

Cambridge University Press  
0521825830 - Principles of Constraint Programming  
Krzysztof R. Apt  
Frontmatter  
[More information](#)

---

# *Principles of Constraint Programming*

Krzysztof R. Apt

CWI, Amsterdam, The Netherlands



Cambridge University Press  
0521825830 - Principles of Constraint Programming  
Krzysztof R. Apt  
Frontmatter  
[More information](#)

---

PUBLISHED BY THE PRESS SYNDICATE OF THE UNIVERSITY OF CAMBRIDGE  
The Pitt Building, Trumpington Street, Cambridge, United Kingdom

CAMBRIDGE UNIVERSITY PRESS  
The Edinburgh Building, Cambridge CB2 2RU, UK  
40 West 20th Street, New York, NY 10011-4211, USA  
477 Williamstown Road, Port Melbourne, VIC 3207, Australia  
Ruiz de Alarcón 13, 28014 Madrid, Spain  
Dock House, The Waterfront, Cape Town 8001, South Africa

<http://www.cambridge.org>

© Cambridge University Press 2003

This book is in copyright. Subject to statutory exception  
and to the provisions of relevant collective licensing agreements,  
no reproduction of any part may take place without  
the written permission of Cambridge University Press.

First published 2003

Printed in the United Kingdom at the University Press, Cambridge

*Typeface* Computer Modern 10/13pt *System* L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>

*A catalogue record for this book is available from the British Library*

ISBN 0 521 82583 0 hardback

Cambridge University Press  
0521825830 - Principles of Constraint Programming  
Krzysztof R. Apt  
Frontmatter  
[More information](#)

---

*To Alma and Daniel*

---

## Contents

<i>Acknowledgements</i>	<i>page xi</i>
<b>1</b> <i>Introduction</i>	1
1.1 Basic characteristics of constraint programming	1
1.2 Applications of constraint programming	3
1.3 A very short history of the subject	5
1.4 Our approach	6
1.5 Organisation of the book	6
<b>2</b> <i>Constraint satisfaction problems: examples</i>	8
2.1 Basic concepts	9
2.2 Constraint satisfaction problems on integers	11
2.3 Constraint satisfaction problems on reals	16
2.4 Boolean constraint satisfaction problems	19
2.5 Symbolic constraint satisfaction problems	21
2.6 Constrained optimization problems	43
2.7 Summary	47
2.8 Exercises	48
2.9 Bibliographic remarks	51
2.10 References	52
<b>3</b> <i>Constraint programming in a nutshell</i>	54
3.1 Equivalence of CSPs	55
3.2 Basic framework for constraint programming	58
3.2.1 PREPROCESS	59
3.2.2 HAPPY	60
3.2.3 ATOMIC	61
3.2.4 SPLIT	61
3.2.5 PROCEED BY CASES	64
3.2.6 CONSTRAINT PROPAGATION	66

vi	<i>Contents</i>	
3.2.7	Constraint propagation algorithms	70
3.3	Example: Boolean constraints	71
3.4	Example: polynomial constraints on integer intervals	74
3.5	Summary	80
3.6	Bibliographic remarks	81
<b>4</b>	<b><i>Some complete constraint solvers</i></b>	<b>82</b>
4.1	A proof theoretical framework	83
4.1.1	Proof rules	84
4.1.2	Derivations	87
4.2	Term equations	92
4.2.1	Terms	93
4.2.2	Substitutions	94
4.2.3	Unifiers and mgus	95
4.2.4	Unification problem and solving of CSPs	98
4.2.5	The <i>UNIF</i> proof system	99
4.2.6	The MARTELLI–MONTANARI algorithm	103
4.3	Linear equations over reals	107
4.3.1	Linear expressions and linear equations	107
4.3.2	Substitutions, unifiers and mgus	110
4.3.3	Linear equations and CSPs	111
4.3.4	The <i>LIN</i> proof system	112
4.3.5	The GAUSS–JORDAN ELIMINATION algorithm	115
4.3.6	The GAUSSIAN ELIMINATION algorithm	118
4.4	Linear inequalities over reals	121
4.4.1	Syntax	121
4.4.2	Linear inequalities and CSPs	122
4.4.3	The <i>INEQ</i> proof system	123
4.4.4	The FOURIER–MOTZKIN ELIMINATION algorithm	124
4.5	Summary	131
4.6	Exercises	131
4.7	Bibliographic remarks	132
4.8	References	133
<b>5</b>	<b><i>Local consistency notions</i></b>	<b>135</b>
5.1	Node consistency	136
5.2	Arc consistency	138
5.3	Hyper-arc consistency	143
5.4	Directional arc consistency	144
5.5	Path consistency	147
5.6	Directional path consistency	155

*Contents*

vii

5.7	<i>k</i> -consistency	157
5.8	Strong <i>k</i> -consistency	164
5.9	Relational consistency	166
5.10	Graphs and CSPs	170
5.11	Summary	175
5.12	Exercises	175
5.13	Bibliographic remarks	176
5.14	References	176
<b>6</b>	<b><i>Some incomplete constraint solvers</i></b>	<b>178</b>
6.1	A useful lemma	180
6.2	Equality and disequality constraints	181
6.3	Boolean constraints	184
6.3.1	Transformation rules	185
6.3.2	Domain reduction rules	186
6.3.3	Example: full adder circuit	188
6.3.4	A characterisation of the system <i>BOOL</i>	191
6.4	Linear constraints on integer intervals	192
6.4.1	Domain reduction rules for inequality constraints	194
6.4.2	Domain reduction rules for equality constraints	196
6.4.3	Rules for disequality constraints	199
6.4.4	Rules for strict inequality constraints	200
6.4.5	Shifting from intervals to finite domains	200
6.4.6	Example: the <i>SEND + MORE = MONEY</i> puzzle	201
6.4.7	Bounds consistency	202
6.4.8	A characterisation of the <i>LINEAR EQUALITY</i> rule	206
6.5	Arithmetic constraints on integer intervals	211
6.5.1	Domain reduction rules: first approach	211
6.5.2	Domain reduction rules: second approach	213
6.5.3	Domain reduction rules: third approach	217
6.5.4	Implementation of the third approach	221
6.5.5	Shifting from intervals to finite domains	223
6.6	Arithmetic constraints on reals	224
6.6.1	Interval arithmetic	226
6.6.2	Domain reduction rules	227
6.6.3	Implementation issues	233
6.6.4	Using floating-point intervals	236
6.6.5	Correctness and efficiency issues	238
6.7	Arithmetic equations over reals	242
6.8	Summary	245

viii	<i>Contents</i>	
6.9	Exercises	245
6.10	Bibliographic remarks	248
6.11	References	251
<b>7</b>	<b><i>Constraint propagation algorithms</i></b>	<b>254</b>
7.1	Generic iteration algorithms	256
7.1.1	Iterations	256
7.1.2	Algorithms for arbitrary partial orderings	261
7.1.3	Algorithms for cartesian products of partial orderings	264
7.2	From partial orderings to CSPs	268
7.3	A node consistency algorithm	269
7.4	An arc consistency algorithm	271
7.5	A hyper-arc consistency algorithm	273
7.6	A directional arc consistency algorithm	275
7.7	A path consistency algorithm	277
7.8	A directional path consistency algorithm	281
7.9	A $k$ -consistency algorithm	283
7.10	A relational consistency algorithm	286
7.11	Implementations of incomplete constraint solvers	287
7.12	Summary	290
7.13	Exercises	291
7.14	Bibliographic remarks	295
7.15	References	297
<b>8</b>	<b><i>Search</i></b>	<b>299</b>
8.1	Search trees	301
8.2	Labeling trees	303
8.2.1	Complete labeling trees	304
8.2.2	Reduced labeling trees	308
8.2.3	<i>prop</i> labeling trees	310
8.3	An example: $SEND + MORE = MONEY$	313
8.4	Instances of <i>prop</i> labeling trees	315
8.4.1	Forward checking	315
8.4.2	Partial look ahead	319
8.4.3	Maintaining arc consistency (MAC)	321
8.5	Search algorithms for the labeling trees	324
8.5.1	Backtrack-free search	325
8.5.2	Backtrack-free search with constraint propagation	327
8.5.3	Backtracking	329
8.5.4	Backtracking with constraint propagation	330
8.6	Instances of backtracking with constraint propagation	332

*Contents*

ix

8.6.1 Forward checking	332
8.6.2 Partial look ahead	333
8.6.3 Maintaining arc consistency (MAC)	334
8.6.4 Searching for all solutions	335
8.7 Search algorithms for finite constrained optimization problems	335
8.7.1 Branch and bound	337
8.7.2 Branch and bound with constraint propagation	339
8.7.3 Branch and bound with constraint propagation and cost constraint	339
8.8 Heuristics for search algorithms	341
8.8.1 Variable selection	341
8.8.2 Value selection	343
8.9 An abstract branch and bound algorithm	344
8.10 Summary	347
8.11 Exercises	347
8.12 Bibliographic remarks	348
8.13 References	349
<b>9 <i>Issues in constraint programming</i></b>	<b>351</b>
9.1 Modeling	352
9.1.1 Choosing the right variables	352
9.1.2 Choosing the right constraints	353
9.1.3 Choosing the right representation	356
9.1.4 Global constraints	358
9.2 Constraint programming languages	359
9.2.1 Constraint logic programming	360
9.2.2 ILOG solver	362
9.2.3 Generation of constraints	363
9.3 Constraint propagation	364
9.4 Constraint solvers	366
9.4.1 Building constraint solvers	366
9.4.2 Incrementality	367
9.4.3 Simplification of constraints	368
9.5 Search	369
9.5.1 Search in modeling languages	369
9.5.2 Depth-first search: backtracking and branch and bound	370
9.5.3 Breadth-first search and limited discrepancy search	371
9.5.4 Local search	372
9.5.5 Search in constraint programming languages	375
9.5.6 Biology-inspired approaches	378



x	<i>Contents</i>	
9.6	Over-constrained problems	379
9.6.1	Partial, weighted and fuzzy CSPs	380
9.6.2	Constraint hierarchies	381
9.6.3	Generalisations	383
9.6.4	Reified constraints	383
9.7	Summary	384
9.8	Bibliographic remarks	384
	<i>Bibliography</i>	387
	<i>Author index</i>	401
	<i>Subject index</i>	404

---

## Acknowledgements

I took to constraint programming late. In fact, I can be more precise. In the summer of 1995, when travelling to a conference, I stopped by at Joxan and Jennifer Jaffar's house in upstate New York. Joxan was working then at the IBM Research Heights and they were moving to Singapore. I still recall the half packed boxes in their hall and a pleasant evening spent on their terrace. I was discussing with Joxan research on constraint logic programming and was confused at that time that constraint programming is the same as constraint logic programming. But he showed me the book Tsang [1993]. Most of this material was new to me, though some isolated concepts I encountered earlier in the most useful book Van Hentenryck [1989].

One year later I thought I should give a course on the subject to learn it better. Providentially, in December 1996, during my visit to Israel, Rachel Ben Eliyahu from the University of Beer-sheva 'sent me over' to Rina Dechter who was staying then on sabbatical at the Tel-Aviv University. A two hour discussion with Rina helped me enormously. Moreover, over the next few months Rina patiently replied to all my, often stupid, questions and provided me with most useful pointers to the literature. This helped me to enter this subject and to start doing research on it.

In the course of the next five years I learned a lot on constraint programming by talking to many colleagues. In addition to those mentioned above I recall in particular illuminating conversations on this subject with Maarten van Emden (during his visits at CWI and my visit at the University of Victoria, Canada, in the summer of 1999), with Andrea Schaerf (during his visits at CWI in the period 1996–1998), François Fages (during the POPL '97 conference in Paris), with, unfortunately deceased in 2000, Igor Shetsov (in the bus to the hotel during the CP '97 conference and later, in 2000, in Amsterdam), François Puget (during a flight from Linz to Frankfurt after the CP '97 conference), Philippe Codognet (during his two short visits in

Amsterdam in, I believe, 1996 and 1999), Alexander Semenov (during his visit to Amsterdam in 2001), and last, but not least, Eric Monfroy, who worked at CWI in Amsterdam from 1996 until 2000.

In 2003 the circle closed, since I finished this book while visiting the National University of Singapore, where Joxan is now Dean of the School of Computing. While in Singapore, I also profited from useful conversations on constraint programming with Martin Henz and Roland Yap. Martin kindly provided me with the tests summarised in Table 9.1 in Chapter 9, obtained using the Oz Explorer. I would like to thank here all those mentioned above for their help in my understanding of the subject.

In addition, I appreciated most helpful comments, corrections and suggestions on various parts of the manuscript sent to me by Sebastian Brand, Maarten van Emden, Pierre Flener, Rosella Gennari, Martin Henz, Willem Jan van Hoeve, Eric Monfroy, Zsofi Ruttkay, Kees Vermeulen and Peter Zoetewij. Also, I have greatly benefited from giving a course based on this material at the School of Computing of the National University of Singapore. Several students in my course, namely Lim Min Kwang, Cui Hang, Wen Conghua, Dong Xiaolan, Ma Lin, Xu Cheng and Damith Chatura Rajapakse provided useful comments. I also taught on this subject for a couple of years at the University of Amsterdam. Strangely, the only comment I ever got on the continuously growing manuscript was from a student who found that I should replace somewhere in the text ‘We now get’ by ‘We get now’.<sup>1</sup>

Finally, Liu Jiang Hong kindly prepared the figures using the Dia drawing program. Nicola Vitacolonna modified the `skak` package in a way that allowed me to take care of the chess board drawings in Chapters 8 and 9 and Hugh Anderson helped me to insert them in the text as `.epsi` files. Also it is a pleasure to acknowledge the help I received from David Tranah of the Cambridge University Press throughout the production of this book.

And last, but not least, I would like to thank my family, Ruth, Alma and Daniel, for their patience with me during writing this book. Alma (now 7) kindly offered to proofread the book but fortunately it’s not needed: this book no errors. Daniel (now 10) got interested in the Dia program. Who knows: perhaps he will be willing to make drawings for my next book?

PS. A comment about the cover: Alma likes penguins very much.

<sup>1</sup> Fortunately, worse comments can happen. Peter Medawar, a recipient of the Nobel Prize in Physiology or Medicine in 1960 mentions in his memoirs that he gave once a lecture during an open day at his University and the only question he got was ‘Where is the women’s toilet?’