DIVISION OF COMPUTER SCIENCE

Formal Methods: No Cure for Faulty Reasoning

M Loomes and R Vinter

Technical Report No. 265

September 1996

Formal Methods: No Cure for Faulty Reasoning

Martin Loomes and Rick Vinter
Faculty of Information Sciences, University of Hertfordshire
Hatfield, United Kingdom

Abstract

Owing to the benefits commonly associated with their use and links with scientific culture, formal methods have become closely identified with the design of safety-critical systems. But, despite the mathematical nature of the logic systems underlying most formal notations, many aspects of formal methods are much less predictable than one might realise. Specifically, it is suggested that the ways in which people interpret and reason about formal descriptions can lead to similar kinds of errors and biases as those exhibited during previous cognitive studies of logical statements in natural language. This paper reports a series of preliminary experiments aimed at testing this hypothesis and several related issues. Early results suggest that, in reality, people frequently depart from fundamental principles of mathematical logic when reasoning about formal specifications, and are content to rely upon probablistic, heuristic methods. Furthermore, they suggest that manipulating such factors as the degrees of thematic and believable content in formal specifications can lead to significant reasoning performance enhancement or degradation. So, although faulty reasoning cannot be cured by formalisation alone, it would appear that the human potential for error can be reduced by avoiding certain expressions and choosing alternative, equivalent forms.

To appear in proceedings of the Safety-critical Systems Symposium 1997, Brighton, 4th-6th February 1997, published by Springer-Verlag.

Introduction

One of the strategies commonly adopted in the design of safety-critical systems is to imagine what can possibly go wrong and to consider ways of preventing or containing the errors. At the technological level we are quite adept at predicting and containing the consequences of failure of the system itself, through a variety of tried and tested techniques. We are less experienced, however, at predicting and avoiding the human errors in the system design process that lead to these failures. For example, the report on the Ariane 5 disaster [Lions96] clearly attributes the failure to faulty design which, with the benefit of hindsight, is simple to understand. However, whilst the recommendations in this report are specific at the technical level in identifying how the design fault could have been found during testing, they are rather vague in terms of improving the design process to avoid such errors arising in future. There are some global approaches that have gained support in recent years, such as adopting particular design methods or documentation standards, but it is far from clear that these really deliver much beyond a certain feel-good factor that comes from doing something rather than nothing.

A particular area that has become identified with safety-critical design is that of formal methods. Indeed, many designers who reject formal methods for their own use on commercial or industrial projects will often add the caveat that they can see their virtue for safety-critical systems. Why is this the case? Partly one suspects it is because formal methods, and mathematics in general, are associated with a scientific culture [Hoare84], and this is the culture that we, as consumers, hope and believe underpins professions such as medicine and engineering where new developments are welcomed but acknowledged as potentially life-threatening. "The appliance of science"[1] conjures up visions of careful, well thought-out and thoroughly analysed technological progress, based on mathematical formalisation and reasoning. Indeed, many advocates of formal methods in the past two decades have drawn extensively on analogies with other technical disciplines to demonstrate how we should seek these scientific and mathematical foundations for software if we aspire to becoming a professional engineering discipline.

Apart from the generally perceived, if nebulous, advantages of working within a scientific culture, adopting formal methods is thought to bring a number of specific benefits. Anyone who asks an undergraduate to list the advantages of formal methods is likely to get an answer containing words such as "unambiguous", "precise" and "correct". This can rapidly become interpreted as implying that the use of formal methods leads to descriptions of a system that are only open to one interpretation, mean exactly what we want them to mean, and where we can show that the system has exactly the properties we require. In fact, of course, it is only the syntax (and possibly a formal semantic interpretation) that is unambiguous and precise. How a reader chooses to interpret the description into some real-world problem domain, reason about the system across this interface and act upon the conclusions is rather less predictable or controllable.

In formalising computer-based systems, we are usually seeking to automate information processing in some form or another, and what needs to be captured is the reasoning process itself, where the obvious models come from logic, which does not currently enjoy the same status and shared culture as the mathematics traditionally applied in other realms of engineering. Logic, which was once a mainstream curriculum topic, is now distributed between mathematics, philosophy and linguistics, making it a complex beast to study and pin down. Thus whilst it may be very sensible to assume that two civil engineers reading a set of equations governing fluid flow will interpret them in the same way and reason about them to the same conclusions, it is far from obvious that this desirable behaviour will necessarily carry over to two software engineers reading formal specifications.

[1] This phrase has become well known in the UK as an advertising slogan for Zanussi washing machines.

> "It (logic) is justified in abstracting - indeed it is under obligation to do so - from all objects of knowledge and their differences, leaving the understanding nothing to deal with save itself and form" Kant [Smith93, p. 18].

Proponents of formal methods often adopt a point of view similar to Kant's, which suggests that formal, abstract, reasoning will be fault-free (save for possible slips which are unlikely to be replicated during subsequent analysis and thus will be easily spotted) as it is liberated from distractions such as intuitions and background knowledge. Whilst this view may be defended from a theoretical perspective, defining formal reasoning as perfect and everything that deviates from it as erroneous, the pragmatics of the situation become rather different. There have been several studies that show actual reasoning performance improves as the task becomes less abstract [Dominowski95, Griggs82, VanDuyne74, Wason71, Wilkins28]. Moreover, there are errors that are made systematically by large numbers of people which are unlikely to be spotted by naive inspection, uninformed of the likely sources of errors.

The COPSE project [Loomes94] was established to explore some of the cognitive and organisational factors that influence software engineers. Whilst most of the work has focused on the organisational and cultural issues, work has also started on the analysis of cognitive issues in the use of formal methods. There are clearly a number of possible starting points for such an investigation, and a number of places where emphasis could be placed. For example, studies of the differences between the use of various notations for formalisation, or differences between problem domains, utilising case studies or demonstrator projects, might yield significant results. The difficulty posed by this sort of high-level case study approach is that there are usually so many factors involved that it proves very hard to devise repeatable experiments, or to explain the results in terms of plausible theories that lead on to practical new experiments. Too often such studies tend to lead to anecdotal evidence, aimed at defending a favoured hypothesis rather than exposing a scientific hypothesis to scrutiny.

One current strand of work at Hertfordshire is attempting to pose and answer, using empirical techniques, a well-founded set of questions based on existing theoretical bodies of knowledge surrounding these issues. In order to achieve this, considerable refinement of the issues has been undertaken to reduce the number of factors under consideration at any one time; hopefully, this has been done without naive over-simplification which would render the results too far removed from real engineering practice. First, we are concentrating primarily on the interpretation of existing specifications, rather than the creative processes that lead to new specifications. Second, we are using a basic set of logical tools for most specifications, rather than an enhanced mathematical tool-set involving structures such as lists, functions and relations. Finally we are using the concrete syntax provided by the Z specification notation [Spivey92]. Although this introduces possible confounding factors by the use of a schema notation, Z is sufficiently popular to ensure that knowledgeable users can be found as participants for the experiments. Within this framework, a number of studies are underway to explore systematic errors in the interpretation of Z specifications and subsequent reasoning errors.

In order to ensure that these studies are based on existing bodies of knowledge, the starting point has been the psychological literature on logical reasoning. There have been many studies carried out over the years in which hard (that is, scientifically repeatable) results have been achieved showing that certain forms of logical expression can lead to faulty reasoning [Braine91, Johnson-Laird72, Lakoff71, Newstead83]. Most of these studies have been carried out using problems posed in natural language, with participants drawn from the general public. The initial question this project set out to explore was whether these results carry over to the realm of software engineers using an established formal notation which they believe they understand. If so, can we use the findings to make available tools and techniques which will aid designers in identifying areas of formal descriptions that are "at risk", and where defensive approaches need to be taken, perhaps by associating metrics with particular forms, or even banning the use of certain syntactic constructs in the specification of safety-critical systems? In this way we hope to develop a technological understanding of formal specification languages and their use to mirror the developing understanding of programming.

The Experiments

A number of areas of potential interest have been identified by analysis of the psychological literature which intersect with reasoning tasks commonly found in software engineering. Examples of these include: the problems of reasoning with implications, the tendency of readers to guess what formal text means based on intuitive interpretation regardless of any formal semantics, preferred styles of expression, the problems of disjunctive and conjunctive reasoning, and syllogistic reasoning with quantification. Space does not permit a detailed discussion of all the experiments and results to date, but this section gives an overview of the approach and highlights a few of the findings which might cause us to reflect on some of the received wisdom concerning the use of formal methods, including the degree of safety we associate with their use. The experiments are loosely clustered into two areas, those concerned with syntactic structure and its interaction with thematic content and those concerned with other features of the specifications under consideration such as their believability and literary style. As the project progresses these two areas will be brought together in more complex experiments exploring the interaction.

Two types of experiment are discussed below: pilot experiments, which were undertaken primarily to help refine the questions and methodology, and the main experiments which constitute the substantive part of the project. The pilot experiments were conducted on small numbers of participants and no claim is made for the statistical significance of their results, although they do suggest some interesting areas for further study. The main experiments are currently being conducted on far larger groups and are intended to provide statistically significant results in the suggested areas. Some of these are nearing completion and tentative results are mentioned.

Syntactic Features

A particularly famous study of human reasoning is the Wason four card problem [Wason66]. Subjects are confronted with a problem similar to that that shown in Figure 1. In abstract, logical, terms the rule is of the form $p \Rightarrow q$, and the four cards represent instances of p, q, $\neg p$ and $\neg q$. The "correct" cards to turn over are A (p) and 7 ($\neg q$), as these are the only instances that can falsify the rule conclusively. Turning over the 4 (q) card may increase our confidence by supplying positive evidence for the rule, but it will not help us to test it. Wason found, and this is a fairly repeatable result which one of the authors has regularly replicated with large groups of students studying logic, that although virtually every participant correctly selects the p case as relevant very few select the $\neg q$ case. Moreover, it is quite common to select both p and q, thus missing one test and carrying out an unnecessary one.

A pack of cards has letters on one side and numbers on the other. Here is a rule: "If there is an A on one side of the card then there is a 4 on the other". Here are four cards from the pack lying on a table.

$$\boxed{\text{A}} \qquad \boxed{\text{S}} \qquad \boxed{4} \qquad \boxed{7}$$

Which card(s) would you need to turn over in order to establish whether the rule is true or false?

Figure 1: Wason's abstract selection task.

One of the claims sometimes made for formal methods is that they help the process of test-set generation. Moreover, we might expect that if we make the implication explicit, by expressing it formally, we would cue the reader into potential problems, especially since these were almost certainly discussed when the notation was first taught. With this in mind, a logically equivalent problem to Wason's task was posed in Z (Figure 2) and carried out by a number of computer scientists with differing levels of Z experience. The aim was to see if there was any substance to this claim, and whether Wason's results would carry over into a formal expression of the task.

The requirements for software operation *InOut* are: "If the operation receives an A as input then it will output a 4". Its formal specification follows.

$$\begin{array}{|l}
\multicolumn{1}{l}{\textit{InOut}} \\ \hline
\textit{in}? : \textit{Letter} \\
\textit{out}! : \mathbb{N} \\ \hline
(\textit{in}? = A) \Rightarrow (\textit{out}! = 4) \\ \hline
\end{array}$$

(A) $\textit{in}? = A$ (B) $\textit{in}? = S$ (C) $\textit{out}! = 4$ (D) $\textit{out}! = 7$

Which inputs and outputs would enable you to test whether *InOut* is working correctly, according to its requirements?

Figure 2: The formalised selection task.

In spite of all the cues given, and the fact that participants were given unlimited time to complete the task, their performance on this experiment was actually worse than that of the "man in the street" on Wason's task, although only marginally so. Generally, a very close correlation between the results of this experiment and Wason's results was noted. No participant correctly recognised the significance of the $\neg q$ case - as compared with 4% in Wason's experiment [Wason72, p. 182]. Every participant correctly identified the p case as necessary, but (as in Wason's experiment) the most popular choice of combination was p and q. One possible explanation of this phenomenon is offered by Evans [Evans72], who suggests that in this sort of reasoning task people are often guilty of a "matching bias", preferring to give answers that contain the same terms as are contained in the problem presentation. In this case, both A (p) and 4 (q) appear in the question so they are preferred terms in the solution. It is important not to read too much into a very simple pilot experiment of this nature, although the results do suggest that we do not necessarily achieve improved reasoning performance by formalisation alone, and we ought to be aware of the possibility of matching bias when we subject formal specifications to tests such as walk-throughs.

One major criticism of the Wason task is that the task is highly abstract: no-one can really imagine why we might have cards of this type in existence. Given the following problem, which is formally identical but has thematic content, people rarely make mistakes.

> Here is a rule: "If a person is drinking alcohol, then the person must be over 18 years of age". There are four young people drinking in the bar and we know just one fact about each: one is drinking a beer, one is drinking lemonade, one is 15 years of age, and one is 20 years of age. Which of the youths would you need to question in order to establish whether the rule is being conformed to?

This suggests that we should explore not only formalisation, but also the degree of abstraction away from thematic content as we formalise. In order to pursue this, a set of experiments has been devised to evaluate reasoning performance in situations with varying degrees of formality and thematic content. Three groups of participants are asked to complete three different types of task. In the first group an abstract formal task is set, and participants reason formally about shapes and colours, with no obvious thematic connection. In the second group a thematic formal task is set, and participants deal with formally presented situations such as the relationship between the safety status of a nuclear reactor and the temperature of its coolant. The third group are set the same tasks as the abstract formal group, but with the tasks being presented in natural language. In all three groups, participants are given a series of questions which comprise a statement about the system involving implication, together with a premise about the system state. Rather than generating test cases as in the Wason experiment, the participants are asked to draw conclusions from the given information or to state that no conclusion is possible, and are also asked to provide confidence ratings on their answers. By manipulating the forms of the presented implication and the premises, we can elicit details of reasoning performance corresponding to different types of logical inference (*modus* *ponens* and *modus tollens*) and fallacious reasoning (*denial of the antecedent* and *affirmation of the consequent*) for all positive and negative combinations of the premises. For example, Figures 3-5 show one of the tasks set to explore the affirmation inference for all three groups. Formally, no valid conclusion can be drawn, but the psychological literature suggests that participants will fallaciously use *affirmation of the consequent* to draw conclusions.

If $colour' \neq blue$ after its execution, what can you say about the value of *shape* before operation *SetColour* has executed?

SetColour

$\Delta ShapeAndColour$

$(shape = circle) \Rightarrow (colour' \neq blue)$
$shape' = shape$

(A) $shape \neq rectangle$
(B) $shape = circle$
(C) $shape \neq circle$
(D) Nothing

Figure 3: An abstract formal logic based task.

If $\neg(reactor\_status! = Ok)$ after its execution, what can you say about *coolertemp* before operation *ReactorTempCheck* has executed?

ReactorTempCheck

$\Xi NuclearPlantStatus$
$reactor\_status! : Report$

$coolertemp > Maxtemp \Rightarrow \neg(reactor\_status! = Ok)$

(A) $coolertemp \leqslant Maxtemp$
(B) $coolertemp > Maxtemp$
(C) $coolertemp > Mintemp$
(D) Nothing

Figure 4: A thematic formal logic based task.

If the shape is a circle then the colour is not blue.
The colour is not blue.

Based on the above description, what can you say about shape?

(A) The shape is not a rectangle
(B) The shape is a circle
(C) The shape is not a circle
(D) Nothing

Figure 5: An abstract natural language based task.

This experiment is still ongoing, and so definitive results cannot be given, but tentative analysis of the data collected so far suggests that many of the results previously observed in the psychological literature carry across to experiments where the tasks are expressed in Z, and the participants are all software engineers trained in formal methods. It also suggests that there are some differences between the groups. For example, as expected, all three groups have very little difficulty with *modus ponens* reasoning. *Modus tollens* reasoning seems to be performed better in the formal groups than in the natural language group and the thematic formal group seems less prone to fallaciously *denying the antecedent.* The natural language group, however, seems less prone to *affirming the consequent.* In general, the formal thematic group seems to be performing slightly better than the other groups, and also seems to have more confidence in the answers given. If these results are born out in the final analysis it would suggest that formalisation can slightly improve reasoning performance with implications, but that certain forms of expression should be avoided and equivalent forms chosen. We should note, however, that even in the performance of the best group, several examples of faulty reasoning occur and the increased confidence of the group could lead to less rigorous testing which offsets any potential benefits.

Similar experiments involving abstract and thematic groups are being carried out to explore reasoning performance with conjunctive and disjunctive forms in Z: an area less well represented in the psychology literature. In particular, some of the contextual dependencies involved in the use of inclusive and exclusive disjunctions are being varied to see if these lead to differences in performance. The literature suggests that people generally find it easier to reason with exclusive forms, but formal logical systems frequently omit this from the primitive syntax, perhaps for reasons of theoretical elegance. These experiments are attempting to isolate features of individual connectives, together with positive and negative instances of propositions. Once reliable results from these experiments are available, further experiments will be carried out with compound forms, involving all combinations of connectives, and also quantifiers will be introduced. There is a significant amount of prior work on faulty reasoning with quantifiers, and the major thrust of this work will be to see if these results carry across to our problem domain, or whether improvements in performance come about with formalisation in Z.

Evaluation and Interpretation of Specifications

Several experiments have been initiated aimed at identifying the cognitive processes involved when readers are asked to evaluate the quality of given specifications or translate them into natural language. In particular, two pilot experiments have been carried out to explore how effective the use of formal specifications is at communicating system concepts, and how engineers themselves assess the quality of the specifications. In the first experiment, for example, we set out to test Gravell's assertion, based on an informal "straw poll" of software engineers opinions, that "To communicate clearly with the majority of readers you should, in general, prefer clarity to brevity" [Gravell90, p. 139]. A group of software engineers were presented with an English description of a simple operation "Toggle", which exchanges the current state of a simple two-way switch, and the following four formal descriptions of the same system.

Toggle

$s, s' : SWITCH$

$s' \neq s$

Concise

Toggle

$s, s' : SWITCH$

$(s = \mathit{off} \land s' = \mathit{on}) \lor$
$(s = \mathit{on} \land s' = \mathit{off})$

Verbose

Toggle

$s, s' : SWITCH$

$s = \mathit{on} \Rightarrow s' = \mathit{off}$
$s = \mathit{off} \Rightarrow s' = \mathit{on}$

Precise

Toggle

$s, s' : SWITCH$

$(s = \mathit{on} \lor s = \mathit{off}) \Rightarrow$
$(s' = \mathit{on} \lor s' = \mathit{off})$

Imprecise

Figure 6: The four styles of formal specification.

These four formal specifications could be classified as concise, precise, verbose and imprecise. The imprecise version under-specifies the system, but the other three are formally equivalent. The participants were asked which version best described the system's behaviour, and to justify their choices. They all rejected the imprecise version, but were split evenly between the other three versions. There was an interesting correlation between age, experience and chosen style, with the older, more experienced engineers preferring the precise style, and the younger, less experienced engineers preferring the concise style. There are many factors that could explain this, including prior experience of specific problems, educational backgrounds or cultural trends. What is significant, however, is that in any project team there are likely to be engineers working with different preferred styles, frequently being asked to work with presentations not in their preferred style. This also highlights the fact that we cannot ignore the individual differences between engineers when attempting to identify "good practice" in design or devise training programs for their development. One possible avenue of exploration is to investigate subsequent reasoning performance based on precise, concise and verbose forms to see if there are any significant differences.

More worrying is the result of another pilot study where participants were shown the specification of a system containing a counter-intuitive clause. They were asked to translate a given schema into natural language in order to test the claim made by Liskov and Berzins that "there is only one way to interpret a formal specification because of the well defined and unambiguous semantics of the specification language" [Liskov79, p. 279]. Whilst it may be true that there is only one formal interpretation of the given schema into some abstract denotational semantics, what is of practical interest is whether real software engineers will reflect this single interpretation in their own understanding of the specification, and the way their interpretation governs their behaviour.

$$
\begin{array}{|l}
\textit{Library} \\
\hline
stock : Copy \pfun Book \\
issued : Copy \pfun Reader \\
shelved : \mathbb{F}\, Copy \\
readers : \mathbb{F}\, Reader \\
\hline
shelved \cup \operatorname{dom} issued = \operatorname{dom} stock \\
shelved \cap \operatorname{dom} issued = \varnothing \\
\operatorname{ran} issued \subseteq readers \\
\neg \exists\, r : readers \bullet \neg(\#(issued \rhd \{r\}) > maxloans) \\
\hline
\end{array}
$$

Original fourth predicate: $\forall\, r : readers \bullet \#(issued \rhd \{r\}) \leq maxloans$
The number of books that any reader borrows must be less than or equal to the maximum number of loans allowed.

Revised fourth predicate: $\neg \exists\, r : readers \bullet \neg(\#(issued \rhd \{r\}) > maxloans)$
The number of books that any reader borrows must be more than the maximum number of loans allowed.

Figure 7: The library specification - modified from [Potter91, p.124].

Most engineers gave correct interpretations of the first three clauses, which were consistent with their intuitions, although we should perhaps be concerned with the 25-33% who erred in each case! However, all of the participants failed to provide a correct interpretation of the final clause. In each case, the participant provided an interpretation which was consistent with an intuitive understanding of library systems, rather than the text as given. The semantics certainly appeared to be unambiguous to the readers, but the meaning was not what the author intended. One possible explanation of this is that readers use the formal specification to obtain linguistic cues regarding the domain of interest, then fit the mentioned terms into relations that are consistent with their intuitions. If this is the case we should be deeply concerned, as it suggests that far from assisting the reasoning process by liberating the engineer from errors caused by faulty intuitive reasoning, we may in fact be obscuring intuitive reasoning by formalisation: achieving a feel-good factor with "the appliance of science" that is not deserved. The fact that every participant made the same mistake suggests that this sort of error is unlikely to be picked up by testing. Clearly this was an extreme example, deliberately chosen to be directly counter-intuitive, but the strength of the result suggests we should take it seriously.

Indeed, there are a number of examples in the psychology literature that suggest people generally tend to abandon logical principles for reasoning in favour of heuristic and probabilistic methods when confronted with arguments containing information relating to strongly-held beliefs [Janis43, Morgan44, Evans83, Oakhill90]. The experiments carried out so far suggest that this phenomenon carries across to our problem domain. A further set of experiments is currently being carried out to explore this systematically. In addition to the two dimensions of formal/informal and abstract/thematic, believable/incredible will be added. Although we are not really interested in the extreme of incredible, eliciting the belief structures and strength of opinion from engineers may provide pointers to potential sources of reasoning errors.

Conclusions

The results of experiments carried out so far, although still tentative, indicate that many of the errors in reasoning that have been noted by psychologists in experiments with ordinary people, working in a natural language, seem to arise just as frequently with software engineers working in Z. Although some improvements have been observed in a few specific situations, the improvements are not as dramatic as one might expect from reading some of the more evangelical literature from proponents of formal methods. There are at least two possible reactions to this. We might start to doubt that formal methods have a role to play in the design of safety-critical systems at all. Perhaps more constructively, however, we might see this as an opportunity to explore how reasoning errors arise, and to develop ways of working that defend against them.

One of the strengths of formalisation in this context is that the grammatical structures are well-defined, and hence we can carry out well-controlled experiments in ways that we cannot easily do with natural languages. Moreover, we can imagine tools that might highlight potential areas of concern in formal specifications, and suggest alternative equivalent logical forms that are less prone to causing errors. We might even learn some lessons from the use of formal methods that we can carry across into less formal reasoning about systems. The main aim of this paper, however, is not to influence the perceptions or use of formal methods. The results are still far too sketchy to warrant this. Rather, the authors' hope is that we will open up the systematic analysis of these sort of issues as a topic of research in Software Engineering. In our opinion, it is dangerous to expend all our effort on the developments of methods and notations in the discipline based on anecdotal evidence or case study material that cannot be easily replicated or generalised.

References

[Braine91] Braine M.D.S. and O'Brien D.P., A theory of If: A lexical entry, reasoning program, and pragmatic principles. *Psychological Review*, *98*, 182-203, 1991.

[Dominowski95] Dominowski R.L., Content effects in Wason's selection task. In S.E. Newstead and J.St.B. Evans (Eds.), *Perspectives on Thinking and Reasoning. Essays in Honour of Peter Wason.* Hove UK: Lawrence Erlbaum Associates, 1995.

[Evans72] Evans J.St.B.T., Interpretation and matching bias in a reasoning task. *Quarterly Journal of Experimental Psychology*, *24*, 193-199, 1972.

[Evans83] Evans J.St.B.T., Barston J.L. and Pollard P., On the conflict between logic and belief in syllogistic reasoning. *Memory and Cognition*, *11* (3), 295-306, 1983.

[Gravell90] Gravell A., What is a good formal specification? In J.E. Nicholls (Ed.), *Z User Workshop, Oxford 1990. Proceedings of the Fifth Annual Z User Meeting, Oxford 17-18 December 1990*, Springer-Verlag, 1990.

[Griggs82] Griggs R.A. and Cox J.R., The elusive thematic materials effect in the Wason selection task. *British Journal of Psychology, 73*, 407-420, 1982.

[Hoare84] Hoare C.A.R., Programming: Sorcery or science, *IEEE Software*, 5-16, April 1984.

[Janis43] Janis L. and Frick F., The relationship between attitudes toward conclusions and errors in judging logical validity of syllogisms. *Journal of Experimental Psychology, 33*, 73-77, 1943.

[Johnson-Laird72] Johnson-Laird P.N. and Tridgell J.M., When negation is easier than affirmation. *Quarterly Journal of Experimental Psychology, 24*, 87-91, 1972.

[Lakoff71] Lakoff R., If's, and's, and but's about conjunction. In C.J. Fillmore and D.T. Langendoen (Eds.), *Studies in Linguistic Semantics.* New York: Holt, Rinehart and Winston, 1971.

[Lions96] Lions J.L., *Ariane 5: Flight 501 Failure. Report by the Inquiry Board.* Paris: European Space Agency, 19 July 1996.

[Liskov79] Liskov B. and Berzins V., An appraisal of program specifications. In P. Wegner (Ed.), *Research Directions in Software Technology*, Cambridge, Mass: MIT Press, 1979.

[Loomes94] Loomes M., Ridley D. and Kornbrot D.E., Cognitive and organisational aspects of design. In F. Redmill and T. Anderson (Eds.), *Technology and Assessment of Safety-critical Systems*, Springer-Verlag, 1994.

[Morgan44] Morgan J.J.B. and Morton J.T., The distortion of syllogistic reasoning produced by personal convictions. *Journal of Social Psychology, 20*, 39-59, 1944.

[Newstead83] Newstead S.E. and Griggs R.A, The language and thought of disjunction. In J.St.B.T.Evans (Ed.), *Thinking and Reasoning. Psychological Approaches*, 76-106, London: Routledge and Kegan Paul, 1983.

[Oakhill90] Oakhill J., Garnham A., and Johnson-Laird P.N., Belief bias effects in syllogistic reasoning. In K.J.Gilhooly, M.T.G. Keane, R.H. Logie, and G. Erdos, *Lines of Thinking: Reflections on the Psychology of Thought. Volume 1. Representation, Reasoning, Analogy and Decision Making*, 125-138,. Chichester: John Wiley and Sons, 1990.

[Potter91] Potter B., Sinclair J. and Till D., *An Introduction to Formal Specification and Z.* Hemel Hempstead: Prentice-Hall, 1991.

[Smith93] Smith N.K. (translator), *Immanuel Kant's Critique of Pure Reason.* Second Edition, London: Macmillan, 1993.

[Spivey92] Spivey J.M., *The Z Notation: A Reference Manual.* Second Edition. Hemel Hempstead, Prentice Hall International, 1992.

[VanDuyne74] Van Duyne P.C., Realism and linguistic complexity in reasoning. *British Journal of Psychology*, 65, 59-67, 1974.

[Wason66] Wason P.C., Reasoning. In B.M. Foss (Ed.), *New Horizons in Psychology. Volume 1*, Reading: Penguin, 1966.

[Wason71] Wason P.C. and Shapiro D., Natural and contrived experience in a reasoning problem. *Quarterly Journal of Experimental Psychology, 23*, 63-71, 1971.

[Wason72] Wason P.C. and Johnson-Laird P.N., *Psychology of Reasoning: Structure and Content.* London: Batsford, 1972.

[Wilkins28] Wilkins M.C., The effect of changed material on ability to do formal syllogistic reasoning. *Journal of Social Psychology, 24*, 149-175, 1928.