A SYNTHESIS OF INTRODUCTORY INFORMATION FOR UNDERGRADUATES CONSIDERING FORMAL METHODS GRADUATE RESEARCH IN 2001

A SYNTHESIS OF INTRODUCTORY INFORMATION FOR UNDERGRADUATES CONSIDERING FORMAL METHODS GRADUATE RESEARCH IN 2001

Prepared for
Ms. Deborah Curtis, Lecturer
Department of Information Systems and Decision Sciences
5245 North Backer Ave. M/S 7
Fresno, CA 93740-8001

Prepared By
Jason J. Simas, Student
Department of Computer Science
5241 N. Maple Ave. M/S MF109
Fresno, CA 93740-8027

May 13, 2001

Department of Computer Science
5241 N. Maple Ave. M/S MF109
Fresno, CA 93740-8027

May 13, 2001

Ms. Deborah Curtis
Lecturer
Department of Information Systems and Decision Sciences
5245 North Backer Ave. M/S 7
Fresno, CA 93740-8001

Dear Ms. Curtis:

Here is the long, formal, informational business report that you requested last March 4.

I wrote a report useful for undergraduates considering graduate school and looking for a research area. Within Computer Science and Applied Logic is an area of research called Formal Methods. It is a challenging yet promising research area.

Thank you for the opportunity to practice and develop my research, synthesis, and communication skills.

Sincerely yours,

Jason J. Simas
Student

TABLE OF CONTENTS

LIST OF FIGURES

Executive Summary

Formal Methods is a research area in Computer Science. Formal methods are advanced or optimized methods for solving problems. Formal Methods research is finding formal methods for solving the problems of computer systems engineering. Throughout human history, our methods have moved in the direction of becoming more formal. One would expect that Formal Methods will grow in popularity.

A formal method is a method which invokes a formalism. A method is a way of solving a problem. A formalism is a system of symbols and symbol manipulations. Mathematics is a formalism; so is first order logic. Mathematics deals with problems of distance and quantity. First order logic deals with problems of truth. The idea of Formal Methods is to construct formalisms that deal with problems of computer systems engineering. The goal of Formal Methods is to widen the scope of problems to which formal methods are applied.

The advancements that have been made in Formal Methods are being applied to some real computer systems. Particularly specification languages are being applied. Specification languages specify the functionality of the system. With a formal specification language, a specification can be checked to have certain desirable properties, before actually building and testing the system.

Currently, research in Formal Methods is waning. Of the 35 top rated universities offering doctorates in Computer Science, only 15 of them have professors conducting research in Formal Methods. This fact is a double edged sword. Finding a graduate school with a rich environment for Formal Methods research is difficult. However, due to the lack of research going on in Formal Methods, the area is less crowded, and a graduate student has many directions open to him.

A SYNTHESIS OF INTRODUCTORY INFORMATION FOR UNDERGRADUATES CONSIDERING FORMAL METHODS GRADUATE RESEARCH IN 2001

INITIAL REMARKS

Formal Methods Research

There are numerous research areas in Computer Science, and the area of Formal Methods is one of them. Among the many things Computer Scientists do, is to invent methods for solving problems of engineering computer systems. These methods can be formal or informal. Formal methods differ from informal methods in that they invoke formalisms. Currently, arguments ensue over the question of applying formal methods to a wider scope of engineering problems. Proponents claim that formal methods can supply higher quality computer systems and can reduce costs of computer system engineering. Opponents rebut claiming that formal methods are difficult to use and aren't applicable for solving industrial sized problems. For now, the opponents appear to have the stronger argument; however formal methods are improving. Given sufficient research and development, arguments for using formal methods on more engineering problems will become stronger.

Overview

In this report, information relevant to undergraduates considering doing graduate research in Formal Methods is presented. Choosing a research area for graduate study requires knowledge of the area under consideration. To enable this, an abstract description of Formal Methods is provided. In United States universities, the amount of research occurring in Formal Methods is small relative to the amount occurring in some other Computer Science research areas. Professors conducting research in Formal Methods is scarce. Information pertinent to finding the best university for Formal Methods is presented.

Methodology

The information provided in this report is a synthesis of information scattered among introductory articles of Formal Methods, software engineering textbooks, university web sites, and university ratings publications. That is the methodology was to research the literature on Formal Methods. Detail has been minimized, and generality has been maximized. The reader of this report should receive a big picture understanding of information relevant to making a choice in pursuing research in Formal Methods.

Terminology

The term, 'Formal Methods', is used in this report to refer to the area of research interested in formulating and applying methods which are formal to the engineering of computer systems. The term, 'formal methods', is used to refer methods for solving problems of engineering computer systems and that are formal.

Before describing Formal Methods, a simple example of how and why methods have evolved to become more formal will be presented. Mathematics is a formalism consisting of symbols (i.e. numbers) and symbol manipulations (addition, subtraction, etc.). A method which invokes math, is properly called a formal method.

BUILDING AN INTUITIVE UNDERSTANDING

An Informal Method

Before math, the determination of quantities or distances was based on an informal method. For example, how might a host determine whether there were enough plates to invite their guests over to eat? The best the host could do would be to imagine each guest in turn, setting aside a plate for each. If the host ran out of plates before imagining all the guests and himself, then the host would know that there weren't enough plates. The method works, but it isn't very powerful. This is brought out by considering another situation. Consider the host invites the guests to come back the following week and invites them bring some of their friends as extra guests. How will the informal plate check method be applied? There are two problems. The host must first get the extra guests into his imagination. The best that can be done is for the guests to write down a list of names of the extra guests. The host could use this in the plate check method. The second problem is running the plate check method, noting that the plate check method must start from the beginning. Once the plates are cleaned and stacked, there is no way of knowing where to begin setting aside plates again. So, the plate check method must begin again with the first plate and the first guest. Other solutions require placing marks on the plates, or setting aside plates until the following week. Obviously, these solutions are less than ideal.

A Formal Method

Mathematics is excellent in solving these types of problems. We take for granted when we apply mathematics to such simple problems, that we are applying a formal method. First, the host can count the number of plates available. This action translates information about the physical world into a mathematical model of the world. The number which results *symbolizes* the quantity of plates available. Once this number is obtained, there is no need for running through the plates again, no matter what situation results. Second, the host could have counted the number of original guests at the first gathering. That number symbolizes the quantity of original guests. The plate problem for the first gathering could have been solved by subtracting the number of plates from the number of guests, if the resultant number was positive number, then there was enough plates for the gathering. For the second gathering, only the number of extra guests and not a list of their names would be required. Perhaps the extra guests was a well known group that someone had already counted, further increasing the ease of dealing with the situation of having more guests. The number of plates subtracted from the number resulting from the addition of the number of original guests with the number of extra guests, would result in a new number. A positive number would imply that enough plates were available.

Lessons Learned

Mathematics is a formalism, and when invoked, produces a formal method. When a formal method is used in the plate problem, work is reused and communication is facilitated. For every new situation, there are some simple manipulations of the previously obtained symbols, which produce the desired solution. Information from the physical world need only be translated into the formal system once. After that, any problems which require this information can be produced. Further, the formalisms used in the problem provide a concise way to communicate the necessary information. A number is much easier to communicate than a list of names. This example only hints at the power of formal methods.

FORMAL METHODS FOR COMPUTER SYSTEMS ENGINEERING

Research Areas

There are many problems which can be solved by computer systems. Building computer systems to solve these problems, poses its own problems. Formal Methods addresses the problems of building computer systems. There are types of computer systems, and there are stages of development of computer systems. Problems which can be solved by formal methods occur in all classes of systems and in all stages of their development. Representative classes of computer systems are hardware systems, software systems, database systems, distributed systems, and real-time systems. Representative stages are specification, design, implementation, and verification. Formal Methods research may occur for any one type, any one stage, or for any one type at any one stage. For instance, a researcher may specialize in Formal Methods for specification of distributed systems.

Evolution of Methods

But why create formal methods for these problems? The answer becomes apparent after considering that computer scientists and computer engineers have to build many systems and have to build them to high standard of quality. With problems as complex as building computer systems, a powerful approach is needed. One of the major problems is simplifying the reasoning that occurs in solving these problems. For any problem that is solved many times, one gets tired of having to reason out the solution each time. One wants a formula to plug the information of the problem into, and receive the answer. There cannot be one formula, because that would mean that only one problem could be solved. Formalisms provide a way of having multiple formulas that can be combined or sequenced so that they can be applied to solve multiple problems. Reasoning then, only occurs in applying the formalism properly, to obtain the desired solution.

On Formalism

By why are these methods called formal? The formalisms have to be well defined so that a person knows whether they can be applied and exactly what the result of the formulism tells them. One of the goals of the formalism is to simplify the reasoning. The reasoning is simplified by allowing the user of the formalism to focus his attention on the symbols, once the information of the problem is translated into them. After the information of the problem is

translated into symbols, the which symbol operations can be applied to them, and what results would come of them if they were applied, is perfectly defined. This simplifies the reasoning involved, reference only the symbols and rules of the system need be referenced.

As has been said, mathematics is an example of a formalism. If we were to solve a problem about how many guests are coming, once the information of the quantity of guests in translated into symbols in math, then no more reference to the problem is necessary. The numbers imply that the addition operation can be applied, and addition will imply a particular answer. Which answer is obtained is dependent not on the content of the problem, but on the symbols and the operations applied to those symbols. Reasoning is simplified because the problem can be forgotten. This is helpful to computer system engineers because the problems of engineering computers are complex. Forgetting about the detail, and only thinking of the formalism is easier. Discovering reusable methods, determining when they are applicable, and describing them so that others can use them is part of the research of Formal Methods.

Formal Methods for Specification

Formal Methods has a future in all areas, but the future for their use is nearer in some than in others. Use of Formal Methods in the specification of computer systems is one area where the research from it is increasingly being applied. Systems are built for customers because customers have work that can be performed by systems. So, systems are built to perform certain work. An early stage of system development is determining and documenting what work the system is to perform. This process is called specifying the system, and the result of it is a specification of the system. A specification is a statement of the requirements of the system. (Wing 1999, 167)

Many system specifications are written in a structured natural language. A structured natural is a language in which there is an order, organization, or format in which information is to be communicated, yet the information is communicated in a natural language. For example, a specification document template would impose a structure on the natural language of the specification. One of the major disadvantages of using a structured natural language as opposed to using a formal language as part of a Formal Method, is that a structured natural language specification is vague if not ambiguous in some cases and as a result is not analyzable by computers (Bowen et alia 1999, 128).

For various reasons, specifying a system generally requires breaking it into parts and specifying its parts. The parts must work together, so their specifications are interdependent. After one part is specified, another must be specified in compliance with the interdependent requirements of the other part. Complex systems have many interdependent parts; specifying them in a way that respects these interdependencies is difficult. There are two reasons for this difficulty. Either the sheer number of interdependencies causes them to be unmanageable or the specifications are vague or ambiguous. A formal language provides for a clear and unambiguous interpretation of a specification.[1] Because of this, specifications written in a formal specification language are analyzable by computers. After one specifies a part, a computer can be used to conduct an automated check on whether the interdependencies have been respected. When the

[1] An example of a formal language, though not a specification language, is the language of math (i.e. x, y, z, +, =).

interdependencies are respected, a specification is said to be consistent. "Ambiguity, incompleteness, and inconsistency can be discovered and corrected more easily, not though ad hoc review but through the application of mathematical analysis" (Pressman 2001, 43). Automated consistency checks is a major advantage of formal methods in specification.

GRADUATE SCHOOLS FOR FORMAL METHODS

In the United States, Formal Methods research in universities is sparse. Of thirty-five top rated Computer Science doctorate programs in the US, only fifteen had resident professors researching Formal Methods. Explanations for the lack of research going on in Formal Methods are speculative. However, computer systems engineering is in an early stage of development (Brookshear 2000, 7). The goal is to merely produce the necessary systems, and not to produce them in the optimal way. Formal methods are optimizations of existing methods. Hence, why Formal Methods is not popular is understandable. Further, that Formal Methods will increase in popularity as optimization becomes important, is reasonable (Pressman 2001, 44).

When choosing a university to research Formal Methods both the number of faculty and their ranking should be considered. Figure 1 tabulates these numbers for the top 35 universities as ranked by the Gourman Report (Gourman 1997, 24). Stanford University and Carnegie Mellon University appear to be on top and on par with respect to graduate programs for Formal Methods. They both have four faculty researching Formal Methods and are ranked within the top three doctorate programs for Computer Science. University of Texas in Austin follows closely.

Figure 1: Number of Faculty For Top Ranked Universities

Name	URL	Faculty	Ranking
Stanford University	www.stanford.edu	4	2
Carnegie Mellon University	www.cmu.edu	4	3
University of Texas, Austin	www.utexas.edu	4	7
University of Pennsylvania	www.upenn.edu	3	24
University of California, Irvine	www.uci.edu	3	34
Massachusetts Institute of Technology	www.mit.edu	2	1
University of North Carolina, Chapel Hill	www.unc.edu	2	30
State University of New York, Stony Brook	www.sunysb.edu	2	31

Further considerations should be given to the prominence of the faculty in the Formal Methods community and its affiliation with any local research facilities. A major player in Formal Methods and author of the article *Software Engineering* in the Encyclopedia of Software Engineering, Dr. Jeanette Wing is at Carnegie Mellon University. Stanford is located near the Software Research Institute (SRI). SRI conducts research in Formal Methods under the guidance of a prominent leader in Formal Methods, John Rushby. Of course there are many other considerations to give depending on the particular person. Barring these personal considerations, Carnegie Mellon and Stanford are the best institutions for graduate work in Formal Methods.

FINAL REMARKS

Formal Methods is a challenging yet promising research area. The challenge comes from the area being so new. Computer systems engineering is a relatively young discipline. Developing formal methods to solve its problems is even younger. The problems confronting Formal Methods are first order. They are problems not of filling in gaps, but of developing the framework. This means the research area is fairly wide open, and ready for young researchers to choose the sub areas within it for which they have the most interest.

Formal methods have been the useful methods of solving the problems which have confronted humans over the years. Mathematics was a formalism invented to solve problems of quantity and distance. Logic was invented to solve problems of truth. Hybrid combinations of mathematics and logic have been constructed and applied to numerous problems. The advantages of formal methods are clear. The claim that formal methods will offer computer systems engineering similar advantages, is reasonable. Because of the fundamental reasons that drive research in Formal Methods, persons considering Formal Methods as a research area for graduate school should feel comfortable with that option.

Roger Pressman, author of the leading textbook on Software Engineering, writes:

> "Although [Formal Methods] is not destined to become a mainstream approach, the formal methods model offers the promise of defect-free software. Yet, the following concerns about its applicability in a business environment have been voiced:
>
> 1. The development of formal models is currently quite time consuming and expensive.
> 2. Because few software developers have the necessary background to apply formal methods, extensive training is required.
> 3. It is difficult to use the models as a communication mechanism for technically unsophisticated customers.
>
> These concerns notwithstanding, it is likely that the formal methods approach will gain adherents among software developers who must build safety-critical software (e.g., developers of aircraft avionics and medical devices) and among developers that would suffer economic hardship should software errors occur." (Pressman 2001, 44).

However, for Formal Methods to be increasingly applied to the engineering of computer systems, research must be invested. Persons choosing to research Formal Methods will likely have a challenging yet promising future.

REFERENCE LIST

Brookshear, J. Glenn. 2000. Computer Science: An Overview. Addison Wesley Longman.

Bowen, J. P. and Michael G. Hinchey. 1999. High-Integrity System Specification and Design. New York: Springer.

Gourman, Jack. 1997. The Gourman Report : A Rating of Graduate and Professional Programs in American and International Universities. New York: Random House.

Pressman, Roger S. 2001. Software Engineering: A Practitioner's Approach. New York: McGraw-Hill.

Wing, Jeannette M. 1999. A Specifier's Introduction to Formal Methods. In High-Integrity System Specification and Design, J. P. Bowen and Michael G. Hinchey, Pp. 167-199. New York: Springer.