Preface

Overview

The objective of this book is to give the reader a flavour of the formal methods field. The goal is to provide a broad and accessible guide to the fundamentals of formal methods, and to show how they may be applied to various areas in computing.

There are many existing books on formal methods, and while many of these provide more in-depth coverage on selected topics, this book is different in that it aims to provide a broad and accessible guide to the reader, as well as showing some of the rich applications of formal methods.

Each chapter of this book could potentially be a book in its own right, and so there are limits to the depth of coverage. However, the author hopes that this book will motivate and stimulate the reader, and encourage further study of the more advanced texts.

Organization and Features

Chapter 1 provides an introduction to the important field of software engineering. The birth of the discipline was at the Garmisch conference in Germany in the late 1960s. The extent to which mathematics should be employed in software engineering remains a topic of active debate.

Chapter 2 discusses software reliability and dependability, and covers topics such as software reliability and software reliability models; the Cleanroom methodology, system availability, safety and security critical systems, and dependability engineering.

Chapter 3 discusses formal methods, which consist of a set of mathematic techniques that provide an extra level of confidence in the correctness of the software. They may be employed to formally state the requirements of the proposed

system, and to derive a program from its mathematical specification. They allow a rigorous proof that the implemented program satisfies its specification to be provided, and they have been mainly applied to the safety critical field.

Chapter 4 provides an introduction to fundamental building blocks in discrete mathematics including sets, relations and functions. A set is a collection of well-defined objects, and it may be finite or infinite. A relation between two sets A and B indicates a relationship between members of the two sets, and is a subset of the Cartesian product of the two sets. A function is a special type of relation such that for each element in A there is at most one element in the co-domain B. Functions may be partial or total and injective, surjective or bijective.

Chapter 5 presents a short history of logic, and we discuss Greek contributions to syllogistic logic, stoic logic, fallacies and paradoxes. Boole's symbolic logic and its application to digital computing are discussed, and we consider Frege's work on predicate logic.

Chapter 6 provides an introduction to propositional and predicate logic. Propositional logic may be used to encode simple arguments that are expressed in natural language, and to determine their validity. The nature of mathematical proof is discussed, and we present proof by truth tables, semantic tableaux and natural deduction. Predicate logic allows complex facts about the world to be represented, and new facts may be determined via deductive reasoning. Predicate calculus includes predicates, variables and quantifiers, and a predicate is a characteristic or property that the subject of a statement can have.

Chapter 7 presents some advanced topics in logic including fuzzy logic, temporal logic, intuitionistic logic, undefined values, theorem provers and the applications of logic to AI. Fuzzy logic is an extension of classical logic that acts as a mathematical model for vagueness. Temporal logic is concerned with the expression of properties that have time dependencies, and it allows properties about the past, present and future to be expressed. Intuitionism was a controversial theory on the foundations of mathematics based on a rejection of the law of the excluded middle, and an insistence on constructive existence. We discuss three approaches to deal with undefined values, including the logic of partial functions; Dijkstra's approach with his cand and cor operators; and Parnas's approach which preserves a classical two-valued logic

Chapter 8 presents the Z specification language, which is one of the more popular formal methods. It was developed at the Programming Research Group at Oxford University in the early 1980s. Z specifications are mathematical, and the use of mathematics ensures precision, and allows inconsistencies and gaps in the specification to be identified. Theorem provers may be employed to demonstrate that the software implementation satisfies its specification.

Chapter 9 presents the Vienna Development Method, which is a popular formal specification language. We describe the history of its development at IBM in Vienna, and the main features of the language and its development method. Chapter 10 discusses the Irish school of VDM, which is a variant of classical VDM. We discuss its constructive mathematical approach, and where it differs from standard VDM.

Chapter 11 presents the unified modelling language (UML), which is a visual modelling language for software systems. It presents several views of the system architecture, and was developed at Rational Corporation as a notation for modelling object-oriented systems. We present various UML diagrams such as use case diagrams, sequence diagrams and activity diagrams.

Chapter 12 focuses on the approach of Dijkstra, Hoare and Parnas. We discuss the calculus of weakest preconditions developed by Dijkstra and the axiomatic semantics of programming languages developed by Hoare. We then discuss the classical engineering approach of Parnas, and his tabular expressions.

Chapter 13 discusses automata theory, including finite-state machines, pushdown automata and Turing machines. Finite-state machines are abstract machines that are in only one state at a time, and the input symbol causes a transition from the current state to the next state. Pushdown automata have greater computational power than finite-state machines, and they contain extra memory in the form of a stack from which symbols may be pushed or popped. The Turing machine is the most powerful model for computation, and this theoretical machine is equivalent to an actual computer in the sense that it can compute exactly the same set of functions.

Chapter 14 discusses model checking which is an automated technique such that given a finite-state model of a system and a formal property, then it systematically checks whether the property is true of false in a given state in the model. It is an effective technique to identify potential design errors, and it increases the confidence in the correctness of the system design.

Chapter 15 discusses the nature of proof and theorem proving, and we discuss automated and interactive theorem provers. We discuss the nature of mathematical proof and formal mathematical proof.

Chapter 16 discusses probability and statistics and includes a discussion on discrete random variables; probability distributions; sample spaces; sampling; the abuse of statistics; variance and standard deviation; and hypothesis testing.

Chapter 17 discusses a selection of tools that are available to support the formal methodist in the performance of the various activities. Tools for VDM, Z, B, UML, theorem provers and model checking are considered.

Chapter 18 discusses technology transfer of formal methods to industry, and is concerned with the practical exploitation of new technology developed by an academic or industrial research group, and the objective is to facilitate its use of the technology in an industrial environment. Chapter 19 summarizes the journey that we have travelled in this book.

Audience

The audience of this book includes computer science students who wish to gain a broad and accessible overview of formal methods and its applications to the computing field. This book will also be of interest to students of mathematics who

are curious as to how formal methods are applied to the computing field. This book will also be of interest to the motivated general reader.

Acknowledgements

I am deeply indebted to family and friends who supported my efforts in this endeavour, and my thanks, as always, to the team at Springer. This book is dedicated to my late aunt (Mrs. Noreen O' Regan), who I always enjoyed visiting in Clonakilty, Co. Cork.

Cork, Ireland Gerard O'Regan

http://www.springer.com/978-3-319-64020-4

Concise Guide to Formal Methods
Theory, Fundamentals and Industry Applications
O'Regan, G.
2017, XXVI, 322 p. 81 illus., 56 illus. in color., Softcover
ISBN: 978-3-319-64020-4