no further abstraction of M_1' however that preserves the correct answers as given by Table 1, and thus M_1' is baseline CTA-optimal. The Appendix contains a more formal definition of CTA-optimality.

Since a TA refers to substituting variables with more abstracted versions (including eliminating the variable), TA-optimality as we define it here corresponds precisely to the problem of proper variable selection for inclusion in the domain model. What this working definition says is that for any set of variables used to model a problem domain, there is a point at which having further temporal detail about the variables is not going to lead to better models (with respect to a modeling formalism and measure of query worth). At this point, we have a TA optimal model. The qualification of "contextual" optimality simply indicates that the optimality applies for a subset of domain queries of interest (the idea being that if some queries are of no interest, a model that does not encompass the useless queries is still good). The present definition of TA-optimality reflects dominance of the worth of a particular abstraction. This definition is biased towards worth functions that reflect the information content of a set of variables with respect to a discrimination task (e.g. diagnosis, prediction), and do not take into account the costs/benefits of developing and using particular models (otherwise a more abstract and less accurate model might give better answers than a less abstract one). It is designed so for simplicity, which facilitates a formal analysis of its properties and implications. Another very important property of TA-optimality defined this way, is that it is decomposable in that it describes a boundary beyond which no further improvement is possible, and thus, can potentially facilitate the development of greedy search algorithms for developing optimal TAs. Developing definitions of TA-optimality that take into account MDSS development and costs is an open research problem that is not dealt with in the present paper.

4.2. *A study of interformalism TA optimality*

Having a well-specified notion of TA optimality, we can now ask whether the specific TAs employed in a specific model are optimal or not. Our motivation comes from the observation that human experts often seem to develop abstractions in an *ad hoc* manner, and thus, these abstractions might be sub-optimal as explained in the previous subsection. Usually such abstractions are chosen on grounds of simplicity or convenience, without proof of optimality. They then find their way into research designs, published studies/textbooks and MDSS knowledge bases. For instance, several temporal abstraction/summarization MDSSs have been developed during the last two decades [8,17,18]. They advanced substantially the state of the art in detecting, constructing, and presenting TAs, but there was relatively less attention paid to the optimality of the produced TAs. It was assumed that the TAs were useful and that the domain expert could indicate which abstractions are good.

The importance of the question of whether the employed TAs are optimal, became apparent in our study of the QMR system. If they are not, maybe they could be replaced by better ones. If they are, then perhaps they could be used as building blocks for other MDSS that employ implicit time modeling as well. So, the emerging question was whether it is possible to determine the optimality of QMR's TAs and how. A related (and more ambitious) question was whether it is possible to determine TA optimality in a formalism-independent manner.

One first approach might be to build a model in a particular formalism in extreme temporal detail and develop an optimal abstract model from it (assuming that we could circumvent the complexity of the construction task—see the next section). Another approach would be to directly prove (i.e. without considering all possible abstractions) that a particular abstraction was optimal or not. We will refer to the latter task as "identification" or "characterization" and to the former one as "development".

Ideally we would like to approach characterization at the most general level (i.e. in a domain and formalism-independent manner, so that our conclusions are not a byproduct of the representation language or domain particulars). To do so, we observe that the last two decades of research in AI have led to representations and corresponding methods that have the guaranteed ability to capture precisely certain function classes. For instance, belief networks can capture any joint probability distribution over an arbitrary set of discrete variables. The same applies to decision trees, certain classes of Neural Networks, etc. This realization is extremely useful because it allows us to focus on capturing some information content that is pertinent to a mapping (i.e. function) rather than on the mechanics of how the content is encoded. Our framework for formalizing a decision problem ties in well with such a function-capturing view of problem-solving. So we give the label in–out–correct (IOC) to this class of algorithms and formalisms and define it as follows: an IOC formalism is a computational framework (a representation and a set of operations) that guarantees the existence of IOC algorithms, that is, algorithms that can capture exactly a given function class (thus the name: they always give the correct output for a specific input). A more formal definition of IO-correctness can be found in the Appendix. In the following, in order to facilitate analysis, we will not consider the relative costs of developing a model in one formalism versus another. We note that IOC formalisms by implying the existence of algorithms implicitly invoke all known constraints on computability of functions. Thus we could be talking about Turing Machines (or equivalents) instead. However, while it is true that all computability results apply to IOC formalisms, at an MDSS engineering level there are several results (e.g. the ones about Neural Networks and belief networks mentioned earlier) that are meaningful at the level of knowledge representation and reasoning rather than at the level of universal computation. As such, IO-correctness separates and characterizes knowledge representation and reasoning formalisms that offer certain important engineering properties. We can now explain in general terms our previous claim that abstraction does not directly affect correctness.

Given an IOC formalism, we could construct CTA-optimal systems. This is so because for each abstracted set of variables we have a corresponding set of queries expressible in terms of those variables, and we can map them to the correct answers (interpreted here as the answers with maximum worth). Since the mapping involves some of the newly-abstracted variables (as well as the non-abstracted ones), this set of abstracted variables immediately suggests a new query context, and constrains the model TA-optimality within this query context. Clearly, if this context excludes queries previously considered useful, the abstracted model is no longer optimal with respect to the original context (or as stated in section 3.2 we lose query completeness, but not in correctness).

We demonstrate this concept concretely using the example from Table 1. Assume the query context is: $Q_{re} = \{ (p(D_i \mid F_{i-1}=M, F_i=H), (p(D_i \mid F_{i-1}=M, F_i=M), (p(D_i \mid F_{i-1}=M, F_i=L), (p(D_i \mid F_{i-1}=L, F_i=H), (p(D_i \mid F_{i-1}=L, F_i=M)\}$. If we replace F_{i-1} values L and M with a single value LM, then we can no longer answer to all queries in Q_{re} correctly. But we can answer all queries in $Q_{re}'=\{(p(D_i \mid F_{i-1}=LM, F_i=H), (p(D_i \mid F_{i-1}=LM, F_i=M)\}$ correctly. Thus, this abstracted model can *sometimes* get correct answers to the original query context, and *always* gets correct answers to a new circumscribed context. Theorem 1, in the Appendix, presents this result in formal terms, while Theorem 2 establishes that belief networks and MTBNs are IOC formalisms.

Now we use the concept of IO-correctness to show by example how we can study a formalism using another formalism, with respect to TA-optimality. In particular, we will first show that we can conclude that a model is not TA-optimal by examining a model involving the same variables, but which is expressed in some other formalism. The idea is to be able to study any model expressed in any particular formalism by utilizing a single formalism (thus enabling formalism-independent study of TA-optimality).

The example to follow is a demonstration that we can detect the non-optimality of TAs expressed in the formalism employed by QMR by showing that these same abstractions are not optimal in an MTBN model. Theorem 3 (part a) in the Appendix proves the validity of the method in general. Assume we implement the problem domain of Table 1 in MTBN form and in the representation of QMR. Assume also that $Q_{re}=\{ (p(D_i \mid F_{i-1}=M, F_i=H), (p(D_i \mid F_{i-1}=M, F_i=M), (p(D_i \mid F_{i-1}=M, F_i=L), (p(D_i \mid F_{i-1}=L, F_i=H), (p(D_i \mid F_{i-1}=L, F_i=M)\}$. Suppose that in QMR terms we get one of the following three outputs: $D_{present}>D_{absent}$, or $D_{absent}>D_{present}$, or $D_{present}=D_{absent}$ (i.e. an ordered list of diagnostic alternatives). Clearly any such ordered lists can be simulated by an MTBN as a probabilistic assignment to values of D. Suppose also that the worths associated with the problem are specific to this list (rather than the exact probabilities associated with each diagnostic alternative). In particular, if $p(D_i=\text{present})>p(D_i=\text{absent})$ according to the table gold standard, then the output $D_{absent}>D_{present}$ has a worth of 1; any other output will be assigned a utility of 0. A similar worth assignment is given for the cases where $p(D_i=$present)$<p(D_i=\text{absent})$, and $p(D_i=\text{present}) = p(D_i=\text{absent})$. If we determine that a particular abstraction is not optimal in the MTBN model, then it will not be CTA-optimal in the QMR model either. For example, if we replace F_i values H and M with one value HM, then we get an answer of $D_{\text{present}}>D_{\text{absent}}$ for the first two queries and the reverse for the remaining ones. Thus two out of four queries get an answer with worth 0 in the abstracted model (vs 1 in the non-abstracted one). QMR will not be able to output a correct answer given the abstracted version of F_i, because if it did then MTBNs would not be IOC (put in other terms we could use the output to construct an TA-optimal MTBN model involving F_i). This methodology for studying TA-optimality can be extended to establish (not only reject) it, in case that both formalisms are IOC. Theorem 3 (part b) in the Appendix proves it for the general case.

In terms of the example of Table 1, if we consider a $Q_{re} = \{ (p(D_i \mid F_{i-1}=M, F_i=H), (p(D_i \mid F_{i-1}=M, F_i=M), (p(D_i \mid F_{i-1}=M, F_i=L), (p(D_i \mid F_{i-1}=L, F_i=H), (p(D_i \mid F_{i-1}=L, F_i=M)\}$, and an abstraction replacing F_{i-1} values H and M with a single value HM, and examine an MTBN model MF_1 utilizing it, we can establish that the same variables would lead to an TA-optimal model MF_2 expressed as a BN. This is because the input and answers of the MTBN model can be used to construct a BN model that is TA-optimal (since BN are IOC—see Theorem 2 in the Appendix).

It seems unlikely to us that in general (i.e. when we do not have some optimal TAs expressed in some formalism as in the previous examples) it is possible to determine TA optimality without actually developing optimal TAs. One alternative is to perform empirical evaluations such as the one reported in [1]. Another possibility is to find restricted classes of worth functions W(.) for which criteria for determining optimality without reference to an existing TA-optimal model exist. Finally, we could develop optimal TAs by searching in the space of all possible TAs. The next section discusses some issues regarding the feasibility of such an approach.

4.3. *Constraining the space of possible abstractions*

The search in the space of TAs would look very much like the search in "concept space" employed in machine learning [19]. The search could be incremental, going from higher to lower abstraction levels, according to available computational resources. A fundamental problem, however, is the infiniteness of the possible abstraction functions.

If we restrict our attention to finite discrete variables and deterministic abstraction functions, we will have a finite space to search. Moreover, it can be shown (see Theorem 8 in the Appendix) that these three conditions (finite values, discrete, deterministic) are necessary for a finite space of TAs. Fortunately, using deterministic functions is a workable constraint as evidenced by the types of abstractions used by humans and existing MDSSs [1].

The following example illustrates the idea of TA equivalence based on which the finiteness result is obtained. Consider the model of Table 1, and assume we wish to abstract over variable F_{i-1}. Recall that F_{i-1} takes three values (H, M, L). There are infinite abstraction functions with a codomain consisting of up to two elements. For example consider the (infinite) functions with codomain {1,2}, {2,3}, {3,4}, etc. From those, there are only three non-equivalent disjunctive abstraction function classes (i.e. in which we abstract by grouping together values): for F_{i-1} the first class assigns HM as the first value, and L as the second one, the second function class assigns HL as the first value and M as the second one, and the third abstraction function class assigns LM as the first value and H as the second one. This is because a function that would map H and M to 1 and L to 2 contains the same information as one that maps H and M to 0 and L to 1, and so on. In other words it does not matter what are the specific values (read labels) of each of the two abstracted values, since they correspond to the same information (i.e. to HM, HL, or LM, according to which equivalence TA class was used) regarding F_{i-1}. Table 2 gives the number of ways of abstracting a variable as the number of values of that variable increases, for the first few numbers of values of the variable. Although (as the table illustrates) the space of possible abstractions grows steeply as a function of the number of values, it is

Table 2. Number of all possible non-equivalent TAs as a function of the number of variable values. This is the same as the number of disjunctive and eliminating TAs (see text)

variable values	abstractions
2	5
3	15
4	52
5	203
6	877
7	4140
8	21147
9	115975

(obviously) not infinite (Theorems 4, 5, 6, 7, and 8 in the Appendix).

We note that in Table 2 (but not in the example for simplicity), we not only consider *disjunctive TAs* (i.e. ones in which we join values together to form the values of the TA), but also *value-eliminating TAs* (i.e. ones in which we drop some of the values of the variables to be abstracted). Disjunctive abstractions in combination with eliminating abstractions describe completely the set of non-equivalent abstractions for finite, discrete, deterministic TAs (Theorem 7 in the Appendix).

4.4. *Some semantic and computational considerations*

In section 4.2, we defined IOC formalisms and models considering only their input–output behavior and not the way domain knowledge is represented internally in the model. We now focus on the semantic interpretation of a model's components in order to clarify further our rationale and show that our results are independent of such considerations. First we distinguish between two types of IOC formalisms/models: the first type is described by the definition we presented in section 4.2, and will be referred as *weak IOC* algorithm/ formalism. The second type *(strong IOC)* is defined having the additional requirement that input–output function is not only implemented correctly, but this is done by decomposing it into meaningful and correct components. "Meaningful" components are constructs isomorphic to essential domain characteristics. For example, in medicine they might be causal mechanisms, physiologic processes, anatomical regions, and so on. "Correct" refers to the requirement that not only should these structural components have a meaningful domain interpretation, but this interpretation should also be valid in its functional parameterization (i.e. the model components interrelate in a way similar to the domain components). For example, a causal mechanism indicating that peptic ulcer causes GI bleeding is meaningful (corresponds well with accepted medical knowledge), but in order to be correct it should also be supplemented by the correct strength of association (expressed as conditional probability, heuristic weight, or other formalism-dependent means).

Strong IOC always implies weak IOC. Therefore since we proved the results of section 4.2 using the weak version, their validity is not altered by the semantic considerations brought forward by strong IOC. Also, trivially, weak IOC does not necessarily imply strong IOC. Usually it is desirable to develop MDSSs using strong-IOC formalisms. The reasons for such a preference is ease of knowledge engineering, explanation, knowledge validation, and updating. Developing strong IOC models is feasible when there exists a rich body of theoretical knowledge about the domain of interest. In cases where such a theory is lacking, weak IOC formalisms may be acceptable. Another obstacle in attaining strong-IOC systems is when domain knowledge exists but is not representable in available formalisms (e.g. higher-order logical statements).

As an example of a weak IOC formalism, consider multi-layered feed-forward neural networks trained with back-propagation. They can learn any function, provided enough training instances have been encountered and enough hidden nodes are used [20], but the internal representation does not correspond to domain knowledge (at least not in any well-defined, and straightforward manner). Belief networks on the other hand, can be used as either weak or strong IOC formalisms. The former was shown by Theorem 3. As an example of the latter, consider causal belief networks in pseudo-nondeterministic domains for which we now the causal mechanisms. We note that whereas Neural Networks are strictly weak IOC formalisms, belief networks can be used as either weak or strong, depending on our knowledge of the domain or other MDSS-building considerations.

In general, the distinctions we made are homologous to the ones between "shallow" and "deep" or "associative" and "causal" models in the AI literature [21]. IOC formalisms are special subclasses of the corresponding classes, however, since weak IOC formalisms can be viewed as associative models that are guaranteed to enable the development of correct problem-solving systems while strong IOC ones guarantee the same and maintain a semantically meaningful (domain-specific) structure.

5. CONCLUSIONS

In this paper we provided arguments in favor of the following two assertions with respect to time-modeling for MDSSs. (a) Time formalisms that can represent not only explicit temporal variables but also temporal abstractions in the same model are likely to be an important tool in developing efficient and expressive MDSS, thus giving us a better handle on the "time problem" in MDSS development. We discussed MTBNs, as an example of such a hybrid-abstraction formalism. There is no reason why we should not expect other types of formalisms to adapt to the need to express simultaneously explicit and abstract parts of a model. In particular, temporal probabilistic logics seem to be good candidates for such extensions, since we can conceivably express abstract variables as indexed ones with unknown time locations, as well as having multiple granularity-level associations to our objects. The appropriate formulations, axiomatizations, and implementations of such logical systems, however, remain to be developed. (b) Temporal abstractions are not all equal, with respect to determining useful problem answers. We need to study TA-optimality and develop methods for developing optimal abstractions. We showed that it is feasible to study TA-optimality in a domain and formalism-independent manner, provided we have a TA-optimal model as reference. Unfortunately, the characterization of TAs as optimal or not without such references is difficult to obtain. Although some quite flexible constraints on the TAs considered yield a finite search space, the space is so large that we need to focus on special TA subclasses and methods for efficient development of TAs.

The extensive research done in developing abstraction and summarization problems in medical informatics was done under the assumption that we knew the abstractions. It was assumed that the main problem was how to construct them from patient-specific data with computer-based methods [8,17,18]. Clearly this is only one (admittedly very important) part of the abstraction problem, as shown in the current paper.

Researchers in AI have provided some initial answers to the problem of exploring methods for simplifying (i.e. abstracting) computational models [22–24]. These methods are not time-aware, however, and thus, they do not exploit the structure of time and temporal reasoning. In the same sense, most of the theoretical analysis regarding the optimality of TAs in section 4 of the present paper applies to all abstractions. We believe that as a practical matter, time modeling *forces* the use of abstraction both in humans and AI systems, otherwise the reasoning models become too big and therefore computationally intractable. This is why abstraction plays such a major role in temporal representation and reasoning. In order to solve the time modeling problem in general, we clearly need to exploit more the structure of time and implement this structure utilizing the tool of TAs.

We believe that several exciting research prospects exist in the field of time modeling with respect to the study of TAs. Research challenges that appear worth pursuing include:

1. Developing algorithms and relevant heuristics for searching in the space of abstractions.

2. Exploring the role of learning using abstractions of the available data, to cope with sample size problems.
3. Modifying existing machine learning methods to learn by searching the space of abstractions (to increase the tractability of learning as well as of inference).
4. Expanding explicit temporal representation and reasoning formalisms so that they can represent and reason efficiently with models involving multiple levels of temporal abstraction as in [11,25].
5. Experimentally investigating the benefits of (iv) for knowledge acquisition and computational tractability.
6. Learning the dynamic structure of processes using abstracted data.
7. Using abstractions to study certain conventions of knowledge representation (such as stable feedback loops [26], selection biases occurring in time [27], etc.)
8. Using existing TA ontologies to develop optimal TAs [1,8].
9. Producing empirical evaluations of implicit-time MDSS in many domains.
10. Enhancing the theoretical understanding of TAs using explicit modeling of costs and benefits involved in the MDSSs development process, and having stochastic definitions of optimality (i.e. admitting errors e with probability p).
11. Identifying TA classes for which stronger theoretical results and more efficient search procedures can be attained.

Summarizing our conclusions we can say that temporal abstractions (especially as used in hybrid-abstraction representations) are a potentially valuable tool in developing MDSSs, since they help us achieve a balance of query completeness and tractability, they can help exploit knowledge that is available in various grades of abstraction, and they can lead to models that are easier to verify and specify.

6. SUMMARY

The utilization of the appropriate level of temporal abstraction is an important aspect of time modeling. We discuss some aspects of the relation of temporal abstraction to temporal implicitness and important knowledge engineering parameters such as model correctness, ease of model specification, knowledge availability, query completeness, inference tractability, and semantic clarity. In particular, temporal abstraction is not always synonymous with implicitness. Abstraction does not influence correctness directly. It may improve the ease of model specification, as well as facilitate knowledge acquisition. Abstraction reduces query completeness and semantic clarity, while enhancing computational tractability. Based on these considerations, we propose that versatile and efficient time-modeling formalisms should encompass ways to represent and reason with both explicit and abstract temporal knowledge. We discuss MTBNs, as an example of such a hybrid formalism.

An important issue in systems utilizing temporal abstractions is how to choose them out of an infinite number of possible candidates. Although many research efforts have concentrated on the automation of specific temporal abstractions, much research needs to be done in understanding and developing provably optimal abstractions. We provide an initial framework for studying this problem in a manner that is independent of the particular problem domain and knowledge representation. Our analysis demonstrates some of the inherent difficulties of this endeavor. We also present several research directions we believe are worth exploring.

Acknowledgements—We are grateful to Dr Yuval Shahar for discussions leading to clarification of some of the ideas discussed in this work. We also thank the anonymous referees for useful suggestions.

REFERENCES

1. C.F. Aliferis, G.F. Cooper, R.A. Miller, B.G. Buchanan, R. Bankowitz and N. Giuse, A temporal analysis of QMR, *J. Am. Med. Inform. Ass.*, **3**, 79–91 (1996).
2. M.G. Kahn, Modeling time in medical decision-support programs, *Med. Dec. Making*, **11**, 249–264 (1991).
3. G.F. Cooper, NESTOR: A computer-based medical diagnostic aid that integrates causal and probabilistic knowledge. Doctoral dissertation, Stanford University (1984).

4. Y. Shoham, Temporal logics in AI: Semantical and ontological considerations, *Artif. Intell.*, **33**, 89–104 (1987).
5. A. Galton, A critical examination of Allen's theory of action and time, *Artif. Intell.*, **42**, 159–188 (1990).
6. A. Afifi and V. Clark, *Computer-aided Multivariate Analysis*, 3rd ed. Chapman and Hall, London (1996).
7. R.E. Neapolitan, *Probabilistic Reasoning in Expert Systems*. Wiley, New York (1990).
8. Y. Shahar, A knowledge-based method for temporal abstraction of clinical data. Doctoral dissertation, Stanford University (1994).
9. R.A. Miller, H.E. Pople and J.D. Myers, INTERNIST-I, an experimental computer-based diagnostic consultant for general internal medicine, *N. Engl. J. Med.*, **307**, 468–476 (1982).
10. J.B. Wyngaarden and L.A. Smith (editors), *Cecil Textbook of Medicine*. W.B. Saunders, Philadelphia (1988).
11. C.F. Aliferis and G.F. Cooper, A new formalism for time modeling in medical decision-support systems. In *Proceedings of 19th Annual Symposium on Computer Applications in Medical Care* (1995).
12. C. Berzuini, R. Bellazi, S. Quaglini and D.J. Spiegelhalter, Bayesian networks for patient monitoring, *Artif. Intell. Med.*, **4**, 243–260 (1992).
13. P. Dagum and A. Galper, Forecasting sleep apnea with dynamic network models. In *Proceedings of Uncertainty in Artificial Intelligence* (1993).
14. K. Kanazawa, A logic and time nets for probabilistic inference. In *Proceedings of the American Association for Artificial Intelligence* (1991).
15. J.X. Polaschek, G.W. Rutledge, S.K. Andersen and L.M. Fagan, Using belief networks to interpret qualitative data in the ICU, *Respiratory Care*, **38**, 60–71 (1993).
16. G. Provan, Tradeoffs in constructing and evaluating temporal influence diagrams. In *Proceedings of Uncertainty in Artificial Intelligence* (1993).
17. I.J. Haimowitz and I.S. Kohane, Automated trend detection with alternate temporal hypotheses. In Proceedings of the 13th International Joint Conference on Artificial Intelligence (1993).
18. I.M. De Zegher-Geets, A.G. Freeman, M.G. Walker, R.L. Blum and G. Wiederhold, Summarization and display of on-line medical records, *M.D. Computing*, **5**, 38–46 (1988).
19. R.S. Michalski, J.G. Carbonell and T.M. Mitchell, *Machine Learning*. Morgan Kaufmann, Palo Alto, CA (1983).
20. J. Hertz, A. Krogh and R.G. Palmer, *Introduction to the Theory of Neural Computation*. Addison-Wesley, Redwood City, CA (1991).
21. B. Chandrasekaran, J.W. Smith and J. Sticklen, "Deep" models and their relation to diagnosis, *Artif. Intell. Med.*, **1**, 29–40 (1989).
22. M.P. Wellman and C.-L. Liu, State-space abstraction for anytime evaluation of probabilistic networks. In Proceedings of Uncertainty in Artificial Intelligence (1994).
23. W.X. Wen, Optimal decomposition of belief networks. In Proceedings of Uncertainty in Artificial Intelligence (1991).
24. M. Baker and T.E. Boult, Pruning Bayesian networks for efficient computation. In Proceedings of Uncertainty in Artificial Intelligence (1991).
25. C. Combi, F. Pinciroli, G. Musazzi and C. Ponti, Managing and displaying different time granularities of clinical information. Proceedings of Annual Symposium of Computer Applications in Medical Care (1994).
26. P. Spirtes, Directed cyclic graphical representations of feedback models. In *Proceedings of Uncertainty in Artificial Intelligence* (1995).
27. G.F. Cooper, Causal discovery from data in the presence of selection bias. Preliminary papers of the 5th International Workshop on Artificial Intelligence and Statistics (1995).
28. R.L. Graham, D.E. Knuth and O. Patashnik, *Concrete Mathematics*. Addison-Wesley, Reading, MA (1989).

APPENDIX

The Appendix presents more formally some of the concepts and results discussed in the paper that were given as examples and intuitive arguments.

(Contextual) TA optimality. Consider a problem D as defined in the text. Let $W(.)$ be a worth function that gives the worth of each possible answer to every query Q_i ($Q_i \in Q_{re}$). We will say that a model M is a contextually optimal *TA* (CTA-optimal) model for D wrt Q_{re}, iff for each $Q_i \in Q_{re}$, M entails answers A_i, and there is no model M' such that M is an abstraction of M' and M' entails A_i' and $W(A_i', Q_j) > W(A_i, Q_j)$ for some i, j. Equivalently we will say that the *TA*s of the variables belonging to M are CTA-optimal for D wrt Q_{re}. A model will be baseline CTA-optimal (BCTA-optimal) iff it is CTA-optimal, and there is no model M'' s.t. M'' is an abstraction of M and for all i, j: $W(A_i', Q_j) = W(A_i'', Q_j)$.

IO-correct algorithm or program with respect to M(.). Consider a function $M(.)$ that maps from a set of inputs I to a set of outputs O. An IO-correct (IOC) algorithm, or program, wrt to $M(.)$ is a program that implements $M(.)$ exactly.

IO-correct formalism with respect to M(.). A formalism F that admits (i.e. is readily associated with, or guarantees the existence of the representation of an IOC algorithm wrt $M(.)$) is an IO-correct (IOC) formalism wrt $M(.)$.

Theorem 1. The existence of IOC formalisms implies that we can construct TA-optimal systems.

Proof (sketch): Consider a mapping M between all queries of interest (Q_{re} for a problem D) to answers such that the worth of each answer is optimal. Clearly M satisfies Contextual TA-optimality. Thus any formalism F which is IOC wrt to the function M will be TA-optimal for the problem and set of queries.

Theorem 2. Belief networks and MTBNs are IOC wrt to any joint probability distribution of discrete variables.

Proof (sketch): For every joint probability distribution (jpd) of discrete variables there is some BN that captures it [7]. The same applies for MTBNs [11]. From these and the definitions of IO-correctness, it follows that belief networks and MTBNs are IOC wrt any jpd of discrete variables.

Theorem 3. (a) If a formalism F_1 for problem D is IOC and the output of models expressed in F_1 can be reduced (i.e. converted) to the output of models expressed in formalism F_2, and the worth function for D is F_2-specific, then by using F_1 we can reject the TA-optimality of abstracting models expressed in F_2 that are not TA-optimal. (b) If two formalisms F_1, F_2 for problem D are IOC and the output of models expressed in F_1 can be reduced to the output of models expressed in F_2, and the worth function for D is F_2-specific, then by using F_1 we can establish or reject the TA-optimality of abstracting models expressed in F_2 that are TA-optimal or not, respectively.

Proof (sketch): (a) If we abstract over a TA-optimal model (expressed in formalism F_1) M_{F1}, then we can infer the possible non-optimality of a similar abstraction over a model M_{F2} (involving the same variables expressed in formalism F_2), by establishing non-optimality of the abstracted version of M_{F1}, M_{F1abs}. This is the case since if MF_{1abs} is not optimal, then necessarily MF_{2abs} is not optimal too (if it were, because F_1 is IOC we would be able to use optimal model MF_{2abs} to construct an optimal model in F_1, which is a contradiction). (b) The proof carries essentially unchanged from the proof of part (a). The following additional step is required for the case when F_2 is IOC for D: if M_{F1abs} is TA-optimal, then F_2 can do no worse than F_1, since we can use M_{F1abs} to develop a TA-optimal model M_{F2abs}. Therefore we cannot only establish non-optimality of F_2 using F_1, but TA-optimality as well.

Example of Application of Theorem 3. Define the general problem D, for the medical diagnosis domain to be a mapping from joint disease-finding instantiations to the joint probability of (Dis,F), and the worth of an answer to be 1 if the joint probability is correct and 0 otherwise. If we restrict our attention to the functionality of QMR that produces rank ordered lists of diagnoses given patient findings, we can reject the TA-optimality of QMR (if it is indeed non-optimal) by using MTBNs as follows: given this definition of the problem and Theorem 3, we need:

1. MTBN is IOC wrt to D (satisfied because of Theorem 2).
2. A jpd of (Dis, F) can be converted to a rank-order of Dis given F. This is satisfied by a trivial constructive proof: use any correct sorting algorithm to construct a sorted list of the diseases Dis_i in the domain each followed by its probability given the evidence of Q_i. Enumerate the list such that equiprobable diseases get the same index.
3. The worth function is rank-order $(Dis \mid F)$ specific. This is satisfied by assumption: any rank-order pair of diseases lists that differs only in the order of tied pairs has the same worth.

Theorem 4. The number of ways $g(n)$ of disjunctively abstracting a finite discrete variable X that can take any one of n possible values is $\Sigma(f(n, m))$, where the sum is taken over all m from 1 to n, and $f(n, m)$ denotes a Stirling number of the second kind. Furthermore, there is a closed form formula for computing $g(n)$.

Proof. By definition, the number of ways to partition n things into m nonempty subsets is given by a Stirling number of the second kind, which we write here as $f(n, m)$. The function $f(n, m)$ can be computed as follows [[28], Equation 6.19]:

$$f(n,m) = \sum_{k=1}^{m} \frac{\binom{m}{k} k^n (-1)^{(m-k)}}{m!}$$

The number of ways of abstracting X is equal to the sum of the number of ways to partition n things into 1, 2,..., and n nonempty subsets. Thus g(n) is equal to:

$$g(n)=\sum_{m=1}^{n} f(n,m).$$

Since the formula for $f(n,m)$ is in closed form, it is straightforward to see that $g(n)$ is in closed form.

Theorem 5. The number of ways of abstracting a finite discrete variable X taking n values, using both disjunctive and value-eliminating abstractions is given by:

$$g'(n)=\sum_{k=0}^{n-1} g(n-k)\binom{n}{k}=g(n-1).$$

Proof (sketch). We create TAs by eliminating k out of n values and then abstracting disjunctively the remaining values. k ranges from 0 to $n-1$. There are (n choose k) ways of eliminating k values from n total values. From Theorem 4, there are $g(n-k)$ ways to abstract disjunctively the remaining values (for each k). The total number of disjunctive and eliminating TAs of n values is equal to the total number of disjunctive TAs of $n+1$ values, since we can derive the former by introducing a pseudo-value "eliminate" and take all possible disjunctive abstractions involving this pseudo-value (taken to mean elimination of the corresponding value or combination of values).

Theorem 6. Consider a model M that contains discrete variables X_1, X_2,..., X_s that can take on n_1, n_2,..., n_s possible values, respectively. The number of possible disjunctive abstractions for M is:

$$g(\prod_{i=1}^{s}(n_i)).$$

And the number of possible disjunctive/value-eliminating abstractions is:

$$g'(\prod_{i=1}^{s}(n_i)).$$

Proof. Consider M to consist of a single variable X whose values are members of the Cartesian product of the values of X_1, X_2,..., X_s. It follows that the number of possible abstractions of X is as given in the statement of the theorem.

Theorem 7. The class of disjunctive and eliminating abstractions over a set of finite discrete variables V contains exactly all non-equivalent abstractions over V.

Proof (sketch). Consider a single variable X that contains all information of V. X has as many values as the Cartesian product over the values of each variable in V. Call this number N. For each number M from 0 to N, consider that although there are infinite deterministic functions that map from an arbitrary subset S of values of X to M values (i.e. they have a codomain with cardinality M), every function that assigns the members of S to M values in the same way (regardless of the meaning or label of the new values) will be mathematically indistinguishable (although semantically not so). The possible ways to make the assignments correspond precisely to taking disjunctions of original values, including allowing value elimination (that is, disjunctions that are not complete relative to S).

Theorem 8. Assuming a set of finite discrete variables V and deterministic abstractions, the number of non-equivalent TA functions over V is finite. Moreover these conditions (finite, discrete, deterministic) are necessary for finiteness of the TAs.

Proof (Sketch). (1) Sufficiency directly follows from Theorems 6 and 7. (2) Necessity is derived by showing that whenever we have a continuous (or infinite discrete) variable, we can devise a family of non-equivalent TAs that contains infinite members. Such a family can be trivially constructed by having each TA preserving all values but two, which are collapsed into a pairwise disjunction. There are is infinite number of such disjuncts and they are not information-equivalent. Similarly we can construct a family of non-equivalent TAs, that contains infinite members whenever we have non-determinism, by abstracting the N values into M abstracted categories based on a probability p of assignment to one of the values in $M.\ p$ can take any of a number of infinite values, and all these are non-equivalent, since they suggest a different probability of the original value(s) given the abstracted variable value assignment.

About the Author—C.F. Aliferis received his MD from Athens University, Athens, Greece in 1990 and his MS in Intelligent Systems/Medical Informatics from the University of Pittsburgh in 1994. He is a Research Fellow in the Section of Medical Informatics at the University of Pittsburgh and is a PhD candidate (Intelligent Systems Program). His main research interest is in the applications of artificial intelligence in medicine.

About the Author—G.F. Cooper received his PhD in Medical Information Science from Stanford University in 1985 and his MD from the same institution in 1986. He is currently an Associate Professor of Medicine and of Intelligent Systems at the University of Pittsburgh and his main research interest is the application of decision theory, probaility theory and artificial intelligence to medical informatics.

About the Author—M.E. Pollack received her PhD in Computer and Information Science from the University of Pennsylvania in 1986. She is currently an Associate Professor of Computer Science and Intelligent Systems at the University of Pittsburgh. Her research interest is artificial intelligence.

About the Author—B.G. Buchanan, PhD, is University Professor of Computer Science, Director of the W.M. Keck Center for Advanced Training in Computational Biology, Co-Director of the Center for Parallel, Distributed and Intelligent Systems, University of Pittsburgh.

About the Author—M.M. Wagner received his MD from New York University School of Medicine in 1979 and his PhD from the University of Pittsburgh Intelligent Systems Program in 1995. He is currently an Assistant Professor of Medicine. His research interest is medical decision-support systems.