Medical Knowledge Base Consistency Checking and Its Application to CADIAG-1/BIN

Wolfgang Moser[†], Klaus-Peter Adlassnig[‡]

[†]GSF-Forschungszentrum für Umwelt und Gesundheit GmbH
Institut für Medizinische Informatik und Systemforschung
Ingolstädter Landstrasse 1, D-8042 Neuherberg, Germany
and
[‡]Universität Wien
Institut für Medizinische Computerwissenschaften
Garnisongasse 13, A-1090 Wien, Austria

Abstract

Knowledge bases of medical expert systems have grown to such an extent that formal methods to verify their consistency seem highly desirable; otherwise, decision results of such expert systems are not reliable and contradictory entries in the knowledge base may even cause totally erroneous conclusions.

This paper presents a new formalization of the finding/finding, finding/disease, and disease/disease relationships of the medical expert system CADIAG-1. This formalization also helps to clarify the differences between the application of propositional logic and of quantificational logic to capture the meaning of some fundamental categorical relationships in the area of medical diagnostics. Moreover, this formalization leads to a very simple yet provable correct and complete algorithm to check the consistency of a medical knowledge base containing a set of these relationships.

1 Introduction

CADIAG-1's knowledge base [1] consists largely of binary finding/finding, finding/disease, and disease/disease relationships. This part of the system, hereafter called CADIAG-1/BIN, contains at present more than 50.000 relationships of this type. Because of this enormous quantity of medical relationships provided by several medical experts, a formal algorithm to check their logical consistency seems highly desirable.

Although such an algorithm has already been presented in [3], a reworking of the underlying formalization seems to be necessary because this algorithm turned out to be incomplete for the detection of *all* possible inconsistencies [5].

After providing an informal account of CADIAG-1/BIN's relationships, some differences between the interpretation of these relationships in terms of propositional and quantificational logic are highlighted. This discussion helps to motivate a new formalization of these relationships which leads directly to a remarkably simple and efficient algorithm for correct and complete consistency checking.

2 Relationships in CADIAG-1/BIN

In CADIAG-1/BIN, two aspects of the relationship between two medical entities E_i and E_j (e.g., symptoms, signs, test results, and diseases) occurring at the same time in a patient are taken into account: (a) the necessity of occurrence of E_i with E_j and (b) the sufficiency of occurrence of E_i to conclude E_j. These two aspects are combined, yielding the following five types of relationships as was proposed in [7]:

1. E_i oc E_j (*obligatory occurrence* and *confirmation*):
   the occurrence of E_i is *necessary* and *sufficient* for the occurrence of E_j in a patient.
2. E_i on E_j (*obligatory occurrence* and *non-confirmation*):
   the occurrence of E_i is *necessary*, but *not sufficient* for the occurrence of E_j in a patient.
3. E_i fc E_j (*facultative occurrence* and *confirmation*):
   the occurrence of E_i is *not necessary*, yet *sufficient* for the occurrence of E_j in a patient.
4. E_i ex E_j (*exclusion*):
   the occurrence of E_i is *not necessary* and *not sufficient* for the occurrence of E_j, yet *sufficient* for the absence of E_j in a patient.
5. E_i fn E_j (*facultative occurrence* and *non-confirmation*):
   the occurrence of E_i is neither *necessary*, nor *sufficient* for the occurrence of E_j, nor *sufficient* for the absence of E_j in a patient.

3 Meaning of the Relationships

By giving informal meaning to the five types of relationships in terms of necessity of occurrence and sufficiency of occurrence, one is tempted to translate these relationships directly into propositional logic. In formalizing (a) "E_i is necessary for E_j" by the implication $\neg e_i \rightarrow \neg e_j$, which is logically equivalent to $e_j \rightarrow e_i$; and (b) "E_i is sufficient for E_j" by $e_i \rightarrow e_j$, the following formal characterization of the relationships is obtained:

1. E_i oc E_j: $e_i \rightarrow e_j \wedge e_j \rightarrow e_i$.
2. E_i on E_j: $e_j \rightarrow e_i$.
3. E_i fc E_j: $e_i \rightarrow e_j$.
4. E_i ex E_j: $e_i \rightarrow \neg e_j$.
5. E_i fn E_j: no formula.

This formalization seems intuitively appealing. It captures what might be concluded from these types of relationships for a single patient.

Consider a knowledge base consisting of the single, validated entry E_i fc E_j. If E_i is present in a patient, e_i is added to the working memory. From e_i and $e_i \rightarrow e_j$, a single application of the modus ponens inference rule derives e_j; thus E_j must occur in this patient, too. However, if E_i is absent in a patient, $\neg e_i$ is added to the working memory, from which neither e_j nor $\neg e_j$ can be derived. Therefore, neither the occurrence nor the absence of E_j in this patient can be concluded in this situation.

Although this formalization leads to intuitively correct conclusions in cases of a single patient, it is too weak to grasp the full meaning of the above-mentioned medical relationships. It is not clear at all, how one should deal with the fn relationships during consistency checking. Another consequence of this formalization for example is that E_i fc E_j is logically entailed by E_i oc E_j.

However, these relationships intuitively contradict each other, because facultative occurrence is supposed to be complementary to obligatory occurrence. These examples indicate that a procedure to test the consistency of CADIAG-1/BIN's knowledge base cannot be built upon this formalization.

Nor are these limitations overcome by adding the corresponding formulas for "not necessary" and "not sufficient" to the above formalization. In fact, this extension results in an inconsistent set of formulas. The formalization of E_i fn E_j yields $\neg(e_i \rightarrow e_j) \wedge \neg(e_j \rightarrow e_i) \wedge \neg(e_i \rightarrow \neg e_j)$. The first two implications of this conjunction already constitute a contradiction. An inconsistent set of formulas is obviously no sensible candidate to attach formal meaning to the relationships.

The failure to give a formal account of these relationships in propositional logic can be traced back to two closely related misconceptions: (a) the actual domain of interpretation of these relationships; and (b) the role of the implication sign in modelling empirical knowledge.

The propositional formalization allows to draw reasonable conclusions for a single patient, while assuming, however, that the respective relationships can be applied to any patient. In this respect the propositional implication $e_i \rightarrow e_j$ is an instantiation for an individual patient of an actually universally quantified implication $\forall X\,(e_i(X) \rightarrow e_j(X))$, where the variable X ranges over all patients.

The merits of changing to a formalization in quantificational logic, and thus from interpretations about a single patient to interpretations about a set of patients, become apparent in the analysis of the relationship E_i fc E_j. This relationship does not refer to a single patient at all. If the occurrence of E_i is not necessary but sufficient for the occurrence of E_j in a patient, at least two different groups of patients must exist: (a) a first group of patients having E_j, but not E_i; and (b) a second group consisting of patients all having E_i. Moreover, all patients in the second group must exhibit E_j as well. It is impossible to claim E_i fc E_j to be true when considering only one single patient.

To ensure that these groups of patients really exist, the corresponding sets of patients have to be non-empty. If one group is allowed to be the empty set, these interpretations refer to situations where the corresponding combination of entities is not present in any patient and therefore unobservable. Obviously the relationships express empirical knowledge that is not justified by non-existing patients but based on observable evidence. To protect against such possible misinterpretations, the formalization must be extended by appropriate existentially quantified formulas. Because the implication $\forall X\,(e_i(X) \rightarrow e_j(X))$ is true even in the case where its premise is false for every X and e_i therefore denotes the empty set, the addition of the existentially quantified formulas is absolutely necessary to ensure the correct meaning of these relationships.

Putting things together, E_i fc E_j can be characterized by the formula $\forall X\,(e_i(X) \rightarrow e_j(X)) \wedge \exists Y\, e_i(Y) \wedge \exists Z\,(\neg e_i(Z) \wedge e_j(Z))$. With this formalization at hand it becomes clear why interpreting the fc relationship as the propositional implication $e_i \rightarrow e_j$ was so appealing. By instantiating the quantificational formula for a single patient, only this part of the instantiated formula is able to actually provide additional information for a patient, namely the occurrence of E_j simultaneously to E_i. Nevertheless, the relationship itself is not a statement about a single patient but about a set of patients.

Instead of presenting the formulas for the other relationships as well, the above discussion is graphically summarized by Venn diagrams [6] in Figure 1. A cross indicates that the corresponding subset should not be empty and a hatched region marks a definitely empty subset. As can be seen from the fc relationship, it is easy to translate a Venn diagram directly into quantificational logic using the one-place (monadic) predicate symbol e_i for the corresponding entity E_i. If CADIAG-1/BIN's knowledge base is formalized in this way, we will end up with a set of formulas denoted by F_m.

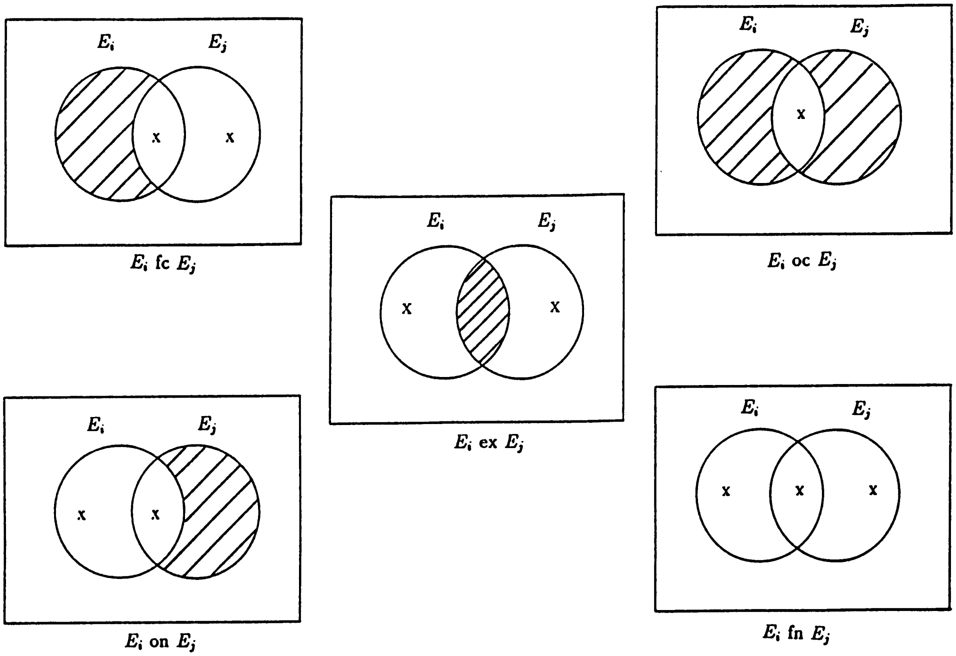

Figure 1: CADIAG-1/BIN's Relationships as Venn Diagrams.

4 A simple formalization

In formalizing the relationships between medical entities in terms of logical relationships between predicate symbols we provide—from a logical point of view—definitions of the five types of relationships in terms of second-order predicate logic.

The question may arise whether it is possible to give an equivalent formalization of these relationships directly in first-order logic, where (a) the relationships are represented themselves by predicate symbols; and (b) the entities are denoted by constant symbols rather than predicate symbols. Such a formalization would have the great advantage to be directly applicable to CADIAG-1/BIN's knowledge base without all relationships having to be translated into a set of logical formulas first.

For the formalization in first-order logic, we introduce the new two-place predicate symbol $\sqsubseteq$. The intended interpretation of $e_i \sqsubseteq e_j$ is that E_i is a subset of E_j.

Bearing this intended interpretation in mind, the following equivalences should be self-evident:

$$\begin{aligned}
\forall X \forall Y\,(X \text{ oc } Y \equiv{} & X \sqsubseteq Y \wedge Y \sqsubseteq X).\\
\forall X \forall Y\,(X \text{ on } Y \equiv{} & \neg X \sqsubseteq Y \wedge Y \sqsubseteq X).\\
\forall X \forall Y\,(X \text{ fc } Y \equiv{} & X \sqsubseteq Y \wedge \neg Y \sqsubseteq X).\\
\forall X \forall Y (X \text{ ex } Y \equiv{} & \neg \exists Z\,(Z \sqsubseteq X \wedge Z \sqsubseteq Y)).\\
\forall X \forall Y\,(X \text{ fn } Y \equiv{} & \neg X \sqsubseteq Y \wedge \neg Y \sqsubseteq X \wedge \exists Z\,(Z \sqsubseteq X \wedge Z \sqsubseteq Y)).
\end{aligned}$$

Formally however, $\sqsubseteq$ is not different from any other predicate symbol in that it can be interpreted in arbitrary ways. By providing the following two axioms, we limit its possible

$$
\begin{array}{rcl}
X \sqsubseteq X & \leftarrow & . \\
X \sqsubseteq Y & \leftarrow & X \sqsubseteq Z, Z \sqsubseteq Y. \\
\\
X \sqsubseteq Y & \leftarrow & X \text{ oc } Y; X \text{ fc } Y. \\
Y \sqsubseteq X & \leftarrow & X \text{ oc } Y; X \text{ on } Y. \\
@f(X,Y) \sqsubseteq X & \leftarrow & X \text{ fn } Y. \\
@f(X,Y) \sqsubseteq Y & \leftarrow & X \text{ fn } Y. \\
\\
false & \leftarrow & (X \text{ on } Y; X \text{ fn } Y), X \sqsubseteq Y. \\
false & \leftarrow & (X \text{ fc } Y; X \text{ fn } Y), Y \sqsubseteq X. \\
false & \leftarrow & X \text{ ex } Y, Z \sqsubseteq X, Z \sqsubseteq Y.
\end{array}
$$

Figure 2: Specification of the Consistency Checker.

interpretations:

$$
\begin{array}{rl}
\text{Reflexivity:} & \forall X\,(X \sqsubseteq X). \\
\text{Transitivity:} & \forall X \forall Y \forall Z\,((X \sqsubseteq Y \wedge Y \sqsubseteq Z) \rightarrow X \sqsubseteq Z).
\end{array}
$$

A formalization of CADIAG-1/BIN in terms of $\sqsubseteq$ will be denoted by $F_{\sqsubseteq}$.

Although this formalization seems to be intuitively correct, it is not obvious at all whether $F_{\sqsubseteq}$ is inconsistent if and only if CADIAG-1/BIN's corresponding formalization in monadic predicate logic F_m is inconsistent. But fortunately the following theorem can be established [5]:

Theorem 1 *F_m has a model if and only if $F_{\sqsubseteq}$ has a model.*

As a set of formulas has then no model if and only if the set of formulas is inconsistent, $F_{\sqsubseteq}$ and F_m are proved to be equivalent with respect to consistency.

5 Consistency Checking

For consistency checking, the relationships between the medical entities in the knowledge base are given in advance. Therefore we are only interested in the left to right direction of the above equivalences of our medical relationships. Transforming these formulas into a set of clauses, where skolemization introduces the new two-place function symbol $@f$, yields a set of horn clauses. Replacing each goal $\leftarrow G$ in this set by a new rule $false \leftarrow G$ results in a set of definite horn clauses equivalent to the PROLOG program depicted in Figure 2 [8].

CADIAG-1/BIN's knowledge base is inconsistent if and only if $false \leftarrow$ is entailed by this set of clauses [5]. The first rule for $false$ should therefore be read in the following way: The underlying knowledge base is inconsistent if Xon$Y$ or Xfn$Y$ is an entry in the knowledge base and if $X \sqsubseteq Y$ is a logical consequence of the other entries. This captures precisely what intuitively constitutes a contradiction for these types of relationships.

From this set of clauses it becomes obvious that consistency checking of CADIAG-1/BIN's knowledge base is reduced to the computation of the reflexive and transitive closure of the $\sqsubseteq$ relation and the lookup of the corresponding $\sqsubseteq$ entries for the $false$ rules. If the entities are stored in a matrix, this computation takes at most time in $O(|E|^3)$, where $|E|$ denotes the number of entities in the knowledge base [4, pp. 550ff].

It should be noted that the clauses in Figure 2 have to be modified for the use with a standard PROLOG system in order to detect all possible inconsistencies. Due to PROLOG's incomplete depth-first, left to right strategy, the system will encounter an infinite loop for every cycle in the $\sqsubseteq$ relation, e.g. for every oc relationship.

6 Conclusions

The paper on hand presents a new formalization of the five types of relationships in CADIAG-1/BIN which leads to a remarkable simple algorithm for consistency checking.

This formalization was only possible by realizing that any knowledge base generally claims statements about different sets of patients to be true. What is true of a single patient is a logical consequence of the placement of this patient within these sets. Hence, only a formalization in quantificational logic as opposed to propositional logic suffices to capture this situation.

Based on this formalization, a program was developed to check the consistency of CADIAG-1/BIN's knowledge base. On the first run, it detected 17 inconsistencies which could be corrected subsequently.

Furthermore, a suitable mapping of binary relationships of some medical expert systems (such as QMR [2]) into the relationship categories of CADIAG-1 makes the developed consistency checking algorithm a broadly applicable one.

References

[1] K.-P. Adlassnig, G. Kolarz, G. Scheithauer, W. Effenberger, and G. Grabner. CADIAG: Approaches to Computer-Assisted Medical Diagnosis. *Comp. Biol. Med.*, 15: 315–335, 1985.

[2] R.A. Bankowitz, M.A. McNeil, S.M. Challinor, R.C Parker, W.N. Kapoor, and R.A. Miller. A Computer-Assisted Medical Diagnostic Consultation Service. *Annals of Internal Medicine*, 110: 824–832, 1989.

[3] F. Barachini and K.-P. Adlassnig. CONSDED: Medical Knowledge Base Consistency Checking. In *Proc. Medical Informatics Europe 87*, Rome, 974–980, 1987.

[4] T.H. Cormen, C.E. Leiserson, and R.L. Rivest. *Introduction to Algorithms.* The MIT Press, Cambridge, Massachusetts, 1990.

[5] W. Moser. *Konsistenzprüfung einer medizinischen Wissensbasis.* Diplomarbeit, Studiengang Medizinische Informatik, Universität Heidelberg und Fachhochschule Heilbronn, Heilbronn, 1990.

[6] W.V.O. Quine. *Methods of Logic.* Holt, Rinehart and Wilson, New York, 1964.

[7] W. Spindelberger and G. Grabner. Ein Computer-Verfahren zur diagnostischen Hilfestellung. In K. Fellinger (Hrsg.) *Computer in der Medizin — Probleme, Erfahrungen, Projekte.* Brüder Hollinek, Wien, 189–221, 1968.

[8] L. Sterling and E. Shapiro. *The Art of Prolog.* The MIT Press, Cambridge, Massachusetts, 1986.