Artificial Intelligence in Medicine 4 (1992) 389–407
Elsevier

Consistency checking of binary categorical relationships in a medical knowledge base

Wolfgang Moser [a] and Klaus-Peter Adlassnig [b]

[a] *GSF Forschungszentrum, Institut für Medizinische Informatik und Systemforschung, Ingolstädter Landstrasse 1, 8042 Neuherberg, Germany*

[b] *Universität Wien, Institut für Medizinische Computerwissenschaften, Währinger Gürtel 18–20, 1090 Wien, Austria*

Received April 1992
Revised May 1992

Abstract

Moser, W. and K.-P. Adlassnig, Consistency checking of binary categorical relationships in a medical knowledge base, Artificial Intelligence in Medicine 4 (1992) 389–407.

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 cause 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 very simple yet provably correct and complete algorithms to check the consistency of a medical knowledge base containing a set of these relationships.

Keywords. Binary categorical relationship; CADIAG-1; consistency; knowledge base.

1. Introduction

The knowledge base of CADIAG-1 [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 of these binary relationships. Because of this high 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 [5], a reworking of the underlying formalization seems to be necessary because this algorithm turned out to be incomplete for

Correspondence to: W. Moser, GSF Forschungszentrum, Institut für Medizinische Informatik und Systemforschung, Ingolstädter Landstrasse 1, 8042 Neuherberg, Germany, e-mail: moser@gsf.de

0933-3657/92/$05.00 © 1992 – Elsevier Science Publishers B.V. All rights reserved

the detection of *all* possible inconsistencies [10].

After providing an informal account of the notion of consistency and of the relationships of CADIAG-1/BIN, the 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 directly leads to remarkably simple and efficient algorithms for correct and complete consistency checking. However, these algorithms are inefficient in cases where a knowledge base known to be consistent has to be extended by new relationships. In modifying our new formalization, we obtain a correct and complete algorithm for such situations and discuss its application to check the consistency of the knowledge base of CADIAG-1/BIN.

2. Consistency

A set of formulas is consistent if and only if there is at least one formula not derivable from this set of formulas. From the contraposition of this definition follows that every formula is derivable from an inconsistent set.

In this paper, we employ first-order predicate logic (FOL) to formalize binary categorical relationships. For a set of formulas of FOL to be inconsistent, it suffices for a formula A as well as for its negation $\neg A$ to be derivable from the set. From A and $\neg A$ an arbitrary formula B can be established in the following way: $\neg A$ could be weakened to $\neg A \vee B$ which can be rewritten as the implication $A \supset B$ and applying the modus ponens inference rule with A will yield B.

It should be noted that consistency is a syntactical characterization of a set of formulas. Yet in modelling knowledge one is interested in the meaning of the formulas rather than in their syntactical properties. For FOL there is a semantic counterpart to the syntactical notion of consistency. Due to the existence of correct and complete calculi for FOL, we are guaranteed that a set of formulas has a model if and only if it is consistent. If and only if the set of formulas is consistent it is possible to attach meaning to the formulas so that all formulas turn out to be true. But one should be aware that such a model has not necessarily to reflect the intended application domain. To test if a model correctly represents the application domain is outside the realm of consistency checking unless the application domain is formalized as well.

In general, the consistency of a set of formulas in FOL is semi-decidable. There can be no algorithm correctly classifying an arbitrary set of formulas either as consistent or as inconsistent. It is only possible to deceive algorithms stopping on an inconsistent set with a correct answer, with no guarantee, however, for termination on a consistent set. In restricting the structure of formulas, subclasses of FOL can be singled out for which their consistency is decidable. One well-known subclass is propositional logic where, for instance, truth tables can be used for consistency checking. However, for propositional logic as well as for most of the other subclasses consistency checking is more of theoretical interest because efficient algorithms consuming a reasonable amount of time and space for a large set of such formulas are either proved or highly conjectured to be impossible. The method of truth tables, for instance, is exponential in the number of propositional variables occurring in the formulas and is therefore not applicable to a large set of such formulas. A comprehensive discussion of computability and complexity aspects of FOL can be found in [6].

Due to its computational complexity, consistency checking of most sets of FOL formulas is outside the scope of any practical algorithm. For such possibly inconsistent sets, it becomes necessary to cautiously deal with contradictions to limit their disastrous consequences in either restricting the rules of inference or dispose with FOL altogether and change to a nonstandard logic system [3, 8]. Fortunately, the formalization of the relationships of CADIAG-1/BIN will yield a set of formulas where efficient consistency checking is still possible.

3. Relationships in CADIAG-1/BIN

In CADIAG-1/BIN, two aspects of the relationship between two medical entities e_i and e_j (i.e. 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 [12]:

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.
   Example: The X-ray finding 'endoprothesis of the knee' is *obligatory occurring* and *confirming* for the disease 'arthoplasty of the knee'.

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.
   Example: The clinical finding 'Herberden's nodes' is *obligatory occurring* and *non-confirming* for the disease 'Herberden's arthrosis'.

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.
   Example: The lab result 'intracellular uric acid crystals in joint effusion' is *facultative occurring*, yet *confirming* for the disease 'gout'.

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.
   Example: The lab finding 'Waaler Rose titer 1:128' *excludes* the disease 'seronegative rheumatoid arthritis'.

5. e_i fn e_j (*facultative occurrence* and *non-confirmation*):
   the occurrence of e_i is neither *necessary*, nor *sufficient* for the occurrence or absence of e_j in a patient.
   Example: The lab result 'elevated erythrocyte sedimentation rate' is *facultative occurring* and *non-confirming* for the disease 'rheumatoid arthritis'.

4. An incomplete formalization 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) 'the occurrence of e_i is necessary (obligatory) for the occurrence of e_j in a patient' by the implication $\neg e_i \supset \neg e_j$, which is logically equivalent to $e_j \supset e_i$; and (b) 'the occurrence of e_i is sufficient (confirming) for the occurrence of e_j in a patient' by $e_i \supset e_j$, the following formal characterization of the relationships is obtained:

1. e_i oc e_j: $e_i \supset e_j \wedge e_j \supset e_i$;

2. e_i on e_j: $e_j \supset e_i$;

3. e_i fc e_j: $e_i \supset e_j$;

4. e_i ex e_j: $e_i \supset \neg e_j$;

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 $e_i \supset e_j$, 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. The formalization of the other relationships could be analyzed in a similar way and would reveal the adequacy of the formalization concerning inferences for a single patient too.

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. One consequence of this formalization is, for instance, 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.

Given the formalization at hand it is not clear at all, how one should deal with the fn relationships during consistency checking. The set $\{e_1$ oc e_2, c_2 ex e_3, e_1 fn $e_3\}$ provides an example of an inconsistent set of relationships involving an fn relationship. If the occurrence of e_1 in a patient is confirming for the presence of e_2 in the patient, and from the presence of e_2 one can conclude that e_3 is absent in the patient, e_1 and e_3 could not occur simultaneously in any patient. However, this is nothing but a proof that the presence of e_1 excludes the occurrence of e_3 in any patient and therefore e_1 ex e_3 is a logical consequence of e_1 oc e_2 and e_2 ex e_3. Obviously, e_1 ex e_3 and e_1 fn e_3 could not hold at the same time. This example clearly indicates that fn relationships have to be taken into account during consistency checking.

The limitations of the formalization are not 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 $\neg(e_i \supset e_j) \wedge \neg(e_j \supset e_i) \wedge \neg(e_i \supset \neg e_j)$. 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.

From the above example one might draw the conclusion that the difficulties in giving a formal account of these relationships in propositional logic are due to the lack of an appropriate formalization of the relations 'not necessary' and 'not sufficient'. But consider the slightly modified set of relationships from above, where the third relationship has been changed from fn to oc: $\{e_1 \text{ oc } e_2, e_2 \text{ ex } e_3, e_1 \text{ oc } e_3\}$. As in the previous example we can conclude from the first two relationships that e_1 ex e_3 must hold. If the formalization of these relationships in propositional logic is complete, the corresponding formulas should form an inconsistent set:

$$\{\ e_1 \supset e_2 \wedge e_2 \supset e_1,$$
$$e_2 \supset \neg e_3,$$
$$e_1 \supset e_3 \wedge e_3 \supset e_1\ \}\ .$$

However, this set is not inconsistent. Since an implication is true whenever its antecedent is false, the interpretation where all three propositional variables denote false, satisfies the formulas. Therefore this set of formulas has at least one model and is consistent.

5. A complete formalization of 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) their ontological presuppositions.

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 $e_i \supset e_j$ is an instantiation for an individual patient of an actually universally quantified implication $\forall X\left(e_i(X) \supset e_j(X)\right)$, 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, should become apparent in the analysis of the relationship e_i fc e_j. This relationship does not refer solely to a single patient. 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 looking at one single patient.

If one of these groups is allowed to be the empty set, these interpretations refer to situations where the corresponding combination of entities is not present in any patient. In such situations the relationships would relate non-empty sets to empty sets of patients. If the first

group of patients having e_j, but not e_i is empty, this interpretation refers to the situation where this combination of entities is not observable for any patient. Claiming in such a situation the occurrence of e_i to be not necessary for the occurrence of e_j in a patient is impossible, because all patients with e_j have e_i, too, and therefore e_i is obligatory for e_j. Likewise, if the second group of patients having e_i is empty, claiming that the occurrence of e_i in a patient is confirming for the occurrence of e_j in the same patient becomes meaningless, as e_i is not related to any patient at all. Situations where one of the corresponding groups of patients denote the empty set are no models of the relationships. To prevent such situations from being models, the formalization must be extended by appropriate existentially quantified formulas.

In the last paragraph we argued that the relationships to be meaningful presuppose the existence of the corresponding entities. This kind of presupposition solves another obstacle of consistency checking in knowledge bases. If a knowledge base comprises general knowledge and if this general knowledge is meant to be applicable to individual patients, the formalization of the general knowledge must account for this application. In case of the relationships of CADIAG-1/BIN, we ensure through the addition of appropriate existentially quantified formulas that it is actually possible to have patients falling under these relationships. Without the addition of the appropriate existentially quantified formulas the knowledge base could be consistent, while never be applicable to any patients, because the entities of the knowledge base denote empty sets.

To illustrate such a situation consider the aforementioned set of relationships $\{e_1 \text{ oc } e_2, e_2 \text{ ex } e_3, e_1 \text{ oc } e_3\}$. If the implications of the propositional formalization are lifted to their corresponding formulas in FOL, the following set is obtained:

$$\Big\{ \forall X\Big(e_1(X) \supset e_2(X)\Big) \wedge \forall X\Big(e_2(X) \supset e_1(X)\Big),$$
$$\forall X\Big(e_2(X) \supset \neg e_3(X)\Big),$$
$$\forall X\Big(e_1(X) \supset e_3(X)\Big) \wedge \forall X\Big(e_3(X) \supset e_1(X)\Big) \Big\}.$$

But this set still has one model. Because the implication $\forall X\Big(e_i(X) \supset e_j(X)\Big)$ is true even in the case where its antecedent is false for every X and e_i therefore denotes the empty set, it suffices to interpret all three entities e_1, e_2, and e_3 as the empty set to obtain a model. However, such an interpretation should not be valid, if the relationships comprise a knowledge base which is a priori applicable to individual patients.

To summarize the foregoing discussion, we formalize

- 'the occurrence of e_i is *sufficient* (*confirming*) for the occurrence of e_j in a patient' by $\forall X\Big(e_i(X) \supset e_j(X)\Big) \wedge \exists X\Big(e_i(X)\Big) \wedge \exists X\Big(e_j(X)\Big)$, which is logically equivalent to $\forall X\Big(e_i(X) \supset e_j(X)\Big) \wedge \exists X\Big(e_i(X)\Big)$;

- 'the occurrence of e_i is *not sufficient* (*non-confirming*) for the occurrence of e_j in a patient' by $\neg\forall X\Big(e_i(X) \supset e_j(X)\Big) \wedge \exists X\Big(e_i(X)\Big) \wedge \exists X\Big(e_j(X)\Big)$, which is logically equivalent to $\exists X\Big(\neg e_i(X) \wedge e_j(X)\Big) \wedge \exists X\Big(e_i(X)\Big)$;

- 'the occurrence of e_i is *necessary* (*obligatory*) for the occurrence of e_j in a patient' by $\forall X\left(e_j(X) \supset e_i(X)\right) \wedge \exists X\left(e_i(X)\right) \wedge \exists X\left(e_j(X)\right)$, which is logically equivalent to $\forall X\left(e_j(X) \supset e_i(X)\right) \wedge \exists X\left(e_j(X)\right)$;
- 'the occurrence of e_i is *not necessary* (*facultative*) for the occurrence of e_j in a patient' by $\neg\forall X\left(e_j(X) \supset e_i(X)\right) \wedge \exists X\left(e_i(X)\right) \wedge \exists X\left(e_j(X)\right)$, which is logically equivalent to $\exists X\left(e_i(X) \wedge \neg e_j(X)\right) \wedge \exists X\left(e_j(X)\right)$.

From these formulas we can assemble the definitions of the relationships:

$$e_i \text{ oc } e_j \stackrel{\text{def}}{=} \forall X\left(e_i(X) \supset e_j(X)\right) \wedge \forall X\left(e_j(X) \supset e_i(X)\right) \wedge \exists X\left(e_i(X)\right);$$

$$e_i \text{ on } e_j \stackrel{\text{def}}{=} \forall X\left(e_j(X) \supset e_i(X)\right) \wedge \exists X\left(e_j(X)\right) \wedge \exists X\left(e_i(X) \wedge \neg e_j(X)\right);$$

$$e_i \text{ fc } e_j \stackrel{\text{def}}{=} \forall X\left(e_i(X) \supset e_j(X)\right) \wedge \exists X\left(e_i(X)\right) \wedge \exists X\left(\neg e_i(X) \wedge e_j(X)\right);$$

$$e_i \text{ ex } e_j \stackrel{\text{def}}{=} \forall X\left(e_i(X) \supset \neg e_j(X)\right) \wedge \exists X\left(e_i(X)\right) \wedge \exists X\left(e_j(X)\right);$$

$$e_i \text{ fn } e_j \stackrel{\text{def}}{=} \exists X\left(e_i(X) \wedge e_j(X)\right) \wedge \exists X\left(e_i(X) \wedge \neg e_j(X)\right) \wedge \exists X\left(\neg e_i(X) \wedge e_j(X)\right).$$

With this formalization at hand it becomes clear why interpreting the relationships as the propositional formulas was so appealing. By instantiating the quantificational formula for a single patient, only the instantiated formulas of the universally quantified implications are able to actually provide additional information for a single patient, namely the occurrence of an entity e_k given in the antecedent of the implication simultaneously to e_l provided by its consequent. Nevertheless, the relationships themselves are not statements about single patients but about non-empty sets of patients.

The above formalization can be graphically summarized by Venn diagrams [11] given in *Fig. 1.* A cross indicates that the corresponding subset is not empty and a hatched region marks a definitely empty subset.

If the knowledge base of CADIAG-1/BIN is formalized in this way, we will end up with a set of formulas using one-place (monadic) predicate symbols denoted by F_m.

6. Some equivalences

With the formalization given in the last section it is easy to prove the following equivalences:

$$e_i \text{ oc } e_j \equiv e_j \text{ oc } e_i;$$

$$e_i \text{ ex } e_j \equiv e_j \text{ ex } e_i;$$

$$e_i \text{ fn } e_j \equiv e_j \text{ fn } e_i;$$

$$e_i \text{ on } e_j \equiv e_j \text{ fc } e_i.$$

Although the relationships are defined with a direction in mind, oc, ex, and fn turn out to be symmetric. Due to the last equivalence, the relationships fc and on could replace each other,

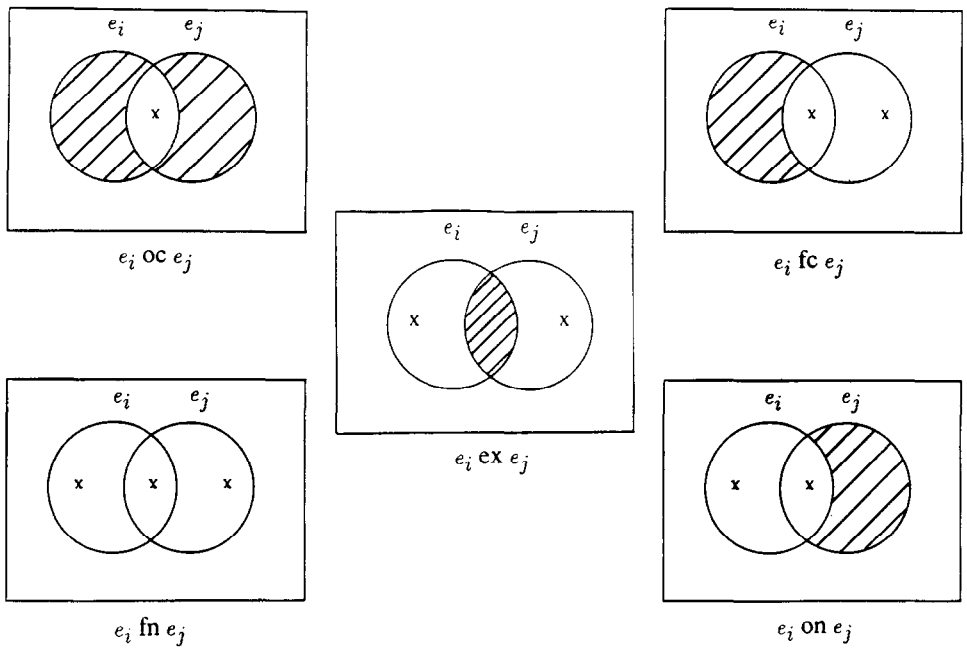

Fig. 1. Relationships of CADIAG-1/BIN as Venn diagrams.

so one of these is redundant from a logical point of view. However, it is more convenient for a medical expert to have both types of relationships available to represent his/her knowledge.

A further remark concerns the relationship e_i ex e_j. This relationship neither forces the set of patients having e_j to be complementary to the set of patients having e_i, nor does it exclude this possibility. Therefore, e_i ex e_j is not equivalent to $(\neg e_i)$ on e_j as one might expect.

7. Consistency checking of F_m

It is straightforward to present a correct and complete algorithm to check the consistency of F_m.

If F_m is translated into its clausal form, a set of Horn clauses with constant symbols introduced via the Skolemization of the existential quantifiers, but without function symbols, is obtained. It is easy to prove [10] that for such sets of Horn clauses positive hyperresolution [9, p. 116ff] ('forward chaining'), i.e. using only already derived facts in the premise of a rule to derive a new fact, with elimination of subsumed facts [9], i.e. only retaining the most general facts, constitutes a decision procedure for consistency checking.

However, there is one major deficiency of this approach which severely restricts its applicability. To check the consistency of the relationships, each of them has to be translated into its corresponding clausal form. This translation would yield over 200.000 clauses for the total of about 50.000 relationships of CADIAG-1/BIN which must be handled during consistency checking.

Instead of trying to optimize the translation of the relationships and their consistency checking, we propose a different approach, namely an optimization of the formalization of the relationships themselves.

8. Simplifying the 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 FOL, 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 the knowledge base of CADIAG-1/BIN without having all relationships to be translated into a set of logical formulas first.

For the formalization in FOL, we introduce the new two-place predicate symbol $\sqsubseteq$ written in infix notation. The intended interpretation of $e_i \sqsubseteq e_j$ is that e_i is a non-empty subset of e_j.

Bearing this intended interpretation and *Fig. 1* in mind, the following equivalences should be self-evident, where $\forall^*$ denotes the universal closure of a formula and negated $\sqsubseteq$ relations are written in infix notation (i.e. $\not\sqsubseteq$).

$$\forall^*(X \text{ oc } Y \equiv X \sqsubseteq Y \wedge Y \sqsubseteq X);$$

$$\forall^*(X \text{ on } Y \equiv X \not\sqsubseteq Y \wedge Y \sqsubseteq X);$$

$$\forall^*(X \text{ fc } Y \equiv X \sqsubseteq Y \wedge Y \not\sqsubseteq X);$$

$$\forall^*\big(X \text{ ex } Y \equiv \neg\exists Z(Z \sqsubseteq X \wedge Z \sqsubseteq Y)\big);$$

$$\forall^*\big(X \text{ fn } Y \equiv X \not\sqsubseteq Y \wedge Y \not\sqsubseteq X \wedge \exists Z(Z \sqsubseteq X \wedge Z \sqsubseteq Y)\big).$$

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 reflecting the properties of the subset relation, we limit its possible models:

$$\text{Reflexivity:} \quad \forall X(X \sqsubseteq X).$$

$$\text{Transitivity:} \quad \forall^*\big((X \sqsubseteq Y \wedge Y \sqsubseteq Z) \supset X \sqsubseteq Z)\big).$$

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 the corresponding formalization of CADIAG-1/BIN in monadic predicate logic F_m is inconsistent.

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.

Proof.

$\Rightarrow$ If F_m has a model with domain D_m and interpretation I_m, we can construct a model of $F_{\sqsubseteq}$ with domain $D_{\sqsubseteq}$ and interpretation $I_{\sqsubseteq}$ in the following way:

$$
\begin{aligned}
D_{\sqsubseteq} &= \{M_i \mid e_i \text{ occurring in } F_m\} \cup \{\mathrm{FN}_{kl} \mid e_k \text{ fn } e_l \text{ formalized in } F_m\}\,; \\
M_i &= \{X \mid X \in D_m \text{ and } e_i(X) \text{ is true under } I_m\}\,; \\
\mathrm{FN}_{kl} &= M_k \cap M_l\,.
\end{aligned}
$$

Due to the existential quantifiers used in the formalization of the relationships each of the sets M_i and FN_{kl} is non-empty.

It is straightforward to show that the interpretation $I_{\sqsubseteq}$ over $D_{\sqsubseteq}$, where each constant symbol e_i in $F_{\sqsubseteq}$ denotes the corresponding set M_i, $M_i \in D_{\sqsubseteq}$, and the predicate symbol $\sqsubseteq$ denotes the ordinary subset relation $\subseteq$ on $D_{\sqsubseteq} \times D_{\sqsubseteq}$ is a model of $F_{\sqsubseteq}$.

If, for instance, e_i fn e_j, i.e.

$$\exists X\big(e_i(X) \wedge e_j(X)\big) \wedge \exists X\big(e_i(X) \wedge \neg e_j(X)\big) \wedge \exists X\big(\neg e_i(X) \wedge e_j(X)\big)$$

is true under I_m,

$$e_i \not\sqsubseteq e_j \wedge e_j \not\sqsubseteq e_i \wedge \exists Z(Z \sqsubseteq e_i \wedge Z \sqsubseteq e_j)$$

is true under $I_{\sqsubseteq}$ as well, because $M_i \not\subseteq M_j$, $M_i \not\subseteq M_j$, $\mathrm{FN}_{ij} \subseteq M_i$ and $\mathrm{FN}_{ij} \subseteq M_j$.

$\Leftarrow$ If $F_{\sqsubseteq}$ has a model with domain $D_{\sqsubseteq}$ and interpretation $I_{\sqsubseteq}$, we can construct a model of F_m with domain D_m and interpretation I_m in the following way:

$$
\begin{aligned}
D_m &= \bigcup\{M_i \mid e_i \text{ occurring in } F_{\sqsubseteq}\}\,; \\
M_i &= \{X \mid X \in D_{\sqsubseteq} \text{ and } X \sqsubseteq e_i \text{ is true under } I_{\sqsubseteq}\}\,.
\end{aligned}
$$

Due to the reflexivity of $\sqsubseteq$ each of the sets M_i is non-empty.

It is straightforward to show that the interpretation I_m over D_m, where each predicate symbol e_i in F_m denotes the corresponding set M_i, $M_i \subseteq D_m$, is a model of F_m.

If, for instance, e_i ex e_j, i.e.

$$\neg\exists Z(Z \sqsubseteq e_i \wedge Z \sqsubseteq e_j)$$

is true under $I_{\sqsubseteq}$,

$$\forall X\big(e_i(X) \supset \neg e_j(X)\big) \wedge \exists X\big(e_i(X)\big) \wedge \exists X\big(e_j(X)\big)$$

is true under I_m as well, because $M_i \cap M_j = \emptyset$ and both M_i and M_j are non-empty. □

9. Consistency checking of $F_{\sqsubseteq}$

For consistency checking, the relationships between the medical entities in the knowledge base KB 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 S_H depicted in *Fig. 2*. It should be noted that the set S_H contains three goals and therefore does *not* directly correspond to clauses used in PROLOG [13] where at most one goal (query) is allowed.

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

Fig. 2. Specification S_H of the consistency checker.

For a set of Horn clauses containing a function symbol positive hyperresolution ('forward chaining') is no longer guaranteed to constitute a decision procedure. Yet, because none of the relationships in the knowledge base of CADIAG-1/BIN contains a variable and therefore no nesting of the function symbol can occur during forward chaining, positive hyperresolution could still be applied to decide the consistency of the knowledge base.

However, for this set of clauses positive hyperresolution is nothing but the computation of the reflexive and transitive closure of the $\sqsubseteq$ relation and the look-up of the corresponding relationships in KB for the goals. Due to the clean interface between the stored relationships and the computation of the reflexive and transitive closure, we can dispose with automated theorem proving techniques altogether. Warshall's algorithm can be applied [7, pp. 550ff] corresponding to an efficient bottom-up computation of the $\sqsubseteq$ relation. This algorithm takes at most time in $O(|E|^3)$, where $|E|$ denotes the number of entities in the knowledge base. For modifications of this algorithm suitable for large knowledge bases KB see [14, pp. 949ff].

Another possibility is to transform the set S_H into a set of definite Horn clauses. Because a set of arbitrary Horn clauses is inconsistent if and only if *one* goal is inconsistent with the subset of definite Horn clauses, each goal $\leftarrow G$ in the set S_H could be replaced by a new rule *false* $\leftarrow G$ to simulate the selection of such a goal. This replacement results in a set of definite Horn clauses S_D corresponding to a PROLOG program. The knowledge base of CADIAG-1/BIN $S_H \cup KB$ is inconsistent if and only if *false* is entailed by the set of clauses $S_D \cup KB$, i.e. $S_D \cup KB \cup \{\leftarrow \mathit{false}\}$ is inconsistent [10]. However, S_D could not directly be applied in a PROLOG system with the query $\leftarrow$ *false* for consistency checking. Due to the SLD resolution employed in PROLOG, the system will encounter an infinite loop for every cycle in the $\sqsubseteq$ relation, for instance, for every oc relationship. However, the required modifications to detect cylces in the $\sqsubseteq$ relation are well known to every PROLOG programmer [2] and will yield a correct and complete top-down algorithm for consistency checking. It should be noted that the worst case of this algorithm is exponential in the number of entities, because there can be an exponential number of different $\sqsubseteq$ paths between two entities e_i and e_j.

10. Extending a consistent knowledge base

Practical applications demand that new relationships can be added to a knowledge base which is already known to be consistent. If the resulting knowledge base becomes inconsistent,

the newly added relationship has to be used during the proof of this fact. To establish the inconsistency of the extended knowledge base

- one of the corresponding $\sqsubseteq$ relations has to be used, if an oc relationship has been added.
- the corresponding instance of the goals, if an ex relationship has been added.
- one of the corresponding $\sqsubseteq$ relations or a corresponding instance of the goals, if an fn, an fc, or an oc relationship has been added.

Obviously, neither the implementation of the consistency checker employing a pure top-down strategy like the modified clause set in PROLOG nor a bottom-up approach like Warshall's algorithm computing the $\sqsubseteq$ relation first, could handle this situation efficiently. If, for instance, a consistent knowledge base is extended by an on relationship, *every* on, fn, fx, and ex relationship in the knowledge base yields a new instance of a goal to be tested by a pure top-down strategy. A bottom-up approach would have to compute the complete reflexive and transitive closure of the $\sqsubseteq$ relation, before establishing the consistency of the extended knowledge base. Instead of trying to optimize the algorithm for consistency checking, we will optimize the formalization itself.

Given a consistent knowledge base $S_H \cup KB$ and an arbitrary relationship R, $S_H \cup KB \cup \{R\}$ is inconsistent if and only if $\neg R$ is a logical consequence of $S_H \cup KB$. Instead of adding R to $S_H \cup KB$ and trying to establish its inconsistency we can try to prove the entailment of $\neg R$ directly from $S_H \cup KB$.

We could prove the entailment of $\neg R$, if we know how to prove each subformula for $\neg R$. Thus we have to solve two related problems:

(a) When are $\sqsubseteq$ relations entailed by $S_H \cup KB$?

(b) When are negated $\sqsubseteq$ relations entailed by $S_H \cup KB$?

How to prove the entailment of $\sqsubseteq$ relations is provided by the axioms of reflexivity and transitivity for the $\sqsubseteq$ relation. Yet there are no explicit axioms for negated $\sqsubseteq$ relations.

If $S_H \cup KB \cup \{e_i \sqsubseteq e_j\}$ is inconsistent and therefore $e_i \not\sqsubseteq e_j$ is a logical consequence of $S_H \cup KB$, at least one of the goals must be used in the proof of the inconsistency.

The general structure of a goal in S_H can be depicted as $\leftarrow ..., X \sqsubseteq Y$. If the instance $\leftarrow ..., e_1 \sqsubseteq e_{j+k}$ is used in the proof of the inconsistency, then $e_1 \sqsubseteq e_{j+k}$ becomes derivable through the addition of the ground fact $e_i \sqsubseteq e_j$ to KB. However, this is only possible if

$$e_1 \sqsubseteq e_2 \sqsubseteq \cdots \sqsubseteq e_i$$

and

$$e_j \sqsubseteq e_{j+1} \sqsubseteq \cdots \sqsubseteq e_{j+k}$$

are already entailed by $S_H \cup KB$. If a fact $e_l \sqsubseteq e_m$ is entailed by $S_H \cup KB$ either this fact or a more general fact containing variables, for instance, $\forall X(e_l \sqsubseteq X)$ is derivable from $S_H \cup KB$ by, for instance, positive hyperresolution.

Instead of adding the fact $e_i \sqsubseteq e_j$ directly to the knowledge base and test for its inconsistency, we can test, if $e_1 \sqsubseteq e_i$, $e_j \sqsubseteq e_{j+k}$ and $\leftarrow ..., e_1 \sqsubseteq e_{j+k}$ are entailed by the knowledge base. Lifting this line of reasoning to the quantificational level and taking the relationships into account, the situation is graphically summarized in *Fig. 3*, where $U \rightarrow V$ denotes that $U \sqsubseteq \cdots \sqsubseteq V$ is entailed by $S_H \cup KB$.

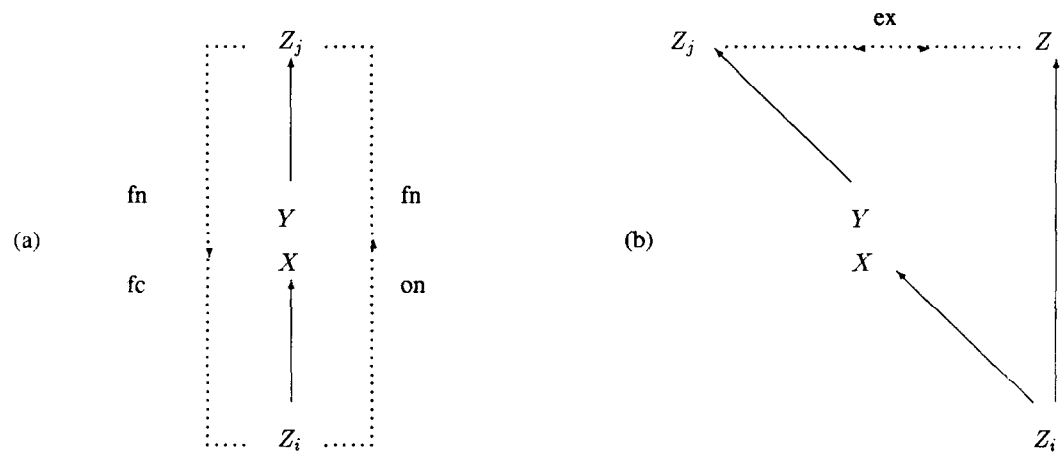

Fig. 3. $X \not\sqsubseteq Y$ is a logical consequence in situations (a) and (b).

If, for instance, $Z_i \sqsubseteq X$, $Y \sqsubseteq Z_j$, and one of the following relationships Z_i fn Z_j, Z_i on Z_j, Z_j fn Z_i, or Z_j fc Z_i are entailed by $S_H \cup KB$ as depicted in *Fig. 3(a)*, the addition of $X \sqsubseteq Y$ would render the knowledge base inconsistent. Therefore, the negation of $X \sqsubseteq Y$ is a logical consequence of $S_H \cup KB$.

Introducing the new two place predicate symbol $\sqsubseteq_{\text{not}}$ where the intended interpretation of $e_i \sqsubseteq_{\text{not}} e_j$ is $e_i \not\sqsubseteq e_j$, the situations in *Fig. 3* can be formalized by the following set of Horn clauses L:

$$
\begin{aligned}
X \sqsubseteq_{\text{not}} Y \quad &\leftarrow \quad Z_i \sqsubseteq X,\ Y \sqsubseteq Z_j,\ Z_i \text{ noSubset } Z_j\ . \\
Z_i \text{ noSubset } Z_j \quad &\leftarrow \quad Z_i \text{ on } Z_j;\ Z_i \text{ fn } Z_j;\ Z_j \text{ fc } Z_i;\ Z_j \text{ fn } Z_i\ . \\
Z_i \text{ noSubset } Z_j \quad &\leftarrow \quad Z_i \sqsubseteq Z,\ (Z_j \text{ ex } Z;\ Z \text{ ex } Z_j)\ . \\
&\leftarrow \quad X \sqsubseteq Y,\ X \sqsubseteq_{\text{not}} Y\ .
\end{aligned}
$$

The replacement of all goals in S_H by the set L yields the new specification S'_H of the consistency checker.

Theorem 2. *$S'_H \cup KB$ is inconsistent if and only if $S_H \cup KB$ is inconsistent.*

Proof.

$\Rightarrow$ The set $L \subseteq S'_H$ can be transformed to the set of goals in S_H. Therefore every proof of the empty clause in $S'_H \cup KB$ can be transformed into a proof of the empty clause in $S_H \cup KB$.

Expanding ('unfold') $X \sqsubseteq_{\text{not}} Y$ in the single goal of S'_H by its definition, yields the goal

$$\leftarrow Z_i \sqsubseteq X, Y \sqsubseteq Z_j, X \sqsubseteq Y, Z_i \text{ noSubset } Z_j\ .$$

Instead of deriving the literals $Z_i \sqsubseteq X$, $X \sqsubseteq Y$ and $Y \sqsubseteq Z_j$ directly, each of them can be established implicity in deriving $Z_i \sqsubseteq Z_j$ by the reflexivity and transitivity of the $\sqsubseteq$ relation. Therefore the first three literals in this goal can be replaced ('folded') with the single literal

$Z_i \sqsubseteq Z_j$ yielding the goal $\leftarrow Z_i \sqsubseteq Z_j$, Z_i noSubset Z_j. If the literal Z_i noSubset Z_j is substituted by its definition, the set of goals from *Fig. 2* will result.

$\Leftarrow$ A similar argument shows that the goals in S_H can be transformed into the set L. □

As the proof of the last theorem clearly indicates the set S'_H is nothing but a rewriting of the set S_H. In using S'_H instead of S_H for consistency checking nothing would be lost or gained. However, the idea behind introducing the set L was to replace questions concerning negated $\sqsubseteq$ relations by questions concerning the corresponding $\sqsubseteq_{\text{not}}$ relation. In testing the entailment of $\sqsubseteq_{\text{not}}$ relations S'_H focuses on the relevant $\sqsubseteq$ relations instead of taking a priori the whole knowledge base into account. In, for instance, the test of the entailment of $e_i \sqsubseteq_{\text{not}} e_j$ the sets

$$\begin{aligned} S_i &= \{e_l \mid e_l \sqsubseteq e_i\}, \\ S_j &= \{e_k \mid e_j \sqsubseteq e_k\}, \\ S_{\text{ex}} &= \{e_m \mid e_n \sqsubseteq e_m,\ e_n \in S_i\} \end{aligned}$$

are involved, which can be computed efficiently by standard algorithms (e.g. depth first search).

The last theorem guarantees that whenever $\forall^*(U \sqsubseteq_{\text{not}} V)$, where U and V denote variables or constant symbols, is entailed by $S'_H \cup KB$ then $\forall^*(U \not\sqsubseteq V)$ is a logical consequence of $S_H \cup KB$. However, this theorem does *not* guarantee that for every negated $\sqsubseteq$ relation the corresponding $\sqsubseteq_{\text{not}}$ relation is entailed by $S'_H \cup KB$. If the knowledge base KB consists of the single relationship e_1 ex e_2, then $\exists X(e_3 \not\sqsubseteq X)$ is a logical consequence of $S_H \cup \{e_1$ ex $e_2\}$, because $e_3 \sqsubseteq e_1$ and $e_3 \sqsubseteq e_2$ could not hold at the same time. Yet, $\exists X(e_3 \sqsubseteq_{\text{not}} X)$ is not derivable from $S'_H \cup \{e_1$ ex $e_2\}$, i.e. $S'_H \cup \{e_1$ ex $e_2\} \cup \{\leftarrow e_3 \sqsubseteq_{\text{not}} X\}$ is *not* inconsistent, because variables occurring in a derived $\sqsubseteq_{\text{not}}$ relation are universally quantified and neither $e_3 \sqsubseteq e_1$ nor $e_3 \sqsubseteq e_2$ is entailed by $S'_H \cup KB$. This example should warn especially PROLOG programmers. Since no definite answer substitution is possible, the goal $\leftarrow e_3 \sqsubseteq_{\text{not}} X$ will fail, although the corresponding negated $\sqsubseteq$ relation is a logical consequence.

Theorem 3. *If $\forall^*(U \not\sqsubseteq V)$ where U and V denote constant symbols or variables, is entailed by a consistent set $S_H \cup KB$, then $\forall^*(U \sqsubseteq_{\text{not}} V)$ is entailed by $S'_H \cup KB$.*

Proof. If $\forall^*(U \not\sqsubseteq V)$ is entailed by $S_H \cup KB$, then its negation $\exists^*(U \sqsubseteq V)$ and $S_H \cup KB$ are inconsistent. Because the negation contains no universal quantifier, its Skolemization yields a ground fact $e_i \sqsubseteq e_j$. Due to the consistency of $S_H \cup KB$, $e_i \sqsubseteq e_j$ must be used in the proof of the inconsistency of $S_H \cup KB \cup \{e_i \sqsubseteq e_j\}$. Yet, situations where through the addition of a ground fact the empty clause becomes derivable, lead to the introduction of the set L. □

Due to Theorem 2 we are guaranteed that the axioms S'_H are correct, and due to Theorem 3 that if $S_H \cup KB$ is consistent the axioms are complete for universally quantified variables occurring in negated $\sqsubseteq$ relations, and complete for existentially quantified variables occurring in negated $\sqsubseteq$ relations as long as a definite substitution of the existentially quantified variables is possible, for instance, $\exists X(X \not\sqsubseteq e_j)$ implies $e_i \not\sqsubseteq e_j$ for some constant symbol e_i.

The formalization S_H was introduced to allow for an efficient testing of the entailment of a negated relationship. The negation of the oc, on, fc, and ex relationships is easy to derive

and is represented by the following set of Horn clauses:

$$X \text{ not\_oc } Y \leftarrow X \sqsubseteq_{\text{not}} Y;\ Y \sqsubseteq_{\text{not}} X\ .$$

$$X \text{ not\_on } Y \leftarrow X \sqsubseteq Y;\ Y \sqsubseteq_{\text{not}} X\ .$$

$$X \text{ not\_fc } Y \leftarrow X \sqsubseteq_{\text{not}} Y;\ Y \sqsubseteq X\ .$$

$$X \text{ not\_ex } Y \leftarrow Z \sqsubseteq X;\ Z \sqsubseteq Y\ .$$

The testing for the negation of the fn relationship is more difficult and will be presented in detail below. The negated definition of the fn relationship

$$\forall^*\Big(X \text{ not\_fn } Y \equiv \neg\Big(X \not\sqsubseteq Y \wedge Y \not\sqsubseteq X \wedge \exists Z(Z \sqsubseteq X \wedge Z \sqsubseteq Y)\Big)\Big)$$

can be rewritten as

$$\forall^*\Big(X \text{ not\_fn } Y \doteq X \sqsubseteq Y \vee Y \sqsubseteq X \vee \neg\exists Z(Z \sqsubseteq X \wedge Z \sqsubseteq Y)\Big)\ .$$

The first two subformulas of the disjunction yield the Horn clauses

$$X \text{ not\_fn } Y \leftarrow X \sqsubseteq Y;\ Y \sqsubseteq Z\ .$$

Obviously, the last subformula of the disjunction $\neg\exists Z(Z \sqsubseteq X \wedge Z \sqsubseteq Y)$ could not be used directly in the body of a rule. But this subformula is nothing but the definition of the ex relationship for X and Y, i.e. X ex Y.

Given a relationship e_i fn e_j, we have therefore to test if e_i ex e_j is entailed by a consistent set $S_H \cup KB$. $\neg\exists Z(Z \sqsubseteq e_i \wedge Z \sqsubseteq e_j)$, is entailed by $S_H \cup KB$, if $S_H \cup KB$ extended by the Skolemized negation of this formula corresponding to the two ground facts $@g(e_i, e_j) \sqsubseteq e_i$ and $@g(e_i, e_j) \sqsubseteq e_j$ is inconsistent. If $S_H \cup KB$ is consistent,

$$S_H \cup KB \cup \{(@g(e_i, e_j) \sqsubseteq e_i\} \cup \{@g(e_i, e_j) \sqsubseteq e_j\}$$

is inconsistent if and only if

(a) $@g(e_i, e_j) \not\sqsubseteq e_i$ is entailed by $S_H \cup KB$, or

(b) $S_H \cup KB \cup \{(@g(e_i, e_j) \sqsubseteq e_i\}$ is consistent and entails $@g(e_i, e_j) \not\sqsubseteq e_j$.

Because the newly introduced constant symbol $@g(e_i, e_j)$ does not occur in $S_H \cup KB$, $@g(e_i, e_j) \not\sqsubseteq e_i$ is a logical consequence of $S_H \cup KB$ if and only if $\forall X(X \not\sqsubseteq e_i)$ is entailed by $S_H \cup KB$. However, if $\forall X(X \not\sqsubseteq e_i)$ is entailed by $S_H \cup KB$, then $S_H \cup KB$ has to be inconsistent, because $e_i \sqsubseteq e_i$ is a logical consequence of the reflexivity of the $\sqsubseteq$ relation. Therefore, to test if $\neg\exists Z(Z \sqsubseteq e_i \wedge Z \not\sqsubseteq e_j)$ is entailed by a *consistent* set $S_H \cup KB$, it suffices to consider case (b) and add $@g(e_i, e_j) \sqsubseteq e_i$ to $S_H \cup KB$ and test if $@g(e_i, e_j) \not\sqsubseteq e_j$ is entailed by this set. The last formula contains no existentially quantified variable. Therefore we can add $@g(e_i, e_j) \sqsubseteq e_i$ to our new formalization $S'_H \cup KB$ and test for the entailment of $@g(e_i, e_j) \sqsubseteq_{\text{not}} e_j$.

A *procedural* specification for testing the relationship not_fn provides the following PROLOG program, where *assert*/1 and *retract*/1 denote the addition and deletion of a clause, respectively, and *fail*/0 initiates backtracking.

$$
\begin{aligned}
X \text{ not\_fn } Y &\leftarrow X \sqsubseteq Y;\ Y \sqsubseteq Z\,. \\
X \text{ not\_fn } Y &\leftarrow \textit{assert\&remove\_on\_backtrack}(@g(X,Y) \sqsubseteq X)\,, \\
&\qquad @g(X,Y) \sqsubseteq_{\text{not}} Y\,, \\
&\qquad \textit{retract}(@g(X,Y) \sqsubseteq X)\,.
\end{aligned}
$$

$$
\begin{aligned}
\textit{assert\&remove\_on\_backtrack}(X) &\leftarrow \textit{assert}(X)\,. \\
\textit{assert\&remove\_on\_backtrack}(X) &\leftarrow \textit{retract}(X), \textit{fail}\,.
\end{aligned}
$$

To summarize this section, the formalization S'_H allows for an efficient testing if the addition of a relationship to a consistent knowledge base would render the extended knowledge base inconsistent. Although only demonstrated for the ex relationship, the new formalization S'_H could be used for the other types of relationships as well to check if a newly to be added relationship is already a logical consequence of the given knowledge base.

11. Consistency checking of CADIAG-1/BIN

To check the consistency of the knowledge base of CADIAG-1/BIN we implemented an algorithm based on the formalization S'_H which actually rebuilds the entire knowledge base anew. After creating a new knowledge base the algorithm

(a) tests if a relationship taken from the old knowledge base of CADIAG-1/BIN is already entailed by the new knowledge base; and

(b) adds this relationship to the new knowledge base if the negated relationship is not a logical consequence of it.

The results are

(a) a consistent knowledge base,

(b) a list of entailed relationships; and

(c) a list of relationships inconsistent with the knowledge base together with their corresponding proofs.

We choose this approach to simulate consistency checking during the knowledge acquisition stage to demonstrate its feasibility. At a first glance such an approach seems to be hopelessly inefficient to check the consistency of an entire knowledge base compared to the algorithms discussed in Section 9. However, one should keep in mind that certain sets of relationships are a priori known to be consistent and therefore must be never checked for consistency. It is easy to prove that e.g. every set consisting solely of relationships e_i fn e_j, $i \neq j$, is consistent and that only fn relationships symmetric to those already contained in the set are logical consequences of it. Therefore to check the consistency of the entire knowledge base of CADIAG-1/BIN, our algorithms do not have to start with an empty new knowledge base, but can use a much larger set of relationships as a starting point for the rebuilding process.

At the first run the consistency checking program detected 17 inconsistencies and marked 530 relationships as logical consequences. Of these 17 inconsistencies four are clearly due to typing errors. Some of the remaining inconsistencies are discussed in more detail below.

Figure 4 depicts a section from the knowledge base of CADIAG-1/BIN that was proved to be inconsistent. The inconsistency may be caused by at least one of the relationships. Which relationship has to be corrected can only be decided semantically, i.e. by the expert physician. In our example here, the fn relationship between 'affection of any joint, presently' and 'coxarthrosis' had to be substituted by an on relationship. The reason is if the 'affection of the hip joint, presently' is obligatory (on relationship) to establish the diagnosis of a 'coxarthrosis' and the finding 'affection of any joint, presently' is considered to be a super-term of 'affection of the hip joint, presently' (fc relationship) then the 'affection of any joint, presently' is also obligatory (on relationship) for 'coxarthrosis'.

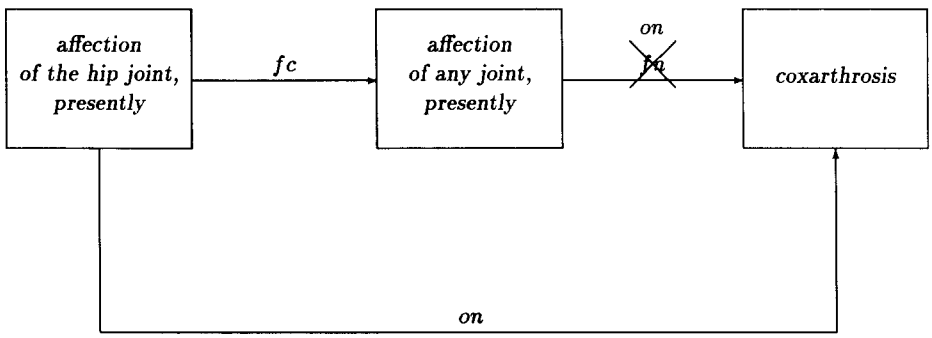

Fig. 4. Example 1: An inconsistent set of relationships.

This inconsistency may be interpreted as a mere inexactness by the physician establishing the knowledge base. However, the same pattern of inconsistency occurred several times. This indicates that the logical concept of hierarchical relationships often do not exactly correspond to conventions established in medicine.

In order to illustrate this, consider the following example: Physicians have always endeavoured to find the most detailed information regarding some pathological sign during the physical examination of the patient. So, for instance, he/she will not write 'varicose veins of lower extremities' in the patient record if there is a 'varicose ulcer'. Since deciding for 'varicose veins' would mean that there is *no* 'ulcer'. But as seen from a logical point of view, 'varicose veins' is a super-term for several alterations of the skin and the subcutaneous tissue including 'ulcer'. Thus 'varicose ulcer' implies 'varicose veins' and both signs are present in the patient. So, it was necessary to realize that all super-terms of obligatory signs for diseases are obligatory for these disease as well. This logical point of view is sometimes in conflict with clinical practice where using the super-term to assess the state of a patient implicitly excludes the subterm to be true for the patient (otherwise it would have been used to describe the state of the patient).

A further error is depicted in *Fig. 5*. In patients with 'arthropathy with hemophilia' the clotting time is sometimes increased and sometimes even highly increased but per definitionem either the one or the other is present (ex relationship). This inconsistency was the starting point to reconsider the concept of exclusions between the sub-categories of a finding, which

usually are increased, highly increased, decreased, and highly decreased in CADIAG-1/BIN. The problem is that there is no general rule to decide whether highly increased means also increased or whether highly increased is different from increased and thus mutually exclusive. We, finally, decided to consider highly increased as being a subconcept of increased, and highly decreased as a subconcept of decreased. In our example, highly increased implies increased (fc relationship) and highly increased may then have an fn relationship to 'arthropathy with hemophilia'.

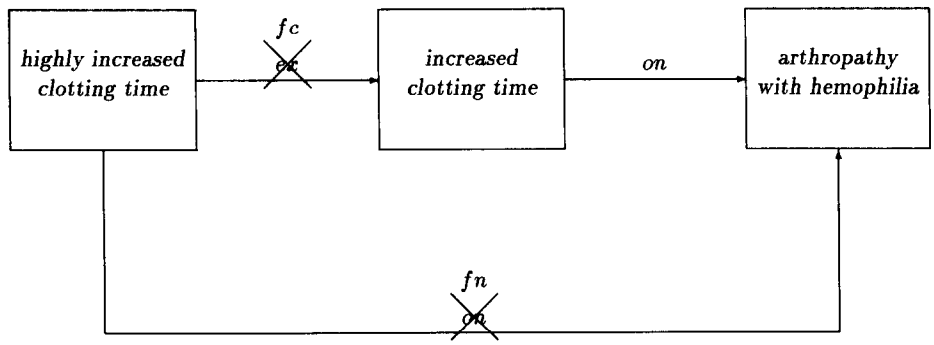

Fig. 5. Example 2: An inconsistent set of relationships.

The medical problem causing the invalid entry into the medical knowledge base depicted in *Fig. 6* is obvious. Patients suffering from 'Morbus Wilson' show a 'normal CU level in urine'. Therefore, an on relationship between 'normal CU in urine' and 'Morbus Wilson' was entered to describe the prototypical concept of the disease. But there are rare clinical cases of 'Morbus Wilson' in which 'increased' or 'highly increased levels CU in urine' are found. Therefore, the on relationship could not be longer kept and was substituted by the weaker relationship fn.

Each of the inconsistencies caused an extended discussion among the participating physicians and computer scientists about various medical concepts and their adequate formalization in CADIAG-1/BIN. But finally, all inconsistencies were removed from the knowledge base.

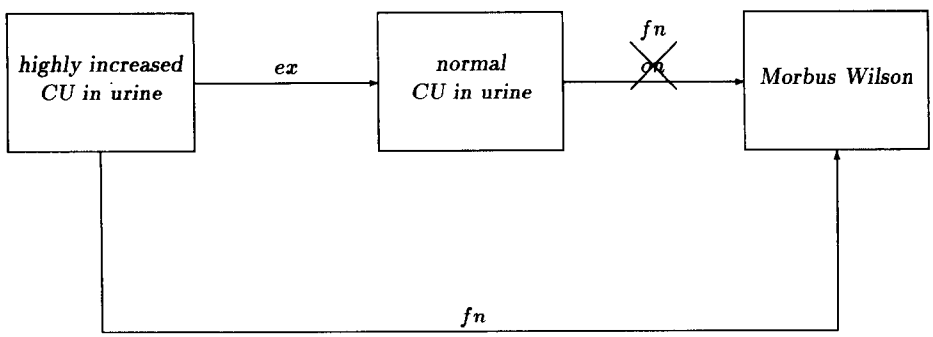

Fig. 6. Example 3: An inconsistent set of relationships.

12. Conclusions

The paper on hand presents a new formalization of the five types of relationships in CADIAG-1/BIN which leads to remarkably simple algorithms 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 the knowledge base of CADIAG-1/BIN. 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 [4]) into the relationship categories of CADIAG-1 makes the developed consistency checking algorithms broadly applicable.

Acknowledgements

The authors are indebted to G. Kolarz, M.D., for his manifold and patient help in discussing all occurring medical issues.

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 (1985) 315–335.

[2] K.R. Apt, R.N. Bol and J.W. Klop, On the safe termination of Prolog programs, in: G. Levi and M. Martelli, eds., *Logic Programming: Proceedings Sixth Internat. Conf.* (MIT Press, Cambridge, MA, 1989) 353–368.

[3] M. Baaz, Automatisches Beweisen in Logiksystemen, in denen Widersprüche behandelt werden können, in: E. Buchberger and J. Retti, eds., *Proc. 3. Österreichische Artificial-Intelligence-Tagung* (Springer, Berlin, 1987) 176–181.

[4] 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 Internal Med.* 110 (1989) 824–832.

[5] F. Barachini and K.-P. Adlassnig, CONSDED: Medical knowledge base consistency checking, in: A. Scrio, R. O'Moore, A. Tardini and F.H. Roger, eds., *Proc. Medical Informatics Europe 87*, Rome (1987) 974–980.

[6] E. Börger, *Berechenbarkeit, Komplexität, Logik* (Friedrich Vieweg & Sohn, Braunschweig, 1986).

[7] T.H. Cormen, C.E. Leierson and R.L. Rivest, *Introduction to Algorithms* (MIT Press, Cambridge, MA, 1990).

[8] N.C.A. da Costa and V.S. Subrahmanian, Paraconsistent logics as a formalism for reasoning about inconsistent knowledge bases, *Artificial Intelligence in Med.* 1 (1989) 167–174.

[9] D.W. Loveland, *Automated Theorem Proving: A Logical Basis* (North-Holland, Amsterdam, 1978).

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

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

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

[13] L. Sterling and E. Shapiro, *The Art of Prolog* (MIT Press, Cambridge, MA, 1986).

[14] J.D. Ullmann, *Principles of Database and Knowledge-Base Systems, Volume II: The New Technologies* (Computer Science Press, Rockville, MD, 1989).