ISSN 0225-0667

DÉPARTEMENT
D'INFORMATIQUE

FACULTÉ DES SCIENCES
ET DE GÉNIE

UNIVERSITÉ LAVAL
QUÉBEC

WORKSHOP NOTES:

International Workshop On Models and Languages For Software Specification And Design

Edited by
Robert G. BABB II
Ali MILI

DIUL-RR-8408

WORKSHOP NOTES

INTERNATIONAL WORKSHOP ON MODELS AND LANGUAGES FOR SOFTWARE SPECIFICATION AND DESIGN

March 30 1984

Orlando, Florida

EDITED BY

Robert G. Babb II

Oregon Graduate Center

Ali Mili

Université Laval

Sponsored by: IEEE COMPUTER SOCIETY LCRST OF JAPAN

UNIVERSITÉ LAVAL

ogc OREGON GRADUATE CENTER

In cooperation with: acm ACM SIGSOFT

TOWARDS A GENERAL THEORY OF PROBLEMS

Paulo A. S. Veloso

This paper discusses some aspects, including motivation and typical results, of a theory of problems that is mathematically rigorous and has wide applicability. Starting from a critical analysis of familiar, but vague, notions a problem is defined as a mathematical structure, its solutions being certain functions. This provides a framework for the precise study of relations between problems, such as analogy, as well as methods for problem transformation, as reduction, and problem-solving methods, as decomposition.

The study of problems has a long history, involving names such as Descartes, Leibnitz, Bolzano, etc. More recently the field of Heuristics has received contributions from mathematicians as Polya [3] and researchers in Artificial Intelligence [2]. These approaches, however, tend to view a problem from a how-to-solve-it viewpoint. Our aim here is more akin to that of metamathematics with its formalization and critical study of the notions of proof, theorem, etc. employed by mathematicians. Thus, we address to the question "what is a solution of a problem" rather than how to obtain it.

In approaching a problem, Polya [3] suggests asking the following questions "what are the data?" ; "what are the possible results?", "what constitutes a satisfactory solution?". In fact, a problem is not precisely formulated until we have unambiguous asnwers to these three questions, as some reflection and simple examples will show. This gives the basis of our proposal for a precise definition of problem. Namely, a problem is defined as a mathematical structure $Q = \langle D, R, q \rangle$, where D and R are nonempty sets (the domains of data and results, respectively) and q is a binary relation from D to R (the problem condition). A solution for Q is a function a from D to R such that for every $d \epsilon D$ $(d, a(d)) \in q$. We call Q solvable iff it has a solution.

This formulation is quite flexible. It includes not only specific problems (such as finding a root of a given polynomial) but also general problems (as finding a root for each polynomial with, say, real coefficients). This formulation was influenced by the concept of "decision problem", but it also includes both "problems to find" and "problems to prove" [3]. In fact, the formulation is general enough so as to encompass problems from a variety of walks of life, such as proving mathematical theorems, geometrical constructions, programming problems, puzzles, classifying biological specimens, selecting flight itineraries, assemblying automobiles from their parts, etc. [1, 5].

An advantage of a formulation as the one above is the immediate availability of powerful mathematical machinery. For instance, subproblem is defined as substructure; likewise the definition of homomorphism together with associated concepts and results carry over. In addition to these mathematical relationships among structures, others stem from their being problems, being closer to a human observer. Such is the case of the notion of analogy, for wich a possible formulation within our framework is as follows. An analogy from a problem $Q = \langle D,R,q \rangle$ to a problem $Q' = \langle D', R' q' \rangle$ consists of two functions $\alpha : D \rightarrow D'$ (linking the data domains) and $\beta : q \rightarrow q'$ (connecting the problem conditions). A typical result about analogies is the following criterion of solvability transfer.

Proposition. Let (α , β) be an analogy from Q to Q' with a surjective α. Then if Q is solvable so is Q'.

The idea of reduction is widely know, variations thereof being called generalization, specialization, etc. (Some precise, albeit specialized, formulations of reduction occur in recursive function theory). Probably the best know examples of reduction are those of geometric problems to algebraic ones achieved by means of Cartesian coordinates. The basic goal in reducing a problem to another one is obtainig a solution of the former from one for the latter. Within our framework this intuitive idea can be formalized as follows. A reduction of a problem $\mathcal{Q}$ to a problem $\mathcal{Q}'$ consists of two functions $\tau: D \to D'$ (the translation map) and $\rho: R' \to R$ (the retrieval map) - such that for any solution $a': D' \to R'$ of $\mathcal{Q}'$ the composite function $\rho . a' . \tau$ is a solution of $\mathcal{Q}$. The following result gives some indication on when a pair of functions consistitutes a reduction.

Proposition. Let $\mathcal{Q}$ and $\mathcal{Q}'$ be problems and $\tau: D \to D'$ and $\rho: R' \to R$. Then (τ,ρ) is a reduction of $\mathcal{Q}$ to $\mathcal{Q}'$ if for all $d \in D$ and $r' \in R'$ $(d,\rho(r')) \in q$ whenever $(\tau(d), r') \in q'$. [4]

It is also possible to indicate when a reduction exists, by means of the concepts of similarily and subsumption. Consider problems $\mathcal{Q}$ and $\mathcal{Q}'$. We say that $(d',r') \in D' \times R'$ is similar to $(d,r) \in D \times R$ iff either $(d',r') \notin q'$ or $(d\ ,r\ ) \in q$. Now, we say that a problem instance $d \in D$ subsumes $d' \in D'$ iff given any $r' \in R'$ there exists $r \in R$ such that (d',r') is similar to (d,r).

Theorem. A sufficient condition for the existence of a reduction of $\mathcal{Q}$ to $\mathcal{Q}'$ is the existence for every problem instance $d \in D$ of a problem instance $d' \in D'$ subsumed by d. [6]

Reduction, as formulated above, can be regarded as a method of problem transformation. On the other hand, a quite common method od problem solving is decomposition. Its underlying idea consists of repeatedly splitting problem instances into smaller problem instances until they are simple enough to permit direct determination of matching results, which are then combined into a result for the original problem instance. These can be precisely formulated as follows. An n-ary decomposition for a problem $\mathcal{Q} = \langle D, R, q \rangle$ consists of n unary functions $\sigma_i: D \to D$, $i = 1,\ldots,n$ (the splitting maps), an n-ary function $\mu: R^n \to R$ (the recombining map), a unary function $\nu: D \to R$ (the direct matching) and a unary relation $\psi \subseteq D$ (the simplicity test) such that the assigment.

$$a(d) = \begin{cases} \nu(d) & \text{if } d \in \psi \\ \mu(a(\sigma_1(d)),\ldots, a(\sigma_n(d))) & \text{otherwise} \end{cases}$$

defines a solution for $\mathcal{Q}$. This formulation views the direct matching ν as a "partial solution" in that it solves $\mathcal{Q}$ only for those data passing the simplicity test. Decomposition extends this partial solution to a total one by means of repeated splittings and recombinations. In this process a funda - mental role is played by a notion of relative simplicity among problem instances. By a comparability relation for $\mathcal{Q}$ we will mean a well-founded binary relation on D.

Proposition. Functions $\sigma_1,\ldots,\sigma_n$, μ, ν and a relation ψ as above constitute a decomposition for $\mathcal{Q}$ if there exists a comparability relation $<$ for $\mathcal{Q}$ such that (i) for all $d \in \psi$, $(d, \nu(d)) \in q$; (ii) for all $d \in (D-\psi)$, $\sigma_1(d) < d,\ldots,\sigma_n(d) < d$ and whenever $(\sigma_1(d), r_1) \in q,\ldots,(\sigma_n(d), r_n) \in q$ then also $(d, \mu(r_1,\ldots,r_n)) \in q$ [4]

In order to see when a decomposition exists let us say that $d_1,\ldots,d_n \in D$ cover $d \in D$ iff given $r_1,\ldots,r_n \in R$ such that $(d_1,r_1) \in q,\ldots, (d_n,r_n) \in q$ there exists $r \in R$ such that $(d,r) \in q$ [6]

Theorem. Assume the existence of a comparability relation $<$ for $\mathcal{Q}$ such that (i) whenever $d \in D$ is minimal then there exists $r \in R$ so that $(d,r) \in q$, (ii) whenever $d \in D$ is not minimal then there exist $d_1 < d,\ldots,d_n < d$ s that $d_1,\ldots,d_n$ cover d. Then $\mathcal{Q}$ has decomposition [6].

In the above we have been dealing with a class of problems that might be called unconstrained problems, in that there is no a priori restriction on what kind of funtions are acceptable as solutions. In contrast geometric problems often call for ruler-and-compass constructions, which eliminates from the outset certain functions as candidate solutions. In order to capture this in a natural way we say that a <u>constrained problem</u> $\hat{Q}$ consists of the previous three ingredients plus a set A of functions from D to R (the set of admissible functions). Accordingly, a <u>solution</u> for $\hat{Q}$ is an admissible function $a \epsilon A$ solving the underlying unconstrained problem. This definition of solution for a constrained problem is a relativization of the previous one (for unconstrained problems). Similary, the concepts of analogy, reduction and decomposition are extended to constrained problems by suitable relativizations.

A further generalization may be useful in some contexts; typically, those involving evaluation of potencial solutions with respect to non-extensional properties, such as clarity, elegance or efficiency. (Think of proofs, constructions or programs, respectively.) Then, a solution should not be simply an assignment of results to data. Rather, we want to view as solution the very description of this assigment. One way to formulate this extension is by adding to the previous three ingredients D,R and q a domain P of function descriptions together with an evaluation map $\lambda: P \times D \to R$ to obtain a generalized problem $\bar{Q}$. Now, a <u>solution</u> for $\bar{Q}$ is a function description $p \epsilon P$ such that for all $d \epsilon D$ $(d, \lambda(p;d)) \epsilon q$.

We have covered only a small part of the theory already developed, but this should give a good idea of its character. An important aspect is under continuous development, namely the study of real problems from diverse disciplines aiming at a better feeling for the adequacies and limitations of our formulations and guiding future directions [7]

<u>References</u>

1. M.A. Lopes. Introduction to a General Theory of Problems (in Portuguese). D. Sc. diss., forthcoming

2. N. J. Nilsson. Problem Solving Methods in Artificial Intelligence. McGraw-Hill, 1971.

3. G. Polya. How to Solve it: a new aspect of the mathematical method. Princeton Univ. Press, 1957.

4. P. A. S. Veloso. Divide-and-conquer via data types. VII Latin-American Conf. Informatics, 1980.

5. P. A. S. Veloso, M.A. Lopes. A Framework for Problem Solving: theory and methodology. Internat. Conf. Appl. Systems Res. and Cybernetics, Acapulco, 1980.

6. P.A.S. Veloso, S. R. M. Veloso. Problem Decomposition: applicability, soundness, completeness, 5th. European Meeting on Cybernetics and Systems Res., Vienna, 1980.

7. P. A. S. Veloso. Outlines of a mathematical theory of problems. Res. Rept., PUC/RJ, 1983.

Paulo A. S. Veloso

Deptº de Informática

Pontifícia Universidade Católica

22453 - Rio de Janeiro - RJ