TEST GENERATION FOR EXPERT SYSTEMS WITH FUZZY TRUTH VALUES

JAAK TEPANDI
Tallinn Technical University
Akadeemia tee 1, Tallinn, Estonia, USSR 200108

ABSTRACT: We develop principles of testing, investigate complexity of testing, and develop methods for test data generation for expert systems (ES) that make use of the approximate reasoning techniques. The contribution adequacy criterion for testing ES is introduced, the problem of finding contribution adequate test sets for ES allowing fuzzy truth values is considered.

1. INTRODUCTION.

Expert systems (ES) have become commercially available and have emerged into areas requiring high reliability: hazardous industrial processes (transportation, chemical, and nuclear power), medicine, and others [Buc86]. This increases the need for the ES reliability. Meanwhile, vendors cannot guarantee quality of ES; verification and validation of ES has been characterized as a difficult task; most research done in this direction includes manual methods and concerns performance validation only [Ram87,Kee87].

In traditional software development life cycle, the urgent need for thorough and systematic testing of safety-related software (as well as for testing almost all other kinds of software) has been widely recognized, and many testing methods have been developed. This is not the case with the ES. So, the goal of the paper is to suggest principles for testing approximate reasoning in ES, to investigate the complexity of such testing, and to develop methods for test data generation. The work is based on our experience in development of a rule-based ES shell and several applications [Tep88], on methods for traditional software analysis and testing [Mye79, Inc87, EWI86, Tep88], as well as on test data generation methods for ES that do not make use of approximate reasoning [Tep89].

We will first introduce the contribution adequacy criterion for testing ES. This criterion may be applied to different types of declarative (or procedural-declarative) programs. We consider the problem of finding contribution adequate test sets for ES allowing fuzzy truth values.

2. THE CONTRIBUTION ADEQUACY CRITERION

By testing, one can not prove that there are no mistakes in the program - it is only possible to show their existence. Therefore, testing can not be exhaustive, and one has to decide, when to stop it. Several criteria have been developed for this purpose. For example, according to the statement adequacy criterion, each program statement must be executed at least once. It is possible to reformulate this criterion for production systems that do not make use of approximate reasoning [Tep89]: when the ES is executed with the given test data, then each rule must be activated at least once with all its premises satisfied (it is assumed that the activation is not cancelled during backtracking). In addition, it is required that each knowledge base statement must be activated at least once - for example, all goals stated in the KB must be evaluated, all explanatory texts must be displayed, etc.

This reformulation is justified for ES without approximate reasoning: usually, when a rule is activated with all its premises satisfied, then it supports a certain conclusion and takes active part of the decision process. In the contrary, in approximate reasoning the decision is often evaluated as aggregation of all information available about it. Therefore, a rule may be activated with the given test data, but its contribution to the reasoning process may be negligible. If there are no useless rules (rules that can not be used for achieving any goal) in the rule base, then the statement adequacy criterion may be fulfilled by evaluating all goals. This is certainly a necessary test, but in no case sufficient. We propose another criterion based on the following idea:

<u>Each component of the rule base should be used effectively, i.e. its contribution should have a crucial impact on the inference process.</u>

More precisely, let us have an expert system E, comprising a set of rules (a rule base) R, a set of possible different facts (propositions) F, a set of goals G ($G \subseteq F$), and an interpreter I. A test for E is a set of facts t from F (given, for example, with their uncertainty measures) , that may be used by the interpreter I to achieve some decision about the goals in G (e.g. to select a goal from G, or to assign fuzzy degrees of truth to the pairs "goal, value"). This decision is the result of t over E=<R,F,G,I>. A test set is a set of tests.

Let D be the set of possible decisions for E. Clearly, not all the decisions are really different from the user's point of view. For example, the conclusions "it is quite possible that the patient has influenza" and "it is very possible that the patient has influenza" seem almost identical, while the conclusions "the patient has influenza" and "the patient has pneumonia" are usually considered different. This motivates the following definition.

Definition 2.1. Let P be a partition of D into disjoint sets, such that each two different elements of P comprise solutions that are different from the user's point of view. We will say that such solutions are significantly different.

Definition 2.2.. A rule r∈R is effectively tested by a test T, if the result of T over <R,F,G,I> and the result of T over <{R\r},F,G,I> are significantly different. The contribution adequacy criterion states that testing of an expert system E may be stopped if each rule of E has been effectively tested. The test set used in such testing is contribution adequate.

Some important questions associated with the contribution adequacy criterion are: does there exist an effective test for a rule? How to find it? When are the algorithms for finding this test feasible (i.e. of polynomial complexity)? How to find a contribution adequate test set? The answers depend on the inference method, on the kind of knowledge the system is able to represent, etc. Therefore, in the next sections we will be more specific about E and its components.

3. THE RULE BASE, THE INTERPRETER, AND OTHER PRELIMINARY ISSUES

3.1. THE RULE BASE

Let r∈R. By PREMISES(r), CONCLUSION(r), UM(r), we will denote correspondingly the set of premises of r, the conclusion of r (the rule r leads to CONCLUSION(r)), and the uncertainty measure (UM) of r. If r,s∈R then by r=s we express the fact that r and s denote the same rule (and similarly for $r \neq s$). With every rule base R, we associate an AND\OR graph GRAPH(R). The AND nodes of the graph correspond to rules, the OR nodes and leaf nodes (nodes without successors) - to, correspondingly, derivable and nonderivable propositions. If $p \in$ PREMISES(r), then there are directed arcs in GRAPH(R) from CONCLUSION(r) (OR node) to r (AND node), and from r to p. A node n2 <u>supports</u> a node n1 if there is a path from n1 to n2 in the graph. We will consider only <u>loopfree</u> graphs, i.e. graphs that do not have any directed paths that begin and end at the same node.

For a node n, by GRAPH(R,n) we will denote the subgraph of GRAPH(R) comprising node n and all nodes supporting it. An AND graph A(R,n) with root n is a subgraph of GRAPH(R,n) with the following properties: (1) n is in A(R,n); (2) if m is an OR node in GRAPH(R,n) and m is in A(R,n), then exactly one of the immediate successors of m is in A(R,n); (3) if m is an AND node and m is in A(R,n) then all immediate successors of m are in A(R,n); (4) every path in A(R,n) ends in a leaf. A graph C(R,n,r) with root n <u>controlled by</u> r is a subgraph of GRAPH(r,n) with the following property: if m belongs to an AND graph A(R,n) that includes r, then m belongs to C(R,n,r). By NODES(G) and LEAVES(G) we will denote correspondingly the sets of all nodes and leaf nodes in a graph G. By UM(R,n,t) we will denote the UM of the node n in rule base R for test t, by UM(R, PREMISES(r), t) - the UM of the conjunction of premises of r for test t.

<u>Definition 3.1.</u> A node n in GRAPH(R) has <u>separated premises</u> if for any two different rules r1, r2 supporting n, it is true that PREMISES(r1)∩PREMISES(r2)=∅. E is an <u>ES with separated premises,</u> if for any two different rules r1, r2 from R, it is true that PREMISES(r1)∩PREMISES(r2)=∅. E is an ES with <u>separated goals</u>, if for any two different goals g1,g2, it is true that LEAVES(GRAPH(R,g1))∩LEAVES(GRAPH(R,g2))=∅. A goal g has a <u>private rule</u> r_p, if r_p leads to g and LEAVES(GRAPH(R,r_p))∩LEAVES(GRAPH(R,r)) =∅ for $r \in R \setminus \{r \in R \mid (r=r_p)$ or (r supports r_p)}. <u>E has private rules</u>, if every goal from G has a private rule.

For example, the sample rule base from [Leu88] , comprising over 60 rules, is an ES with private rules; all the goals from this ES have separated premises.

3.2. THE INTERPRETER

The interpreter I comprises three components: operations with UM, inference control, and the selection of the decisions.

We assume that there exist the order and equivalence relations (usual notations $=, <, \leq, >, \geq$ will be used) on the set V of UM. The notations f_{min} and f_{max} are used correspondingly for the minimum ("zero", "impossible") and maximum ("one", "sure") UM values. The test including only propositions set to f_{max} (f_{min}) will be denoted by t_{max} (t_{min}).

Operations with UM include AND , OR, a "Modus Ponens generating function" [Tri85] INFERENCE, COMPATIBILITY, and COMPOSITION [God88]; we shall consider only the first three of them. To develop test data generation algorithms, we must use the following properties of the operators: reducability to logical connectives (this is usually satisfied); completeness with respect to the truth values, for moving backward from conclusions to premises (properties (C1)-(C4) below - they are mostly satisfied for systems with uncertainties); strict monotonicity, for the generation of tests resulting in different UM values (properties (M1),(M2)); equality preserving, for determining the existence of an effective test (Lemma 3.1).

It is reasonable to require that if we allow the UM to take values only from the set {0,1}, then the aggregation of the individual premises of a rule reduces to logical AND, the deduction of a conclusion from given premises - to *modus ponens* (defined when the UM of the rule equals 1), and combination of conclusions - to the logical OR. For example, operators and systems described in [God88,Haj84,Tep88], satisfy these requirements. We also assume that the INFERENCE operator is based on the R-implication [Tri85,God88], i.e. satisfies the axioms for AND.

Definition 3.2. Let I make use of AND, INFERENCE, and OR operators. If I satisfies the properties that (C1) for the given truth value f_r of a rule r, for a given maximum possible truth value f_p of the premise of r, and for a given truth value $f_c \leq INFERENCE(f_p, f_r)$, it is possible to find a value $f \leq f_p$ such that $INFERENCE(f, f_r) = f_c$; (C2) for given f_1 and f_2, and for $f \leq AND(f_1, f_2)$, it is possible to find truth values $f_1' \leq f_1$ and

$f_2' \le f_2$ such that $AND(f_1', f_2') = f$; (C3) for given f_1 and f_2, and for $f \le OR(f_1, f_2)$, it is possible to find truth values $f_1' \le f_1$ and $f_2' \le f_2$ such that $OR(f_1', f_2') = f$; (C4) for given $f_2 > f_1$ it is possible to find f such that $OR(f, f_1) = f_2$ - then we will say that I (or the corresponding operator) <u>is complete with respect to (wrt) the truth values</u>.

<u>Definition 3.3.</u> Let $f' < f''$. If I satisfies the properties that (M1) if $f > f_{min}$, then $AND(f,f') < AND(f,f'')$ and $INFERENCE(f,f') < INFERENCE(f,f'')$; (M2) if $f < f_{max}$, then $OR(f,f') < OR(f,f'')$ - then we will say that I (or the corresponding operator) is <u>strictly monotonous</u>.

<u>Lemma 3.1.</u> (Equality preserving). Let the OR operator be complete wrt truth values and $f_0 > f_{min}$. If $OR(f, f_0) = f$ and $f' > f$, then $OR(f', f_0) = f'$.

As for the inference control, we will propose that the UM for each goal is evaluated to the degree it can be used for the selection of decisions. We consider the following possibilities for the selection of the decision.

<u>Definition 3.4.</u> In <u>threshold-driven goal selection,</u> a goal is selected if its UM is equal to or exceeds a given threshold value. In <u>maximum-driven goal selection</u>, a goal (or a set of goals) with maximum UM values are selected. In the last case, if all the UM receive the f_{min} values, then the result (the set of selected goals) is the empty set. If the empty set is accepted as a decision of E, then we will say that E <u>accepts empty results</u>. Two results G1 and G2 are considered significantly different if $G1 \ne G2$.

<u>Lemma 3.2.</u> Let $E = \langle R,F,G,I \rangle$ be an ES not accepting empty results, and $r \in R$. An effective test for r may exist only if $|G| > 1$.

Due to this lemma, in the following we will assume, that if E does not accept empty results, then $|G| > 1$. We also assume that if r is the only rule leading to g, then $UM(R \setminus \{r\}, g, t) = f_{min}$ for any t.

3.3. OTHER PRELIMINARIES

Before testing, it is reasonable to perform preliminary static analysis of the ES to eliminate loops, missing rules (there are no rules for some goal), useless rules (rules not leading to any goal, or rules that can not contribute to any goal whatever values their premises are given), subsuming rules (rule r1 subsumes rule r2, if they lead to the same goal, and r1 has weaker or equal premises and stronger or equal conclusions, with comparision to r2), redundant rules (rules that give the same result for all the tests), useless goals (goals that cannot be selected as a decision whatever values are given to their premises) [Suw84,Cra87].

Definition 3.5. If the static analysis is performed, we will say that the ES is statically correct.

Given an ES E and a rule r∈R, we will consider the following problems:

(P1) Does there exist an effective test for r?

(P2) Generate an effective test for r, if it exists.

(P3) Does there exist a contribution adequate test set for E?

(P4) Generate a contribution adequate test set for E.

Finally, we will assume that the problems are coded in a standard way in the sense of [Aho76].

4. TESTING SYSTEMS WITH VAGUENESS IN FACTS AND RULES

As an example of a fuzzy ES (FES), we will consider a system E=⟨R,F,G,I⟩, where the truth values are assumed to belong to a predetermined set of values [God88, Mar88] (for convenience, we will in this section also use the abbreviation UMD. In such systems the operations component of I comprises, as minimum, the truth tables for conjunction, modus ponens, and disjunction operators. The property of being complete wrt truth values is often not held in FES (e.g. due to linguistic approximation), therefore in general we do not assume it.

Proposition 4.1. For a FES E=⟨R,F,G,I⟩ and r∈R, the problems P1-P4 are NP-complete.

It is interesting to ask whether the static analysis of FES makes the test generation easier. In an arbitrary rule base, the number of rules may be exponential in the number of propositions |F|. Proposition 4.1 assumes standard coding of the problems, therefore the problem length is also exponential in |F|. If the possible number of different rules in a subsumption-free FES were polynomial in the number of different facts, then the test generation for the statically correct FES would be substantially simpler. The following proposition shows that this is not so.

Proposition 4.2. Let $E=\langle R, F\cup\{g\}, \{g\}, I\rangle$ be a statically correct FES such that $g \notin F$. For any given integer $k>0$ there exists an integer m such that if $|F|>m$ then the possible number of different rules in R exceeds $|F|^k$.

Therefore one must look for subclasses of FES, where effective test generation is feasible. If I is complete wrt truth values, the propositions 4.3-4.7 hold.

Proposition 4.3. Let $E=\langle R,G,F,I\rangle$ be a statically correct FES with threshold-driven goal selection and with separated premises, and rule r support a goal g. The problems P1 and P2 may be solved in time linear in $N1=|NODES(GRAPH(R,g))|+|F|$. The problems P3 and P4 may be solved in time linear in $N1*|R|$.

Proposition 4.4. Let $E= \langle R,F,G,I\rangle$ be a statically correct FES accepting empty results, using maximum-driven goal selection, with $|G|>1$, with strictly monotonous OR operator, having private rules, and with goals having separated premises. An effective test for a rule r exists always, and may be generated in time linear in $N=|NODES(GRAPH(R))|$.

Proposition 4.5. Let $E=\langle R,F,G,I\rangle$ be a statically correct FES with separated goals, with maximum-driven goal selection, and with $|G|>1$. Then there exists a contribution adequate test set for E.

Proposition 4.6. Let $E=\langle R,F,G,I\rangle$ be a statically correct FES using threshold-driven goal selection, and including at most one rule leading to each $p \in F$. Then t_{max} is an effective test for $r \in R$.

Proposition 4.7. Let $E=\langle R,F,G,I\rangle$ be a statically correct FES with maximum-driven goal selection, such that there are no goals supporting other goals. Let $r \in R$ and R include $|G|$ rules - exactly one rule for each goal. Let (P) be the property that for each rule $r_0 \in R$, $r_0 \neq r$, it is true that $PREMISES(r_0) \setminus PREMISES(r) \neq \emptyset$, or that $UM(r_0) \leq UM(r)$. If the inference operator of E is strictly monotonous, then an effective test for r exists, if and only if (P) holds. Recognition and generation of the effective test may in this case be performed in time linear in N. If the inference operator of E is not strictly monotonous, then the property (P) is not necessary.

If I is not complete wrt truth values, then Propositions 4.6 and 4.7 remain true, but 4.3-4.5 do not hold. The propositions 4.8 and 4.9 hold for this case.

Proposition 4.8. Let E=<R,F,G,I> be a statically correct FES with maximum-driven goal selection, accepting empty results, and with separated premises. Let $r \in R$ support a goal g. An effective test for r may be generated in time linear in |NODES(C(R,g,r))|+|F|.

Proposition 4.9. Let E=<R,F,G,I> be a statically correct FES with separated premises. Then it is possible to establish whether an effective test for a rule r exists and, if it exists, to generate it in time linear in $N' = |NODES(GRAPH(R))| * |V|^2$.

ACKNOWLEDMENT

The author would like to thank the Basic Laboratory of AI of the Institute of Technical Cybernetics, Slovak Academy of Sciences, for support and pleasant working environment.

REFERENCES

Aho76 Aho, A.V., Hopcroft, J.E., Ullman J. E. The Design and Analysis of Computer Algorithms. Addison-Wesley, London, 1976.

Buc86 Buchanan, B. G. Expert Systems: Working Systems and the Research Literature. Expert Systems, 3, 1986, No.1, pp.32-51.

Cra87 Cragun, B. J., Stendel, H. J. A Decision-Table-Based Processor for Checking Completeness and Consistency in Rule-Based Expert Systems. Int. J. Man-Mach Stud., 26, 1987, No. 5, pp.633-648.

EWI86 EWICS TC7 / Techniques for Verification and Validation of Safety- Related Software. Computers & Standards, 4, 1986, No.2, pp. 101-112.

God88 Godo, L.L., Lopez de Mantaras, R., Sierra, C., Verdaguer, A. Managing Linguistically Expressed Uncertainty in MILORD Application to Medical Diagnosis. AI Communications, Vol. 1, No. 1, 1988, pp. 14-31.

Haj84 Hajek, P. A Theory of Propagation of Uncertain Knowledge in Consulting Systems. In New Enchancements in GUHA Software (P. Hajek, ed.), 37-47, 1984.

Inc87 Ince, D.G. The Automatic Generation of Test Data. The Computer J., 30, 1987, No.1, pp. 63-69.

Kee87 O'Keefe, R.M. , Balci, O. , Smith, E.P. Validating Expert

System Performance. IEEE Expert, 2, 1987, No.4, pp. 81-90.
Leu88 Leung, K.S., Lam, W. Fuzzy Concepts in Expert Systems. Computer, 21, No.9, September 1988, pp. 43-56.
Mar88 Martin-Clouaire, R. Approximate Reasoning in the Possibility Theory Framework: Where are We Practically? Bulletin for Studies and Exchanges on Fuzziness and its Applications, 34, 180-185, 1988.
Mye79 Myers G.J. The Art of Software Testing. New York, Wiley, 1979.
Ram87 Ramamoorthy, C.V. - Shekhar, S. - Garg, V.: Software Development Support for AI Programs. Computer, 20, 1987, No.1, pp. 30-37.
Suw84 Suwa, M., Scott, A. C.,Shortliffe, E. H. Completeness and Consistency in a Rule-Based System. In: Rule-Based Expert Systems/Buchanan, B.G. and Shortliffe, E. H., Eds. London, Addison-Wesley, 1984, pp. 159-170.
Tep88 Tepandi, J. A Knowledge-Based Approach to the Specification-Based Program Testing. Computers and Artificial Intelligence, 7, 1988, No.1, pp. 39-48.
Tep89 Tepandi, J. The Generation of Test Data for Rule-Based Expert systems. - In: Proc. of Int. Symp. Algorithms '89, April 17-21, 1989, Vysoke Tatry. - Bratislava, 1989, pp. 310-312.
Tri85 Trillas,E., Valverde, L. On mode and implication in approximate reasoning, In: Expert Systems, M.M.Gupta et al. (eds.), North-Holland, 1985.