QUANTITATIVE DEDUCTION AND ITS FIXPOINT THEORY

M.H. van Emden

Department of Computer Science
University of Waterloo
Waterloo, Ontario, Canada
N2L 3G1

Research Report CS-85-15
June 1985

Quantitative deduction and its fixpoint theory

M.H. van Emden

Department of Computer Science
University of Waterloo
Waterloo, Ontario N2L 3G1
Canada

Abstract

Logic programming provides a model for rule-based reasoning in expert systems. The advantage of this formal model is that it makes available many results from the semantics and proof theory of first order predicate logic. A disadvantage is that in expert systems one often wants to use, instead of the usual two truth values, an entire continuum of "uncertainties" in between. That is, instead of the usual "qualitative" deduction, a form of "quantitative" deduction is required.

In this paper I present an approach to generalizing the Tarskian semantics of Horn clause rules to justify a form of quantitative deduction. Each clause receives a numerical attenuation factor. Herbrand interpretations, which are subsets of the Herbrand base, are generalized to subsets which are fuzzy in the sense of Zadeh. I show that as result the fixpoint method in the semantics of Horn clause rules can be developed in much the same way for the quantitative case.

As for proof theory, the interesting phenomenon is that a proof should be viewed as a two-person game. The value of the game turns out to be the truth value of the atomic formula to be proved, evaluated in the minimal fixpoint of the rule set. The analog of the Prolog interpreter for quantitative deduction becomes a search of the game tree (=proof tree) using the alpha-beta heuristic well-known in game theory.

1. Introduction

Developers of expert systems have found the usual logical reasoning in terms of the truth values **true** and **false** insufficient for their purpose. They have implemented several quantitative alternatives, where these truth values are replaced by probabilities or other measures of uncertainty. Of the different alternatives none has been sufficiently convincing to replace the others, nor has it become clear which method is preferable in a given situation.

This state of affairs is perhaps unavoidable, at least temporarily. It is a consequence of working backward from the application. We should complement this approach with another one working forward from a well-established, coherent method of world description and reasoning to obtain a quantitative alternative to existing methods. A difference with working backward from the application is that coherence, rather than applicability, has priority.

We have chosen the Horn clause subset of first-order predicate logic as a well-established, coherent method of world description and reasoning. It is well-established by having a semantics in the sense of Tarski. It is coherent by the existence of correctness and completeness results of its proof procedures. The latter include proof procedures that are understandable by humans in terms of problem reduction and that are computer-implementable with sufficient speed. Sets of Horn clauses used in this way are also known as "logic programs" written in the pure subset of the Prolog language. In this paper we will interchangeably use "rule sets" and "logic programs" for sets of positive Horn clauses, both in their qualitative and quantitative forms.

Having chosen our starting point, in what way will we now choose our extension? We regard the truth value **false** as the real number 0, **true** as the real number 1. We extend the concept of truth value to include all real numbers in between. There are two concepts in Horn clause logic that we generalise from the qualitative to the quantitative. The first is the implication in a rule, consisting of a condition and a conclusion. The implication can be regarded as transferring truth from the condition to the conclusion. We associate with each implication an *attenuation factor* f which can be thought of as contributing $f \times t$ to the truth value of the conclusion if t is the truth value of the condition. Of course the precise definition of the meaning of rules is determined by the mathematical definition, to be given later on, of when a rule is true in a given interpretation.

The second concept to be generalised is that of the *interpretation*, an abstract kind of possible world, in which a given rule set is true or false, according to a precise definition. These interpretations are usually thought of as relational structures, specifying which individuals stand in which relation. Each relation by itself is readily thought of as equivalent to a set of tuples of individuals: namely the set of exactly those tuples among which the relation holds. It is perhaps less familiar to think of several relations among individuals as a set, but this has important advantages. A set I of variable-free atomic formulas can be used to specify a relational structure, and hence an interpretation. For example, we can consider, for some given P, the set R of all tuples $(t_1, \ldots, t_n)$ such that the formula $P(t_1, \ldots, t_n)$ is in I. Then R is one of the relations of I, when I is regarded as a relational structure.

Now, in the usual relational structure, two individuals either stand in a given relation, or they do not. It may be useful to allow other alternatives. We obtain quantitative interpretations by regarding them as *fuzzy* subsets (in the sense of Zadeh [10]) of the set of all variable-free atomic formulas. A result is that a relational structure can specify that a relation between individuals holds with a certain strength, characterised by any natural number between 0 and 1. Strength 0 (1) then can be taken to correspond to the absence (presence) of the relation in the conventional sense.

Because quantitative interpretations can be operated on in much the same way as the usual ones, we can transfer the existing results on the semantics of logic programs into close analogs which hold for the quantitative version of the theory. In particular, the powerful method of fixpoints is used in a similar fashion. Sometimes not only the theorems, but also the proofs can be adopted unchanged. At other times interesting differences emerge, such as in the approximation theorem for least fixpoints.

In proof theory we also find strong similarities and some interesting differences. In the first place, we have to make it clear what it means to generalise proofs to the quantitative case. We consider the simple situation where the question, of which the answer is to be proved, is a single variable-free atomic formula A. It is supplied with a truth value t between 0 and 1 and we expect a proof to prove that A has membership value at least equal to t in the minimal model of the rule set. As before, the qualitative case is a special case: there the supplied truth value is implicitly 1; from the proof we conclude that A belongs (in the non-fuzzy sense) to the minimal model.

Proofs using the usual Horn clauses are found by searching the and/or tree associated with the set of clauses. The Prolog interpreter performs this search depth-first, from left to right. It is well-known that and/or trees can be regarded as game trees for a suitably chosen game. This game is determined, of course, by the rules generating the and/or tree. The values 0 and 1 correspond to **false** and **true** in the and/or tree. In the game tree they correspond to loss and win for one of the players, say, White.

Usually, however, game trees are too large to be evaluated to the end of the game. As a result *heuristic* game values have to be used: these are real numbers that can be normalised to lie between 0 and 1, giving a degree of expectation that White can win from the associated position. There is a well-known algorithm for searching game trees with heuristic game values; it is called ***alpha-beta***.

When we translate back from game trees with heuristic game values, we obtain and/or trees with fuzzy truth values. This suggests that a version of alpha-beta search is suitable for answering questions to quantitative rule sets. A quick check will show that in the special case of rules with all factors equal to 1, the alpha-beta search gives the same behaviour as a Prolog interpreter.

Now some remarks on related work. In a sense, [3] is most closely related: all results in it find their quantitative counterparts in the present work. In another sense, Shapiro's [8] is most closely related, as it is the only existing paper giving a quantitative treatment of the semantics and proof theory of logic programs. Shapiro gives a general method for computing uncertainties, of which ours $(2.1'.c)$ is a special case. Our decision to forego Shapiro's level of generality is richly rewarded in terms of results. In fact, Shapiro's only semantical result is his quantitative version of the fact that the intersection of Herbrand models is a model, for Horn clauses. Of course, this result in [8] is proved for a whole class of quantitative schemes; ours only for one.

Finally, a word about the organisation of this paper. In this investigation we have been guided by a close analogy between the qualitative and the quantitative case of rule-based reasoning. To make this analogy as clear as possible, we often first remind the reader of existing definitions or results in the qualitative case and then present the quantitative analogy. In such situations we use a numbering scheme to reflect the analogy: if the qualitative item is numbered $(n.m)$, then the corresponding quantitative item is numbered $(n.m')$. Independently of this parallel presentation, numbers with a prime reference quantitative items.

2. Syntax and semantics of finite sets of rules

Syntax is quickly disposed of. Qualitative rules have a conclusion and a condition. We will write such a rule in the style

$$A \leftarrow B_1 \,\&\, \cdots \,\&\, B_n \,, \quad n \geq 0 \,.$$

A quantitative rule also has a conclusion and a condition (with the same syntax as in the qualitative case). In addition it has a *factor*, a real number f in the interval $(0,1]$. This component is called thus because it will appear from the meaning of a rule, defined later on, that the rule contributes to the conclusion a truth value which is f times the truth value of the condition. We write the rule as

$$A \xleftarrow{\;f\;} B_1 \,\&\, \cdots \,\&\, B_n \,, \quad n \geq 0 \,.$$

The intention is to show the factor f "embedded" in the arrow.

Having disposed of syntax, we now turn our attention to semantics. In both the qualitative and the quantitative case, the Herbrand base B_P of a set P of rules is defined as the set of all variable-free atomic formulas that can be formed with the symbols contained in P.

In the qualitative case, a Herbrand interpretation is defined as a mapping $B_P \rightarrow$ {**false**, **true**}, where the range is the set of truth-values. A Herbrand interpretation is often regarded as a subset of B_P, namely of those atomic formulas mapping to **true**. In the quantitative case the range is the interval $[0,1]$ of real numbers. Here a Herbrand interpretation can still be regarded as a subset of B_P, provided we consider the subset to be "fuzzy" in the sense of Zadeh [10]. By identifying 1 with **true** and 0 with **false**, we make qualitative interpretations a special case of quantitative ones. The mapping $B_P \rightarrow [0,1]$ can be thought of as the membership function characterising a "fuzzy" subset I of B_P.

All Herbrand interpretations, qualitative or quantitative, of a given set P of rules can be specified by a function *val* taking as arguments a variable-free atomic formula A, and an interpretation I and having as result $val(A, I)$, the value of the membership function for I at the argument A.

Now that we know what a Herbrand interpretation is, the first thing to straighten out is when a rule set is true in a given interpretation I. For the qualitative case we recall [3] the following

Definition (2.1)

a) A rule set is true in I iff every one of its rules is true in I.

b) A rule is true in I iff every one of its variable-free instances is true in I.

c) A variable-free instance $A \leftarrow B_1 \ \& \ \cdots \ \& \ B_n$ is true in I iff A is true in I or at least one of $B_1, \ldots, B_n$ is false in I. (This last "or" is the usual non-exclusive one.)

For the quantitative case we have the

Definition (2.1′)

a) A rule set is true in I iff every one of its rules is true in I.

b) A rule is true in I iff every one of its variable-free instances is true in I.

c) A variable-free instance $A \xleftarrow{f} B_1 \ \& \ \cdots \ \& \ B_n$ of a rule is true I iff

$$val(A, I) \geq f \times \min \{val(B_i, I) \mid i \in \{1, \ldots, n\}\}.$$

(We define $\min \varnothing = 1$.) □

It will be seen that parts a) and b) are the same as in the qualitative case. Note that for rules with $f = 1$ and I such that $val(A, I) = 0$ or $val(A, I) = 1$ for all A, c) is also the same as in the qualitative case.

Definition (2.2, 2.2′)

A Herbrand interpretation I such that a rule set P is true in I, is called a *Herbrand model* of P. □

In this paper we use the turnstile symbol (usually meaning logical implication) in a different way.

Definition (2.3)

For all rule sets P and all $A \in B_P$, $P \models \{A \leftarrow\}$ iff the right-hand side is true in every Herbrand model of P. □

Let us now consider the quantitative version:

Definition (2.3′)

For all rule sets P, all $A \in B_P$, and all $f \in [0, 1]$, $P \models \{A \leftarrow f - \}$ iff the right-hand side is true in every Herbrand model of P. □

In the quantitative case we have that $P \models \{A \leftarrow f - \}$ implies $P \models \{A \leftarrow f' - \}$ for any $f' \leq f$. Thus we should be careful to make as strong as possible a statement by making the f as large as possible. In the quantitative case one cannot fail to do so: it corresponds to $f = 1$.

Let us denote by $M(P)$ the set of Herbrand models of a rule set P. $\cap M(P)$ is also defined in the quantitative case if we adopt Zadeh's rule [10] for intersections:

$$val(A, \cap S') = inf\{val(A, S) : S \in S'\}$$

where S' is a family of Herbrand interpretations and inf is the greatest lower bound. In the qualitative case we found [3] as characterisation of the intersection of all Herbrand models:

$$\cap M(P) = \{A : A \in B_P \;\&\; P \models \{A \leftarrow\}\}. \tag{2.4}$$

Its quantitative analog is given in the following:

Theorem (2.4′)

$$val(A, \cap M(P)) = sup\,\{x \mid P \models \{A \leftarrow x - \}\}$$

where *sup* is the least upper bound.

Proof If P is a rule set, I a model of P, $A \in B_P$, and $P \models \{A \leftarrow x -\}$, then $\{A \leftarrow x -\}$ is true in I and, by (2.1′), $val(A,I) \geq x$. Therefore,

$$val(A, I) \geq sup\ \{x \mid P \models \{A \leftarrow x - \}\}$$

for any model $I \in M(P)$ and

$$val(A, \cap M(P)) \geq sup\ \{x \mid P \models A \leftarrow x - \}$$

Strict inequality in the above relation is impossible, as we have $P \models A \leftarrow val(A, \cap M(P)) -$ for all P and all $A \in B_P$. □

The method followed in [1,3,5] is to associate with each rule set P a mapping T_P from interpretations to interpretations and to show that fixpoints of T_P are models of P. Then various mathematical results about T_P can be used to discover properties of models.

Here we follow the same method. First a reminder of the definition in the qualitative case:

Definition (2.5)

$$T_P(I) = \{A \mid A \leftarrow B_1 \ \& \ \cdots \ \& \ B_n \text{ is a variable-free instance of a rule in } P \text{ and } B_1 \in I, \ldots, B_n \in I\}$$

For the quantitative case we use:

Definition (2.5′)

For every $A \in B_P$

$$val(A, T_P(I)) = sup\ \{f \times \min\{val(B_i, I) \mid i \in \{1, \ldots, n\}\} \mid A \leftarrow f - B_1 \ \& \ \cdots \ \& \ B_n \text{ is a variable free instance of a rule in } P\}$$

□

In the qualitative case the partial order of set inclusion among interpretations plays an important role. For the quantitative case we adopt Zadeh's definition [10] of inclusion among fuzzy sets. For two interpretations $I_1 \in B_P$, $I_2 \in B_P$ this gives

$$I_1 \subseteq I_2 \text{ iff } val(A, I_1) \leq val(A, I_2) \text{ for all } A \in B_P .$$

Just as in the qualitative case we define

$$I_1 = I_2 \text{ iff } I_1 \subseteq I_2 \text{ and } I_2 \subseteq I_1 .$$

It follows immediately from definition (2.5′) that T_P is a monotone function, for any rule set P. That is, $I_1 \subseteq I_2$ implies $T_P(I_1) \subseteq T_P(I_2)$. It is well-known that monotonicity implies that the least fixpoint $lfp(T_P)$ of T_P, namely

$$\cap \{I : T_P(I) = I\}$$

exists and is equal to

$$\cap \{I : T_P(I) \subseteq I\},$$

and dually for greatest fixpoints (see for example [1] or [5]).

A useful connection between models and fixpoints is established by theorem 2.6 which was first stated and proved for the qualitative case in [3]. It makes just as much sense, and is just as true, in the quantitative case.

Theorem (2.6, 2.6′)

For every rule set P and for every $I \subseteq B_P$, P is true in I iff $T_P(I) \subseteq I$.

Proof (If)

$I \supseteq T_P(I) \Rightarrow val(A, I) \geq val(A, T_P(I))$ for any $A \in B_P$. Moreover,

$$val(A, T_P(I)) \geq f \times \min \{val(B_i, I) \mid i \in \{1, \ldots, n\}\}$$

for any variable-free instance $A \xleftarrow{f} B_1 \,\&\, \cdots \,\&\, B_n$ of a rule in P, by (2.5′). Hence

$$val(A, I) \geq f \times \min\{val(B_i, I) \mid i \in \{1, \ldots, n\}\}$$

and this implies that P is true in I by (2.1′).

(Only if)

Let a rule set P and $A \in B_P$ be given. P is true in I implies that for all variable-free instances $A \xleftarrow{f} B_1 \;\&\; \cdots \;\&\; B_n$ of a rule in P we have

$$val(A, I) \geq f \times \min\{val(B_i, I) \mid i \in \{I, \ldots, n\}\}.$$

Hence

$$val(A, I) \geq sup\{f \times \min\{val(B_i, I) \mid i \in 1, \ldots, n\}\}$$
$$\mid A \xleftarrow{f} B_1 \;\&\; \cdots \;\&\; B_n \text{ is a variable-free instance of a rule in } P$$
$$\}$$

and $val(A, I) \geq val(A, T_P(I))$ by definition 2.5′. □

Theorem 2.6 enables us to study fixpoints of T_P to discover properties of Herbrand models. It implies, for example, that $\cap M(P) = lfp(T_P)$, the least fixpoint of T_P. It follows that $\cap M(P)$ is itself a Herbrand model because the monotonicity of T_P implies that $lfp(T_P) = \cap \{I : T_P(I) = I\}$ is itself a fixpoint of T_P.

Another important property is that

$$lfp(T_P) = \cup \{T^n(\varnothing) \mid n \in \mathbf{N}\}, \qquad (2.7, 2.7')$$

where $\mathbf{N}$ is the set of natural numbers.

Again, this both makes sense and is true in the qualitative as well as in the quantitative case if, in the latter, we interpret $\varnothing$ to mean the interpretation such that $val(A, \varnothing) = 0$ for all $A \in B_P$. It is easy to prove 2.7 and 2.7′ from the continuity of T_P.

Theorem (2.8, 2.8′)

T_P is continuous, i.e.

$$\cup \{T_P(I_j) \mid j \in \mathbf{N}\} = T_P(\cup \{I_j \mid j \in \mathbf{N}\})$$

for all sequences $I_1 \subsetneq I_2 \subsetneq \cdots$ of Herbrand interpretations.

Proof

We prove the theorem for the quantitative case. Let A be an atomic formula: $A \in B_P$. Let $\cup I$ be $\cup \{I_j \mid j \in \mathbf{N}\}$. According to the definition (2.5′) of T_P:

$val(A, T(\cup I)) =$

$sup \, \{f \times \min \{val(B_k, \cup I) \mid k \in \{1, \ldots, n\}\}$

$\mid A \leftarrow f - B_1 \,\&\, \cdots \,\&\, B_n$ is a variable−free instance of a rule in P

$\}$.

Let us use α to enumerate the variable-free instances of rules in P having A as conclusion. Thus $A \leftarrow f_\alpha - B_{\alpha 1}, \& \cdots \& B_{\alpha n_\alpha}$ is the α-th such variable-free instance. Then we can shorten the above expression to:

$$val(A, T(\cup I)) = sup_\alpha(f_\alpha \times \min_k val(B_{\alpha k}, \cup I)) .$$

Using $val(B_{\alpha k}, \cup I) = sup_j(val(B_{\alpha k}, I_j))$ where j indexes the monotone sequence $I_1 \subseteq I_2 \subseteq \cdots$ of Herbrand interpretations, we have

$$\begin{aligned} val(A, T(\cup I)) &= sup_\alpha \, sup_j \, f_\alpha \times \min_k(val(B_{\alpha k}, I_j)) \\ &= sup_\alpha \, sup_j \, v_{\alpha j} \end{aligned}$$

where $v_{\alpha j} = f_\alpha \times \min_k(val(B_{\alpha k}, I_j))$.

Using the same notation we find

$$val(A, \cup \{T(I_j) \mid j \in \mathbf{N}\}) = sup_j \, sup_\alpha \, v_{\alpha j} .$$

It remains to show that

$$sup_\alpha\ sup_j\ v_{\alpha j} = sup_j\ sup_\alpha\ v_{\alpha j}\ .$$

The set of all $v_{\alpha j}$ is bounded above and therefore has a least upper bound, say, v. Of course, $sup_\alpha\ sup_j\ v_{\alpha j}$ is an upper bound for the set of all $v_{\alpha j}$. Hence,

$$sup_\alpha\ sup_j\ v_{\alpha j} \geq v\ .$$

On the other hand, we have $sup_j\ v_{\alpha j} \leq v$ for all α. Hence $sup_\alpha\ sup_j\ v_{\alpha j} \leq v$. Thus we found that $sup_\alpha\ sup_j\ v_{\alpha j} = v$. Similarly one shows that $sup_j\ sup_\alpha\ v_{\alpha j} = v$, which completes the proof of the continuity of T_P. □

An important task of this section on semantics is to establish a theorem that can serve as foundation for the completeness result in the next section on proof theory. A completeness result for a proof method is of the form: if an assertion is true, then it can be proved according to the method. In qualitative case we consider an assertion of the form $A \leftarrow$, with $A \in B_P$, and we assume that it is true in all Herbrand models of the set P of rules. That is, we assume that $A \in \cap M(P)$. We know that

$$\cap M(P) = lfp(T_P) = \cup\ \{T_P^n\ (\emptyset)\ |\ n\ \in \mathbf{N}\}\ .$$

From the assumption that A is in the infinite union of the increasing sequence of sets $\emptyset \subseteq T_P(\emptyset) \subseteq T_P^2(\emptyset) \subseteq\ \cdots$ it follows immediately that there exists an $N \in \mathbf{N}$ such that $A \in T_P^N(\emptyset)$. T can be regarded as an operator adding one-step modus-ponens consequences to its argument set. Hence, without going into any details of proof theory, it should be plausible that from the fact that A is in a finite power of T applied to $\emptyset$, one can show that a finite proof of A exists according to a given proof method, not necessarily modus ponens.

In the quantitative case we follow roughly the same path. Instead of assuming $A \in \cap M(P)$, we now assume $val(A, \cap M(P) = \alpha$ and we ultimately want to show that $A \leftarrow\!\alpha\!-$ can be derived from P. Theorem (2.8′) helps us prove that in the quantitative case also $\cap M(P) = \cup\ \{T_P^n(\emptyset)\ |\ n\ \in \mathbf{N}\}$. But we would also like to draw the stronger conclusion from $val(A, \cap M(P)) = \alpha$ that there exists an $N \in \mathbf{N}$ such that $val(A, T_P^N(\emptyset)) = \alpha$. However, this seems to take more work than in the qualitative case. Here is one way of doing it.

Lemma (2.10′)

For any finite set P of rules, any $A \in B_P$, and any real $\epsilon > 0$

$$\{val(A, T^n(\emptyset)) \mid n \in \mathbf{N} \text{ and } val(A, T^n(\emptyset)) \geq \epsilon\}$$

is finite.

Proof Let F be the set of factors of rules in P. Note that F is finite by our assumption about P. Let m be the greatest element of F such that $m < 1$. The real number $val(A, T_P^n(\emptyset))$ is a product of a sequence of elements of F. In this sequence, at most q elements can be less than 1, if q is the smallest integer such that $m^q < \epsilon$. Because 1 can occur in the sequence any number of times, the sequence can have any length. Thus the number of different products $\geq \epsilon$ of the sequences of elements of F is not greater than $|F|^q$. □

For the qualitative case, we have the following

Theorem (2.11)

For all sets P of rules, and all $A \in B_P$, $A \in \cap M(P)$ implies that there exists an $N \in \mathbf{N}$ such that $A \in T_P^N(\emptyset)$.

Proof $\cap M(P) = lfp(T_P) = \cup \{T_P^n(\emptyset) \mid n \in \mathbf{N}\}$. For A to be in this infinite union it is necessary that $A \in T_P^n(\emptyset)$ for some $n \in \mathbf{N}$. □

Its quantitative analog is:

Theorem (2.11′)

For all finite sets P of rules, and all $A \in B_P$, there exists an $N \in \mathbf{N}$ such that $val(A, \cap M(P)) = val(A, T_P^N(\emptyset))$.

Proof If $v = val(A, \cap M(P))$ is zero, then the N obviously exists: $N = 0$ will do. Suppose now that $v > 0$.

$$\cap M(P) = (\text{by } 2.6') \; lfp(T_P) = (\text{by } 2.8') \; \cup\{T_P^n(\emptyset) \mid n \in \mathbf{N}\}.$$

Hence,

$$val(A, \cap M(P)) = sup\,\{val(A,\ T_P^n(\emptyset))\ |\ n\ \in \mathbf{N}\} =$$

$$sup\,\{val(A,\ T_P^n(\emptyset))\ |\ n\ \in \mathbf{N}\ \text{ and }\ val(A,\ T_P^n(\emptyset)) \geq \epsilon\}$$

for any ϵ smaller than v. If we choose such an ϵ positive, which we can do by our assumption about v, then the latter set is finite by lemma (2.10′). Hence the least upper bound is attained for some $N \in \mathbf{N}$. □

The following example shows that the condition of P being finite is essential. Let

$$P = \{p(x) \xleftarrow{1} q(x),\ p(x) \xleftarrow{1} p(s(x))\} \cup$$
$$\{q(n) \xleftarrow{1-2^{-n}}\ |\ n\ \in \mathbf{N}\}$$

Consider

$$val(p(s^i(0)),\ T^n(\emptyset)) = \max\{val(q(s^i(0)),\ T^{n-1}(\emptyset))\ \ (= 1-2^{-i})$$
$$,\ val(p(s^{i+1}(0)),\ T^{n-1}(\emptyset))\ \}$$

For $n \geq 3$, the contribution from the second rule is greater for all i. Hence

$$val(p(s^i(0),\ T^n(\emptyset)) = 1-2^{2-i-n} \qquad \text{for}\ i \geq 0,\ n \geq 0,\ i+n \geq 3.$$

When we keep i fixed and let n increase, this quantity does not attain its least upper bound.

3. Proof theory for quantitative rules

Suppose we have the following rules

$$A \xleftarrow{\cdot 5} B \ \&\ F \qquad\qquad C \xleftarrow{\cdot 45}$$
$$A \xleftarrow{\cdot 5} C \ \&\ D \qquad\qquad D \xleftarrow{1}$$
$$B \xleftarrow{\cdot 2} \qquad\qquad E \xleftarrow{\cdot 5}$$

$$F \xleftarrow{\cdot 9} E$$

Suppose we want to prove that the truth value of A in the least model is at least ·2. The first rule tells us to try and prove the same for B & F with truth value at least ·4; and, hence, that we must both prove B and F with truth value at least ·4. We see that the best we can do for B is ·2. We therefore do not even try to prove F, and conclude that the first rule for A gets us nowhere. The second rule then allows us to prove that the truth value of A in the least model is at least $\cdot 225$.

In this section we need to describe precisely the proof procedure of which the above is an example and we justify its result using the semantics of quantitative rule-based reasoning as presented in the previous section.

As in the qualitative case, the proof procedure in quantitative deduction is a search of an and/or tree. This tree, determined by a set P of rules and an initial atom G, is defined as follows.

- There are two kinds of nodes: and-nodes and or-nodes.
- Each or-node is labelled by a single atomic formula.
- Each and-node is labelled by a rule from P and by a substitution.
- The descendants of every or-node are all and-nodes, and vice versa.
- The root is an or-node labelled by G.
- For every rule R in P with a left-hand side unifying with the atomic formula A (with most general unifier θ) in an or-node, there is an and-node descendant of the or-node labelled with R and θ. An or-node with no descendants is called a *failure node*.
- For every atomic formula B in the right-hand side of the rule labelling an and-node, there is a descendant or-node labelled with B. An and-node with no descendants is called a success node.

With each node of the and/or tree of a set P of rules we associate a real number. We call it the *value* of that node. The value of a success node is the factor of its associated rule. The value of a nonterminal and-node is $f \times m$ where m is the minimum of the values of its descendants and f is the factor of the rule labelling the and-node. The value of a failure node is 0. The value of a nonterminal or-node is the maximum of the values of its descendants.

A *proof tree* is a subtree of an and/or tree defined as follows. The root of the proof tree is the root of the and/or tree. An or-node of the proof tree which also occurs in the and/or tree, has *one* descendant in the proof tree which is one of the descendants of that node in the and/or tree. An and-node in the proof tree which also occurs in the and/or tree, has as descendants in the proof tree *all* of the descendants of that node in the and/or tree. Furthermore, all terminal nodes in a proof tree are success nodes. We assign values to proof tree nodes in the same way as we do to nodes in an and/or tree.

In the qualitative case, correctness of the (SLD-resolution) proof procedure says in its most elementary form: if $A \in B_P$ is proved, then $A \in \cap M(P)$. We could express correctness like this: results of the proof procedure are ***not more true*** than they are in the minimal model $\cap M(P)$. Formulated in this way, it immediately suggests the form of a corresponding correctness property in the quantitative case limited to finite and/or trees:

Theorem (3.1′)

For every set P of rules with a finite and/or tree and every $A \in B_P$, the value of the root in the and/or tree with A as root is not greater than $val(A, \cap M(P))$.

Proof Observe first that the value of the root in the and/or tree is the maximum of the values of the roots of its constituent proof trees. It can easily be verified that the value of the root of a proof tree with A as root is not greater than $T_P^{n+1}(\varnothing)$ where n is the length of a longest path from the root to a terminal node. Here one unit of path length is from or-node to or-node along the path. □

The above theorem says that the result of an and/or tree is *not too true*, and this we argued to be analogous to correctness in the qualitative case. Similarly, completeness is the property of being *true enough.*

Theorem (3.2′)

For every set P of rules with a finite and/or tree and every $A \in B_P$, the value v of the root in the and/or tree with A as root is at least $val(A, \cap M(P))$.

Proof We prove by induction on n that $v \geq val(A, T^n(\varnothing))$ for all $n \in \mathbf{N}$. When we have shown that, we can conclude that

$$v \geq sup\,\{val(A, T^n(\varnothing)) \mid n \in \mathbf{N}\} = val(A, \cup\,\{T^n(\varnothing) \mid n \in \mathbf{N}\}) = val(A, \cap M(P)) .$$

To start the inductive proof of $v \geq val(A, T^n(\varnothing))$, observe that it is true for $n = 0$. To prove the induction step, suppose that it holds for a certain value n_0 of n.

$$
\begin{aligned}
&val(A,\, T^{n_0+1}(\varnothing)) \\
&= sup\ \{f \times \min\ \{val(B_k,\, T^{n_0}(\varnothing)) \mid k \in \mathrm{N}\} \\
&\qquad \mid A \xleftarrow{f} B_1 \,\&\, \cdots \,\&\, B_n \ \text{ is a variable-free instance of a rule in } P \\
&\qquad \}
\end{aligned}
$$

By lemma (2.9′), the set over which the supremum is taken, is finite. Therefore the supremum must be attained for a variable-free instance, say,

$$A \xleftarrow{f} B_1 \,\&\, \cdots \,\&\, B_n,$$

of a rule in P, say, $R = A' \xleftarrow{f} B'_1 \,\&\, \cdots \,\&\, B'_n$. Thus we have

$$val(A,\, T^{n_0+1}(\varnothing)) = f \times \min\ \{val(B_k,\, T^{n_0}(\varnothing)) \mid k \in \mathrm{N}\}\,. \tag{3.3′}$$

Let us now consider the and/or tree for P having A as root. One of the descendants of the root must be the rule R. Because its left-hand side A' has A as variable-free instance, there is a most general unifier θ of A and A'. Hence one of the descendants of the root is the node (R, θ) labelled with R and θ. Its descendants are $B'_1\,\theta, \ldots, B'_k\,\theta$ with values $v'_1, \ldots, v'_k$ and having $B_1, \ldots, B_k$ respectively as variable-free instances.

By the induction hypothesis, $B_1, \ldots, B_k$ are roots of and/or trees having values $v_1, \ldots, v_k$ such that $v_i \geq val(B_i,\, T^{n_0}(\varnothing))$, $i = 1, \ldots, k$. Because $B'_1\,\theta$ has B_i as instance, we must have $v'_i \geq v_i$. For the value v of the entire and/or tree, with A as root, we have $v = f \times \min\ \{v'_i \mid i = 1,\ldots,k\}$ and hence

$$v \geq f \times \min\ \{val(B_i,\, T^{n_0}(\varnothing)) \mid i = 1, \ldots, k\}\,.$$

By (3.3′) we conclude that $v \geq val(A,\, T^{n_0+1}(\varnothing))$, which completes the induction step of the proof. □

4. Game-theoretic aspects of rule-based reasoning

Finally, after the fixpoint theory and the proof theory of quantitative deduction, we consider its game-theoretic aspects. We first review the main concepts of two-person games because these have close parallels to rule-based reasoning (see, for example, [6,7]), both qualitative and quantitative. These parallels suggest an algorithm for a quantitative version of a Prolog interpreter.

There are two players, White and Black, and there is a state (for example the disposition of pieces on a board; or of matches over heaps, as in Nim). Starting from the initial state, players take turns making a move, that is, changing the state according to the rules of the game. If no move exists for the player whose turn it is to move, then that player has lost, the other has won, and the game is over.

Optimal ways of playing a game can in theory be analyzed by means of the game tree. The nodes of a game tree are states, together with an indication of which player's turn it is to move. The root node is the initial state, with White to move. The descendants in the game tree of a White-to-move node n are all states resulting from the moves by White starting in n. These descendants then have the indication that it is Black to move.

Let us consider the game of Nim as an example. The state consists of a set of heaps of matches. A move consists of selecting a non-empty heap and removing at least one match from it. The game tree in figure 4.1 has as root a state in which there are two heaps of matches, one with 2, one with 1 match. If a node is a white circle, then it is White's turn to move.

How does a game tree, in principle at least, allow one to determine an optimal move? In the terminal nodes the rules determine which player has won. If it is White, then attach a 1 as value to the node, otherwise a 0. A nonterminal node of which all descendants have a value, obtains the maximum of the values of its descendants if it is a White-to-move node; otherwise the minimum. In this way the root in figure 4.1 obtains 1 as value, indicating that White can win against any play by Black (we say that it is a *forced win* for White). It can also be seen that there is just one initial move allowing White to win against any play by Black.

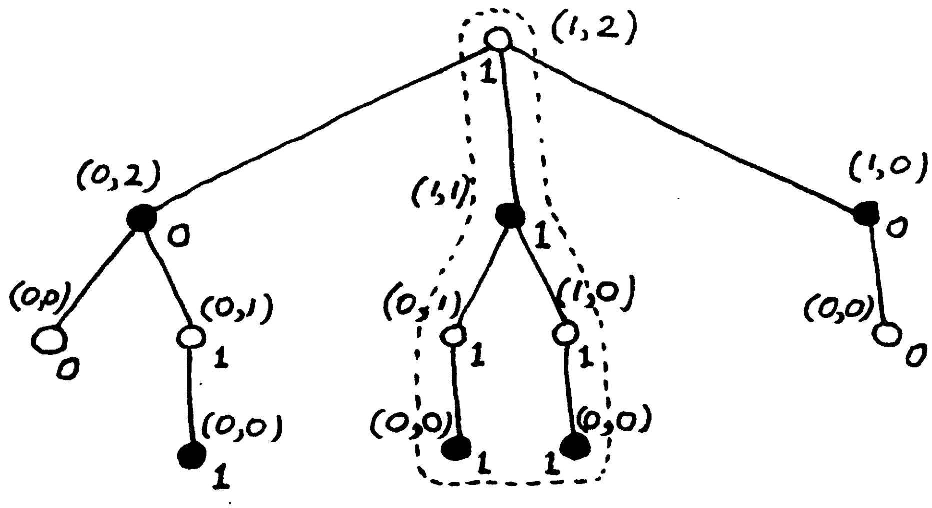

Figure 4.1

A game tree for the game of Nim. The sizes of the heaps are in parentheses. The colour of the node indicates whose turn it is to move (White or Black). The single numbers indicate the value of the game at the node. The dotted outline indicates the forcing tree.

How would White remember that sequence of moves to make in order to realise a forced win? Of course it can consult the game tree. But not all of it is needed. Of White's moves, all except one optimal move can be discarded, because White can choose its move. All of Black's countermoves have to be kept, because White has no control over Black's choice of move. What remains of the game tree is called the *forcing tree*. See figure 4.1 for an example.

To discover a forcing tree one need not always traverse the entire game tree. Consider in figure 4.1, for example, the node P. After noting that the move to P by Black causes White to lose, we need not traverse any sibling subtrees of P: we already know that the value of Q is 0 without visiting other descendants of Q (this is a so-called *beta cut-off*). Similar considerations, with the roles of White and Black reversed, allow us to skip other parts of the game tree. For example, one never has to look beyond the first forcing tree encountered (this is a so-called *alpha cut-off*). In figure 4.1 this eliminates the rightmost subtree of the root (this is an alpha cut-off). The process indicated here is called *alpha-beta pruning*.

If the game tree is too large to traverse completely, then it may be possible to obtain approximate game values. Thus, let us consider a subtree of the game tree by removing all descendants of certain nodes. Then, in a terminal node in the subtree, the game may not have ended. Such a node receives as value a number between 0 and 1, indicating the degree of expectation that its exact value is 1. In this way all terminal nodes receive values between 0 and 1 and all other nodes can be evaluated according the max-min rule given above.

To introduce the game-theoretic aspects of rule-based reasoning we first give an example of rules giving an and/or tree as close as possible to the game tree in figure 4.1. Let $W(x, y)$ mean that White can win starting in a state of two heaps, with x matches on the one heap and y matches on the other heap. The rules of Nim then state that

$$W(1, 2) \xleftarrow{1} W(0, 0) \,\&\, W(0, 1)$$

$$W(1, 2) \xleftarrow{1} W(0, 1) \,\&\, W(1, 0)$$

$$W(1, 2) \xleftarrow{1} W(0, 0)\ .$$

The and/or tree for these rules and the root $W(1, 2)$ is shown in figure 4.2. We have omitted rules saying that White can

win from (0, 1) and from (1, 0). It may not be possible to do this in a way yielding an and/or tree isomorphic to the game tree in figure 4.1.

It is not necessary for game trees to be exactly translatable to and/or trees for our purpose. What matters is that:

- or-nodes correspond to White-to-move nodes, and-nodes to Black-to-move nodes
- if all factors are 1 in quantitative rules with non-empty right-hand sides, then the values of nodes in an and/or tree are formed according to the same rules as in a game tree with heuristic approximations to true game values
- the proof tree in an and/or tree is the analog of the forcing tree in the game tree

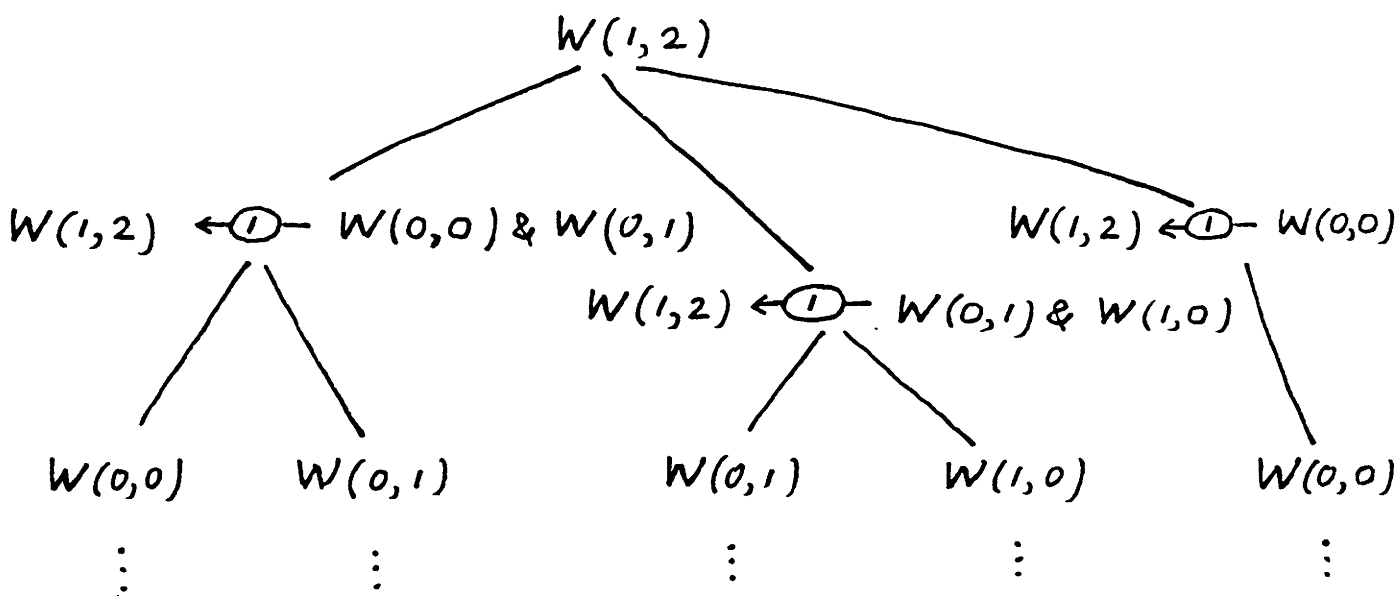

Figure 4.2

And/or tree corresponding to the upper part of the game tree in figure 4.1.

- when searching for a proof tree, in an and/or tree alpha-beta pruning is applicable in the same way as when searching for a forcing tree in a game tree. In the and/or tree an obvious modification takes care of the case when there are factors different from 1.

The last point is for us the main justification of comparing rule-based reasoning to two-person games: the well-known algorithm for alpha-beta pruning (see Winston [9] for a Lisp version; van Emden and Clark [4] for a Prolog version) can be used to interpret quantitative rules in principle with the same efficiency as a Prolog machine interprets qualitative rules. The only change required is to multiply the truth value by a suitable factor.

If an and/or tree is like a game tree, then what is the game behind the and/or tree? This must be a game with or-nodes as states in which White moves; with and-nodes as states which Black moves. White's move consists of selecting a rule matching the or-node in which it is to move. Black's move consists of selecting an atom from the condition in the rule labelling the and-node in which it is to move. In the qualitative case, White wins when Black is to move in a state having a rule with no condition. Black wins when White is to move in a state having an atom matching no rule. Note that neither a win by White nor a win by Black in itself implies the existence or otherwise of a proof. But if, in this game, White can win against *any play* by Black, then and only then a proof exists. This is implied by the correspondence, noted above, between proof trees and forcing trees.

5. Concluding remarks

Rule sets in expert systems are typically not recursive. Yet, conceptually at least, our restriction to finite and/or trees is disappointing. We would like to have as few differences as possible between expert systems and logic programs, where infinite and/or trees are common.

We have required our and/or trees to be finite because, in general, the min-max algorithm only works for finite trees. There are two exceptions. The first is the special case where truth values are only 0 and 1, that is, the qualitative case. If, in this case, an infinite and/or tree is scanned from left to right and there is a success node to the left of the leftmost infinite branch, then this branch is cut off by alpha-beta pruning. As a result, the value of the root is determined by a terminating algorithm. This describes the behaviour of a Prolog interpreter instructed to find one solution only.

Let us now consider the other special case where we can handle infinite and/or trees. This is the case where all factors of rules are less than 1. Here it is important that the search algorithm for and/or trees is not quite the same as alpha-beta search of game trees. There is one difference, conceptually trivial and easy to implement, which has interesting consequences. This is that in a game tree, black-to-move nodes receive as game value the minimum of the values of their descendants. The corresponding ***and*** nodes receive as value the minimum multiplied by the factor of the rule associated with the ***and*** node. When searching the tree from the top down, each ***or*** node has associated with it a certain threshold. The subtree below this node has to deliver a value at least as large as this threshold to be worth exploring. The threshold of the next *or* node down the tree will be $t \div f$, where f is the factor of the intervening rule. Thus, if factors are always less than 1 (and there are only finitely many rules), infinite branches, the bane of Prolog execution, are avoided: as one goes down the tree, thresholds must ultimately increase beyond 1, at which point the branch can be safely abandoned.

What is the role of "negation by failure" in quantitative deduction? Proof-theoretically, the phenomenon certainly occurs: if the initial goal is a variable-free atomic formula A with threshold greater than $val(A, \cap M(P))$ and if the and/or tree is favourable (by being finite or because the rules have suitable factors), then search will terminate without a proof tree being found.

Negation by failure usually means a semantical characterisation of this phenomenon. K.L. Clark [2] has shown for logic programs that the negation of a goal giving rise to finite failure is a logical implication of a strengthened version of the logic program. Note that logical implication means truth in *all* models, not just Herbrand models. In our treatment we only consider Herbrand models. This restriction seems to be essential: our models have to be sets so that we can make the transition to the quantitative case by changing these sets to be fuzzy. As we have warned our readers, the turnstile symbol ($\models$) is used in a nonstandard way: it means truth in all Herbrand models rather than in all models. Because of our restriction to Herbrand models we cannot contemplate a semantical characterisation of negation by failure in the sense of Clark.

Other characterisations of negation by failure are possible. For example, [1] proves that atoms giving rise to finite failed and/or trees are contained in the complement of the greatest fixpoint of the mapping T_P associated with the rule set P. (Lloyd [5] is a convenient source for this result as well as for negation by failure.) It is plausible that a similar result can be proved for the quantitative case.

6. References

[1] K.R. Apt and M.H. van Emden: Contributions to the theory of logic programming. Journal of the ACM, **29** (1982), pp 841-862.

[2] K.L. Clark: Negation by failure; pp 293-324 in: Logic and Databases, H. Gallaire and J. Minker (eds.), Plenum Press, 1978.

[3] M.H. van Emden and R.A. Kowalski: The semantics of logic as a programming language. Journal of the ACM, **23** (1976), 733-742.

[4] M.H. van Emden and K.L. Clark: The logic of two-person games. Chapter 12 (pp 320-340) in: micro-PROLOG: Programming in Logic by K.L. Clark and F.G. McCabe; Prentice Hall, 1984.

[5] J.W. Lloyd: Foundations of Logic Programming. Springer, 1984.

[6] N.J. Nilsson: Problem-Solving Methods in Artificial Intelligence. McGraw-Hill, 1971.

[7] N.J. Nilsson: Principles of Artificial Intelligence. Tioga Publishing Company, 1980.

[8] E.Y. Shapiro: Logic programs with uncertainties: a tool for implementing rule-based systems; Proc. 8th I.J.C.A.I., A. Bundy (ed.), pp 529-532. William Kaufmann, 1983.

[9] P.H. Winston: Artificial Intelligence. Addison Wesley, 1978.

[10] L.A. Zadeh: Fuzzy Sets; Information and Control **8** (1965), pp 338-353.