

1. Record Nr.	UNISA996466439203316
Titolo	Automated Deduction – CADE 27 [[electronic resource]] : 27th International Conference on Automated Deduction, Natal, Brazil, August 27–30, 2019, Proceedings / / edited by Pascal Fontaine
Pubbl/distr/stampa	Cham : , : Springer International Publishing : , : Imprint : Springer, , 2019
ISBN	3-030-29436-6
Edizione	[1st ed. 2019.]
Descrizione fisica	1 online resource (XXIII, 582 p. 1901 illus., 56 illus. in color.)
Collana	Lecture Notes in Artificial Intelligence ; ; 11716
Disciplina	511.36028563
Soggetti	Artificial intelligence Software engineering Computer system failures Mathematical logic Computer logic Artificial Intelligence Software Engineering System Performance and Evaluation Mathematical Logic and Formal Languages Logics and Meanings of Programs
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Nota di bibliografia	Includes bibliographical references and index.
Nota di contenuto	Automated Reasoning for Security Protocols -- Computer Deduction and (Formal) Proofs in Mathematics -- From Counter-Model-based Quantifier Instantiation to Quantifier Elimination in SMT -- The CADE-27 ATP System Competition - CASC-27 -- Unification modulo Lists with Reverse - Relation with Certain Word Equations -- On the Width of Regular Classes of Finite Structures -- Extending SMT solvers to Higher-Order Logic -- Superposition with Lambdas -- Restricted Combinatory Unification -- dLi: Definite Descriptions in Differential Dynamic Logic -- SPASS-SATT { A CDCL(LA) Solver -- GRUNGE: A Grand Unified ATP Challenge -- Model Completeness, Covers and Superposition -- A Tableaux Calculus for Default Intuitionistic Logic -- NIL: Learning Nonlinear Interpolants -- ENIGMA-NG: Efficient Neural

and Gradient-Boosted Inference Guidance for E -- Towards Physical Hybrid Systems -- SCL -- Clause Learning from Simple Models -- Names are not just Sound and Smoke: Word Embeddings for Axiom Selection -- Computing Expected Runtimes for Constant Probability Programs -- Automatic Generation of Logical Models with AGES -- Automata Terms in a Lazy WSkS Decision Procedure -- Confluence by Critical Pair Analysis Revisited -- Composing Proof Terms -- Combining ProVerif and Automated Theorem Provers for Security Protocol Verification -- Towards Bit-Width-Independent Proofs in SMT Solvers -- On Invariant Synthesis for Parametric Systems -- The Aspect Calculus -- Uniform Substitution At One Fell Swoop -- A Formally Verified Abstract Account of Gödel's Incompleteness Theorems -- Old or Heavy? Decaying Gracefully with Age/Weight Shapes -- Induction in Saturation-Based Proof Search -- Faster, Higher, Stronger: E 2.3 -- Certified Equational Reasoning via Ordered Completion -- JGXYZ - An ATP System for Gap and Glut Logics -- GKC: a Reasoning System for Large Knowledge Bases -- Optimization Modulo the Theory of Floating-Point Numbers -- FAME(Q): An Automated Tool for Forgetting in Description Logics with Qualified Number Restrictions. .

Sommario/riassunto

This book constitutes the proceeding of the 27th International Conference on Automated Deduction, CADE 27, held in Natal, Brazil, in August 2019. The 27 full papers and 7 system descriptions presented were carefully reviewed and selected from 65 submissions. CADE is the major forum for the presentation of research in all aspects of automated deduction, including foundations, applications, implementations, and practical experience. .

2. Record Nr.	UNINA9910861995303321
Autore	Eberly Dave
Titolo	Robust and Error-Free Geometric Computing
Pubbl/distr/stampa	[Place of publication not identified] : , : CRC Press (Unlimited) : , : CRC Press, , 2020
ISBN	1-000-05664-3 0-429-33050-2 1-000-05662-7
Edizione	[1st ed.]
Descrizione fisica	1 online resource (xxiv, 363 pages)
Disciplina	516.08 516.00285
Soggetti	Geometric programming
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Nota di bibliografia	Includes bibliographical references and index.
Nota di contenuto	Cover -- Half Title -- Title Page -- Copyright Page -- Contents -- Preface -- Trademarks -- List of Figures -- List of Tables -- Listings -- 1 Introduction -- 1.1 Eigendecomposition for 3×3 Symmetric Matrices -- 1.1.1 Computing the Eigenvalues -- 1.1.2 Computing the Eigenvectors -- 1.1.3 A Nonrobust Floating-Point Implementation -- 1.1.4 A Robust Floating-Point Implementation -- 1.1.4.1 Computing the Eigenvalues -- 1.1.4.2 Computing the Eigenvectors -- 1.1.5 An Error-Free Implementation -- 1.2 Distance between Line Segments -- 1.2.1 Nonparallel Segments -- 1.2.2 Parallel Segments -- 1.2.3 A Nonrobust Floating-Point Implementation -- 1.2.4 A Robust Floating-Point Implementation -- 1.2.4.1 Conjugate Gradient Method -- 1.2.4.2 Constrained Conjugate Gradient Method -- 1.2.5 An Error-Free Implementation -- 2 Floating-Point Arithmetic -- 2.1 Binary Encodings -- 2.2 Binary Encoding of 32-bit Floating-Point Numbers -- 2.3 Binary Encoding of 64-bit Floating-Point Numbers -- 2.4 Rounding of Floating-Point Numbers -- 2.4.1 Round to Nearest with Ties to Even -- 2.4.2 Round to Nearest with Ties to Away -- 2.4.3 Round toward Zero -- 2.4.4 Round toward Positive -- 2.4.5 Round toward Negative -- 2.4.6 Rounding Support in C++ -- 3 Arbitrary-Precision Arithmetic -- 3.1 Binary Scientific Notation -- 3.2 Binary Scientific Numbers -- 3.2.1 Addition -- 3.2.2 Subtraction -- 3.2.3 Multiplication -- 3.3 Binary

Scientific Rationals -- 3.3.1 Addition and Subtraction -- 3.3.2
 Multiplication and Division -- 3.4 Conversions -- 3.4.1 Floating-Point
 Number to BSNumer -- 3.4.2 BSNumer to Floating-Point Number --
 3.4.3 BSRational to BSNumer of Specified Precision -- 3.4.4 BSNumer
 to BSNumer of Specified Precision -- 3.5 Performance Considerations
 -- 3.5.1 Static Computation of Maximum Precision -- 3.5.1.1 Addition
 and Subtraction -- 3.5.1.2 Multiplication.
 3.5.2 Dynamic Computation of Maximum Precision -- 3.5.3 Memory
 Management -- 4 Interval Arithmetic -- 4.1 Arithmetic Operations --
 4.2 Signs of Determinants -- 4.3 Primal Queries -- 4.3.1 Queries in 2D
 -- 4.3.2 Queries in 3D -- 5 Quadratic-Field Arithmetic -- 5.1 Sources
 of Rounding Errors -- 5.1.1 Rounding Errors when Normalizing Vectors
 -- 5.1.2 Errors in Roots to Quadratic Equations -- 5.1.3 Intersection of
 Line and Cone Frustum -- 5.2 Real Quadratic Fields -- 5.2.1 Arithmetic
 Operations -- 5.2.2 Allowing for Non-Square-Free d -- 5.2.3 Allowing
 for Rational d -- 5.3 Comparisons of Quadratic Field Numbers -- 5.4
 Quadratic Fields with Multiple Square Roots -- 5.4.1 Arithmetic
 Operations -- 5.4.2 Composition of Quadratic Fields -- 5.5 Estimating
 a Quadratic Field Number -- 5.5.1 Estimating a Rational Number --
 5.5.2 Estimating the Square Root of a Rational Number -- 5.5.3
 Estimating a 1-Root Quadratic Field Number -- 5.5.4 Estimating a 2-
 Root Quadratic Field Number -- 5.5.4.1 Two Nonzero Radical
 Coefficients -- 5.5.4.2 Three Nonzero Radical Coefficients -- 6
 Numerical Methods -- 6.1 Root Finding -- 6.1.1 Function Evaluation --
 6.1.2 Bisection -- 6.1.3 Newton's Method -- 6.1.4 Hybrid Newton-
 Bisection Method -- 6.1.5 Arbitrary-Precision Newton's Method -- 6.2
 Polynomial Root Finding -- 6.2.1 Discriminants -- 6.2.2 Preprocessing
 the Polynomials -- 6.2.3 Quadratic Polynomial -- 6.2.4 Cubic
 Polynomial -- 6.2.4.1 Nonsimple Real Roots -- 6.2.4.2 One Simple Real
 Root -- 6.2.4.3 Three Simple Real Roots -- 6.2.5 Quartic Polynomial --
 6.2.5.1 Processing the Root Zero -- 6.2.5.2 The Biquadratic Case --
 6.2.5.3 Multiplicity Vector (3, 1, 0, 0) -- 6.2.5.4 Multiplicity Vector (2,
 2, 0, 0) -- 6.2.5.5 Multiplicity Vector (2, 1, 1, 0) -- 6.2.5.6 Multiplicity
 Vector (1, 1, 1, 1) -- 6.2.6 High-Degree Polynomials -- 6.2.6.1
 Bounding Root Sequences by Derivatives.
 6.2.6.2 Bounding Roots by Sturm Sequences -- 6.2.6.3 Root Counting
 by Descartes' Rule of Signs -- 6.2.6.4 Real-Root Isolation -- 6.3 Linear
 Algebra -- 6.3.1 Systems of Linear Equations -- 6.3.2
 Eigendecomposition for 2×2 Symmetric Matrices -- 6.3.3
 Eigendecomposition for 3×3 Symmetric Matrices -- 6.3.4 3D Rotation
 Matrices with Rational Elements -- 7 Distance Queries -- 7.1
 Introduction -- 7.1.1 The Quadratic Programming Problem -- 7.1.2
 The Linear Complementarity Problem -- 7.1.3 The Convex Quadratic
 Programming Problem -- 7.1.3.1 Eliminating Unconstrained Variables
 -- 7.1.3.2 Reduction for Equality Constraints -- 7.2 Lemke's Method --
 7.2.1 Terms and Framework -- 7.2.2 LCP with a Unique Solution --
 7.2.3 LCP with Infinitely Many Solutions -- 7.2.4 LCP with No Solution
 -- 7.2.5 LCP with a Cycle -- 7.2.6 Avoiding Cycles when Constant
 Terms are Zero -- 7.3 Formulating a Geometric Query as a CQP --
 7.3.1 Distance Between Oriented Boxes -- 7.3.2 Test-Intersection of
 Triangle and Cylinder -- 7.4 Implementation Details -- 7.4.1 The LCP
 Solver -- 7.4.2 Distance Between Oriented Boxes in 3D -- 7.4.3 Test-
 Intersection of Triangle and Cylinder in 3D -- 7.4.4 Accuracy Problems
 with Floating-Point Arithmetic -- 7.4.5 Dealing with Vector
 Normalization -- 8 Intersection Queries -- 8.1 Method of Separating
 Axes -- 8.1.1 Separation by Projection onto a Line -- 8.1.2 Separation
 of Convex Polygons in 2D -- 8.1.3 Separation of Convex Polyhedra in
 3D -- 8.1.4 Separation of Convex Polygons in 3D -- 8.1.5 Separation

of Moving Convex Objects -- 8.1.6 Contact Set for Moving Convex Objects -- 8.2 Triangles Moving with Constant Linear Velocity -- 8.2.1 Two Moving Triangles in 2D -- 8.2.2 Two Moving Triangles in 3D -- 8.3 Linear Component and Sphere -- 8.3.1 Test-Intersection Queries -- 8.3.1.1 Line and Sphere -- 8.3.1.2 Ray and Sphere -- 8.3.1.3 Segment and Sphere. 8.3.2 Find-Intersection Queries -- 8.3.2.1 Line and Sphere -- 8.3.2.2 Ray and Sphere -- 8.3.2.3 Segment and Sphere -- 8.4 Linear Component and Box -- 8.4.1 Test-Intersection Queries -- 8.4.1.1 Lines and Boxes -- 8.4.1.2 Rays and Boxes -- 8.4.1.3 Segments and Boxes -- 8.4.2 Find-Intersection Queries -- 8.4.2.1 Liang-Barsky Clipping -- 8.4.2.2 Lines and Boxes -- 8.4.2.3 Rays and Boxes -- 8.4.2.4 Segments and Boxes -- 8.5 Line and Cone -- 8.5.1 Definition of Cones -- 8.5.2 Practical Matters for Representing Infinity -- 8.5.3 Definition of a Line, Ray and Segment -- 8.5.4 Intersection with a Line -- 8.5.4.1 Case $c_2 = 0$ -- 8.5.4.2 Case $c_2 = 0$ and $c_1 = 0$ -- 8.5.4.3 Case $c_2 = 0$ and $c_1 = 0$ -- 8.5.5 Clamping to the Cone Height Range -- 8.5.6 Pseudocode for Error-Free Arithmetic -- 8.5.6.1 Intersection of Intervals -- 8.5.6.2 Line-Cone Query -- 8.5.7 Intersection with a Ray -- 8.5.8 Intersection with a Segment -- 8.5.9 Implementation using Quadratic-Field Arithmetic -- 8.6 Intersection of Ellipses -- 8.6.1 Ellipse Representations -- 8.6.1.1 The Standard Form for an Ellipse -- 8.6.1.2 Conversion to a Quadratic Equation -- 8.6.2 Test-Intersection Query for Ellipses -- 8.6.3 Find-Intersection Query for Ellipses -- 8.6.3.1 Case $d_4 = 0$ and $e(x) = 0$ -- 8.6.3.2 Case $d_4 = 0$ and $e(x) = 0$ -- 8.6.3.3 Case $d_4 = 0$, $d_2 = 0$ and $e_2 = 0$ -- 8.6.3.4 Case $d_4 = 0$, $d_2 = 0$ and $e_2 = 0$ -- 8.6.3.5 Case $d_4 = 0$ and $d_2 = 0$ -- 8.7 Intersection of Ellipsoids -- 8.7.1 Ellipsoid Representations -- 8.7.1.1 The Standard Form for an Ellipsoid -- 8.7.1.2 Conversion to a Quadratic Equation -- 8.7.2 Test-Intersection Query for Ellipsoids -- 8.7.3 Find-Intersection Query for Ellipsoids -- 8.7.3.1 Two Spheres -- 8.7.3.2 Sphere-Ellipsoid: 3-Zero Center -- 8.7.3.3 Sphere-Ellipsoid: 2-Zero Center -- 8.7.3.4 Sphere-Ellipsoid: 1-Zero Center -- 8.7.3.5 Sphere-Ellipsoid: No-Zero Center. 8.7.3.6 Reduction to a Sphere-Ellipsoid Query -- 9 Computational Geometry Algorithms -- 9.1 Convex Hull of Points in 2D -- 9.1.1 Incremental Construction -- 9.1.2 Divide-and-Conquer Method -- 9.2 Convex Hull of Points in 3D -- 9.2.1 Incremental Construction -- 9.2.2 Divide-and-Conquer Method -- 9.3 Delaunay Triangulation -- 9.3.1 Incremental Construction -- 9.3.1.1 Inserting Points -- 9.3.1.2 Linear Walks and Intrinsic Dimension -- 9.3.1.3 The Insertion Step -- 9.3.2 Construction by Convex Hull -- 9.4 Minimum-Area Circle of Points -- 9.5 Minimum-Volume Sphere of Points -- 9.6 Minimum-Area Rectangle of Points -- 9.6.1 The Exhaustive Search Algorithm -- 9.6.2 The Rotating Calipers Algorithm -- 9.6.2.1 Computing the Initial Rectangle -- 9.6.2.2 Updating the Rectangle -- 9.6.2.3 Distinct Supporting Vertices -- 9.6.2.4 Duplicate Supporting Vertices -- 9.6.2.5 Multiple Polygon Edges of Minimum Angle -- 9.6.2.6 The General Update Step -- 9.6.3 A Robust Implementation -- 9.6.3.1 Avoiding Normalization -- 9.6.3.2 Indirect Comparisons of Angles -- 9.6.3.3 Updating the Support Information -- 9.6.3.4 Conversion to a Floating-Point Rectangle -- 9.7 Minimum-Volume Box of Points -- 9.7.1 Processing Hull Faces -- 9.7.1.1 Comparing Areas -- 9.7.1.2 Comparing Volumes -- 9.7.1.3 Comparing Angles -- 9.7.2 Processing Hull Edges -- 9.7.3 Conversion to a Floating-Point Box -- Bibliography -- Index.

both the descriptions and source code for hard-to-find
implementations of geometric algorithms. The code is robust, as
required by the applications they work on"--
