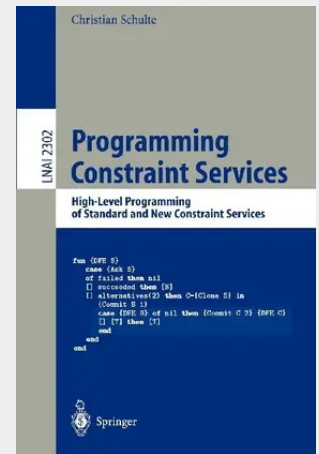


Programming Constraint Services

High-Level Programming of Standard and New Constraint Services

Constraint Programming is an approach for modeling and solving combinatorial problems that has proven successful in many applications. It builds on techniques developed in Artificial Intelligence, Logic Programming, and Operations Research. Key techniques are constraint propagation and heuristic search. Constraint Programming is based on an abstraction that decomposes a problem solver into a reusable constraint engine and a declarative program modeling the problem. The constraint engine implements the required propagation and search algorithms. It can be realized as a library for a general purpose programming language (e.g. C++), as an extension of an existing language (e.g. Prolog), or as a system with its own dedicated language. The present book is concerned with the architecture and implementation of constraint engines. It presents a new, concurrent architecture that is far superior to the sequential architecture underlying Prolog. The new architecture is based on concurrent search with copying and recomputation rather than sequential search with trailing and backtracking. One advantage of the concurrent approach is that it accommodates any search strategy. Furthermore, it considerably simplifies the implementation of constraint propagation algorithms since it eliminates the need to account for trailing and backtracking. The book investigates an expressive generalization of the concurrent architecture that accommodates propagation-preserving combinators (known as deep guard combinators) for negation, disjunction, implication, and reification of constraint propagators. Such combinators are beyond the scope of Prolog's technology. In the concurrent approach they can be obtained with a reflective encapsulation primitive.

Constraint Programming is an approach for modeling and solving combinatorial problems that has proven successful in many applications. It builds on techniques developed in Artificial Intelligence, Logic Programming, and Operations Research. Key techniques are constraint propagation and heuristic search. Constraint Programming is based on an abstraction that decomposes a problem solver into a reusable constraint engine and a declarative program modeling the problem. The constraint engine implements the required propagation and search algorithms. It can be realized as a library for a general purpose programming language (e.g. C++), as an extension of an existing language (e.g. Prolog), or as a system with its own dedicated language. The present book is concerned with the architecture and implementation of constraint engines. It presents a new, concurrent architecture that is far superior to the sequential architecture underlying Prolog. The new architecture is based on concurrent search with copying and recomputation rather than sequential search with trailing and backtracking. One advantage of the concurrent approach is that it accommodates any search strategy. Furthermore, it considerably simplifies the implementation of constraint propagation algorithms since it eliminates the need to account for trailing and backtracking. The book investigates an expressive generalization of the concurrent architecture that accommodates propagation-preserving combinators (known as deep guard combinators) for negation, disjunction, implication, and reification of constraint propagators. Such combinators are beyond the scope of Prolog's technology. In the concurrent approach they can be obtained with a reflective encapsulation primitive.



53,49 €

49,99 € (zzgl. MwSt.)

Lieferfrist: bis zu 10 Tage

Artikelnummer: 9783540433712

Medium: Buch

ISBN: 978-3-540-43371-2

Verlag: Springer

Erscheinungstermin: 20.03.2002

Sprache(n): Englisch

Auflage: 1. Auflage 2002

Serie: Lecture Notes in Artificial Intelligence

Produktform: Kartoniert

Gewicht: 295 g

Seiten: 176

Format (B x H): 155 x 235 mm

