Foundations of Software Technology and Theoretical Computer Science

Lecture Notes in Computer Science

Edited by G. Goos and J. Hartmanis

472

K.V. Nori C.E. Veni Madhavan (Eds.)

FST and TCS 10

Foundations of Software Technology and Theoretical Computer Science

Tenth Conference, Bangalore, India
December 17–19, 1990
Proceedings

Springer-Verlag
Berlin Heidelberg New York London
Paris Tokyo Hong Kong Barcelona

Volume Editors
Kesav V. Nori
Tata Research Development and Design Centre
1, Mangaldas Road, Pune 411 001, India

C. E. Veni Madhavan
Department of Computer Science and Automation
Indian Institute of Science
Bangalore 560 012, India

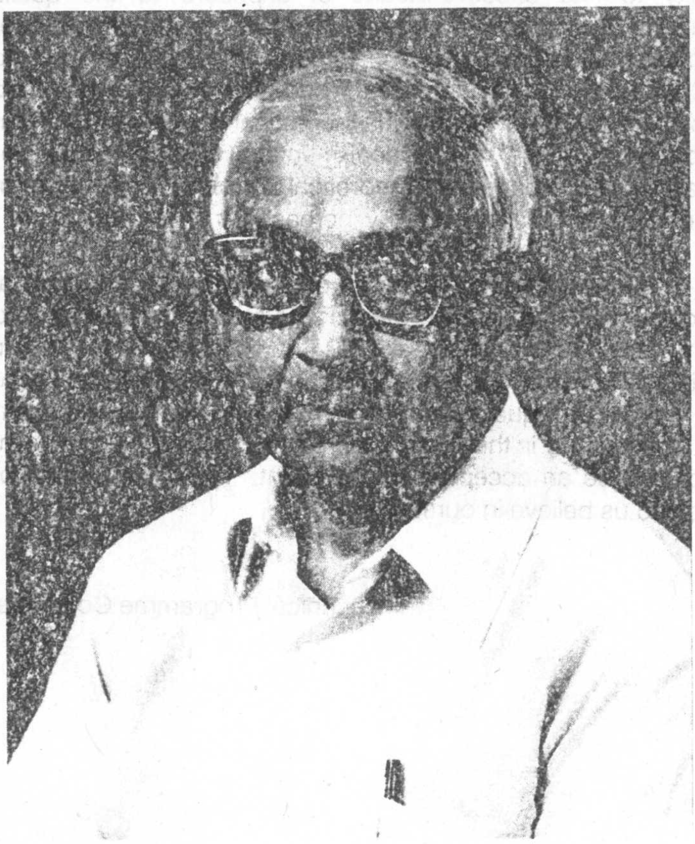
Conference Sponsors
Centre for Development of Advanced Computing
Computer Maintenance Corporation
Indian Institute of Science
Tata Consultancy Services
Tata Institute of Fundamental Research
Tata Research Development and Design Centre

CR Subject Classification (1987): B.7.2, F.2.2, F.3.1–2, F.4.1–3, G.2.1–2, H.2.4, I.2.2–4, I.2.7, I.2.10, I.3.5

ISBN 3-540-53487-3 Springer-Verlag Berlin Heidelberg New York
ISBN 0-387-53487-3 Springer-Verlag New York Berlin Heidelberg

Printed in Germany

Printing and binding: Druckhaus Beltz, Hemsbach/Bergstr.
2145/3140-543210 – Printed on acid-free paper



PROFESSOR R. NARASIMHAN

DEDICATION

There are many reasons to celebrate in this tenth year of FST&TCS conferences. This conference is organised and run by the Computer Science Research community in India without support from any professional societies. It is an expression of solidarity of this community to find effective means of exposure to the quality of research elsewhere, and to be visible on account of merit. Its purpose has been to provide a forum for professional interaction between members of this research community and their counterparts in different parts of the world. The effects of this sustained exposure is locally visible: the quality of teaching in theoretical aspects of Computer Science has uniformly improved in all the major Departments of Computer Science in India; also, the number and quality of papers in Theoretical Computer Science reporting on original research carried out in India has steadily increased.

The Tenth Conference on Foundations of Software Technology and Theoretical Computer Science is dedicated to Professor R. Narasimhan on the eve of his retirement from the Computer Science Group at TIFR. It was his vision of finding means to integrate individual researchers into a community, his emphasis on quality, his ideas on methods of maintaining quality in a visible manner, and his constant support to the Conference Committees in their endeavours to implement his ideas, which have helped make this conference an accepted annual event. We will always be beholden to him for having helped us believe in ourselves.

Technical Programme Committee, FST&TCS 10

ACKNOWLEDGEMENTS

We owe a debt of gratitude to many who have contributed to this conference:

- to all the authors of the 116 submitted papers for providing us with delectable material;
- to all the authors of the 26 selected papers for giving us an excellent programme;
- to all the reviewers for their discerning comments;
- to all Invited Speakers for accepting our invitations;
- to Patricia Lobo, Deenaz Bulsara, A. M. Farooqi, P. R. Naik, and Y. Pajnigar, S. Shelar, C. Kumbhar for efficiently managing the Conference Secretariat;
- to Faculty and Students of the Department of Computer Science and Automation of IISc for providing organisational support;
- to the funding agencies for their encouragement and support.

December, 1990

K. V. Nori
C. E. Veni Madhavan

Conference Advisory Committee

D Bjorner (Denmark)

A Chandra (IBM Resarch)

S Crespi Reghizzi (Milan)

Z Galil (Columbia)

D Gries (Cornell)

M Joseph (Warwick)

A Joshi (Pennsylvania)

R Kannan (CMU)

U Montanari (Pisa)

R Narasimhan (TIFR)

M Nivat (Paris)

R Parikh (New York)

S Rao Kosaraju (Johns Hopkins)

S Sahni (Minnesota)

W A Wulf (Virginia)

Technical Programme Committee

G P Bhattacharjee (IIT Kharagpur)

S Biswas (IIT Kanpur)

K Krithivasan (IIT Madras)

S N Maheshwari (IIT Delhi)

K V Nori (TRDDC Pune)

R K Shyamasundar (TIFR)

Y N Srikant (IISc Bangalore)

P S Thiagarajan (SPIC Foundation)

C E Veni Madhavan (IISc Bangalore)

G Venkatesh (IIT Bombay)

LIST OF REVIEWERS

G S Adhar, Univ. of North Carolina, USA
W R Adrion, Univ. of Massachusetts, USA
A Aggarwal, IBM-T J Watson Res Ctr, USA
A Aggarwal, Univ. of Maryland, USA
R K Ahuja, IIT Kanpur, India
M Ahuja, Ohio State Univ., USA
J Anderson, Comp Resources Int'l, Denmark
A Arora, MCTC, USA
K Bagchi, Aalborg Univ., Denmark
S Balaji, ISRO B'lore, India
S Banerjee, IISc B'lore, India
R Barua, ISI Calcutta, India
G Barua, IIT Kanpur, India
K Berkling, Syracuse Univ., USA
S Bettayeb, Louisiana State Univ., USA
P Bonatti, Corso Italia, Pisa, Italy
E Bevers, Univ. of Leuven, Belgium
B Bhargava, Purdue Univ., USA
P C P Bhatt, IIT Delhi, India
B K Bhattacharya, Simon Fraser Univ., Canada
B B Bhattacharya, ISI Calcutta, India
S Biswas, IIT Kanpur, India
D Bjorner, Tech Univ. of Denmark, Denmark
J R S Blair, Univ. of Tennessee, USA
J Camilleri, Computel Labs, Cambridge, UK
D M Campbell, Brigham Young Univ., USA
D L Carver, Louisiana State Univ., USA
J Case, Univ. of Delaware, USA
M Cerioli, Univ. of Genoa, Italy
D R Chand, Framingham, MA, USA
V Chandru, IISc B'lore, India
S Chaudhari, Rutgers NJ, USA
J Cheriyan, Univ. of Saarland, FRG
B Courcelle, France
K Culik II, Univ. of South Carolina, USA
D D'Souza, R M I T, Australia
P Dasgupta, Georgia Inst. of Tech., USA
A Datta, IIT Madras, India
R W Dawes, Queen's Univ., Canada
A Dawar, Univ. of Pennsylvania, USA
P J de Bruin, Univ. of Saarland, FRG
W P de Roever, Kiel Univ., FRG
A Deb, Memorial Univ., Canada
N C Debnath, Winona State Univ., USA
S K Debray, Univ. of Arizona, USA
A H Dekker, Griffith Univ., Australia
N Deo, Univ. of Central Florida, USA
J S Deogun, Univ. of Nebraska-Lincoln, USA
S C Desarkar, IIT Kharagpur, India
J Desharnais, Univ. Laval, Canada
T K Dey, Purdue Univ., USA
A A Diwan, TIFR Bombay, India
Dubashi, Cornell Univ., USA
P Dublish, Univ. of Hildesheim, FRG
R Dwyer, St. Univ. of North Carolina, USA
W Fey, Univ. of Berlin, FRG
J Fiadeiro, Imperial College, UK
M. Gabbrielli, Univ. of Pisa, Italy
G Gambosi, IASI-CNR, Italy
A V Ganapati, IISc B'lore, India
M Garzon, Memphis State Univ., USA
E V Gestel Univ. of Leuven, Belgium
S K Ghosh, TIFR Bombay, India
D Goelman, Villanova Univ., USA
Md Gouda, MCTC, USA
R Greenlaw, College of Eng., Durham, NH, USA
J Gunawardena, Hewlett-Packard Labs, UK
R Gupta, Simon Fraser Univ., Canada
K Havelund, Tech Univ. of Denmark, Denmark
A Haxthauser, Louisiana State Univ., USA
L S Heath, Virginia PI & State Univ., USA
H Heller, Siemens, FRG
M Hermann, CRIN, France
Z Hong, Nanjing Univ., PR China
G Hotz, Univ. of Saarland, FRG
W J Hsu, Michigan State Univ., USA
S S Iyengar, Louisiana State Univ., USA
S Jain, Univ. of Rochester, USA
R Janardan, Univ. of Minnesota, USA
J M Janas, Bundeswehr Univ. Munich, FRG
L S Jensen, Univ. of Denmark, Denmark
Y J Jiang, Imperial College, UK
M Joseph, Univ. of Warwick, UK
A K Joshi, Univ. of Pennsylvania, USA
J Kamper, IBM, FRG
K Kanchanasut, Univ. of Melbourne, Australia
R Kannan, CMU, USA
S Kapoor, IIT Delhi, India
H Karnick, IIT Kanpur, India
V N Kasyanov, USSR Academy of Sc., USSR
V B Kaujalgi, IIM B'lore, India
C Kenyon, Laboratoire d'Informatique, France
S Khuller, Cornell Univ., USA
C M R Kintala, AT & T Bell Labs, USA
S Krishnamoorthy, Georgia Tech, USA
K Krithivasan, IIT Madras, India
S Kumar, IIT Kanpur, India
A Kumar, IIT Delhi, India

V S Lakshmanan, IISc B'lore, India
C Levcopoulos, Lund Univ., Sweden
G Levi, Corso Italia, Pisa, Italy
C P Lewington, Univ. of Essex, UK
A Lingas, Lund Univ., Sweden
K Lodaya, Matscience Madras, India
H H Lovengreen, Univ. of Denmark, Denmark
S M Mahajan, BARC Bombay, India
S Mahajan, Simon Fraser Univ., Canada
A Mahanti, Univ. of Maryland, USA
M Maher, IBM-TJ Watson Res Ctr, USA
S N Maheshwari, IIT Delhi
A K Majumdar, IIT Kharagpur
F M Malvestuto, Studi-DOC, ENEA Italy
A Marchetti, Rome, Italy
Mehndiratta, IIT Bombay, India
A K Mittal, IIT Kanpur, India
C K Mohan, New York, USA
S Morasca, Milan Polytechnic, Italy
A Moslemie, Univ. of Helsinki, Finland
R Motwani, Stanford Univ., USA
I S Mumick, Stanford Univ. USA
D V J Murphy, Univ. of Glasgow, UK
C R Muthukrishnan, IIT Madras, India
K N R Nair, Univ. of Poona, India
A Nakamura, Hiroshima Univ., Japan
N K Nanda, AIT Bangkok, Thailand
G Narasimhan, Memphis State Univ., USA
K V Nori, TRDDC Pune, India
L Narayan, Univ. of Rochester, USA
S Olariu, Old Dominion Univ., USA
B J Oommen, Carleton Univ., USA
A K Pal, IIM Calcutta, India
S P Pal, IISc B'lore, India
M Palis, Univ. of Pennsylvania, USA
C P Rangan, IIT Madras, India
G E Pantziou, Computer Tech. Inst., Greece
M A Papalaskari, Villanova Univ., USA
J Paredaens, Univ. of Antwerp, Belgium
A Paulraj, C-DAC, B'lore, India
A Pettorosi, IASI-CNR, Italy
M Pezze, Milan Polytechnic, Italy
I Phillips, Imperial College, UK
B C Pierce, IBM-TJ Watson Res Ctr, USA
B Plateau, INPG, France
S K Prasad, Univ. of Central Florida, USA
M Protasi, Univ. of Rome, Italy
S Purushothaman, Penn. State Univ., USA
P Raghavan, IBM-TJ Watson Res Ctr, USA
C S Raghavendra, USC, USA
V Rajasekar, IISc B'lore, India
S Rajasekaran, Univ. of Penn., USA
K S Rajasethupathy, SUNY-Brockport, USA
S Ramakrishnan, Bowling Green Univ., USA
R Raman, Univ. of Rochester, USA
R Ramanujam, IMSc Madras, India
S Ramesh, IIT Bombay, India
E Ritter, Univ. of Cambridge, UK
N S V Rao, Old Dominion Univ., USA
K S Rao, John Hopkins Univ., USA
S Rengarajan, IISc B'lore, India
C Retore, Imperial College, UK
K Ross, IBM-TJ Watson Res Ctr, USA
H V Sahasrabuddhe, Univ. of Poona, India
R Sangal, IIT Kanpur, India
A Saoudi, Univ. of Paris VII, France
H Saran, Univ. of California, USA
S Sarkar, IBM Corporation, USA
S Saxena, IIT Kanpur, India
B Schieber, IBM-TJ Watson Res. Ctr., USA
S Sen, Duke Univ., USA
A S Sethi, Univ. of Delaware, USA
S G Shiva, Univ. of Alabama, USA
S Singh, IIT Delhi, India
M Sinha, IIT Delhi, India
G Sivakumar, Univ. of Delaware, USA
A Somani, Univ. of Washington, USA
N Sopankar, Univ. of Texas, USA
N Soundarajan, Ohio State Univ., USA
P Spirakis, Computer Tech. Inst., Greece
P Sreenivasa Kumar, IISc B'lore, India
Y N Srikant, IISc B'lore, India
P K Srimani, Colorado State Univ., USA
M Subramaniam, TRDDC Pune, India
V S Sunderam, Emory Univ., USA
R Sunderraman, Wichita State Univ., USA
S Sur-Kolay, ISI Calcutta, India
R K Shyamasundar, TIFR Bombay, India
G Taubenfeld, Yale Univ., USA
P S Thiagarajan, SPIC Sc Foundn., India
P Tiwari, IBM-TJ Watson Res Ctr, USA
A Tyagi, Univ. of North Carolina, USA
R S Valiveti, Carleton Univ., USA
V Vazarani, Cornell Univ., USA
C E Veni Madhavan, IISc B'lore, India
S Venkatesan, Univ. of Texas, USA
G Venkatesh, IIT Bombay, India
K Vidyasankar, Memorial Univ., USA
V Vinay, IISc B'lore, India
D S L Wei, Univ. of Pennsylvania, USA
R Wilkinson, RMIT, Australia
M F Worboys, Univ. of Leicester, UK
R Yerneni, Tandem Computers, USA
J H You, Univ. of Alberta, Canada
C D Zaroliagis, CTI, Greece
E Zucca, Univ. of Genoa, Italy

TABLE OF CONTENTS

Reasoning About Linear Constraints Using Parametric Queries

Tien Huynh Leo Joskowicz Catherine Lassez Jean-Louis Lassez
IBM T.J. Watson Research Center
P.O.Box 704
Yorktown Heights, NY 10598

Abstract

We address the problem of building intelligent query systems to reason about linear arithmetic constraints. The central issue is the development of tools for testing solvability, for constraints representation, for incremental updates and for intelligent feedback. The concept of parametric queries introduced in the context of constraint logic programming provides the starting point for this study. The relevance of this approach is illustrated by examples from the domain of spatial reasoning.

1 Introduction

Linear arithmetic constraints are key elements in applications such as Operations Research, Constructive Solid Geometry, Robotics, CAD/CAM, Spreadsheets, Model-based Reasoning, Theorem Proving and Program Verification. Constraints handling techniques have been incorporated in a number of programming systems including CLP($\Re$), CHIP, CAL, Prolog III, BNR-Prolog, Mathematica and Trilogy. More recently constraints have been introduced in committed choice languages [M], [S] and in database querying languages [KKR]. Here we address the general problem of designing systems to reason about linear arithmetic using the constraints as the basic entity. The domain knowledge is expressed as constraints, and queries are asked on these constraints to extract new information. These queries are asked in an interactive context where the constraints in store can be dynamically added and deleted. Apart from the problem of finding suitable query-answering mechanisms, this raises the complex problem of coordinating the various operations.

To answer queries, efficient mechanisms are required to deal with subsumption, implicit equalities, redundancy, canonical representation, projections, and incrementally updating. For each of these problems, one can import algorithms from Symbolic Computation, Computational Geometry, Automated Reasoning and Operations Research. However, a collection of disconnected algorithms only makes an ad-hoc system. A general system requires a coherent underlying theory and an integrated implementation. In this paper, we informally present an integrated framework based on the concept of *parametric queries* which provides a unifying formalism and leads to a general query-answering method.

The rest of the paper is organized as follows. In the next section, we present examples of the use of constraints in the domain of spatial reasoning. These examples illustrate how the various kinds of information in this particular domain translate naturally into constraints and what the required constraint-handling operations are. In section 3 we define the general concept of parametric queries that provides a framework to express the different aspects of reasoning with constraints. We also show how a basic but somewhat ignored result of Linear Programming leads to a method to answer these queries. In section 4, we discuss the key issue of constraints representation and present a natural canonical form that fits the requirements of both the interaction and the processing of queries. In section 5, we address the problem of dynamically updating the constraints in store and we show the importance of proper feedback information for maintaining a coherent system of constraints. In the last section, we show how linear programming generalizes naturally into a symbolic computation method to answer parametric queries. As a particular case, we show how a variant of the dual simplex leads to a solver which provides, for free, information on the algebraic properties of the constraints and the geometric structure of the associated polyhedron.

2 Spatial reasoning: a case study

Spatial reasoning covers a wide spectrum of tasks and domains, such as planning the motion of a robot's arm, analyzing and designing machines, and solving commonsense problems. The fundamental question underlying these tasks is: given an assembly of rigid objects, determine how they move and interact with each other in space. Object geometry determines the motion properties of an assembly. For example, Figure 1 shows a simple puzzle consisting of an enclosing frame containing three identical red and blue blocks. Given that the blocks have opposite orientations, we want to determine if the blocks can move, how many blocks can move at a time or if the blue and red blocks can exchange positions.

Finding the motion properties of objects in an assembly can be formulated as a constraint

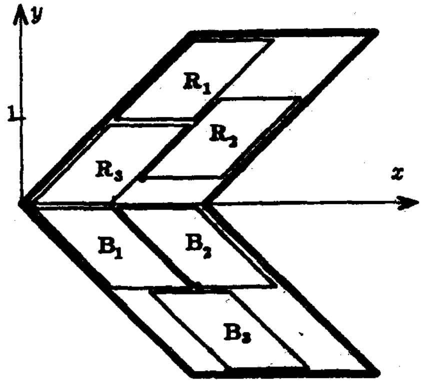


Figure 1: Geometric description of a puzzle consisting of a fixed frame containing three identical red R_i (upper) and blue (lower) B_i unit length diamond-shape blocks.

satisfaction problem [Ja]. Every moving object is assigned a reference point and one motion parameter for each of its potential degrees of freedom. Specific parameter values define the objects' *configuration*, i.e., its position and orientation in space. The space generated by all objects' motion parameters is called a *configuration space*. For example, each block in the puzzle has its reference point attached to its lower left corner and two translational parameters: x_i and y_i for the red blocks, and u_i and v_i for the blue blocks. The parameters' axes are aligned with the cartesian x-y coordinate frame and define a 12-dimensional configuration space. The configuration depicted in Figure 1 corresponds to the point:

$$\{x_1 = 1,\ y_1 = 1,\ x_2 = 1.5,\ y_2 = 0.5,\ x_3 = 0,\ y_3 = 0,$$
$$u_1 = 1,\ v_1 = -1,\ u_2 = 2,\ v_2 = -1,\ u_3 = 2.5,\ v_3 = -2\}.$$

Since rigid objects cannot deform or interpenetrate, their motions are constrained by the contacts between them. Each pairwise contact defines a *motion constraint* on the objects' degrees of freedom. Motion constraints are non-linear inequality functions on the objects' motion parameters. Their exact form is determined by the object geometry and degrees of freedom. Translating polyhedral objects generate linear constraints[1]. The set of motion constraints for each object pair is computed by examining all possible contacts between the objects' faces, edges, and vertices [LW].

In the puzzle, the frame restricts red blocks to slide along its edges. The corresponding motion constraints are graphically depicted in Figure 2(a). The axes are the motion parameters

[1]Other non-linear constraints of curved geometry and rotations can sometimes be approximated by linear inequalities [Jb].

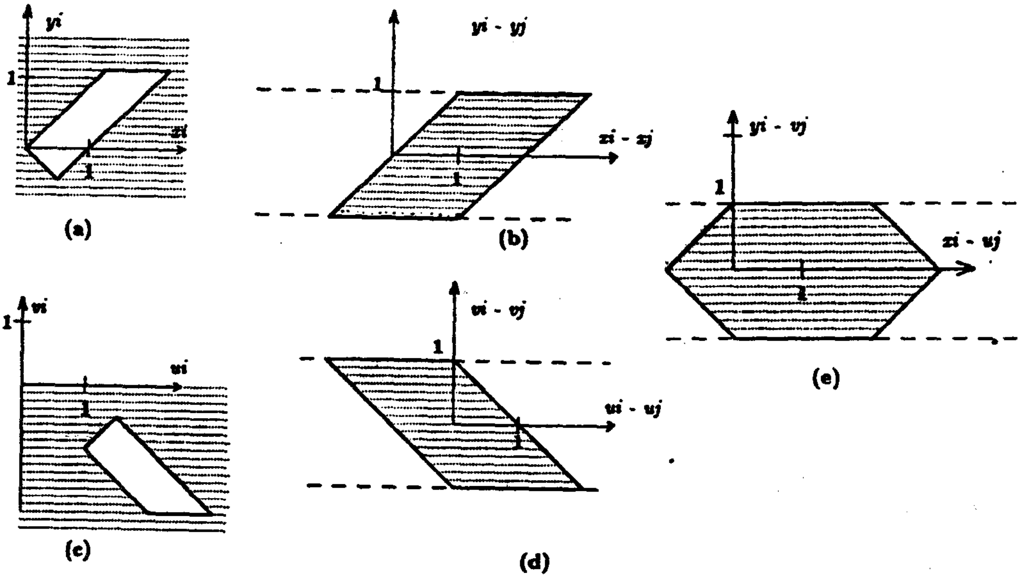


Figure 2: Graphical representation of the local motion constraints: (a) the frame and a red block; (b) two red blocks; (c) the frame and a blue block; (d) two blue blocks, and (e) a red and a blue block. Shaded areas correspond to forbidden object configurations.

of block R_i, x_i and y_i. Each of the trapezoid's boundaries corresponds to a contact: the left line corresponds to the red block's left edge sliding along the frame's upper left edge. The red blocks are constrained to slide around each others' contours, as shown in Figure 2(b). Motion constraints are described by disjunctions and conjunctions of linear constraints. Each conjunctive clause defines a convex region. The union of all conjunctive clauses defines the set of legal configurations. The frame and red block motion constraints are described by a single region:

$$\{y_i \leq 1, \quad x_i - y_i \geq 0, \quad x_i + y_i \geq 0, \quad x_i - y_i \leq 1\}$$

for $i = 1, 2, 3$, whereas the motion constraints of two red blocks R_i and R_j are described by four regions, partitioned as indicated by the dashed lines in the figure:

$$\begin{array}{llllll} \{(x_i - x_j) - (y_i - y_j) \geq 1, & y_i - y_j > -1, & y_i - y_j > -1\} & \vee & y_i - y_j \geq 1 & \vee \\ \{(x_i - x_j) - (y_i - y_j) \leq -1, & y_i - y_j > -1, & y_i - y_j > -1\} & \vee & y_i - y_j \leq -1 & \end{array}$$

for $i, j = 1, 2, 3$ and $i \neq j$. Similar sets of constraints describe the remaining three interactions, shown in Figure 2. Physical object motions must be within the pairwise object contacts. That is, they must satisfy the set of all pairwise motion constraints.

Properties of the objects' motions can be extracted from queries on the set S of all motion constraints. These questions include, among many others:

1. *Assembly:* can the objects be assembled? If not, which object must be removed?
2. *Freedom:* is an object fixed? How many degrees of freedom does it have? Can an object move in a fixed direction?
3. *Planning:* can we get from one assembly configuration to another?
4. *Editing:* what happens if the shape of an object is modified, or a new object is added? Do the assembly's motion properties change?

To determine if objects can be assembled, we test if S is solvable. A solvable S is a non-empty configuration space which contains at least one point corresponding to a feasible object configuration. If S is unsolvable, the constraints causing its unsolvability define the objects that interpenetrate. Removing one of these objects deletes the problematic constraints. To determine an objects' degrees of freedom, we find the values of its motion parameters. Each constant parameter removes a degree of freedom. When they are all constant, the object is fixed. To find the motion range, we project S on the space of the variable motion parameters. Equality relations of the type $\alpha x + \beta y = \gamma$ where x and y are the motion parameters of an object define an axis of motion. To determine if we can go from one configuration to another, we test if the corresponding configuration space points belong to the same connected region. Finally, to determine what happens as a consequence of a structural change, we must replace the motion constraints of the original object with the motion constraints of the new object. To test for assembly equivalence, we must test if their constraints are equivalent.

The set S generally contains many redundant inequalities and implicit equalities. Removing them and putting S in a canonical form directly answers many of the above questions. For example, the region of the puzzle's configuration space containing the configuration in Figure 1 is defined by the set S of 45 inequalities:

$$\bigcap_{i\neq j}\{y_i \le 1,\ x_i - y_i \ge 0,\ x_i + y_i \ge 0,\ x_i - y_i \le 1,$$
$$v_i \ge -2,\ u_i + v_i \le 1,\ u_i - v_i \ge 2,\ u_i + v_i \ge 0,\ y_i - v_j \ge 1\}$$
$$\cap \{y_1 - y_2 > -1,\ \ y_1 - y_2 < 1,\ \ (x_1 - x_2) - (y_1 - y_2) \le -1,$$
$$y_1 - y_3 \ge 1,$$
$$y_2 - y_3 > -1,\ \ y_2 - y_3 < 1,\ \ (x_2 - x_3) - (y_2 - y_3) \ge 1,$$
$$v_1 - v_2 > -1,\ \ v_1 - v_2 < 1,\ \ (u_1 - u_2) + (v_1 - v_2) \le -1,$$
$$v_1 - v_3 \ge 1,$$
$$v_2 - v_3 \ge 1\}$$

for $i, j = 1, 2, 3$. The canonical representation S' is much more compact and informative:

$$\begin{array}{llll}\{x_1 = 1, & y_1 = 1, & x_3 = 0, & y_3 = 0,\\ u_1 = 1, & v_1 = -1, & u_2 = 2, & v_2 = -1,\\ x_2 \le 2, & y_2 \ge 0, & x_2 - y_2 = 1, & \\ u_3 \ge 2, & u_3 \le 3, & v_3 = -2\} & \end{array}$$

These 10 equalities and 6 inequalities provide direct answers to some of the above queries. First, S is solvable and thus contains feasible object configurations. All blocks except R_2 and B_3 are fixed. B_3 has a single degree of freedom, whereas R_2 moves along the axis defined by $x_2 - y_2 = 1$. Further, since there is no functional relationship between the parameters of R_2 and B_3, their motions are independent. The other configuration space regions are similar and can be constructed incrementally [JS]. By examining them, we conclude that only one red and one blue object can move independently at a time, that the red and blue blocks do not mix, and that the clockwise order of the blocks cannot be reversed.

Editing queries require updating the motion constraints. Suppose we remove block R_3 and want to know if this affects the motion range of B_2. We delete all the motion constraints related to R_3 from S, and project the result onto the u_2-v_2 plane. The resulting projection is:

$$\{u_2 \ge 1.5,\ \ u_2 \le 2,\ \ u_2 = 1 - v_2\}$$

that is, B_2 is no longer fixed but can slide into the upper part of the puzzle.

The following sections show how the various operations discussed here naturally translate into basic operations of the proposed constraint based system and the importance of a suitable canonical form for the constraints.