Bull. ERS, No.11 (1977)

BRIEF NOTE

ON LOGICAL TREATMENT AND EVALUATION OF REGULATORY STATEMENTS

by

Taijiro TSUTSUMI[I)] and Heki SHIBATA[II)]

Abstract

This paper is dealing with logical treatment and evaluation of regulatory statements, for example, regulatory codes, regulatory guide so on. Based on an idea of "regulatory tree" as the dual form of fault tree, we can examine a regulatory statement system on its inconsistency, duplication, lack of statement, redundancy and so on by using "resolution principle". And we can also evaluate them with some indexes similar to those of the communication theory. Through such investigation the authors felt the necessity of introducing "fuzzy technique", however, they are still on the way.

1. Introduction

This paper is dealing with the problem of logical treatment and evaluation of regulatory statements such as regulatory code, regulatory guide and so on. At first the authors try to establish the "ideal regulatory tree". This tree has a some relation to the fault tree of the system for which we want to establish the regulatory statements. This tree also has final concrete statement boxes in the lowest level. However, we usually do not describe the whole boxes in the regulatory statements, and cut in a certain level of the tree. If the regulatory statement system includes the whole contents of an ideal regulatory tree, we call this RSS as a closed form code. On the other hand, if the RSS does not include the complete contents, then we call them as an open form code.

Statements can be analyzed logically by using LISP Program. Based on ideal regulatory tree, we can evaluate "concreteness" of a regulatory statement, or degree how its form is "open". Sometimes in a particular regulatory statement system, levels of concreteness or severity are different in each statement. We need to check their balance statement by statement. And also we can check their inconsistency, duplication, lack of statement and redundancy.

In an engineering sense, the mathematical result like a figure of reliability of the system seems to express the degree of

---

I) Graduate Student

II) Professor of the Institute of Industrial Science, University of Tokyo

its safety without any argument. However, the "safety" is the concept which is defined in the community, and not by engineers. Sometimes the governmental authority would be a representative of the community. So, the concept of "safety" is subjective one, whereas the figure of reliability is objective one. To say "this system is safe", the community should make their own judgement. The judgement is sometimes not logical, and quite subjective, that is fuzzy. How to control this is one of the problems.

2. Ideal Regulatory Tree

By putting the purpose of a regulation, the proposition, in a top box, we can figure out a regulatory tree as a dual form of the fault tree which has a top event which the regulation wants to control. For example, let us assume that the purpose of a regulation is a proposition "Earthquake-resistant strength of piping is adequate from an aseismic design point of view". In this case, the fault tree is shown in Fig. 1 and the dual tree, that is the ideal regulatory tree, is shown in Fig. 2. If we can describe every events related to the top event, unfavourable event for the safety as a tree, then the dual tree of this tree is ideal one as the corresponding regulatory tree. And by describing this ideal regulatory tree to regulatory statements, we can obtain an ideal regulatory code.

Boxes in the lowest level should describe the each regulation, led from the top event through AND gates and OR gates, in the most concrete form. For example, in Fig. 2 the boxes (a) and (b) are those in the lowest level.

To check existent regulatory codes, this ideal regulatory tree should be standard. By making logical treatment on both, we can evaluate the existent regulatory code as the authors will describe later.

3. Logical Treatment of Regulatory Statement by Computer

It is rather easy problem, if we can convert statements to a standard form in a certain language system like LISP 1.5. Here, the authors employed the form of the first-order logic. Some standard predicates in their case are as follows:

i) Noun to express "thing" or "position".
exp) PIPING (x) : x is a piping.

ii) Noun to describe the relation X to Y.
exp) MAIN-STRUCTURE ($x \mid y$) : x is a main structure of y.

iii) Verb.
exp) SET ($x \mid y$, z , ...) : x sets y at z ... .

iv) Noun to describe "attribute".
exp) LENGTH (x | 10, m, GT) : The length of x is greater than 10 meters.

We combine such predicates and logical symbols under quantifiers, then the statement can be transformed to a computer code. As the inference rule, the authors used "resolution principle". In this stage we can introduce "fuzzy logic" in it.

4. Evaluation of Regulatory Statement --- Logical Treatment

Logical treatment on coded statement is done by inference algorithm. There were several techniques for such inference rules. Robinson developed "resolution principle"[1] in 1965, and this method has a system of very simple algorithm and is applicable to an ordinary computer. The authors employed this algorithm for the study. We treat only well-formed formulas in the first-order logic. At first the well-formed formulas should be transformed into clause form. When an unsatisfiable set of clauses is given, the resolution principle always generates the empty clause, NIL, from the set of clauses. NIL means "inconsistency" of the original statements. This deduction process makes a kind of tree, it is called "deduction tree". To explain the resolution principle, let us consider a simple example. Consider the following set of propositions:

i) Material of piping is cast iron.
: $(\forall x)$ [PIPING(x) $\rightarrow$ MATERIAL(x | *cast iron*)]

ii) Material of piping is not cast iron.
: $(\forall x)$ [PIPING(x) $\rightarrow \sim$ MATERIAL(x | *cast iron*)]

iii) There is at least one pipe.
: $(\exists x)$ [PIPING(x)].

The clause form of the each proposition is as follows:

iv) $\sim$ PIPING(x) $\vee$ MATERIAL(x | *cast iron*)

v) $\sim$ PIPING(x) $\vee$ $\sim$ MATERIAL(x | *cast iron*)

vi) PIPING(a)

In this case "a" is a constant. We obtain the deduction tree shown in Fig. 3 by the resolution principle. We can apply "fuzzy logic" in this process, as the authors will mention later.

Through this logical treatment, we can test the characteristics of statements such as inconsistency, duplication, redundancy and so on. However there is some limitation. If the statement describe a some possibility like "In a case that it has a possibility of subjecting to ---, ---", we should treat such a conditional statement as "proposition". They can mention other cases in addition to "possibility".

Usually such regulatory statements are supported by very deep engineering knowledges. Although a closed form code tries to describe all including such knowledges, there is limitation. When we figure out the regulatory tree, sometimes we wonder whether a statement should be in a box of the tree or a supporting knowledge outside of the tree. Such knowledges can be defined in several ways.

5. Closed Form Code and Open Form Code

The authors described on "closed form code" in the last part of the previous chapter. Closed form code should contain all statements which are necessary to apply the code to an actual system. There is no allowance for engineering design in this regard. On the other hand, "open form code" is describing 1) general principle and applied theoretical basis, 2) limitation of application and basis of judgement at branching points in the design procedure, 3) actual values of parameters for the design, 4) some design criteria and limitation. And all other actual procedures are remained to be free for the designer. Through such open form regulation, the designer has a possibility of engineering development compare to the design under a close form regulation. However, it has a possibility to give the designer a chance that some of them try to escape from the real purpose of the regulation.

6. Evaluation of Regulatory Statements --- Quantitative Approach

As the authors described, RT can be made from the fault tree which has a top event reduced from the event we want to control. RT may have a quantitative nature as well as the corresponding fault tree. Therefore we may categolize the factor of importance of each box in this regard.

Additionally to this we can also evaluate regulatory statements quantitatively in some point of view. For example, the degree of "open" can be evaluate by the following procedure: we make the regulatory tree including boxes for supporting engineering knowledges, and evaluate the degree of "abstruction" of each box in "bit". If we compare the regulatory statements with the regulatory tree, then we can draw the boundary line cutting the tree, and evaluate the degree of "open", or "concreteness" of the regulatory statements. In some sense, the evaluation of degree of abstruction is a matter of subject of applicants. However, the authors consider that it can be expressed by evaluating information quantity of each statement in a box.

Required cost, that is, "severity" is another performance of the code. The procedure of evaluation is almost similar to that of "concreteness", however, the way of evaluating on each statement has not established well.

7. Application of Fuzzy Set Theory

As already mentioned, sometimes the statement itself is not deterministic, and it expresses a fuzzy proposition. In such a case, we need fuzzy algorithm[2]. The reason why the fuzzy proposition is introduced into the RS is considered as follows: Most of statements which describe engineering procedures and facts are deterministic. However, in the stages of judgement, they become fuzzy. Some numerical results obtained through engineering procedures are objective, contrary judgements on "safety related affair" is quite subjective. Between objective quantity and subjective statement, we need fuzzyness.

The authors understand that the fuzzyness is a problem of distribution of individual judgements in a community and the membership function $\mu(x)$ is a stochastic parameter of the distributions as shown in Fig. 4[3]. If there is a strong but minor opponent group, some portion of distributions make bi-peak form. Although in such a case, we could give a certain algorithm to treat it, it becomes no real solution.

8. Acknowledgement

Some part of this short article is a part of Doctor thesis[4] of Mr. Tsutsumi. The authors like to express their great gratitudes to Dr. Etoh, Hara and Fujita for their kind discussions.

9. References

(1) Robinson, J.A., "A Machine Oriented Logic Based on the Resolution Principle", *J. ACM*, 12, 1 (1965) 23.

(2) Zadeh, L.A., "Communication Fuzzy Algorithms", *Information and Control*, 12, (1968) 94.

(3) Shibata, H., "A Comment of Fuzzy Set", *Summary of Papers on General Fuzzy Problems* (WG on Fuzzy Systems), 2, (1976) 101.

(4) Tsutsumi, T., "Systematic Analysis and Computer Logical Treatment of Engineering Regulatory Statements", Doctor Thesis of University of Tokyo (1977), (under examination).

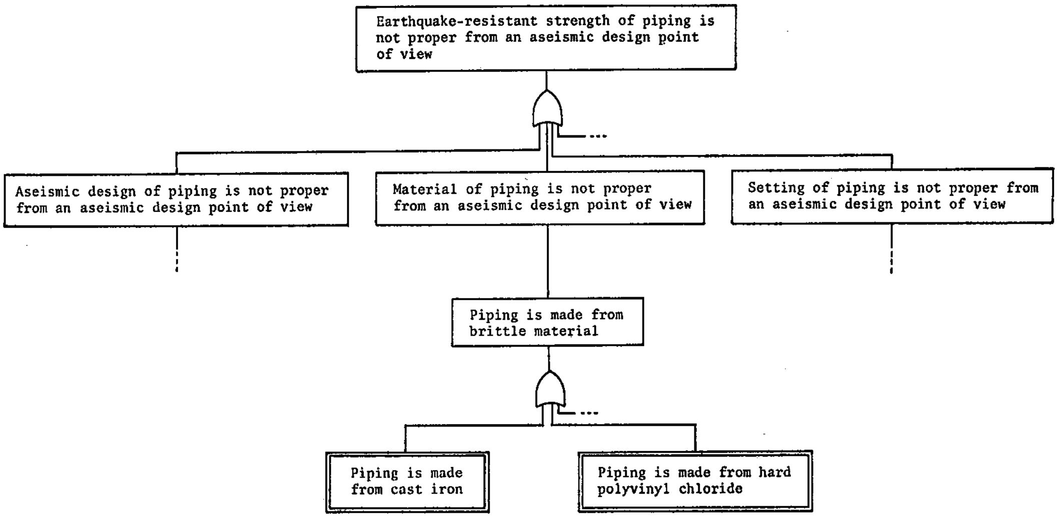

Fig.1 Fault Tree

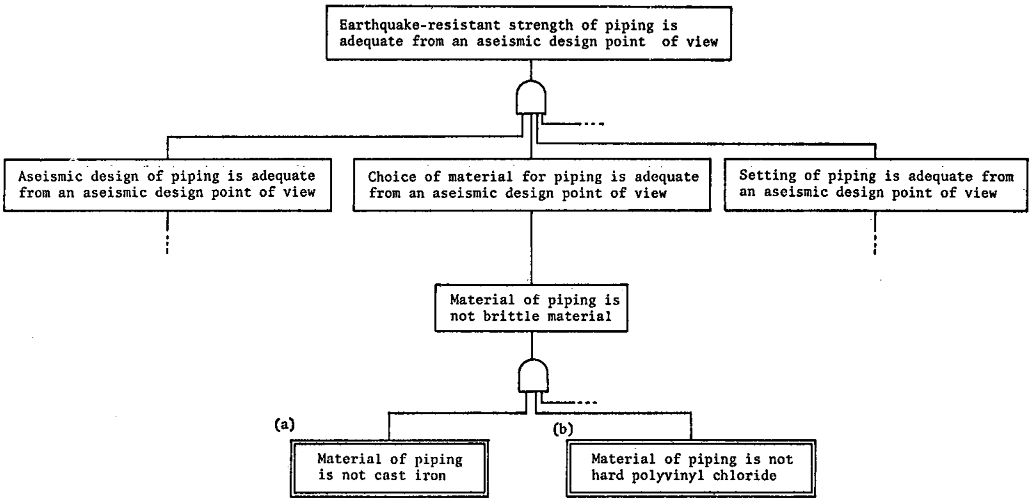

Fig.2 Ideal Regulatory Tree

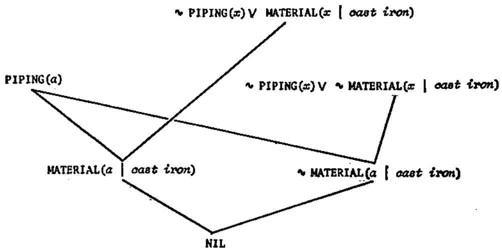

Fig.3 Deduction Tree

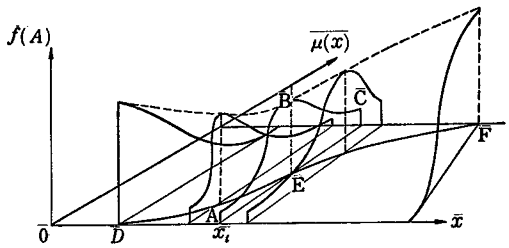

Fig.4 Schematic Relation of Member-ship Function and their Distribution to Variable