Centrum voor Wiskunde en Informatica
Centre for Mathematics and Computer Science

M. Bezem

Consistency of rule-based expert systems

Computer Science/Department of Software Technology Report CS-R8736 July

The Centre for Mathematics and Computer Science is a research institute of the Stichting Mathematisch Centrum, which was founded on February 11, 1946, as a nonprofit institution aiming at the promotion of mathematics, computer science, and their applications. It is sponsored by the Dutch Government through the Netherlands Organization for the Advancement of Pure Research (Z.W.O.).

Consistency of Rule-based Expert Systems

Marc Bezem
Centre for Mathematics and Computer Science
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

Consistency of a knowledge-based system has become a topic of growing concern. Every notion of consistency presupposes a notion of semantics. We present a theoretical framework in which both the semantics and the consistency of a knowledge base can be studied. This framework is based on first order many-sorted predicate logic and is sufficiently rich to capture an interesting class of rule-based expert systems and deductive databases. We also provide criteria which allow us to isolate cases in which the consistency test is feasible.

1980 Mathematics Subject Classification: 68T30, 68T15.
1987 CR Categories: I.2.3, I.2.4, F.4.1.
Key Words & Phrases: knowledge-based systems, rule-based expert systems, knowledge representation, consistency.
Note: The work in this document was conducted as part of the PRISMA project, a joint effort with Philips Research Eindhoven, partially supported by the Dutch "Stimulerings-projectteam Informatica-onderzoek" (SPIN).

Introduction

The plan of this paper is as follows. First the reader is introduced to the knowledge representation used in rule-based expert systems. We shall indicate some semantical problems in relation to this knowledge representation. Then we explain in an informal way how many-sorted predicate logic comes in. In the next section we describe syntax and semantics of many-sorted predicate logic. We assume some knowledge of first order logic. Thereafter we shall be able to characterize rule-based expert systems as first order theories. The Tarski semantics solves the semantical problems mentioned above. Furthermore we shall derive several results on decidability and consistency of rule-based expert systems. Unfortunately, some natural equality and ordering axioms are not in Horn format (see [Re] for a discussion on the domain closure axiom). Hence testing consistency with a standard theorem prover would be very inefficient. In the last section we describe a technical device, a certain kind of null value, which allows feasible consistency testing in the presence of equality and ordering axioms, which are not in Horn format. This kind of null value, being quite different from null values as described in [IL], appears to be new. We shall focus our attention on rule-based expert systems, but the techniques can also be applied to deductive databases.

Contents

Report CS-R8736
Centre for Mathematics and Computer Science
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

1. Rule-based expert systems

1.1. In rule-based expert systems shells such as EMYCIN [BS] or DELFI2 [L], knowledge about some specific domain can be expressed in *facts* and in *rules* of the form

IF <antecedent> THEN <consequent>.

Facts are so-called object-attribute-value triples, or *<o,a,v>* triples for short. The antecedent of a rule is a conjunction of disjunctions of conditions, and conditions are definite statements, such as *same*, *notsame* and *less_than*, about *<o,a,v>* triples. We restrict ourselves to rules having as consequent a conjunction of conclusions of the form *conclude <o,a,v>*. In most cases so-called certainty factors are associated with the facts and the rules. Certainty factors range from 1.00 (definitely true) to −1.00 (definitely false). The certainty factor of a fact expresses a measure of certainty about that fact, whereas the certainty factor of a rule scales the measure of certainty about the consequent with respect to the measure of certainty about the antecedent. In DELFI2 an *object tree* (called *context tree* in MYCIN) is used to state properties of and relations between different objects, which cannot be expressed by the rules. The nodes of this tree are objects, labeled by their attributes and respective values. The path from a node to the root of this tree constitutes the context of that node. In other words: the objects occurring in the subtree of a node are sub-objects of the object belonging to that node. Furthermore it is stated in the object tree whether an attribute is singlevalued or multivalued.

The interpretation of the knowledge in rule-based expert systems is more operational than declarative: *same <o,a,v>* is true if and only if *<o,a,v>* occurs as fact (with certainty factor > 0.2), *conclude <o,a,v>* has the effect that *<o,a,v>* is added as fact (with appropriate certainty factor), and if the antecedent of a rule evaluates to true, then that rule may be *fired*, i.e. all conclusions occurring in the consequent are executed. We remark that *same <o,a,v>* and *conclude <o,a,v>* have the same declarative meaning as the fact *<o,a,v>*, i.e. attribute *a* of object *o* has value *v*.

1.2. Consider the following real-life example extracted from HEPAR, an expert system for the diagnosis of liver and biliary disease, built with DELFI2.

IF same <patient,complaint,colicky_pain>

THEN conclude <patient,pain,colicky> (1.00)

IF same <patient,abd_pain,yes> AND

same <pain,character,continuous>

THEN conclude <patient,pain,colicky> (−1.00)

IF same <patient,complaint,abdominal_pain> OR

same <patient,pain,colicky>

THEN conclude <patient,abd_pain,yes> (1.00)

These three rules (from a rule base consisting of over 400 rules) show two *objects*, patient and pain, four *attributes*, namely complaint, pain and abd_pain of patient and character of pain, as well as several *values*.

1.3. A first observation, which can be made on the three rules above, is their inconsistency in the presence of the facts *<patient,complaint,colicky_pain>* (1.00) and *<pain,character,continuous>* (1.00). With some effort even the (sound, but incomplete) inference engine did hit upon the contradiction *<patient,pain,colicky>* (1.00 and −1.00).

A second observation is the following. Five items refer to pain: the values colicky_pain and abdominal_pain, the attributes pain and abd_pain of the object patient, and the object pain, which is a sub-object of patient, as stated by the object tree of HEPAR. The interrelations between these items do not seem to be expressible by the formalism.

These observations show a defect of the knowledge representation used in rule-based expert systems, namely the lack of a clear semantics.

1.4. Basically our approach amounts to interpreting $<o,a,v>$ by $a(o,v)$ in the multivalued, and by $a(o) = v$ in the singlevalued case. Here o and v are constants for elements of a domain O of objects and a domain V of values. In the multivalued case a denotes a *relation*, i.e. a subset of $O \times V$, and in the singlevalued case a *function* from O to V. If o is a sub-object of o', then o' is added as argument of a.

1.5. The following examples show how to extend our interpretation of $<o,a,v>$ triples to atoms.

conclude $<$*patient,complaint,abdominal_pain*$>$

becomes

complaint (*patient,abdominal_pain*)

less_than $<$*patient,temperature*, 36.8$>$

becomes

temperature (*patient*) $<$ 36.8

same $<$*pain,character,continuous*$>$

becomes

character (*patient,pain*) $=$ *continuous*

Note that the fact that pain is a sub-object of patient is expressed by adding *patient* as an argument of the function character.

At this state of affairs we prefer not to incorporate any uncertainty handling in our formalism.

1.6. Under the interpretation described above, a rule-based expert system becomes a theory in first order many-sorted predicate logic, in short: a many-sorted theory. To keep this paper self-contained we give a short, introductory description of the syntax and semantics of many-sorted predicate logic in the next section. In Section 3 we shall characterize some types of expert systems as certain many-sorted theories. This approach has the following advantages:

- The declarative semantics of the expert system becomes perfectly clear, being the Tarski semantics of the associated many-sorted theory.
- Logical concepts such as decidability, consistency etc. get a clear meaning in relation to the expert system.
- Theorem proving techniques for testing consistency, such as resolution, become available for the expert system.

2. Many-Sorted Predicate Logic

2.1. The syntax of many-sorted predicate logic extends the syntax of ordinary, one-sorted, predicate logic by having a finite set of *sorts* Σ, instead of just one sort. Moreover we have *variables* x_i^σ and *constants* c_i^σ for all sorts $\sigma \in \Sigma$. Furthermore we have finitely many *function symbols* $f_i^{\sigma_1 \times \cdots \times \sigma_m \rightarrow \sigma_0}$, where the notion of *type* $\sigma_1 \times \cdots \times \sigma_m \rightarrow \sigma_0$ replaces the notion of *arity* from the one-sorted case. We also have finitely many *proposition symbols* p_i and *predicate symbols* $P_i^{\sigma_1 \times \cdots \times \sigma_m}$ of type $\sigma_1 \times \cdots \times \sigma_m$. *Terms* are formed from variables and constants by function application (respecting the sorts). *Atoms* are either proposition symbols or the application of a predicate symbol to terms of appropriate sorts. With the help of propositional connectives and quantifiers, atoms are combined into *formulas*. The sets Σ, CONS, FUNC, PROP and PRED of, respectively, sorts, constants, function symbols, proposition symbols and predicate symbols, form together the *similarity type* of some specific many-sorted predicate calculus.

2.2. A *many-sorted structure* $\mathfrak{M}$ consists of:
(a) A non-empty set A_σ for each $\sigma \in \Sigma$, called the *domain* of sort σ. We abbreviate $\bigcup_{\sigma \in \Sigma} A_\sigma$ by A.
(b) For each constant c_i^σ an element $\overline{c}_i \in A_\sigma$.
(c) For each function symbol $f_i^{\sigma_1 \times \cdots \times \sigma_m \rightarrow \sigma_0}$ a mapping $\overline{f}_i : A_{\sigma_1} \times \cdots \times A_{\sigma_m} \rightarrow A_{\sigma_0}$.
(d) For each proposition symbol p_i a truth value $\overline{p}_i$
(e) For each predicate symbol $P_i^{\sigma_1 \times \cdots \times \sigma_m}$ a mapping $\overline{P}_i : A_{\sigma_1} \times \cdots \times A_{\sigma_m} \rightarrow \{TRUE, FALSE\}$.

2.3. An *assignment* in $\mathfrak{M}$ is a mapping a assigning to each variable x_i^σ an element $a(x_i^\sigma)$ of A_σ.

2.4. The *interpretation* in $\mathfrak{M}$ of a term t under an assignment a, denoted by $I_a^{\mathfrak{M}}(t)$ or $\overline{t}$ for short, is inductively defined as follows:
(a) $\overline{x_i^\sigma} = a(x_i^\sigma)$
(b) $\overline{c_i^\sigma} = \overline{c}_i$
(c) $\overline{f_i^{\sigma_1 \times \cdots \times \sigma_m \rightarrow \sigma_0}(t_1,\ldots,t_m)} = \overline{f}_i(\overline{t}_1, \ldots, \overline{t}_m)$
The *truth value* in $\mathfrak{M}$ of an atom $P_i^{\sigma_1 \times \cdots \times \sigma_m}(t_1,\ldots,t_m)$ under an assignment a is given by $\overline{P}_i(\overline{t}_1, \ldots, \overline{t}_m)$.

2.5. The *truth value* in $\mathfrak{M}$ of a formula F under an assignment a, denoted by $\mathcal{V}_a^{\mathfrak{M}}(F)$, is inductively defined as follows:
(a) If F is an atom, then $\mathcal{V}_a^{\mathfrak{M}}(F)$ is given by 2.4.
(b) $\mathcal{V}_a^{\mathfrak{M}}$ respects the truth tables of the propositional connectives.
(c) $\mathcal{V}_a^{\mathfrak{M}}(\forall x_i^\sigma F) = TRUE$ if and only if for all assignments a', which differ at most on x_i^σ from a, we have $\mathcal{V}_{a'}^{\mathfrak{M}}(F) = TRUE$.
(d) $\mathcal{V}_a^{\mathfrak{M}}(\exists x_i^\sigma F) = TRUE$ if and only if there exists an assignment a', which differs at most on x_i^σ from a, such that $\mathcal{V}_{a'}^{\mathfrak{M}}(F) = TRUE$.

2.6. A formula F is *true* in $\mathfrak{M}$, denoted by $\models_{\mathfrak{M}} F$, if $\mathcal{V}_a^{\mathfrak{M}}(F) = TRUE$ for all assignments a.

2.7. A *sentence* (or *closed* formula) is a formula without free variables (i.e. variables which are not in the scope of a quantifier). It will be clear that for sentences S the truth value $\mathcal{V}_a^{\mathfrak{M}}(S)$ does not depend on the assignment a. As a consequence we have either $\models_{\mathfrak{M}} S$ or $\models_{\mathfrak{M}} \neg S$ for all sentences S. Let SENT denote the set of sentences.

2.8. Let $\Gamma \subset$ SENT. $\mathfrak{M}$ is called a *model* for Γ, denoted by $\models_{\mathfrak{M}} \Gamma$, if $\models_{\mathfrak{M}} S$ for every $S \in \Gamma$.

2.9. $S \in$ SENT is called a (*semantical*) *consequence* of $\Gamma \subset$ SENT if for all many-sorted structures $\mathfrak{M}$

we have: if $\models_{\mathfrak{M}}\Gamma$, then $\models_{\mathfrak{M}}S$. This will be denoted by $\Gamma \models S$ (or $\models S$ if Γ is empty). Furthermore we define the *theory* of Γ as the set $Th(\Gamma) = \{S \in \text{SENT} \mid \Gamma \models S\}$.

2.10. $\Gamma \subset$ SENT is called *consistent* if Γ has a model. $Th(\Gamma)$ is called *decidable* if there exists a mechanical decision procedure to decide whether a given sentence S is a semantical consequence of Γ or not.

2.11. Two many-sorted structures are called *elementarily equivalent* if exactly the same sentences are true in both structures.

2.12. REMARKS.

2.12.1. We refrain from giving an axiomatization of many-sorted predicate logic since our main concern will be model theory. Most textbooks on mathematical logic provide a complete axiomatization of ordinary (one-sorted) predicate logic. It suffices to generalize the quantifier rules in order to obtain an axiomatization of many-sorted predicate logic.

2.12.2. Of course, one-sorted predicate logic is a special case of many-sorted predicate logic. As a consequence, the latter is as undecidable as the former. More precisely: $\models S$ is undecidable, provided that the similarity type is rich enough (CHURCH, TURING, 1936, see also [M, 16.58]).

2.12.3. Conversely, many-sorted predicate logic can be embedded in one-sorted predicate logic by adding unary predicate symbols $S^{\sigma}(x)$, expressing that x is of sort σ, and replacing inductively in formulas $\forall x_i^{\sigma} F$ (resp. $\exists x_i^{\sigma} F$) by $\forall x(S^{\sigma}(x) \rightarrow F)$ (resp. $\exists x(S^{\sigma}(x) \wedge F)$). Let A' be the one-sorted sentence obtained from $A \in$ SENT in this way. It can be proved (see [M]) that $\models A$ if and only if $\Gamma \models A'$, where $\Gamma = \{\exists x S^{\sigma}(x) \mid \sigma \in \Sigma\}$ expresses the fact that the domains are non-empty. This embedding allows us to generalize immediately many results on one-sorted predicate calculus to the many-sorted case (e.g. the compactness theorem). We shall not make use of this possibility in the present paper.

3. RULE-BASED EXPERT SYSTEMS AS MANY-SORTED THEORIES

3.0. We propose the following terminology for certain kinds of many-sorted theories:
- Indexed propositional expert systems.
- Universally quantified expert systems.

3.1. An *indexed propositional expert system* is a many-sorted theory axiomatized by:

(a) *Explicit axioms* (the rule base and the fact base), which are boolean combinations of atoms of the form $P^{\sigma_1 \times \cdots \times \sigma_m}(c,...,c')$ or of the form $f^{(\sigma_1 \times \cdots \times \sigma_m) \rightarrow \sigma_0}(c,...,c') =_{\sigma_0} c''$ (resp. $<_{\sigma_0} c''$, $>_{\sigma_0} c''$), with constants $c,...,c',c''$ of appropriate sorts. Such atoms (here and below called *constant-atoms*, or *c-atoms* for short) may be viewed as indexed propositions, which explains the name. Note that we conform to the convention to denote $=$, $<$ and $>$ as infix predicates.

(b) *Implicit axioms* for equality of each sort and ordering of each sort for which an ordering is appropriate. The axioms for $=_{\sigma}$, equality of sort σ, are (loosely omitting sort super- and subscripts):

$$\forall x(x=x),$$
$$\forall x_1,x_2(x_1=x_2 \rightarrow x_2=x_1),$$
$$\forall x_1,x_2,x_3((x_1=x_2 \wedge x_2=x_3) \rightarrow x_1=x_3),$$
$$\forall x_1,x_2((x_1=x_2 \wedge F(x_1)) \rightarrow F(x_2)),$$
$$\forall x(x=c_0 \vee \cdots \vee x=c_n),$$
$$\neg c_i=c_j \text{ for } 0 \leqslant i \neq j \leqslant n.$$

These axioms express that $=$ is a congruence relation on a finite domain, where every element has exactly one name. Let EQ denote the set of equality axioms for all sorts σ, then we have by definition that either $EQ \models c_i = c_j$ or $EQ \models \neg c_i = c_j$ for all i,j. The axioms for $<$ and $>$ are:

$\forall x_1, x_2, x_3((x_1 < x_2 \wedge x_2 < x_3) \rightarrow x_1 < x_3)$,

$\forall x(\neg x < x)$, and (possibly)

$\forall x_1, x_2(x_1 < x_2 \vee x_1 = x_2 \vee x_2 < x_1)$,

$\forall x_1, x_2(x_1 < x_2 \leftrightarrow x_2 > x_1)$,

$\Delta \subset \{c_i < c_j \mid 0 \leqslant i \neq j \leqslant n\} \cup \{\neg c_i < c_j \mid 0 \leqslant i, j \leqslant n\}$ (to be explained below).

These axioms express that $<$ is a transitive, irreflexive and (possibly) total ordering with inverse $>$. Let O denote the set of ordering axioms. We require that Δ is such that either $O \models c_i < c_j$ or $O \models \neg c_i < c_j$ for all i,j.

The idea behind the implicit axioms is that $=$ and $<$ are provided by the system and have a fixed meaning, whereas the other predicates are user-defined.

3.2. A *universally quantified expert system* differs from an indexed propositional one by allowing not only constants, but also variables in the explicit axioms. All explicit axioms are assumed to be universally closed.

3.3. Among the theories that do not fall under 3.1 and 3.2 are many theorem provers. A theorem prover might be an undecidable theory. The theoretical observations below show that, from a logical point of view, expert systems as defined in 3.1 and 3.2 are very simple, decidable theories.

3.4. LEMMA. *For every* $S \in$ SENT *there exists a boolean combination* S' *of closed atoms such that* $EQ \models S \leftrightarrow S'$.

PROOF. Replace inductively every subformula $\forall x F(x)$ by $F(c_0) \wedge \cdots \wedge F(c_n)$ and $\exists x F(x)$ by $F(c_0) \vee \cdots \vee F(c_n)$ where $\forall x(x = c_0 \vee \cdots \vee x = c_n) \in EQ$. □

3.5. LEMMA. *For every closed atom* A *there exists a boolean combination* A' *of c-atoms such that* $EQ \models A \leftrightarrow A'$.

PROOF by giving a typical example. Let A be for instance $P(f(f'(c^{\sigma_0}), c^{\sigma_1}), c^{\sigma_2})$ with f of type $\sigma_2 \times \sigma_1 \rightarrow \sigma_0$ and f' of type $\sigma_0 \rightarrow \sigma_2$. Let A^3 be the sentence $\exists x^{\sigma_2} \exists y^{\sigma_0}[f'(c^{\sigma_0}) = x^{\sigma_2} \wedge f(x^{\sigma_2}, c^{\sigma_1}) = y^{\sigma_0} \wedge P(y^{\sigma_0}, c^{\sigma_2})]$. Now apply Lemma 3.4 to A^3 and obtain A'. □

3.6. Both lemmas above can cause combinatorial explosions. Therefore they are only of theoretical use. They tell us that, in the presence of EQ, boolean combinations of c-atoms have the same expressive power as full many-sorted predicate logic.

3.7. LEMMA. *If* $EQ \subset \Gamma \subset$ SENT, *then every model of* Γ *is elementarily equivalent to a model whose domains consist of exactly the interpretations of all constants.*

PROOF. Let $\mathfrak{M}$ be a model of Γ with $EQ \subset \Gamma \subset$ SENT. By EQ the interpretations of the equality predicates in $\mathfrak{M}$ are equivalence relations which are congruences with respect to the interpretation of all other predicate and function symbols in $\mathfrak{M}$. It follows that $\mathfrak{M}$ and $\mathfrak{M}/_=$, the quotient structure of $\mathfrak{M}$ modulo equality, are elementarily equivalent. By the axioms $\forall x(x = c_0 \vee \cdots \vee x = c_n)$ and

$\neg c_i = c_j$ $(i \neq j)$ from *EQ*, it follows that the domains of $\mathfrak{M}/_=$ consist of exactly the interpretations of all constants. □

3.8. LEMMA. *If* $EQ \subset \Gamma \subset$ SENT, *then* $Th(\Gamma)$ *is decidable.*

PROOF. Let Γ be such that $EQ \subset \Gamma \subset$ SENT. Then we can apply Lemma 3.7 and observe that there are just finitely many non-isomorphic $\mathfrak{M}/_=$'s. In other words: up to dividing out $=$ and renaming there are just finitely many different models of Γ. Moreover all models of Γ are finite. Hence we can, for any given $S \in$ SENT, test in finite time whether S holds in all models of Γ or not. □

4. TESTING CONSISTENCY

4.1. Definition 2.10 and Lemma 3.7 suggest the following procedure for testing the consistency of theories Γ with $EQ \subset \Gamma$: generate all many-sorted structures whose domains consist of exactly the interpretations of all constants, and test each time whether the many-sorted structure is a model of Γ or not. This procedure is in general not feasible. A first step towards a feasible consistency test is the alternative characterization of consistency for indexed propositional expert systems, described in the following paragraphs.

Let Γ be an axiomatization of an indexed proposition expert system (see 3.1). Let $P_1,...,P_m$ enlist all c-atoms of the form $P(c,...,c')$ occurring in Γ, and let $t_1,...,t_n$ enlist all terms $f(c,...,c')$ occurring in Γ. The idea behind the following construction is that it is not necessary to have an entire many-sorted structure to be able to interpret Γ.

Let $A_\sigma = \{c_0^\sigma,...,c_n^\sigma\}$ be the set of all constants of sort $\sigma \in \Sigma$. Equality of sort σ is interpreted by syntactical identity on A_σ. Then all equality axioms from *EQ* are satisfied. The (eventual) ordering on A_σ is induced by O, i.e. $c_i < c_j$ if and only if $O \models c_i < c_j$. We need the following notions:

- A *truth valuation* of $P_1,...,P_m$ is an assignment of either *TRUE* or *FALSE* to each P_i $(1 \leq i \leq m)$.
- A *valuation* of $t_1,...,t_n$ is an assignment of an element of the appropriate domain A_σ to each t_j $(1 \leq j \leq n)$. It will be clear that a truth valuation of $P_1,...,P_m$ and a valuation of $t_1,...,t_n$ suffice for an interpretation of Γ. Each model of Γ yields a truth valuation of $P_1,...,P_m$ and a valuation of $t_1,...,t_n$. Conversely, each truth valuation of $P_1,...,P_m$ and valuation of $t_1,...,t_n$ for which every explicit axiom of Γ is true, can be extended in an arbitrary way to a many-sorted structure $\mathfrak{M} \models \Gamma$. Thus we have the following

THEOREM. *Let conditions be as above. Then we have:* Γ *is consistent if and only if there exists a truth valuation of* $P_1,...,P_m$ *and a valuation of* $t_1,...,t_n$ *for which every explicit axiom of* Γ *is true.*

4.2. Theorem 4.1 suggests a simple algorithm for testing the consistency of an indexed propositional expert system Γ: generate all valuations and truth valuations and test each time whether the explicit axioms of Γ are satisfied or not. This clearly leads to combinatorial explosions. A second step towards a feasible consistency test is imposing language restrictions on our expert systems.

A common language restriction is Horn format. A *clause* is a finite disjunction of atoms and negations of atoms (so called positive and negative *literals*). A *conjunctive normal form* is a finite conjunction of clauses. A *Horn clause* is a clause containing at most one positive literal. A *unit clause* is a clause containing one positive literal. The connection between Horn clauses and production rules is easily seen by the equivalence of $(A_1 \wedge \cdots \wedge A_k) \rightarrow B$ and $A_1^- \vee \cdots \vee A_k^- \vee B^+$, where the superscripts $+$ and $-$ denote whether a literal occurs positively or negatively. However, the implicit axioms $\forall x(x = c_0 \vee \cdots \vee x = c_n)$ and $\forall x_1,x_2(x_1 < x_2 \vee x_1 = x_2 \vee x_2 < x_1)$ are not Horn clauses. As a consequence we can only require the explicit axioms to be Horn clauses. This is not sufficient for feasible consistency checking, as will be demonstrated in the next paragraph. The idea is to reduce the satisfiability problem for propositional logic, which is known to be NP-complete (see [GJ]), to the consistency problem of indexed propositional expert systems, all whose explicit axioms are Horn clauses.

Let σ be a sort with exactly two constants: c_0^σ and c_1^σ. Then we have $\forall x(x=c_0 \vee x=c_1) \in EQ$. For any propositional atom A, let f_A be a function symbol of type $\sigma \rightarrow \sigma$. It is not difficult to see that

$$\{\neg f_A(c_0)=c_0 \vee A, f_A(c_0)=c_0 \vee \neg A\} \bigcup EQ \models (\neg f_A(c_0)=c_1) \leftrightarrow A.$$

So the *positive* literal A is equivalent to the *negative* literal $\neg f_A(c_0)=c_1$ in any Γ containing the two Horn clauses $\neg f_A(c_0)=c_0 \vee A$ and $f_A(c_0)=c_0 \vee \neg A$ and the equality axioms EQ. Let C be any propositional conjunctive normal form. Let Γ_C be the indexed propositional expert system with explicit axioms $\neg f_A(c_0)=c_0 \vee A$ and $f_A(c_0)=c_0 \vee \neg A$ for all positive literals A occurring in C. Then Γ_C is consistent and satisfies $\Gamma_C \models C \leftrightarrow H_C$, where H_C is the set (conjunction) of Horn clauses obtained from C by replacing all positive literals A by their equivalent negative literal. Moreover we have that C is satisfiable if and only if $\Gamma_C \bigcup H_C$ is consistent. This yields a polynomial reduction of the satisfiability problem for propositional logic to the consistency problem of indexed propositional expert systems, all whose explicit axioms are Horn clauses. A similar reduction could be established using $\forall x_1,x_2(x_1<x_2 \vee x_1=x_2 \vee x_2<x_1)$ instead of $\forall x(x=c_0 \vee x=c_1)$.

4.3. In the previous subsection we showed that testing consistency is NP-hard without further language restrictions. The problem is to specify language restrictions, which are strong enough to guarantee a feasible consistency test, and still allow enough expressivity for some given application. This third and final step towards a feasible consistency test will be achieved in the following theorem by the introduction of constants *undefined*, which are different from (wrt. $=$) and incomparable with (wrt. $<$) any other constant.

Theorem. *Let Γ be the axiomatization of an indexed propositional expert system satisfying the following conditions:*
(1) All explicit axioms of Γ are Horn clauses.
(2) For every sort $\sigma \in \Sigma$ there exists at least one constant, which does not occur in the explicit axioms of Γ. Such a constant , say c_0^σ, will be denoted by undefined$^\sigma$. So we have $\forall x(x=undefined \vee x=c_1 \vee \cdots \vee x=c_n) \in EQ$ and $\neg undefined=c_i \in EQ$ for all $1 \leqslant i \leqslant n$.
(3) For every sort $\sigma \in \Sigma$ the (eventual) ordering of sort σ is partial with respect to undefined$^\sigma$. More precisely: we require that $O \models \neg undefined<c_i$ and $O \models \neg undefined>c_i$ for all $1 \leqslant i \leqslant n$.
(4) The orderings $<$ and $>$ do not occur in the positive literals occurring in the explicit axioms of Γ.
Then the consistency of Γ can be tested in polynomial time.

Proof. Enlist the explicit axioms of Γ as follows (the super- and subscripted capitals denote c-atoms):

$$A_1^+$$
$$\cdot$$
$$\cdot$$
$$A_p^+$$
$$B_{1,1}^- \vee \cdots \vee B_{1,m_1}^-$$
$$\cdot$$
$$\cdot$$
$$B_{q,1}^- \vee \cdots \vee B_{q,m_q}^-$$
$$C_{1,1}^- \vee \cdots \vee C_{1,n_1}^- \vee D_1^+$$
$$\cdot$$
$$\cdot$$
$$C_{r,1}^- \vee \cdots \vee C_{r,n_r}^- \vee D_r^+$$

Now apply the following well-known algorithm, which is in fact a special case of hyper-resolution (see [R]).

WHILE cancellations possible AND no clause empty

DO

cancel all clauses that contain a literal $B_{i,j}^-$, $C_{i,j}^-$ *or* D_k^+ *which is implied by a unit clause*;

cancel all literals $B_{i,j}^-$*'s* and $C_{i,j}^-$*'s whose complement is implied by a unit clause*

(and possibly get new unit clauses D_k^+ !)

OD;

IF empty clause occurs OR unit clauses $f(c,...,c') = c_i$ and $f(c,...,c')=c_j$ *occur with* $i \neq j$

THEN Γ *is inconsistent*

ELSE Γ *is consistent*

Note that the WHILE-loop terminates since the number of c-atoms involved is strictly decreasing. By the cancellations the explicit axioms of Γ are transformed into an equivalent set of Horn clauses having the property that no unit clause implies nor refutes any negative literal. Since we have either $EQ \models c_i=c_j$ or $EQ \models \neg c_i=c_j$ and either $O \models c_i<c_j$ or $O \models \neg c_i<c_j$, it follows that no term $f(c,...,c')$ occurs both in a unit clause and in a negative literal. As a consequence the consistency of Γ in the ELSE-part above can be seen by applying Theorem 4.1 with the following truth valuation of $P_1,...,P_m$ and valuation of $t_1,...,t_n$:

$$P_i = \begin{cases} \textit{TRUE} \text{ if } P_i \textit{ occurs as unit clause} \\ \textit{FALSE otherwise} \end{cases}$$

$$t_j = \begin{cases} c_i \text{ if } t_j=c_i \textit{ occurs as unit clause} \\ \textit{undefined otherwise} \end{cases}$$

The algorithm is clearly polynomial (quadratic in the number of occurrences of literals). □

REMARKS.

4.3.1. As follows by close inspection of the proof above, it would suffice to require the following weakening of condition (2): for every sort σ for which a function symbol f of type $...\rightarrow\sigma$ occurs in a negative literal occurring in the explicit axioms of Γ, there exists at least one constant which does not occur on the right-hand side of an equation occurring in such a literal. However, we think it is more systematic to require condition (2) as it stands.

4.3.2. Condition (4) can not be missed, which can be seen as follows. Assume for some sort σ we have exactly three constants different from *undefined*, which are totally ordered by $c_1<c_2<c_3$. Then the unit clause $f(c)>c_1$ is equivalent to $f(c)=c_2 \vee f(c)=c_3$, which enables a similar construction as in the second paragraph of 4.2.

4.3.3. In view of condition (1), occurrences of *notsame* $<o,a,v>$ in the antecedent of a rule can be problematic. This problem can be overcome by postponing the consistency test until those occurrences evaluate to either *TRUE* or *FALSE*. Then the knowledge base can be transformed into an equivalent knowledge base satisfying (1).

4.4. Let us briefly discuss the semantical consequences of the conditions (2) and (3) from the previous theorem, since they may slightly deviate from the intended meaning of the knowledge base. It is possible that the consistency of Γ essentially depends on the valuation $f(c,...,c')=$ *undefined*, i.e. that any valuation $f(c,...,c')=c_i$ $(1 \leqslant i \leqslant n)$ would not yield a model for Γ. One could say that $f(c,...,c')=$ *undefined* possibly saves the expert system from inconsistencies by preventing production rules with occurrences of $f(c,...,c')$ in the antecedent from firing. Since such rules have obviously not been used in the inference, this may be considered an advantage. On the other hand, this may be considered a disadvantage in cases where $f(c,...,c')=$ *undefined* is not realistic (e.g. *temperature* (*patient*) = *undefined*). In those cases we suggest to add the appropriate unit clause $f(c,...,c')=c_i$ and to test consistency again.

4.5. It is not difficult to generalize the algorithm of 4.3 to universally quantified expert systems, although some care has to be taken in quantifying x in clauses containing literals of the form $\neg f(c,...,c')=x$. In those cases only restricted quantification of the form $\forall x \neq$ *undefined* is allowed. Of course, hyper-resolution can become very inefficient (from $P(c_0)$, $P(c_1)$, $\neg P(x_1) \vee \cdots \vee \neg P(x_n) \vee Q(x_1,...,x_n)$, for instance, 2^n instances of Q are generated), so we suggest limited use of variables (or, preferably, the use of a more efficient algorithm).

References

[BS] B.G. Buchanan, E.H. Shortliffe, *Rule-based expert systems: the Mycin experiments of the Stanford Heuristic Programming Project*. Addison-Wesley, Reading, Massachusetts (1984).

[GJ] M.R. Garey, D.S. Johnson, *Computers and intractability: a guide to the theory of NP-completeness*. Freeman, San Francisco, California (1979).

[IL] T. Imielinski, W. Lipski jr., *Incomplete information in relational databases*. Journal of the ACM 31, 4, pp. 761-791 (1984).

[L] P.J.F. Lucas, *Knowledge representation and inference in rule-based expert systems*. Report CS-R8613, Centre for Mathematics and Computer Science, Amsterdam (1986).

[M] J.D. Monk, *Mathematical Logic*, Springer Verlag, Berlin (1976).

[R] J.A. Robinson, *Automatic deduction with hyper-resolution*. International Journal of Computer Math. 1, pp. 227-234 (1965).

[Re] R. Reiter, *Equality and domain closure in first order databases*. Journal of the ACM 27, 2, pp. 235-249 (1980).