ADVANCES IN ARTIFICIAL INTELLIGENCE

edited by
TIM O'SHEA

North-Holland

Advances in Artificial Intelligence, T. O'Shea (Ed.)
Elsevier Science Publishers B.V. (North-Holland)

THE UBIQUITOUS DIALECTIC

Edwina L. Rissland

Department of Computer and Information Science
University of Massachusetts
Amherst, MA 01003

In this paper, we discuss the central role played by examples in reasoning in various fields including mathematics, law, linguistics, and computer science. In particular, we consider the dialectic between proposing a concept, conjecture, or proposition and testing and refining it with examples. We provide several examples of this ubiquitous process.

1. INTRODUCTION

Examples are an important component of an expert's knowledge of his field. In mathematics, examples can be said to be as important to understanding as the traditionally exalted definitions, theorems, and proofs (Rissland 1978). In fact, some mathematical areas developed in response to troublesome counter-examples like modern real function theory which has been called "the branch of mathematics which deals with counter-examples" (Munroe 1953).

In the law, examples – that is, legal cases – are the basis from which the law derives (at least in common law systems like those of the United States and England): the law is made through court decisions by consideration of specific problems in specific cases. The cases lead to rule-like decisions which are then refined (or perhaps refuted, i.e., overturned) in subsequent cases.

In linguistics, for instance in the study of syntax, linguistic rules are derived from study of examples of actual language and are then subjected to testing on more examples. Some examples are taken from the infinite store of run of the mill sentences available to every natural speaker; others, like certain difficult garden path sentences, are fabricated and used as counter-examples are in mathematics.

In computer science, examples – for instance, test data and already existing code – are used at the heart of the programming effort. There is an "inevitable intertwining" between the specification, attempted implementation, and the test data of a program (Swartout and Balzer 1982).

Similar observations were made by Kuhn about theory evolution in scientific disciplines like physics and astronomy (Kuhn 1970, 1974). In all these disciplines, the theories and concepts would not be so finely tuned nor so convincing without the use of examples. The examples are often used in a dialectic between a proposed idea, proposition, or theory and the examples used to test it out:

> *An idea is proposed. It is tried out on examples. Some examples motivate tightening or refining it; others necessitate radical redefinition or abandonment. This leads to a new idea and the process repeats itself.*

In the rest of this paper, we give examples of this ubiquitous dialectic and point out some of the similarities and differences in it as it occurs in different fields.

This research was supported in part by grants IST-8017343 and IST-8212238 of the National Science Foundation.

2. MATHEMATICS: CONJECTURE and COUNTER-EXAMPLE

The opening chapter of Lakatos' wonderful book *Proofs and Refutations*, provides a quintessential example of the dialectic. The subject is Euler's formula relating the number of vertices (V), edges (E), and faces (F) of certain polyhedra in the equation V - E + F = 2. The discussion - placed in a hypothetical classroom among students and a teacher - opens with the primitive conjecture:

CONJ-1: For all polyhedra, V - E + F = 2.

After discussion of a proposed proof of CONJ-1, which includes presentation of a "local" (i.e., to the proof) counter-example and a patch of the proof, student Alpha offers the first "global" (i.e., to the conjecture) counter-example, the so-called "hollow cube", for which V - E + F = 4:

CEG-1 : A pair of cubes, one of which is inside, but does not touch the other.

In response, student Delta shifts the burden of the problem onto the definition of polyhedron, "It is a fake criticism. This pair of nested cubes is not a polyhedron at all." Gamma, another student, tries to save CONJ-1 by offering the following definition of polyhedron:

DEF-1: A polyhedron is a solid whose surface consists of polygonal faces.

Delta offers the alternative definition:

DEF-2: A polyhedron is a surface consisting of a system of polygons.

Alpha then states triumphantly, "So! My counter-example has bred a new concept of polyhedron." For the moment, DEF-2 becomes the accepted definition. (We can note that new problems may have been introduced with undefined terms like "surface" and "system" and familiar, but yet undefined terms like "polygon".) Regardless, Alpha gives two new counter-examples meeting the new definition, but violating the conjecture since for both V - E + F = 3:

CEG-1a: Two tetrahedra which have an edge in common.

CEG-2b: Two tetradedra which have a vertex in common.

Delta then parries by modifying DEF-2 by adding two technical conditions which exclude CEG-2a and CEG-2b to arrive at the third definition:

DEF-3: "A polyhedron is a system of polygons arranged in such a way that (1) exactly two polygons meet at every edge and (2) it is possible to get from the inside of any polygon to the inside of any other..."

Alpha caustically then says, "Why don't you just define a polyhedron as a system of polygons for which the equation V - E + F = 2 holds?" The class calls this the "perfect definition" or "Def.P". The teacher interposes with the comment "definitions are frequently proposed and argued about when counter-examples emerge." He then asks that the discussion continue under DEF-3 to which Gamma quickly offers the "urchin" or "small stellated dodecahedron as a counter-example since for it V - E + F = -6. Delta next offers a fourth definition which is a further refinement of DEF-3 and Alpha then counters with the "picture frame" counter-example for which V - E + F = 0.

And so continues what Lakatos calls the "fundamental dialectic of proofs and refutations".

3. THE LAW: DOCTRINES and CASES

In the law, the dialectic is between evolving doctrines and the litigated cases to which the doctrine is applied. This cycle is often simulated in classroom discussions in law school with the proposing of a doctrinal proposition and the probing of it with hypothetical cases ("hypos") (Rissland 1982, 1983, 1984). It is also central to developing arguments in preparation for litigation (Rissland, Valcarce & Ashley 1984).

In common law systems, courts express decisions – which superficially look like rules but, in fact, are more like heuristics in the sense that they do not always obtain – as "holdings" which are arrived at through *stare decisis*, the doctrine of precedent. Applying a holding to a new case – primarily its "fact situation" – usually provokes revision or, at least, further interpretation. Revisions often involve narrowing the applicability (or to use the language of logic, "antecedent" conditions) or the outcome ("consequences") of the rule, for instance by redefinition of the ingredient concepts. Revision is strikingly similar to that seen in mathematics: outright "monster barring", "dividing and conquering" by distinguishing subclasses, addition of technical conditions, etc.

The following is a record of an actual discussion from a first year torts class on the subject of "foreseeability", "liability", and "negligence". It arose from consideration of the English case of *Stone v. Bolton* (Gregory, Kalven & Epstein 1977), in which the plaintiff, Miss Bessie Stone, was struck on the head by a cricket ball hit out of the cricket grounds as she walked on the street outside of them. It was said to be one of the longest balls ever hit and that no more than six to ten balls had been hit outside in thirty years. The grounds were surrounded by a high wall.

The initial doctrinal proposition before the class was:

PROP-1: Foreseeable harm implies liability.

This first proposition was then tested on the following hypos:

HYPO-1: Defendant builds a dam which caves in and injures plaintiff.

HYPO-2: Defendant has an attack of epilepsy while driving a car and injures plaintiff.

HYPO-2a: Defendant in HYPO-2 has never before had an attack of epilepsy.

HYPO-2b: Defendant in HYPO-2 had one attack of epilepsy fifteen years ago.

HYPO-2c: Defendant in HYPO-2 has had several attacks of epilepsy.

After a few more hypos, the class considered the refined proposition:

PROP-2: If this is the first time, then one is not liable.

Discussion continued with hypos, derived from the original case through modifications:

HYPO-5: The cricket club only opened last year.

HYPO-6: It would cost the entire Gross National Product to build a "high" fence.

HYPO-6a: Defendants in HYPO-6 choose not to build a "high" fence.

HYPO-6b: HYPO-6 + after one ball is hit outside the grounds, defendants raise the fence height by 3 meters. A year later, Bessie Stone is hit.

HYPO-6c: The ball in HYPO-6b that hit Ms. Stone was hit by the Babe Ruth of cricket.

This example exhibits many features of the dialectic: the putting forth of a proposition, the testing of it with one or more examples, the refining of it in response to experience with the examples. It has the added features that the examples come in sequences which are generated by successive modifications and that the sequences may arise from a real initial case (Rissland 1983).

This is in slight contrast to the mathematics example where only one or two examples are offered in response to a proposition or definition. This might be because of the difference in the standards of "truth" in the two fields: mathematical truth is basically absolute and binary; legal truth is subjective and relative. There are no QEDs in law, only interpretation, argument, and more argument. Thus, the need for strings of convincing or undermining examples.

4. LINGUISTICS: RULES and SENTENCES

In linguistics, rules of syntax are often motivated and then refined by consideration of rules and example sentences. As in law, one has an unlimited number of examples readily available from one's personal experience; this is different from mathematics where one must generate examples, often with difficulty. This familiarity is often used by teachers in socratic dialogues.

The following is typical of interchanges that occur in introductory courses. The teacher asks the class to formulate a rule for turning an English sentence into a yes-no question. The students respond with an example pair of sentences and a rule:

SENTS-1: John will go to class.
Will John go to class?
RULE-1: Interchange the subject and the first verb.

This rule is then tested with another example sentence pair and the rule is refined:

SENTS-2: John hasn't been here.
Hasn't John been here?
RULE-2: RULE-1 + "bring any negative along with the verb".

The teacher points out with an example that RULE-2 can yield sentences different from what is desired and the class revises the rule again:

SENTS-3: John has not been here.
Has not John been here?
Has John not been here?
RULE-3: RULE-2 except "only a contracted negative stays with the verb".

The discussion continues with consideration of main versus auxilliary verbs and RULE-3:

SENTS-4: John ran to the store.
*Ran John to the store?
Did John run to the store?

Surprised, the students ask, "Where did the <u>did</u> come from?"

Note that while the teacher does not have a monopoly on the examples, the expert does have a more thorough and more structured knowledge of them. In general, expert knowledge is more structured than the novice's. The point for this discussion is that examples are critical and indispensible elements in the dialectic.

5. COMPUTER SCIENCE: PROGRAMS and DATA

These same observations apply to the writing of computer programs. One starts out with a problem or description of what the program is supposed to do, codes up a first version, tests it out, and then debugs, i.e., revises, the code.

The dialectic also occurs on the level of the program specification since it is nearly impossible to write a complete specification (DeMillo, Lipton & Perlis 1979). Some argue it is, in fact, wrongheaded to do so (Selfridge 1983). This leads one to seek alternative ways to describe what one wants in a program: using examples to illustrate what is wanted in certain cases is one way of doing this. Formal specification and illustrative examples are complementary descriptions which compensate for each others weaknesses.

One can see this dynamic intertwining of specification, program, and example data particularly well in the context of interactive (interpretive) programming environments, like those for LISP. Instead of reproducing an example record here, the author suggests that the reader try programming a little problem and keeping track of the data used to debug the program, the versions of the program, and the versions of the specifications.

In looking at expert programmers, one will see not only heavy use of test data, but also use of different classes of examples. For instance, any LISP hacker knows to test a program on NIL, the empty list since NIL is a standard trouble-maker, or counter-example, in the LISP domain; novices must be taught this (e.g., see Friedman's *Little LISPer*). Experts know to try a variety of testing examples that span the possibilities: from simple "start-up" cases like (A), to standard "reference" cases like (A B C), to more complex ones like ((A B) ((C))). The more complex ones are often generated by modifications of simpler ones with respect to structural features like list length and depth of atoms (Rissland and Soloway 1980).

6. CONCLUSIONS

In this paper, we have discussed the dialectic of proposing a concept, conjecture or proposition, testing it with examples, and then revising it in light of the experience with the examples. We have given examples from several diverse disciplines which we feel indicate its ubiquity.

In this ubiquitous dialectic, examples play a critical role. They are essential to the evolution of the concepts, conjectures, or propositions. The examples sometimes must be generated, other times they are well-known. Sometimes they come in sequences, as in the law; sometimes only one or two at a time, as in mathematics. Features of the examples used include their taxonomic class, such as known counter-example or simple case.

We have not discussed how one decides what examples to use in this dialectic process nor how to find or generate them. The latter is the subject of our past work on constrained example generation ("CEG") (Rissland 1980, 1981). The former is a complex question which we are currently investigating.

7. REFERENCES

[1] DeMillo, R.A., Lipton, R.J., and Perlis, A.J., "Social Progress and Proofs of Theorems and Programs". *CACM*, Vol. 22, No. 5, May 1979.

[2] Gregory, C.O., Kalven, H., and Epstein, R.A., *Cases and Materials on Torts*. Little, Brown and Company, Boston, 1977.

[3] Kuhn, T.S., *The Structure of Scientific Revolutions.* Second Edition, University of Chicago Press, 1970.

[4]__________, "Second Thoughts on Paradigms". In Suppe (Ed), *The Structure of Scientific Theories*, University of Chicago Press, 1974.

[5] Lakatos, I., *Proofs and Refutations.* Cambridge University Press, London, 1976.

[6] Munroe, M.E., *Introduction to Measure and Integration.* Addison-Wesley, Reading, MA, 1953.

[7] Rissland, E.L., *Constrained Example Generation.* Technical Report 81-24, Department of Computer and Information Science, University of Massachusetts, Amherst, Massachusetts, 1981.

[8] __________, "Example Generation". In *Proceedings Third National Conference of the Canadian Society for Computational Studies of Intelligence*, Victoria, B.C., May 1980.

[9] __________, "Examples in Legal Reasoning: Legal Hypotheticals". In *Proceedings IJCAI-83*, Karlsruhe, W. Germany, August 1983.

[10] __________, "Examples in the Legal Domain: Hypotheticals in Contract Law". In *Proceedings Fourth Annual Cognitive Science Conference*, Ann Arbor, Michigan, August 1982.

[11] __________, "Learning How to Argue: Using Hypotheticals". In *Proceedings First Annual Conference on Theoretical Issues in Conceptual Information Processing*, Atlanta, Georgia, March, 1984.

[12] __________, "Understanding Understanding Mathematics", *Cognitive Science*, Vol.2, No.4, 1978.

[13] Rissland, E.L. and Soloway, E.M., "Overview of an Example Generation System", in *Proceedings First National Conference on Artificial Intelligence*, Stanford University, August 1980.

[14] Rissland, E.L., Valcarce, E.M., and Ashley, K.D., "Explaining and Arguing with Examples". In *Proceedings AAAI-84*, Austin, Texas, August 1984.

[15] Selfridge, O.G., "Non-Algorithmic Software: A Proposal", submitted for publication, 1983.

[16] Swartout, W. and Balzer, R., "On the Inevitable Intertwining of Specification and Programming", *CACM*, Vol.25, No.7, July 1982.

ISBN: 0 444 87611 1