1.

Record Nr.	UNINA9910141494003321
Autore	Marlet Renaud
Titolo	Program specialization [[electronic resource] /] / Renaud Marlet
Pubbl/distr/stampa	Hoboken, N.J., : John Wiley & Sons, Inc., 2013
ISBN	1-118-57698-5 1-299-14643-0 1-118-57700-0 1-118-57686-1
Descrizione fisica	1 online resource (560 p.)
Collana	Computer engineering series
Disciplina	005.1
Soggetti	Program transformation (Computer programming)
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	Description based upon print version of record.
Nota di bibliografia	Includes bibliographical references and index.
Nota di contenuto	Title Page; Contents; Chapter 1. Main Principles of Program Specialization; 1.1. Specialized program; 1.1.1. Program specialization; 1.1.2. Context of specialization; 1.1.3. Specialization of a fragment of program; 1.1.4. Partial computations; 1.1.5. Range of specializations; 1.1.6. Equivalence between the specialized program and the generic program; 1.2. Specializing to improve performance; 1.2.1. Execution time; 1.2.2. Memory space; 1.2.3. Effect of the compiler; 1.2.4. Opacity of the code generated; 1.2.5. Effect of the memory cache; 1.3. Automatic specialization; 1.3.1. Specializer 1.3.2. Operation of specialization1.3.3. Execution times; 1.3.4. Advantages and disadvantages to automatic specialization; 1.4. Main applications of specialization; 1.4.1. Application 1: compiling using an interpreter; 1.4.2. Application 2: transforming an interpreter into a compiler; 1.4.3. Application 3: creating a compiler generator; 1.5. Specialization times; 1.5.1. Compile-time specialization; 1.5.2. Runtime specialization; 1.5.3. Specialization server; 1.5.4. Specialized code cache; 1.6. Financial viability of specialization; 1.6.1. Specialization gain; 1.6.2. Specialization time 1.6.3. Size of the specializer1.6.4. Specialization before execution; 1.6.5. Runtime specialization and break-even point; Chapter 2. Specialization Techniques; 2.1. Transforming specialization programs;

2.1.1. Partial evaluation; 2.1.2. Specialization strategies; 2.1.3. Formulation of specialization using general transformations; 2.1.4. Formulation of specialization using ad hoc transformations; 2.1.5. Techniques for executing precomputations; 2.1.6. Speculative specialization; 2.1.7. Interprocedural specialization; 2.1.8. Polyvariant specialization; 2.2. Termination of specialization

2.2.1. Online control2.2.2. Offline control; 2.3. Correctness of specialization; 2.3.1. Soundness, completeness and correctness; 2.3.2. Remedying laziness; 2.3.3. Execution error handling; 2.3.4. Portability; 2.3.5. Pre-processor; 2.4. Other forms of specialization; 2.4.1. Driving and supercompilation; 2.4.2. Generalized partial computation; 2.4.3. Configurable partial computation; 2.4.4. Program slicing; 2.4.5. Comparison with a compiler; 2.4.6. Comparison with a multilevel language; Chapter 3. Offline Specialization; 3.1. Main principles of offline specialization

3.1.1. Specification of input binding times3.1.2. Binding-time analysis; 3.1.3. Specialization by binding-time interpretation; 3.1.4. Action analysis; 3.1.5. Specialization by action interpretation; 3.1.6. Generating extension; 3.1.7. Compiler generator; 3.1.8. Generation of a specialized program; 3.1.9. Offline specializer; 3.1.10. Correction of offline specialization; 3.1.11. Specialization grammar; 3.1.12. Polyvariant offline specialization; 3.2. Compared advantages of offline specialization; 3.2.1. Evaluation a priori of the specialization degree

3.2.2. Visualization of specialization information

Sommario/riassunto

This book presents the principles and techniques of program specialization - a general method to make programs faster (and possibly smaller) when some inputs can be known in advance. As an illustration, it describes the architecture of Tempo, an offline program specializer for C that can also specialize code at runtime, and provides figures for concrete applications in various domains. Technical details address issues related to program analysis precision, value reification, incomplete program specialization, strategies to exploit specialized program, incremental specialization, and data speci