applications

communication protocols

processors (CPUs)

IBM's CICS

kernel of a secure distributed operating system

compilers

safety-critical: medical systems, nuclear control

railways: French — safety-critical, Chinese — all aspects

aerospace — attitude monitors

instrumentation systems

AT&T switching system

Airbus cabin communication

AAMP5 commercial microprocessor

programs are

commands to a computer → execution

mathematical expressions → theory of programming

why theory? → proof, calculation, precision, understanding

theory = formalism + rules of proof, calculation, manipulation

formal ⧧ careful, detailed

informal ⧧ sloppy, sketchy

formal = using formulas (mathematical expressions)

informal = using a natural language (English) start informal (with discussion)

end formal (with program)

then test, but

- how do you know if the program is working?
- what about the inputs you didn't test?

proof tells whether program is correct for all inputs

~~proof / verification after development~~

program development, with proof at each step

program modification, with proof

other theories

Hoare triples $P\{S\}R$ or $\{P\}S\{R\}$

Dijkstra's weakest preconditions $wp(S, R)$

Vienna Development Method (VDM)

Z and B

temporal logic $\Box$ $\Diamond$

process algebras (CSP, CCS, mu-calculus, pi-calculus, ...)

event traces, interleaved histories

model checking

- exhaustive automated testing
- up to 10^{60} states $\approx$ 2^{200} states $\approx$ 200 bits $\approx$ 6 variables
- abstraction, proof (not automated)

this theory

simpler

- just boolean expressions

more general

- includes terminating and nonterminating computation
- includes sequential and parallel computation
- includes stand-alone and interactive computation
- includes time and space bounds and real time
- includes probabilistic computations

prerequisites

basic boolean algebra (true, false, not, and, or)

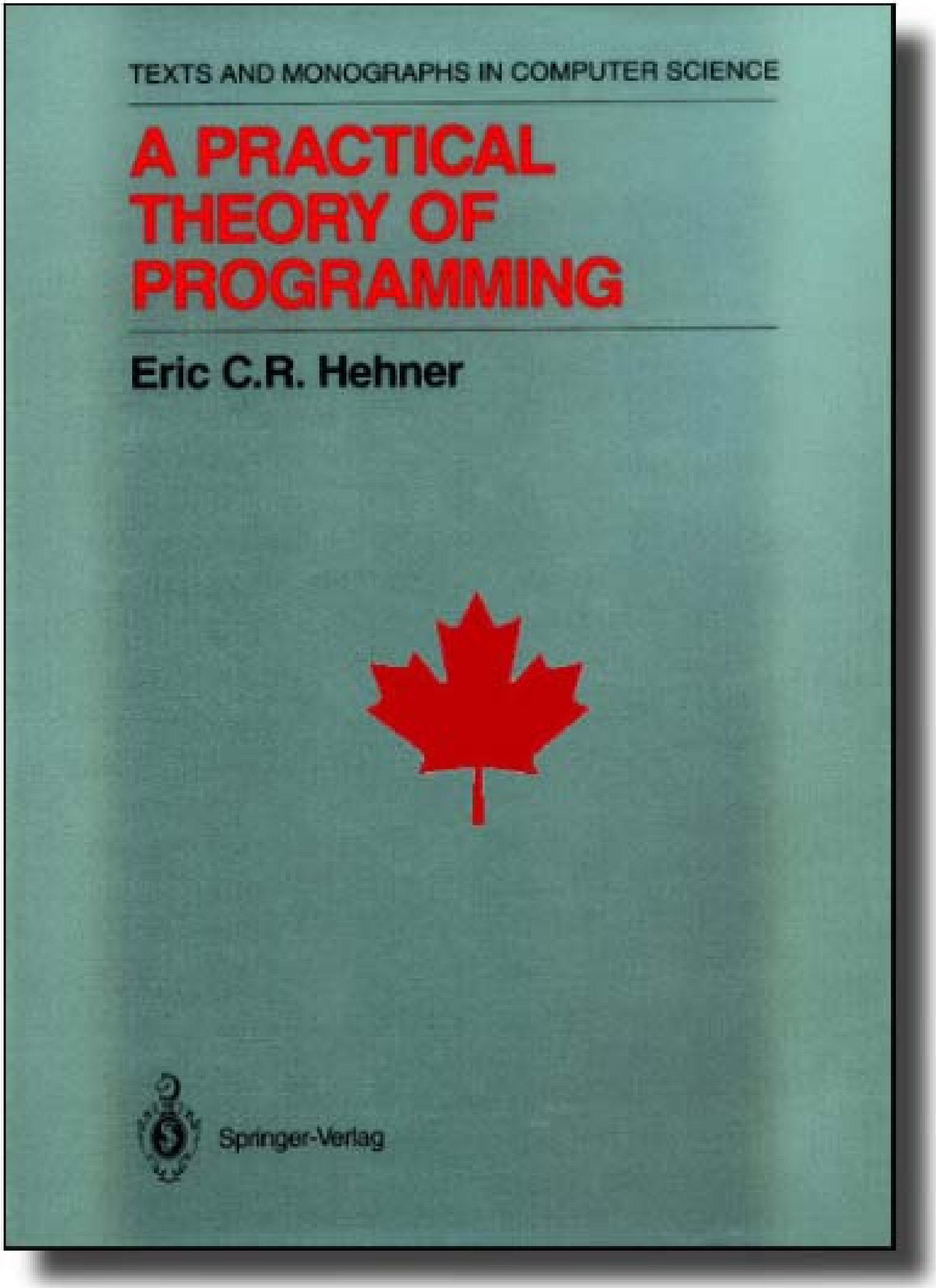
intermediate programming, any language

assignment statement, if statement

reading

C.A.R.Hoare, J.Misra: *Verified Software: Theories, Tools, Experiments*

www.cs.utoronto.ca/~hehner/vstte-hoare-misra.pdf



TEXTS AND MONOGRAPHS IN COMPUTER SCIENCE

A PRACTICAL THEORY OF PROGRAMMING

Eric C.R. Hehner

Springer-Verlag

TEXTBOOK

available

FREE

at

www.cs.utoronto.ca/~hehner