On the Classical Content of Monadic $\mathsf{G}^{\sim}$ and its Application to a Fuzzy Medical Expert System

Agata Ciabattoni
Institute of Computer Languages
Theory and Logic group
Technical University of Vienna
Favoritenstrasse 9-11, 1040 Wien, Austria
agata@logic.at

Pavel Rusnok
Section on Medical Expert and
Knowledge-Based Systems
Medical University of Vienna
Spitalgasse 23, 1090 Wien, Austria
pavel.rusnok@meduniwien.ac.at

Abstract

The satisfiability problem for monadic infinite-valued Gödel logic is known to be undecidable. We identify a fragment of this logic extended with strong negation whose satisfiability is not only decidable but it is decidable within classical logic. We use this fragment to formalize the rules of CADIAG-2, a well performing fuzzy expert system assisting in the differential diagnosis in internal medicine. A (classical) satisfiability check of the resulting formulas allowed the detection of some errors in the rules of the system.

1. Introduction

Gödel logics are one of the oldest families of many-valued logics. Introduced by Gödel in 1932 to show that Intuitionistic logic does not have a characteristic finite matrix, they naturally turn up in a number of different contexts; among them relevance logics (Dunn and Meyer 1971), the provability logic of Heyting arithmetic (Visser 1982) and strong equivalence in logic programming (Lifschitz et al. 2001). Infinite-valued Gödel logic G was recognized as one of the main formal systems for reasoning under vagueness, see (Hájek 1998). In this context a special role is played by its monadic fragment, i.e., the fragment of first-order G with no function symbols and in which all predicates are unary. This fragment provides indeed a formalization of the popular concept of a *fuzzy IF-THEN rule*, like: "IF $A(x)$ and $B(x)$ THEN $C(x)$", where the predicates A, B, and C are fuzzy, i.e., they apply to x possibly only to some degree.

The addition of a classical, involutive, negation $\sim$ to G defines a more expressive logic denoted $\mathsf{G}^{\sim}$. See, e.g., (Esteva et al. 2000; Flaminio and Marchioni 2006) for a mathematical investigation of G (and of t-norm[1] based logics, in general) extended with $\sim$.

In contrast to classical logic, the satisfiability problem for monadic G is undecidable (Baaz et al. 2009). This makes problematic the actual use of this logic (with and without $\sim$) to formalize and reason about real-life systems.

In this paper we identify a non-trivial fragment of monadic $\mathsf{G}^{\sim}$ – we denote it by $\Sigma_{cl}^{\mathsf{G}^{\sim}}$ – whose satisfiability is not only decidable but it is decidable within classical logic. $\Sigma_{cl}^{\mathsf{G}^{\sim}}$ is powerful enough to formalize interesting features of certain fuzzy rule-based systems which seem to lack answer set semantics in the style, e.g., of (Mateis 1999; Straccia 2008; Łukasiewicz 2008; Janssen et al. 2009). These systems can instead be analyzed within $\mathsf{G}^{\sim}$ and some of their properties can be checked using well established tools for classical logic.

As a case study, we consider CADIAG-2 (Computer-Assisted DIAGnosis), a large data-driven fuzzy medical expert system built for computer-based consultation in internal medicine. The system, designed and implemented at the Medical University of Vienna, is integrated into the information system of the Vienna General Hospital, see (Adlassnig et al. 1986; Adlassnig and Kolarz 1986a; Adlassnig et al. 1985). CADIAG-2's knowledge base contains more than 20.000 rules expressing relationships between medical entities, i.e., patient's symptoms, signs, laboratory test results, clinical findings and diagnoses. CADIAG-2's rules are classified into four types: c_d "confirming to the degree d", me "mutually exclusive", ao "always occurring" and oc "obligatory confirming". These follow the general patterns below:

c_d IF A THEN B with the degree c

me IF A THEN not B

ao IF not A THEN not B

oc IF A THEN B AND IF not A THEN not B

where $c \in (0, 1]$, B is an atomic entity and A is built from atomic entities using "not", "and" and "or". In contrast with its predecessor system CADIAG-1, CADIAG-2 deals with possibly graded entities, e.g., the symptoms having "strong abdominal pain" or "suspicion of pancreatic tumor by CT"; these are represented as suitable numbers in $[0, 1]$ mostly calculated using predefined fuzzy sets. As shown in (Ciabattoni and Vetterlein 2010), the logic closest to the natural interpretation of the entities and rules of CADIAG-2 is $\mathsf{G}^{\sim}$.

CADIAG-1's entities and rules are boolean (crisp). A translation of the binary rules of the system into suitable formulas of monadic classical logic allowed the detection of some errors. See (Moser and Adlassnig 1992) for details.

Here we perform a similar investigation for the fuzzy system CADIAG-2. We first formalize its knowledge base as formulas of $\mathsf{G}^{\sim}$. By using the fragment $\Sigma_{cl}^{\mathsf{G}^{\sim}}$ of $\mathsf{G}^{\sim}$ we show that most of its rules, including all binary ones, can be ana-

[1] T-norms are the main tool in fuzzy logic to combine vague information.

lyzed within classical logic. The actual check is performed using the (counter)model generator *Mace4* and the theorem prover *Prover9* of (McCune). This allowed the identification of some errors in the knowledge representation of various rules of CADIAG-2.

2. Preliminaries on $\mathsf{G}^\sim$

Infinite-valued first-order Gödel logic G, sometimes also called Intuitionistic fuzzy logic (Takeuti and Titani 1984), arises from Intuitionistic logic by adding the axioms $(A \to C) \vee (C \to A)$ and $\forall x(A \vee B^x) \to (\forall x A) \vee B^x$, where B^x means that the variable x does not occur free in B.

Since (Hájek 1998), G has been recognized to be one of the main formalizations of Fuzzy Logic. See, e.g., (Baaz et al. 2007) for more information about Gödel logic—its winding history, importance and variants.

An *interpretation* v for Gödel logic maps constants and object variables to elements of a *domain* $\mathcal{D}$, n-ary function symbols to functions from $\mathcal{D}^n$ into $\mathcal{D}$, and n-ary predicate symbols p to functions from $\mathcal{D}^n$ into $[0,1]$.

The interpretation v extends in the usual way to a function mapping all the terms of the language to an element of the domain. v evaluates atomic formulas $q \equiv p(t_1, \ldots, t_n)$ as

$$v(q) = v(p)(v(t_1), \ldots, v(t_n))^2$$

Extension to all formulas is given by

$$v(A \to B) = \begin{cases} 1 & \text{if } v(A) \le v(B) \\ v(B) & \text{otherwise,} \end{cases}$$

$$v(\neg A) = \begin{cases} 1 & \text{if } v(A) = 0 \\ 0 & \text{otherwise,} \end{cases}$$

$$v(A \wedge B) = \min(v(A), v(B))$$
$$v(A \vee B) = \max(v(A), v(B))$$

To assist a concise formulation of the semantics of quantifiers we define the *distribution* of a formula A and a free variable x with respect to an interpretation v as $\mathrm{Distr}(A(x)) = \{v'(A(x)) : v' \sim_x v\}$, where $v' \sim_x v$ means that v' is exactly as v with the possible exception of the domain element assigned to x. The semantics of quantifiers is given by the infimum and supremum of the corresponding distribution, that is

$$v(\forall x A(x)) = \inf \mathrm{Distr}(A(x))$$
$$v(\exists x A(x)) = \sup \mathrm{Distr}(A(x)).$$

Given any predicate p, in G we can express, that for each interpretation there exists an element of the domain for which

$(= 0)$ p is equal to 0: take the formula $\exists x \neg p(x)$,

$(\neq 0)$ p is different from 0: take the formula $\exists x \neg\neg p(x)$

while we cannot express that "p is equal to 1" (notice that the formula $\exists x p(x)$ can take value 1 in an interpretation v even when for no element d of the domain $v(p(d)) = 1$).

Gödel logic was extended with a classical, involutive negation $\sim$ in (Esteva et al. 2000; Flaminio and Marchioni 2006). An interpretation $v_{\mathsf{G}^\sim}$ in the resulting logic –we refer to it as $\mathsf{G}^\sim$– extends any G interpretation v with

$$v_{\mathsf{G}^\sim}(\sim A) = 1 - v_{\mathsf{G}^\sim}(A).$$

Clearly an interpretation in classical logic is a particular case of a $\mathsf{G}^\sim$ interpretation.

$\mathsf{G}^\sim$ is strictly more expressive than G. E.g., in $\mathsf{G}^\sim$ we can express that given any predicate p, for each interpretation there exists an element of the domain for which

$(= 1)$ p is equal to 1: take the formula $\exists x \neg \sim p(x)$

$(\neq 1)$ p is different from 1: take the formula $\exists x \neg\neg \sim p(x)$

The following laws hold in $\mathsf{G}^\sim$ ($X \leftrightarrow Y$ abbreviates the formula $X \to Y \wedge Y \to X$):

Proposition 1 *For every formulas A, B, C of $\mathsf{G}^\sim$ we have*

1. $\sim (A \wedge B) \leftrightarrow \sim A \vee \sim B$
2. $\sim (A \vee B) \leftrightarrow \sim A \wedge \sim B$
3. $\sim\sim A \leftrightarrow A$
4. $(A \wedge (B \vee C)) \leftrightarrow ((A \wedge B) \vee (A \wedge C))$

As in the classical case, a formula A is *satisfiable* in G (resp. $\mathsf{G}^\sim$), if there is an interpretation v of G (resp. $\mathsf{G}^\sim$) that *satisfies* A, i.e., such that $v(A) = 1$. Let Γ be a set of formulas. Γ is *unsatisfiable* in G (resp. $\mathsf{G}^\sim$) if there is no interpretation v of G (resp. of $\mathsf{G}^\sim$) that satisfies A, for all $A \in \Gamma$.

Henceforth we will indicate with SAT the satisfiability problem for a logic or a set of formulas.

3. A classical decidable fragment of $\mathsf{G}^\sim$

As is well known, SAT in classical monadic logic is decidable. This does not hold anymore for monadic G that was shown in (Baaz et al. 2009) to be undecidable in presence of at least three different predicate symbols. The aim of this section is to identify a useful fragment of monadic $\mathsf{G}^\sim$

(dec) for which SAT is a decidable problem and

($\mathrm{sat}_{\mathrm{CL}}$) with the property of being satisfiable in $\mathsf{G}^\sim$ if and only if it is satisfiable in classical logic.

As shown in Section 4. , the identified fragment is powerful enough to formalize important features of the rules of an existing fuzzy medical expert system.

First notice that in presence of the classical negation $\sim$, the undecidability result for G can be strengthened as follows:

Proposition 2 *SAT for monadic formulas of $\mathsf{G}^\sim$ is undecidable in presence of at least two predicate symbols.*

Proof: Proceeds as the proof in (Baaz et al. 2009) of the undecidability of SAT for monadic G extended with the modality Δ of (Baaz 1996), being Δ derivable in $\mathsf{G}^\sim$.

The claim indeed follows by a faithful embedding in monadic $\mathsf{G}^\sim$ of the classical theory **CE** of two equivalence relations ($\equiv_1$ and $\equiv_2$), known to be undecidable since (Rogers 1956). The idea is $(*)$ to translate each atomic **CE** formula of the form $x \equiv_i y$ into a formula $\neg \sim (p_i(x) \leftrightarrow p_i(y))$, $i = 1, 2$, of the monadic fragment of $\mathsf{G}^\sim$, where p_1 and p_2 are different unary predicate symbols. Notice that for

[2] To simplify the notation we will write $v(p(d_1, \ldots, d_n))$ where d_i stands for $v(t_i)$ for $i = 1, \ldots, n$.

each interpretation $v_{\mathsf{G}^\sim}$ of $\mathsf{G}^\sim$ the above formula of $\mathsf{G}^\sim$ can only take value 0 or value 1. More precisely, for each $\mathsf{G}^\sim$ interpretation v

$$v(\neg \sim (p_i(x) \leftrightarrow p_i(y))) = \begin{cases} 1 & \text{if } v(p_i(x)) = v(p_i(y)) \\ 0 & \text{otherwise,} \end{cases}$$

Let S be a generic formula in **CE** (that, without loss of generality, we can assume to be prenex that is in which quantifiers are always in front)

$$\mathsf{Q}^* \bigwedge (\bigwedge_j x_j \equiv y_j \rightarrow \bigvee_k x'_k \equiv y'_k),$$

where each occurrence of $\equiv$ is either $\equiv_1$ or $\equiv_2$, and Q^* is a string $(\mathsf{Q}_1\mathsf{z}_1)\ldots(\mathsf{Q}_n\mathsf{z}_n)$ of n quantifier occurrences. I.e., for all $l = 1, \ldots, n$, $\mathsf{Q}_l \in \{\forall, \exists\}$, and $\mathsf{z}_l \in \{x_l, y_l, x'_l, y'_l\}$. Let $S^\sharp$ be the formula in $\mathsf{G}^\sim$ obtained by making in S the replacement $(*)$ above. We show that S is satisfiable in classical logic if and only if $S^\sharp$ is satisfiable in $\mathsf{G}^\sim$.
($\Leftarrow$) Assume that there exists an interpretation $v_{\mathsf{G}^\sim}$ in $\mathsf{G}^\sim$ such that $v_{\mathsf{G}^\sim}(S^\sharp) = 1$. Define the interpretation v_{CL} in classical logic as ($i = 1, 2$)

$$v_{CL}(x \equiv_i y) = 1 \text{ iff } v_{\mathsf{G}^\sim}(p_i(x)) = v_{\mathsf{G}^\sim}(p_i(y))$$

It is easy to see that $v_{CL}(S) = 1$.
($\Rightarrow$) Assume that there exists an interpretation v_{CL} in classical logic such that $v_{CL}(S) = 1$. By the downwards Löwenheim-Skolem theorem we can assume its domain $\mathcal{D}$ to be countable. Therefore so is the set of equivalence classes $[x]_i = \{y \mid v_{CL}(x \equiv_i y) = 1\}$ with respect to the two equivalence relations ($i = 1, 2$). An interpretation $v_{\mathsf{G}^\sim}$ of $\mathsf{G}^\sim$ that satisfies $S^\sharp$ is hence defined as

$$v_{\mathsf{G}^\sim}(p_i(x)) = \lambda([x]_i),$$

where λ is an injective function indexing the equivalence classes, i.e., $\lambda : \{[x]_i : x \in \mathcal{D}\} \rightarrow [0, 1]$. Note indeed that $v_{\mathsf{G}^\sim}(\neg \sim (p_i(x) \leftrightarrow p_i(y))) = 1$ if and only if $v_{CL}(x \equiv_i y) = 1$, and that both are equal to 0, otherwise. □

The same undecidability result also holds for the prenex[3] monadic fragment of $\mathsf{G}^\sim$.

Corollary 3 *SAT for prenex monadic formulas of $\mathsf{G}^\sim$ is undecidable in presence of at least two predicate symbols.*

Proof: Note that $S^\sharp$ in the above proof is a prenex formula. □

The identification of the fragment of monadic $\mathsf{G}^\sim$, powerful enough for formalizing real systems and at the same time satisfying the properties (dec) and (sat_{CL}) above, is driven by the following considerations: (a) the one variable fragment of monadic logic is often enough to capture rule-based systems. Though very simple, already in Gödel logic (with no additional negation $\sim$) this fragment does not fulfill the property (sat_{CL}) above. A simple counterexample is the formula

$$\neg\forall x p(x) \wedge \forall x \neg\neg p(x)$$

[3]Note that G (and therefore $\mathsf{G}^\sim$) does not admit equivalent prenex formulas as in classical logic, see e.g. (Baaz et al. 2007).

that has no model in classical logic while it is satisfiable in G (any interpretation v which assigns to the predicate $p(x)$ a decreasing sequence to 0 satisfies the formula). Let us therefore consider formulas built from conjunction and disjunction of one variable monadic formulas that are prenex. (b) Note however that the prenex formula

$$\exists x(\neg\neg \sim p(x) \wedge \neg\neg p(x))$$

is satisfiable in $\mathsf{G}^\sim$ while it is not in classical logic (any $\mathsf{G}^\sim$ interpretation $v_{\mathsf{G}^\sim}$ in which $0 < v_{\mathsf{G}^\sim}(p(d)) < 1$, for an element d, satisfies the formula).

This leads to the following definition:

Definition 4 *We call $\Sigma_{cl}^{\mathsf{G}^\sim}$ each set containing formulas of monadic $\mathsf{G}^\sim$ generated by the following BNF grammar*

$$\mathcal{A} := \mathcal{Q}_i x P(x) \mid \mathcal{Q}_j x \neg P^\sim(x) \mid \mathcal{A} \vee \mathcal{A} \mid \mathcal{A} \wedge \mathcal{A}$$

where $\mathcal{Q}_i, \mathcal{Q}_j \in \{\forall, \exists\}$ and $P(x)$, $P^\sim(x)$ are as follows ($p_k(x)$ is atomic for all $k \in \mathcal{N}$):

$$\begin{aligned} P(x) :=\ & p_k(x) \mid P(x) \vee P(x) \mid P(x) \wedge P(x) \mid \\ & \mid P(x) \rightarrow P(x) \mid \neg P(x) \\ P^\sim(x) :=\ & p_k(x) \mid \sim p_k(x) \mid P^\sim(x) \vee P^\sim(x) \mid \\ & \mid P^\sim(x) \wedge P^\sim(x) \mid \sim P^\sim(x) \end{aligned}$$

For our decidability proof, we first need the following technical lemma.

Lemma 5 *Let $P(x)$ and $P^\sim(x)$ be as in Definition 4 and $v_{\mathsf{G}^\sim}$ be any interpretation in $\mathsf{G}^\sim$. We can find an interpretation v_{CL} in classical logic such that*

1. *$v_{\mathsf{G}^\sim}(P(x)) = 0$ if and only if $v_{CL}(P(x)) = 0$ and $v_{\mathsf{G}^\sim}(P(x)) > 0$ if and only if $v_{CL}(P(x)) = 1$.*
2. *If $v_{\mathsf{G}^\sim}(P^\sim(x)) = 0$ then $v_{CL}(P^\sim(x)) = 0$.*

Proof: Given any interpretation $v_{\mathsf{G}^\sim}$ in $\mathsf{G}^\sim$, let v_{CL} be the interpretation of classical logic defined as follows: for each atomic formula $p(x)$

$$v_{CL}(p(x)) = \begin{cases} 1 & \text{if } v_{\mathsf{G}^\sim}(p(x)) > 0 \\ 0 & \text{otherwise.} \end{cases}$$

The claims follow by induction on $P(x)$ and $P^\sim(x)$. Both claims trivially hold when $P(x)$ and $P^\sim(x)$ are atomic.

Claim 1: we consider as an example the case when $P(x)$ is $P_1(x) \rightarrow P_2(x)$, other cases being very similar. By the definition of interpretation in $\mathsf{G}^\sim$, $v_{\mathsf{G}^\sim}(P(x)) = 0$ if and only if $v_{\mathsf{G}^\sim}(P_2(x)) = 0$ and $v_{\mathsf{G}^\sim}(P_1(x)) > 0$. By the i.h. $v_{\mathsf{G}^\sim}(P_2(x)) = 0$ if and only if $v_{CL}(P_2(x)) = 0$ and $v_{\mathsf{G}^\sim}(P_1(x)) > 0$ if and only if $v_{CL}(P_1(x)) = 1$; it follows that $v_{\mathsf{G}^\sim}(P(x)) = 0$ if and only if $v_{CL}(P(x)) = 0$. The case $v_{\mathsf{G}^\sim}(P(x)) > 0$ is similar.

Claim 2: assume $P^\sim(x) = \sim p_k(x)$. Then $v_{\mathsf{G}^\sim}(P^\sim(x)) = 0$ if and only if $v_{\mathsf{G}^\sim}(p_k(x)) = 1$ and this implies $v_{CL}(p_k(x)) = 1$ (being $p_k(x)$ atomic); hence $v_{CL}(P^\sim(x)) = 0$. The cases $P^\sim(x) := P_1^\sim(x) \vee P_2^\sim(x)$ or $P^\sim(x) := P_1^\sim(x) \wedge P_2^\sim(x)$ are easy. When $P^\sim(x) = \sim P_1^\sim(x)$ we first push $\sim$ in front of the atoms using the laws of De Morgan and double negation (Proposition 1); the claim then follows by the previous cases. □

Remark 6 *Claim 2 above does not hold for implicative formulas. E.g., the interpretation that assigns* $v_{\mathsf{G}^\sim}(p(x)) = 0.5$ *and* $v_{\mathsf{G}^\sim}(q(x)) = 0$ *is such that* $v_{\mathsf{G}^\sim}(\sim p(x) \to q(x)) = 0$ *while* $v_{CL}(\sim p(x) \to q(x)) = 1$.

Theorem 7 $\Sigma_{cl}^{\mathsf{G}^\sim}$ *is satisfiable in* $\mathsf{G}^\sim$ *if and only if* $\Sigma_{cl}^{\mathsf{G}^\sim}$ *is satisfiable in classical logic.*

Proof: Let Q be any formula in $\Sigma_{cl}^{\mathsf{G}^\sim}$. If Q is satisfiable in monadic classical logic then Q is obviously satisfiable in $\mathsf{G}^\sim$. For the converse direction, consider any interpretation $v_{\mathsf{G}^\sim}$ of $\mathsf{G}^\sim$ such that $v_{\mathsf{G}^\sim}(Q) = 1$. Let v_{CL} be the interpretation in classical logic defined as in the proof of the previous lemma. By $v_{\mathsf{G}^\sim}(Q) = 1$ it follows that some (depending on the nesting of the connectives $\wedge$ and $\vee$, see Proposition 1.4) of the formulas $\mathcal{Q}_i x P(x)$ and $\mathcal{Q}_j x \neg P^\sim(x)$ (cf. Definition 4) evaluate to 1 under $v_{\mathsf{G}^\sim}$. We show that the same formulas evaluate to 1 also under v_{CL}, hence $v_{CL}(Q) = 1$. Consider indeed a formula of the kind $\mathcal{Q}_i x P(x)$. If $\mathcal{Q}_i = \forall$ then for all elements d in the domain $v_{\mathsf{G}^\sim}(P(d)) = 1$ and if $\mathcal{Q}_i = \exists$ then for at least one element d, $v_{\mathsf{G}^\sim}(P(d)) > 0$. By Lemma 5.1 it follows that $v_{CL}(P(d)) = 1$ and hence $v_{CL}(\mathcal{Q}_i x P(x)) = 1$.

Consider a formula of the kind $\mathcal{Q}_j x \neg P^\sim(x)$. By similar considerations as above we have that $v_{\mathsf{G}^\sim}(P^\sim(d)) = 0$ (for all d, when $\mathcal{Q}_j = \forall$ and for at least one d, when $\mathcal{Q}_j = \exists$). The claim follows by Lemma 5.2. □

Corollary 8 *SAT of* $\Sigma_{cl}^{\mathsf{G}^\sim}$ *in* $\mathsf{G}^\sim$ *is decidable.*

Proof: Trivially follows from the theorem above being formulas in $\Sigma_{cl}^{\mathsf{G}^\sim}$ monadic. □

4. An application: CADIAG-2

The mathematical results in the previous section are applied here to check the knowledge base in CADIAG-2, a large and well performing fuzzy medical expert system.

The section is organized as follows: We first describe CADIAG-2. Section 4.2 introduces a logical formalization of CADIAG-2's rules within the logic $\mathsf{G}^\sim$. Section 4.3 analyzes them by checking the consistency of their logical formalization. By using the fragment $\Sigma_{cl}^{\mathsf{G}^\sim}$ of $\mathsf{G}^\sim$ we show that most of the logical formulas representing the system's rules are satisfiable in $\mathsf{G}^\sim$ if and only if they are satisfiable in classical logic. This allows us to perform a consistency check on these formulas using the existing systems *Prover9* and *Mace4* that work for classical logic. Our analysis allowed the detection of some errors among the 20.000 rules of CADIAG-2.

4.1 System Description

CADIAG-2 is a fuzzy expert system assisting in the differential diagnosis in internal medicine. The system has been successfully tested in rheumatology and gastroenterology over 470 patients; the overall accuracy for confirmation and hypotheses generation was calculated to be 90%, see (Adlassnig et al. 1985) for more details.

CADIAG-2 consists of two principal parts: a knowledge base (KB for short) and an inference engine. The KB contains more than 20.000 rules expressing causal relationships between symptoms and/or diagnoses. The inference engine proposes differential diagnoses on the basis of the patient data and the KB.

Let $S_1, \ldots, S_i \ldots S_{1781}$ denote the symptoms and $D_1, \ldots, D_j, \ldots D_{342}$ the diagnoses appearing in the KB. A symptom or a diagnose is called a *basic entity*. CADIAG-2 considers all statements about symptoms to be graded. Namely, to each symptom, there is associated a degree of presence, expressed by an element of the real unit interval $[0, 1]$. A real number in $[0, 1]$ is also associated to diagnoses, though its meaning is better understood in this case as a degree of certainty. Compound entities are defined by means of conjunction, disjunction, involutive negation and the derivable connectives "at least n out of m" and "at most n out of m"; conjunction, disjunction and negation are computed by the system as minimum, maximum and the function $1 - x$.

Let α stand for a possibly compound entity and β for a basic entity. The general definition of a rule in CADIAG-2's KB is as follows:

Definition 9 *A* rule *in CADIAG-2's KB is a 4-tuple*

$$R = \langle \alpha, \beta, soc, foo \rangle,$$

where α *(*β*) is the antecedent (the succedent) of the rule and* soc *(*strength of confirmation*) and* foo *(*frequency of occurrence*) are numbers in the real unit interval* $[0, 1]$.

If α *is a basic entity* R *is called* binary.

The numbers soc and foo are used by the inference engine to determine the value of β according to the value of α. The interpretation of soc and foo proposed in (Adlassnig et al. 1986), is the following: given a set of patients P

$$soc = \frac{\Sigma_a \min\{\alpha(a), \beta(a)\}}{\Sigma_a \alpha(a)}, \tag{1}$$

$$foo = \frac{\Sigma_a \min\{\alpha(a), \beta(a)\}}{\Sigma_a \beta(a)}, \tag{2}$$

where $\alpha(a)$ and $\beta(a)$ are the degrees to which the entities α and β apply to a patient a and the sum Σ_a ranges over all patients in P. The database associated with CADIAG-2 did not contain enough patients for calculating all numbers soc and foo by (1)-(2). For this reason most of these values were estimated by different means, including clinical experience of physicians and books.

CADIAG-2's rules can be classified into four groups: c_d, standing for "confirming to the degree d", me, standing for "mutually exclusive", ao, standing for "always occurring" and oc, standing for "obligatory confirming". The classification is based on the possible values of foo and soc. The following cases can arise:

- when $0 < soc = d \leq 1$ and $0 < foo < 1$ then $R = \mathrm{c}_d$
- when $soc = 0$ and $foo = 0$ then $R = \mathrm{me}$
- when $0 < soc < 1$ and $foo = 1$ then $R = \mathrm{ao}$
- when $soc = 1$ and $foo = 1$ then $R = \mathrm{oc}$

Henceforth we will consider the particular case of rules c_d when $d = 1$ and denote it by c_1.

Examples for each type of rules are:

c_d: $\langle S0641, D069, 0.3, 0.5\rangle$, where $S0641$ stands for "strongly reduced number of thrombocytes" and $D069$ for "systemic lupus erythematosus". The rule can be expressed in natural language as:

IF *strongly reduced number of thrombocytes* THEN *systemic lupus erythematosus* with the degree $d = 0.3$.

me: $\langle S1483, D073, 0, 0\rangle$, where $S1483$ stands for "positive rheumatoid factor" and $D073$ for "seronegative rheumatoid arthritis". The rule can be expressed in natural language as:

IF *positive rheumatoid factor* THEN NOT *seronegative rheumatoid arthritis*

ao: $\langle S0601, D238, 0.001, 1\rangle$, where $S0601$ stands for "Waaler-Rose test, negative" and $D238$ for "juvenile chronic arthritis, polyarticular form, seronegative". The rule can be expressed in natural language as:

IF NOT *Waaler-Rose test, negative* THEN NOT *juvenile chronic arthritis,* *polyarticular form, seronegative*

oc: $\langle D001$ AND $S0293$ AND $S0645, D002, 1, 1\rangle$, where $D001$ stands for "rheumatoid arthritis", $S0293$ for "splenomegaly", $S0645$ for "leukopenia under $4\text{giga}/l$" and $D002$ for "Felty's syndrome". The rule can be expressed in natural language as:

IF (*rheumatoid arthritis* AND *splenomegaly* AND *leukopenia under* $4\text{giga}/l$) THEN *Felty's syndrome* AND IF NOT (*rheumatoid arthritis* AND *splenomegaly* AND *leukopenia under* $4\text{giga}/l$) THEN NOT *Felty's syndrome*

We sketch below how CADIAG-2 determines the values of diagnoses for a given patient. See (Adlassnig et al. 1986; Adlassnig and Kolarz 1986a; Ciabattoni and Vetterlein 2010) for more details. Firstly, the patient's data are transformed into degrees of presence for various symptoms. This is done using the fuzzy and crisp sets defined in the system. The values of compound entities are then calculated according to the system's operations on the basis of the available knowledge.

The rules in CADIAG-2's KB are applied systematically one by one, in an arbitrary order. As we will see, the used rule changes the value of only one specific basic entity; the values of the compound entities built over this one, are then updated. The process is completed when, by the use of any of the rules, there are no changes. The system eventually stops. Let α_p and β_p denote the actual values of the entities α and β before the application of a rule $R = \langle \alpha, \beta, soc, foo\rangle$. Note that, in contrast with α_p, β_p might not be defined. R proposes a new value b for its consequent β, according to the rule's type:

- If R=c_d and $\alpha_p > 0$ then $b = \min\{\alpha_p, soc\}$
- If R=me and $\alpha_p = 1$ then $b = 0$
- If R=ao and $\alpha_p = 0$ then $b = 0$
- If R=oc then $b = \min\{\alpha_p, 1\}$

Notice that a rule oc behaves as a rule c_d or a rule ao, depending on the value α_p.

If β_p is undefined then the value assigned to β, denoted by β_{p+1}, is b. Otherwise, the following function is used to calculate β_{p+1}:

$$\beta_{p+1} = \max^*(\beta_p, b) \quad = \quad \begin{cases} \beta_p & \text{if } 0 < b \leq \beta_p \leq 1 \\ & \text{or } 0 < b < 1 \text{ and } \beta_p = 0 \\ b & \text{otherwise,} \end{cases}$$

Observe that both 0 and 1 are maximal in this context. The situations when ($\beta_p = 0$ and $b = 1$) or ($\beta_b = 1$ and $b = 0$) are called *runtime inconsistencies*. They cause a stop of the system with an error message.

CADIAG-2 is written in Java programming language. Some numbers about its KB: there are over 600 predefined fuzzy sets, 20036 rules of type c_d, 948 rules of type me and 486 rules of type ao. Among the rules, 21407 are binary.

4.2 From rules to $\mathsf{G}^\sim$ formulas

We formalize the rules of CADIAG-2 as suitable formulas in $\mathsf{G}^\sim$. Our formalization is mainly based on the intended interpretation of the two numbers *foo* and *soc* (cf. the equations (1) and (2)) that characterize the rule types in the system. Ideally those numbers should be determined by statistical means on a sufficiently large number of patients. As mentioned before, only some of them have been determined in this way. Intuitively, the actual numbers *foo* and *soc* in the system are correct if *there can be* a set of patients that fulfills them. This notion of "experimental consistency" leads to the following definition.

Definition 10 *Let SP be a set of patients. We say that SP* models *a rule $R = \langle \alpha, \beta, soc, foo\rangle$ of CADIAG-2 if soc and foo are calculated according to the equations (1) and (2).*

SP models *CADIAG-2's KB if SP models all its rules.*

The insufficiency of propositional logic for formalizing the system's rules is in this context clear. The remaining question is: in which logic should we formalize them? In the case of CADIAG-1's rules, classical first-order logic was enough as the basic entities of the system were boolean (crisp): either true or false. For CADIAG-2's rules we have to take into account that the involved entities take their values in $[0, 1]$. This suggests the use of a many-valued logic (fuzzy logic, in the sense of (Hájek 1998)). As observed in (Ciabattoni and Vetterlein 2010), the many-valued logic closest to the natural interpretation of CADIAG-2's entities and rules is $\mathsf{G}^\sim$.

Notice that in this logic, statements of the form $\alpha(a) \leq \beta(a)$ can be modeled in a natural way by using the implication. Moreover given two formulas P and Q, we can express that there exists an element of the domain for which

($<$) the value of Q is strictly less than the value of P; take the formula

$$\exists x((P(x) \to Q(x)) \to Q(x)) \wedge \neg\neg \sim Q(x)),$$

denoted by

$$\exists x(Q(x) \prec P(x));$$

indeed $v((P(x) \to Q(x)) \to Q(x)) = 1$ iff either $v(Q(x)) < v(P(x))$, or $v(Q(x)) = v(P(x)) = 1$. Hence the need of excluding that $v(Q(x)) = 1$ with the formula $\neg\neg \sim Q(x)$.

These facts, together with the formulas $(= 0)$, $(\neq 0)$, $(= 1)$ and $(\neq 1)$ in Section 2., are the main ingredients to transform the rules of CADIAG-2 into suitable formulas of $\mathsf{G}^\sim$. These formulas are then used to check the existence of a set of patients modeling CADIAG-2's KB ("experimental consistency").

Formula Representation

We define below the formulas formalizing CADIAG-2's KB.

CADIAG-2's basic entities can be seen as unary predicates in $\mathsf{G}^\sim$ and we shall identify $S_i(a)$ with the proposition "the symptom S_i is present in a patient a" and, similarly, we will identify $D_j(a)$ with the proposition "the diagnosis D_j applies to a patient a".

Let $R = \langle \alpha, \beta, soc, foo \rangle$ be any rule. Henceforth A (resp. B) will denote the formula representation of the compound entity α (resp. of the basic entity β) in $\mathsf{G}^\sim$. For every rule R of CADIAG-2, the equations (1)-(2) imply that $\alpha(a)$ and $\beta(a)$ are different from zero at least for one patient, otherwise *soc* and *foo* would be undefined. This is expressed by the $\mathsf{G}^\sim$ formulas

$$(DEF_{soc})\ \exists x \neg\neg A(x) \quad \text{and} \quad (DEF_{foo})\ \exists x \neg\neg B(x).$$

Rule me

From $soc = foo = 0$ we conclude that $\Sigma_a \min\{\alpha(a), \beta(a)\} = 0$, i.e., for each patient a either $\alpha(a)$ or $\beta(a)$ is equal to 0. This can be expressed by the $\mathsf{G}^\sim$ formula $\forall x(\neg A(x) \vee \neg B(x))$. Hence the formula representing a rule me is of the form:
$\forall x(\neg A(x) \vee \neg B(x)) \wedge \exists x \neg\neg A(x) \wedge \exists x \neg\neg B(x)$.

Rule c_d

In the general case, we know that in a c_d rule $foo, soc > 0$, i.e., $\Sigma_a \min\{\alpha(a), \beta(a)\} > 0$. This means that there is at least one patient for whom both $\alpha(a)$ and $\beta(a)$ are different from zero; this is accounted by the formula

$$(POS)\ \exists x(\neg\neg A(x) \wedge \neg\neg B(x)).$$

We distinguish two cases:

- $d = soc < 1$. Then $\Sigma_a \min\{\alpha(a), \beta(a)\} < \Sigma_a \alpha(a)$, which means that there exists a patient a such that $\beta(a) < \alpha(a)$. This can be expressed by the formula $\exists x(B(x) \prec A(x))$. Similarly from $foo < 1$ follows that there exists a patient a such that $\alpha(a) < \beta(a)$ and this is expressed by the formula $\exists x(A(x) \prec B(x))$. Observe, that (DEF$_{soc}$) and (DEF$_{foo}$) are implied by (POS). Therefore the formula representation of a rule c_d is:
  $\exists x(\neg\neg A(x) \wedge \neg\neg B(x)) \wedge$
  $\exists x(((A(x) \to B(x)) \to B(x)) \wedge \neg\neg \sim B(x)) \wedge$
  $\exists x(((B(x) \to A(x)) \to A(x)) \wedge \neg\neg \sim A(x))$
- $d = soc = 1$. Here we know that $\Sigma_a \min\{\alpha(a), \beta(a)\} = \Sigma_a \alpha(a)$, which means $\alpha(a) \leq \beta(a)$ for all patients in the database. This is accounted by the implicative formula (I) $\forall x(A(x) \to B(x))$. We add the formula (II) $\exists x(A(x) \prec B(x))$, which arises from $foo < 1$. In this case the formulas (POS) and (DEF$_{foo}$) are implied by (I) and (DEF$_{soc}$). This leads to the following formula representation of a rule c_1:
  $\forall x(A(x) \to B(x)) \wedge \exists x \neg\neg A(x) \wedge$
  $\exists x(((B(x) \to A(x)) \to A(x)) \wedge \neg\neg \sim A(x))$.

Rule ao

From $foo = 1$ we have $\Sigma_a \min\{\alpha(a), \beta(a)\} = \Sigma_a \beta(a)$, which implies that for all patients a, $\beta(a) \leq \alpha(a)$. This is accounted by the formula (I) $\forall x(B(x) \to A(x))$. From $1 > soc > 0$ we have, (II) $\exists x(\neg\neg A(x) \wedge \neg\neg B(x))$ and (III) $\exists x(B(x) \prec A(x))$. Note that the formulas (I) and (DEF$_{foo}$) imply (II) and (DEF$_{soc}$). These considerations lead to the following formula representation of a rule ao:
$\forall x(B(x) \to A(x)) \wedge \exists x \neg\neg B(x) \wedge$
$\exists x(((A(x) \to B(x)) \to B(x)) \wedge \neg\neg \sim B(x))$.

Rule oc

The formulas (I) $\forall x(B(x) \to A(x))$ and (II) $\forall x(A(x) \to B(x))$ formalize $foo = 1$ and $soc = 1$, respectively (see the cases ao and c_1 above). Note that (I), (II) and (DEF$_{soc}$) imply (DEF$_{foo}$). Therefore the formula representation of a rule oc is:
$\forall x(B(x) \to A(x)) \wedge \forall x(A(x) \to B(x)) \wedge \exists x \neg\neg A(x)$.

Let Σ^{KB} be the set of formulas of $\mathsf{G}^\sim$ obtained by instantiating the schemata below with the (formula representation of the) entities of CADIAG-2's, according to the system's KB:

(c_d)	$\exists x(\neg\neg A(x) \wedge \neg\neg B(x)) \wedge$ $\exists x(((A(x) \to B(x)) \to B(x)) \wedge \neg\neg \sim B(x)) \wedge$ $\exists x(((B(x) \to A(x)) \to A(x)) \wedge \neg\neg \sim A(x))$
(c_1)	$\forall x(A(x) \to B(x)) \wedge \exists x \neg\neg A(x) \wedge$ $\exists x(((B(x) \to A(x)) \to A(x)) \wedge \neg\neg \sim A(x))$
(me)	$\forall x(\neg A(x) \vee \neg B(x)) \wedge \exists x \neg\neg A(x) \wedge \exists x \neg\neg B(x)$
(ao)	$\forall x(B(x) \to A(x)) \wedge \exists x \neg\neg B(x) \wedge$ $\exists x(((A(x) \to B(x)) \to B(x)) \wedge \neg\neg \sim B(x))$
(oc)	$\forall x(B(x) \to A(x)) \wedge \forall x(A(x) \to B(x)) \wedge$ $\exists x \neg\neg A(x)$

where $B(x)$, $B_i(x)$ are atomic and $A(x)$ is a formula generated by the following BNF grammar:
$A(x) := B_i(x) \mid \sim B_i(x) \mid A(x) \vee A(x) \mid A(x) \wedge A(x)$.

Proposition 11 *If there exists a set of patients that models CADIAG-2's KB then Σ^{KB} is satisfiable in $\mathsf{G}^{\sim}$.*

Proof: Let SP be such a set of patients (cf. Definition 10). We define an interpretation v_{SP} in $\mathsf{G}^{\sim}$ that satisfies Σ^{KB}. We set the domain $\mathcal{D}$ of this interpretation to be SP. We denote with $\bar{S}_1(a), \ldots, \bar{S}_{1781}(a), \bar{D}_1(a), \ldots, \bar{D}_{342}(a)$ the degrees to which the basic entities of CADIAG-2 hold for a patient $a \in SP$. For every patient $a \in SP$ and every basic entity S_i we set $v_{SP}(S_i(a)) = \bar{S}_i(a)$ and analogously for every basic entity D_i. v_{SP} is extended to compound formulas by the truth functions of $\mathsf{G}^{\sim}$. We have $v_{SP}(A(a)) = \alpha(a)$ for all $a \in \mathcal{D}$. From this and the fact that SP models CADIAG-2's KB, for every rule $R = \langle \alpha, \beta, soc, foo \rangle$ the following equations hold:

$$soc = \frac{\Sigma_a \min\{v_{SP}(A(a)), v_{SP}(B(a))\}}{\Sigma_a v_{SP}(A(a))},$$

$$foo = \frac{\Sigma_a \min\{v_{SP}(A(a)), v_{SP}(B(a))\}}{\Sigma_a v_{SP}(B(a))},$$

where Σ_a ranges over all elements in SP. It is easy to see that v_{SP} satisfies Σ^{KB}. We consider as an example the cases of the formulas (me) and (c_d) representing the rules of type me and c_d ($d < 1$), the other cases being similar.

If the set of patients SP models a rule of type me then $\Sigma_a \min\{v_{SP}(A(a)), v_{SP}(B(a))\} = 0$, i.e., for each patient a either $v_{SP}(A(a)) = 0$ or $v_{SP}(B(a)) = 0$. The existence of patients a and b such that $v_{SP}(A(a)) > 0$ and $v_{SP}(B(b)) > 0$ follows from the fact that *soc* and *foo* are defined. Hence v_{SP} satisfies the formula (me).

If SP models a rule of type c_d ($d = soc < 1$) then $\Sigma_a \min\{v_{SP}(A(a)), v_{SP}(B(a))\} < \Sigma_a v_{SP}(A(a))$ and hence there exists a patient a_1 such that $v_{SP}(B(a_1)) < v_{SP}(A(a_1))$. Analogously from $foo < 1$ we conclude the existence of a patient a_2 such that $v_{SP}(B(a_2)) > v_{SP}(A(a_2))$. Finally, from $\Sigma_a \min\{v_{SP}(A(a)), v_{SP}(B(a))\} > 0$ we conclude that there exists a patient a_3 such that $v_{SP}(A(a_3)) > 0$ and $v_{SP}(B(a_3)) > 0$. Hence v_{SP} satisfies the formula (c_d). □

Remark 12 *The converse direction of the above proposition does not hold. The reason being that the formulas (c_d) representing the c_d rules, with $d < 1$, do not take into account the actual value $d = soc$. This way, rules that assign, e.g., two different numbers $0 < n, m < 1$ to a same diagnosis are considered "correct", in accordance with the intended interpretation of CADIAG-2. Recall, indeed, that the value $t \in (0, 1)$ of any entity in the system is any time improvable to a larger one $t' \in (0, 1]$, in which case the former value is no longer used.*

4.3 Rules Checking

In this section we analyze CADIAG-2's KB by checking the satisfiability of large subsets of its formula representation. Our analysis allowed the identification of some errors in the system's KB.

To actually do the check using existing tools for classical logic we first manipulate the rules of Σ^{KB} to make them fitting into the fragment $\Sigma_{cl}^{\mathsf{G}^{\sim}}$ of $\mathsf{G}^{\sim}$ (cf. Section 3.). To this aim we perform the following surgery to the formulas in Σ^{KB} by removing:

(res1) all formulas (c_d), (c_1), (ao) and (oc) in which $A(x)$ contains $\sim$

(res2) the conjunct $\exists x \neg\neg A(x)$ from (me) and the formulas $\neg\neg \sim A(x)$ and $\neg\neg \sim B(x)$, if any, from (c_d), (c_1) and (ao).

We call Σ_{-}^{KB} the resulting set of formulas. More precisely, Σ_{-}^{KB} consists of the formulas of $\mathsf{G}^{\sim}$ obtained by instantiating the schemata below with the (formula representation of the) entities of CADIAG-2's, according to the KB$\backslash\{R = \langle \alpha, \beta, soc, foo \rangle \mid \alpha$ contains $\sim$ and $soc \neq 0\}$.

$(\mathrm{c}_d)'$ $\exists x(\neg\neg C(x) \wedge \neg\neg B(x)) \wedge$
$\exists x((C(x) \to B(x)) \to B(x)) \wedge$
$\exists x((B(x) \to C(x)) \to C(x))$

$(\mathrm{c}_1)'$ $\forall x(C(x) \to B(x)) \wedge \exists x \neg\neg C(x) \wedge$
$\exists x((B(x) \to C(x)) \to C(x))$

$(\mathrm{me})'$ $\forall x(\neg A(x) \vee \neg B(x)) \wedge \exists x \neg\neg B(x)$

$(\mathrm{ao})'$ $\forall x(B(x) \to C(x)) \wedge \exists x \neg\neg B(x) \wedge$
$\exists x((C(x) \to B(x)) \to B(x))$

$(\mathrm{oc})'$ $\forall x(B(x) \to C(x)) \wedge \forall x(C(x) \to B(x)) \wedge$
$\exists x \neg\neg C(x)$

where $B(x)$, $B_i(x)$ are atomic while $C(x)$ and $A(x)$ are formulas generated by the following BNF grammars:

- $C(x) := B_i(x) \mid C(x) \wedge C(x) \mid C(x) \vee C(x)$.
- $A(x) := B_i(x) \mid \sim B_i(x) \mid A(x) \wedge A(x) \mid A(x) \vee A(x)$.

It is easy to see that $\Sigma_{-}^{KB} \subset \Sigma_{cl}^{\mathsf{G}^{\sim}}$. Moreover

Proposition 13 *If Σ_{-}^{KB} is unsatisfiable in classical logic then there is no set of patients that models CADIAG-2's KB.*

Proof: If Σ_{-}^{KB} is unsatisfiable in $\mathsf{G}^{\sim}$ then so is Σ^{KB}. The claim follows by Theorem 7 and Proposition 11. □

Though Σ_{-}^{KB} is a reduced version of the formulas representing CADIAG-2's KB, and therefore it can express only some features of the system's rules, Σ_{-}^{KB} turned out to be unsatisfiable in classical logic. The satisfiability check of Σ_{-}^{KB} was implemented using the theorem prover *Prover9* and the (counter)model generator *Mace4*, developed at the University of New Mexico (McCune).

Prover9 identifies unsatisfiable formulas by proving the negation of their conjunction. In the case of Σ_{-}^{KB} such input would have been too large for *Prover9*. We therefore made a step-wise selection of the input formulas. We first represented CADIAG-2's KB as a graph, whose vertices are the basic entities of the system and whose edges connect two entities if there is a rule in the KB in which they both appear. We say that two entities are connected by a k-path when there is a path between them containing at most k edges. For each basic entity X and $k \in \mathcal{N}$ we gave as input to Prover9 the formulas of Σ_{-}^{KB} containing at least one predicate representing an entity connected to X by a k-path.

For $k = 1$ *Prover9* detected one set of unsatisfiable formulas that turned out to be the only one present in Σ_{-}^{KB}, as *Mace4* found a model for the remaining formulas. The rules of CADIAG-2 responsible for that are:

$\langle \text{S0118}, \text{D025}, 0, 0\rangle$,
$\langle \text{D025}, \text{D049}, 0.001, 1\rangle$,
$\langle \text{S0118}, \text{D049}, 0.99, 0.01\rangle$,

where S0118 stands for "Present difficulties, Nerves, Chorea Minor" D025 for "Reactive arthritis" and D049 for "Rheumatic fever". A natural language representation of these rules is:

me, IF *Present difficulties, Nerves, Chorea Minor* THEN NOT *Reactive arthritis*

ao, IF NOT *Reactive arthritis* THEN NOT *Rheumatic fever*

c_d, IF *Present difficulties, Nerves, Chorea Minor* THEN *Rheumatic fever* with the degree 0.99.

Though these rules cannot produce a runtime inconsistency, from the same fact (S0118 with degree 1) they derive two almost opposite conclusions, i.e., the diagnose "Rheumatic fever" applies with degree of certainty 0 (meaning that it is excluded) and 0.99 (meaning that it is almost sure). The consulted physicians asserted that, in fact, the first and the third rules above cannot be confirmed (e.g. 0.99 is a too high degree) and therefore they have to be corrected or deleted from the system.

Binary Rules

Most of the rules in CADIAG-2's KB are binary (cf. Definition 9). By using the fragment $\Sigma_{cl}^{\mathsf{G}^\sim}$ of $\mathsf{G}^\sim$ we show here that SAT for the formalization of such rules can be checked within classical logic.

Henceforth we denote by $(\Sigma^{KB})^{at}$ the formula representation for the binary rules of CADIAG-2. More precisely $(\Sigma^{KB})^{at}$ consists of all formulas (c_d), (c_1), (me), (ao), (oc) of Σ^{KB} in which both $A(x)$ and $B(x)$ are *atomic*.

Lemma 14 *$(\Sigma^{KB})^{at}$ is satisfiable in $\mathsf{G}^\sim$ if and only if it is satisfiable in classical logic.*

Proof: If $(\Sigma^{KB})^{at}$ is classically satisfiable then it is satisfiable in $\mathsf{G}^\sim$. For the converse direction, let $v_{\mathsf{G}^\sim}$ be an interpretation in $\mathsf{G}^\sim$ satisfying $(\Sigma^{KB})^{at}$. We denote by $\{X \prec Y\}$ the set of all conjuncts of the form $\exists x(A(x) \prec B(x))$ in $(\Sigma^{KB})^{at}$. Assume that there are n such conjuncts. Let $(\Sigma^{KB})^{at} \setminus \{X \prec Y\}$ be $(\Sigma^{KB})^{at}$ in which we remove all of them. $(\Sigma^{KB})^{at} \setminus \{X \prec Y\}$ is in $\Sigma_{cl}^{\mathsf{G}^\sim}$ and therefore is classically satisfiable by Theorem 7. Let v_{CL} be an interpretation in classical logic that satisfies $(\Sigma^{KB})^{at} \setminus \{X \prec Y\}$ and let $\mathcal{D}$ be its domain. By stepwise extending $\mathcal{D}$ with a new domain element for each formula in $\{X \prec Y\}$ we construct a classical interpretation v_{CL}^n that satisfies $(\Sigma^{KB})^{at}$. Let $v_{CL}^0 = v_{CL}$ and

$$(\Sigma^{KB})_0^{at} = (\Sigma^{KB})^{at} \setminus \{X \prec Y\}$$

Assume to fix ideas that at step i ($i = 1, \ldots, n$) we are dealing with the conjunct $\exists x(A(x) \prec B(x))$. Let

$$(\Sigma^{KB})_i^{at} = (\Sigma^{KB})_{i-1}^{at} \cup \{\exists x(A(x) \prec B(x))\}.$$

Consider the interpretation $v_{\mathsf{G}^\sim}$ of $\mathsf{G}^\sim$ that satisfies $(\Sigma^{KB})^{at}$ and take an element $d^\sim$ in the domain of $v_{\mathsf{G}^\sim}$ that makes true the formula $\exists x(A(x) \prec B(x))$, that is for which $v_{\mathsf{G}^\sim}(A(d^\sim)) < v_{\mathsf{G}^\sim}(B(d^\sim))$. v_{CL}^i is defined as follows:

1. we add a new element d_i to the domain $\mathcal{D} \cup \{d_1, \ldots, d_{i-1}\}$ of the classical interpretation v_{CL}^{i-1} satisfying $(\Sigma^{KB})_{i-1}^{at}$
2. we assign the following values to each atomic formula F in $(\Sigma^{KB})^{at}$:

   $v_{CL}^i(F(d)) = v_{CL}^{i-1}(F(d))$ for all $d \in \mathcal{D} \cup \{d_1, \ldots, d_{i-1}\}$

   (i.e., v_{CL}^i and v_{CL}^{i-1} coincide for the domain elements in common) and

$$v_{CL}^i(F(d_i)) \quad = \quad \begin{cases} 0 & \text{if } v_{\mathsf{G}^\sim}(F(d^\sim)) \leq v_{\mathsf{G}^\sim}(A(d^\sim)) \\ 1 & \text{otherwise,} \end{cases}$$

Clearly v_{CL}^i satisfies $\exists x(A(x) \prec B(x))$. Observe, that for all atomic formulas C and D, if $v_{\mathsf{G}^\sim}(C(d^\sim) \to D(d^\sim)) = 1$ then also $v_{CL}^i(C(d_i) \to D(d_i)) = 1$ and if $v_{\mathsf{G}^\sim}(\neg C(d^\sim) \vee \neg D(d^\sim)) = 1$ then also $v_{CL}^i(\neg C(d_i) \vee \neg D(d_i))$. It is therefore easy to see that v_{CL}^i satisfies $(\Sigma^{KB})_i^{at}$.

The interpretation v_{CL}^n obtained at the final step satisfies $(\Sigma^{KB})_n^{at}$, that coincides with $(\Sigma^{KB})^{at}$.

□

Proposition 15 *If $(\Sigma^{KB})^{at}$ is unsatisfiable in classical logic then there is no set of patients that models CADIAG-2's KB.*

The satisfiability check of $(\Sigma^{KB})^{at}$ was implemented using *Prover9* and *Mace4*, as in the case of Σ_{-}^{KB}. We found ten groups of inconsistent rules (one of them was already found while checking the satisfiability of Σ_{-}^{KB}). As an example we present here a group of unsatisfiable rules:

$\langle S1017,\ D013,\ 1.0,\ 0.0010\rangle$
$\langle D013,\ D011,\ 1.0,\ 0.0010\rangle$
$\langle S1017,\ D011,\ 0.1,\ 0.2\rangle$

If we assign the value 1 to $S1017$ then, using the first two rules, the system infers the value 1 for both $D013$ and $D011$. The third rule proposes instead the value 0.1 for $D011$. Though the third rule does not influence the final result ($\max\{1, 0.1\} = 1$), its *soc* does not seem to be correct. Indeed the consulted physicians asserted that the right value for *soc* in this rule should be 1; it is very likely that a typing error occurred.

Final Remark

The satisfiability check performed over the two sets of formulas Σ_{-}^{KB} and $(\Sigma^{KB})^{at}$ representing CADIAG-2's KB allowed the detection of 10 groups of erroneous rules. Notice that

- The restriction **(res1)** in Σ_{-}^{KB} leaves out from our investigation 25 rules, out of the (more than) 20.000 rules in CADIAG-2's KB. Moreover, due to the restriction **(res2)** Σ_{-}^{KB} does not capture all the features of CADIAG-2's rules.
- By restricting to binary rules (i.e. considering $(\Sigma^{KB})^{at}$), 63 rules are left out from our investigation. Observe that due to the use of the (derivable) connectives "at least n out of m" and "at most n out of m", some of the remaining rules turn out to be rather complex. Indeed when expressed in disjunctive normal form, the antecedents of these rules can contain up to 30.000.000 disjuncts and each disjunct consists of 16 literals.

We believe that a satisfiability check of Σ^{KB} might allow the discovery of further errors in the knowledge representation of CADIAG-2. This check should be feasible, in view of our conjecture that SAT in $\mathsf{G}^{\sim}$ of formulas of the form

$$\bigwedge \exists x P_i(x) \wedge \bigwedge \forall x P_j(x)$$

is decidable, where $P_i(x)$ and $P_j(x)$ are monadic and quantifier-free formulas of $\mathsf{G}^{\sim}$; our future work include the design of suitable tools to perform this check. Alternative formalizations of CADIAG-2's rules will be also considered. E.g., based on a suitable adaptation (generalization) of the Kripke-Kleene semantics as, e.g., in (Straccia 2006). Such formalizations might capture different aspects of the system's rules.

5. Acknowledgments

This research was supported by the Vienna Science and Technology Fund (WWTF) Grant MA07-016.

References

Adlassnig, K.-P.; Kolarz, G.; Scheithauer, W.; and Grabner, G. 1986. Approach to a hospital-based application of a medical expert system, *Med. Inform.* 11: 205–223.

Adlassnig, K.-P.; and Kolarz, G. 1986. Representation and semiautomatic acquisition of medical knowledge in CADIAG-1 and CADIAG-2, *Computers and Biomedical Research* 19: 63–79.

Adlassnig, K.-P.; Kolarz, G.; Scheithauer, W.; Effenberger, H.; and Grabner, G. 1985. CADIAG: Approaches to computer-assisted medical diagnosis, *Comput. Biol. Med.* 15(5): 315–335.

Baaz, M. 1996. Infinite-valued Gödel logics with 0-1-projections and relativizations. In *Proceedings Gödel 96. Kurt Gödel's Legacy*. LNL 6, 23–33, Berlin: Springer-Verlag.

Baaz, M.; Ciabattoni, A.; and Preining, N. 2009. SAT in Monadic Gödel Logics: a borderline between decidability and undecidability. In *Proceedings of WOLLIC 2009*. LNAI 5514: 113–123.

Baaz, M.; Preining, N.; and Zach, R. 2007. First-order Gdel logics. *Annals of Pure and Applied Logic* 147: 23–47.

Ciabattoni, A.; and Vetterlein, T. 2010. On the (fuzzy) logical content of CADIAG-2. *Fuzzy Sets and Systems*. To appear; online doi: 10.1016/j.fss.2009.09.011

Dunn, J.M.; Meyer, R.K. 1971. Algebraic completeness results for Dummett's LC and its extensions. *Z. Math. Logik Grundlagen Math* 17: 225–230.

Esteva, F.; Godo, L.; Hájek, P.; and Navara. M. 2000. Residuated fuzzy logics with an involutive negation. *Archive for Mathematical Logic* 39(2): 103–124.

Flaminio, T.; and Marchioni, E. 2006. T-norm-based logics with an independent involutive negation. *Fuzzy Sets and Systems* 157(24): 3125–3144.

Hájek, P. 1998. "Methamatematics of Fuzzy Logic", Kluwer.

Janssen, J.; Schockaert, S.; Vermeir, D.; De Cock, M. 2009. General Fuzzy Answer Set Programs. In *Proceedings of WILF 2009*, LNCS 5571: 352–359.

Lifschitz, V.; Pearce, D.; and Valverde, A. 2001. Strongly Equivalent Logic Programs. *ACM Transaction on Computational Logic* 2(4): 526–541.

Łukasiewicz, T. 2008. Fuzzy Description Logic Programs under the Answer Set Semantics for the Semantic Web. *Fundamenta Informaticae* 82(3): 289–310.

Mateis, C. 1999. Extending Disjunctive Logic Programming by T-norms. In *Proceedings of LPNMR 1999*. LNCS 1730: 290–304.

McCune, W. Prover9 and Mace4. `http://www.cs.unm.edu/~mccune/prover9/`. (Accessed 9th of Sept., 2009)

Moser, W.; and Adlassnig, K.-P. 1992. Consistency checking of binary categorical relationships in a medical knowledge base, *Artificial Intelligence in Medicine* 8: 389–407.

Rogers, H. 1956. Certain logical reduction and decision problems. *Annals of Mathematics* 64: 264–284.

Straccia, U. 2006. Query Answering under the Any-World Assumption for Normal Logic Programs. In *Proceedings of KR-06*, 329–339, AAAI Press.

Straccia, U. 2008. Managing Uncertainty and Vagueness in Description Logics, Logic Programs and Description Logic Programs. In *Reasoning Web, 4th International Summer School, Tutorial Lectures*. LNCS 5224: 54–103.

Takeuti, G.; and Titani, T. 1984. Intuitionistic fuzzy logic and intuitionistic fuzzy set theory. *J. of Symbolic Logic* 49: 851–866.

Visser, A. 1982. On the completeness principle: a study of provability in Heyting's Arithmetic. *Annals of Math. Logic* 22: 263–295.