RWTH Aachen

Department of Computer Science
Technical Report

Demonstration of a Branch-and-Bound Algorithm for Global Optimization using McCormick Relaxations

Callum Corbett, Uwe Naumann, and Alexander Mitsos

ISSN 0935–3232 · Aachener Informatik Berichte · AIB-2011-24

RWTH Aachen · Department of Computer Science · November 2011

Demonstration of a Branch-and-Bound Algorithm for Global Optimization using McCormick Relaxations

Callum Corbett, Uwe Naumann, and Alexander Mitsos
naumann@stce.rwth-aachen.de

LuFG Informatik 12: Software and Tools for Computational Engineering

Abstract. This report is meant to demonstrate the actions performed by a branch-and-bound algorithm for global optimization on a simple minimization problem. McCormick relaxations are used to construct piecewise affine underestimators of the objective function.

1 The branch-and-bound algorithm

The branch-and-bound algorithm is used in global optimization to find the global minimum of an objective function $f(x)$, $x \in \mathbb{R}^n$, by bounding its value between an upper and a lower bound. After calculating the bounds, the optimization variables' set is partitioned (branching) and the procedure is repeated on all subsets. In two cases a subset is not partitioned any further: either if its bounds have converged or if it can be fathomed. When the values of the lower and upper bound are within an absolute or relative tolerance defined by the user, they are considered to have converged. If the lower bound of a subset attains a higher value than the upper bound this subset is fathomed. If ub^i is the upper bound of the subset S^i, then $\exists x^i \in S^i : f(x^i) \leq ub^i$. A lower bound lb^j of a subset S^j with a higher value than this upper bound, $lb^j > ub^i$, results in $\exists x^i \in S^i : f(x^j) \geq lb^j > ub^i \geq f(x^i)\ \forall x^j \in S^j$. Hence, the global minimum cannot be within the subset S^j as all values within the subset are definitely higher than a value previously found in the subset S^i. Typically, only one upper bound is stored, the lowest one found so far, while the lower bounds are different for each subset. The reason for this is that the upper bound is a value that is actually taken by the function at a specific point, while the lower bounds are underestimators. This means that an interval with a smaller lower bound than another does not necessarily lead to a lower value, as the bound may be lower due to a weaker relaxation.

To illustrate the workflow of the branch-and-bound algorithm, a simple example with only one optimization variable is considered here: Minimize the objective function

$$f(x) = \sqrt{|x|} + 0.01 \cdot x^3$$

with $x \in [-7, 5]$. As can be seen in Figure 1, this function attains its minimum $f^* \approx -0.784$ at $x = -7$. Additionally, a suboptimal local minimum exists at $x = 0$ with an objective value of 0. The relative tolerance is set to zero and the absolute tolerance to 0.5. This means that the solver will consider a subset to have converged (i.e., not consider any further subdivisions on this subset) if the difference between the upper and lower bound is less than or equal to 0.5. Note that the value 0.5 is only used for illustrative purposes, i.e., to keep the number of iterations small.

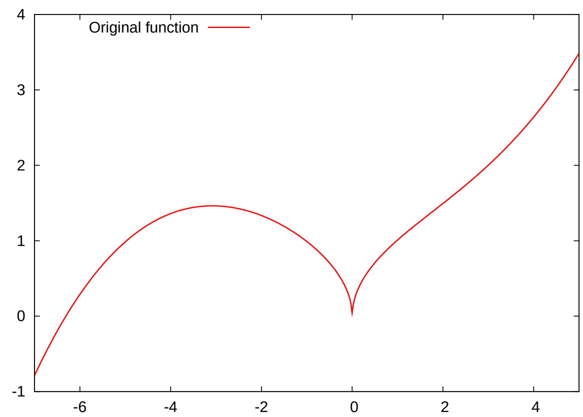

Fig. 1. The function to be minimized

In this example, the lower bound is calculated by means of McCormick relaxations and natural interval extensions, see [1]. The McCormick relaxations are convex underestimators that are propagated throughout the calculation of the objective function. In addition, the subgradient of the convex underestimator at a given point is propagated and its linearization is used as an affine relaxation. The values of the affine relaxation are always less than or equal to the values of the convex relaxation at any given point. As the values of the convex relaxation are always less than or equal to the corresponding objective function values, the minimum of the affine relaxation can be used as a lower bound. Additionally, the value of the natural interval extension is compared to the minimal value of an affine relaxation of the convex underestimator. By choosing the maximum of the values, the tighter of both underestimators is chosen. An alternative is to run a local solver on the convex relaxation to find its minimum. This generally leads to a tighter lower bound with the tradeoff of running a more costly local solver in each iteration.

The upper bound is computed by a cheap point evaluation at the arithmetic mean of the part of the optimization variable's set currently considered. Typically, a more costly local solver is used for the upper bound as it usually results in a better upper bound. For demonstration purposes the point evaluation shall suffice.

In the following, the 9 iterations needed by the branch-and-bound solver are illustrated and explained. Each iteration selects a subset according to the best-bound selection heuristic. The best-bound heuristic selects the subset with the lowest lower bound (new subsets are assumed to have the lower bound of their parent set) and if more than one exists uses a breadth-first approach. The solver then calculates a lower bound on the current subset. If the lower bound lies above the global upper bound (the lowest upper bound found so far), the subset is fathomed and the next iteration started. In the other case, the upper bound is computed. Finally, if the subset has not converged, it is branched upon, creating two new subsets. The solver then repeats the procedure until all subsets have either converged or been fathomed.

In the first iteration, the entire set $x \in [-7, 5]$ is considered. The results are plotted in Figure 2. The tighter lower bound in this case is the natural interval extension with approximately -3.43 as opposed to -3.64 for the minimum of the affine relaxation of the convex underestimator. It should be noted that the convex underestimator is not differentiable in the point $x = 0$. This is the reason for the use of subgradients and not gradients. For the upper bound, the point evaluation at $x = \frac{-7+5}{2} = -1$, $f(-1) = 0.99$, is used.

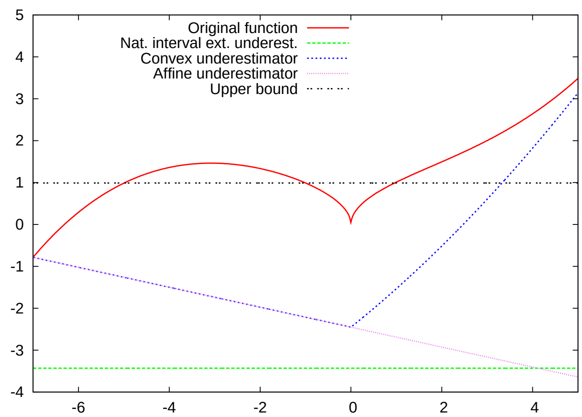

Fig. 2. Functions used in the first iteration

Now the branching on the optimization variable's set can be performed. There are various heuristics on where to branch, e.g., if local solvers were used either for the upper bound or on the convex relaxation for a lower bound, either of their solution points could serve as branching point. Here, the arithmetic mean is used as branching point resulting in the subsets $[-7, -1]$ and $[-1, 5]$.

From the two subsets available, see Figure 3, the second iteration calculates the lower and upper bound of the subset $[-7,-1]$. The result of these iterations

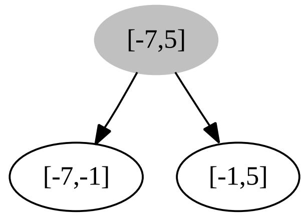

Fig. 3. Subsets before the second iteration

can be seen in Figure 4. As the upper bounds of the interval is greater than the

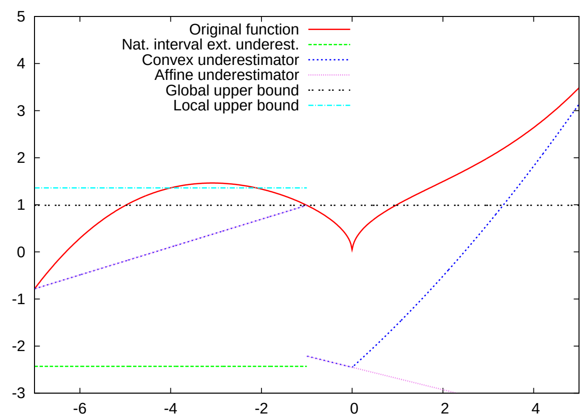

Fig. 4. Functions used in the second iteration

upper bound found previously, it is disregarded. It is also worth noting, that the affine relaxation is equal to the convex underestimator, which is the secant of the concave original function. This underestimator is the tightest possible convex relaxation and illustrates the strength of the McCormick relaxations. Finally, the subset is bisected into $[-7,-4]$ and $[-4,-1]$

The subset $[-1, 5]$ is selected for the third iteration, see Figure 5. On the sub-

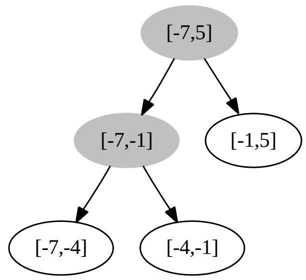

Fig. 5. Subsets before the second iteration

set [-1,5], the convex underestimator is not differentiable in two points, although only one non-differentiable point exists for the original function, see Figure 6. This again shows the necessity of subgradients as opposed to gradients, as also differentiable (parts of) functions can lead to non-differentiable relaxations. The upper bound does not improve on this subset as it is higher than the one found in the first iteration.

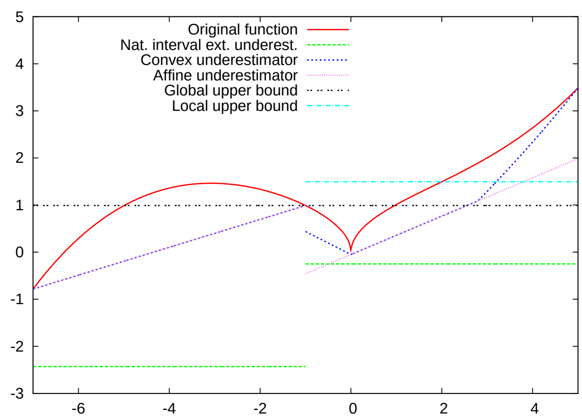

Fig. 6. Functions used in the third iteration

For the fourth iteration the solver has four possible subsets: $[-7,-4]$, $[-4,-1]$, $[-1,2]$ and $[2,5]$, see Figure 7. The interval $[-7,-4]$ is chosen. Again, the convex

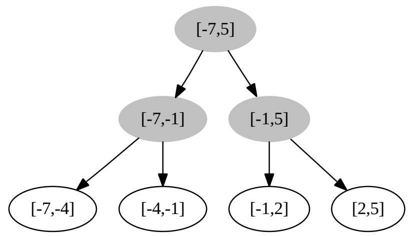

Fig. 7. Subsets before the fourth iteration

relaxation and the affine relaxation are the tightest possible convex relaxations, the secant of the concave original function. The upper bound calculated on this interval lies below the global upper bound. This means the global upper bound is updated with this new, lower value. Figure 8 shows the results of the calculations before the global upper bound is updated.

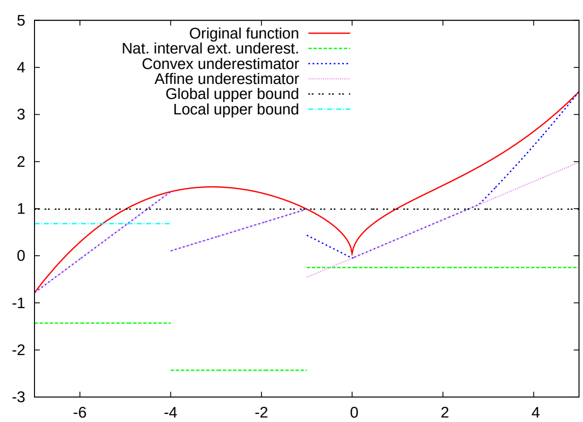

Fig. 8. Functions used in the fourth iteration

After updating the global upper bound to the new, lower value, the fifth iteration is performed on the subset $[-4, -1]$, Figure 9. Figure 10 illustrates the results of the computation. The lower bound of the subset $[-4, -1]$ (the value

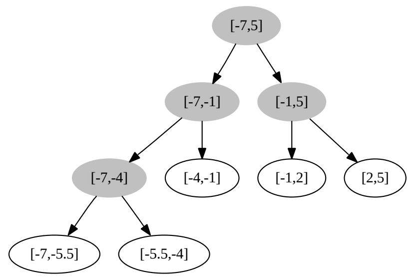

Fig. 9. Subsets before the fifth iteration

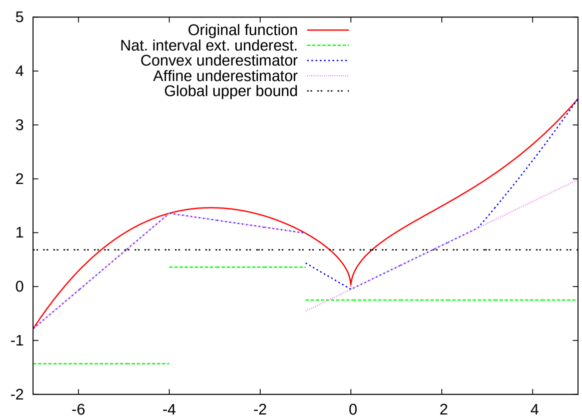

Fig. 10. Functions used by the fifth iteration

of the affine relaxation at $x = -1$) lies above the global upper bound. Due to this, the entire interval is fathomed as the global minimum cannot be within it.

The sixth iteration calculates the bounds on the subset $[-7, -5.5]$, which lowers the upper bound to ≈ 0.05. The lower bound of the subset $[-5.5, -4]$ calculated in the seventh iteration lies above the new upper bound, so it is fathomed. Figure 11 shows the state of the subsets after the seventh iteration, Figure 12 the results of the computation.

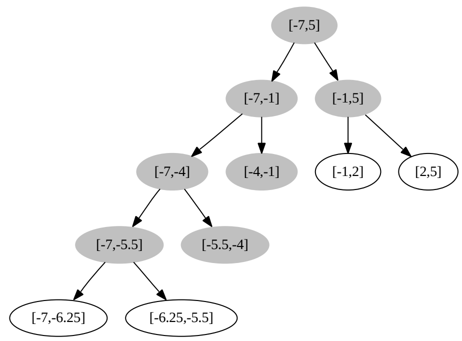

Fig. 11. Branching after the seventh iteration

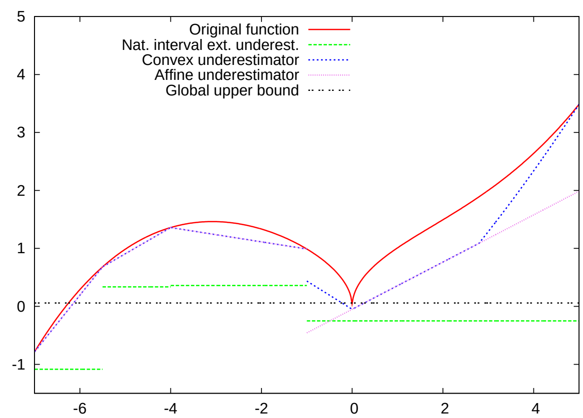

Fig. 12. Functions used in the sixth and seventh iteration

Calculating the bounds on the subset $[-7, -6.25]$ in the eighth iteration results in an upper bound that fathoms all other subsets except for $[-6.25, -5.5]$. The ninth iteration considers the interval $[-6.25, -5.5]$ though, witch leads to a lower bound above the upper bound calculated in the previous iteration, see Figures 13 and 14. The optimization variables value is now narrowed down to $x \in [-7, -6.25]$. All other intervals have been fathomed due to the low upper bound (≈ -0.334) found in the eighth iteration.

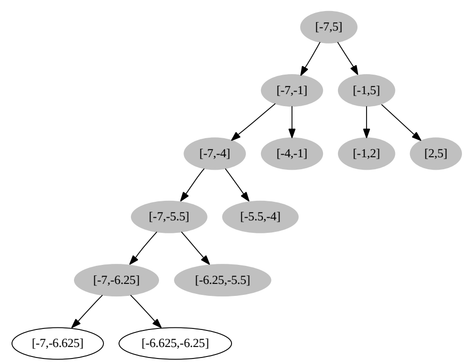

Fig. 13. Branching after the nineth iteration

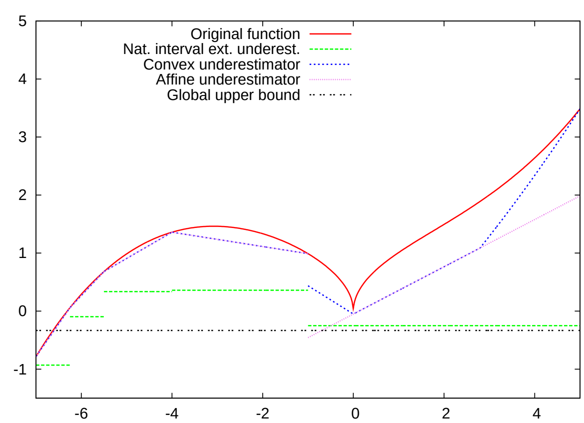

Fig. 14. Functions used in the eighth and ninth iteration

The subset $[-7, -6.25]$ has converged, as the difference between the upper bound ≈ -0.334 and the lower bound ≈ -0.784 is about 0.45, which is less than the absolute tolerance of 0.5. As all other subsets have been fathomed, the solver is finished and the example ends here.

References

1. A. Mitsos, B. Chachuat, and P. I. Barton. McCormick-based relaxations of algorithms. *In press: SIAM Journal on Optimization*, 20(2):573–601, 2009.

Aachener Informatik-Berichte

This list contains all technical reports published during the past three years. A complete list of reports dating back to 1987 is available from:

`http://aib.informatik.rwth-aachen.de/`

To obtain copies please consult the above URL or send your request to:

Informatik-Bibliothek, RWTH Aachen, Ahornstr. 55, 52056 Aachen,
Email: `biblio@informatik.rwth-aachen.de`

2008-01 *	Fachgruppe Informatik: Jahresbericht 2007
2008-02	Henrik Bohnenkamp, Marielle Stoelinga: Quantitative Testing
2008-03	Carsten Fuhs, Jürgen Giesl, Aart Middeldorp, Peter Schneider-Kamp, René Thiemann, Harald Zankl: Maximal Termination
2008-04	Uwe Naumann, Jan Riehme: Sensitivity Analysis in Sisyphe with the AD-Enabled NAGWare Fortran Compiler
2008-05	Frank G. Radmacher: An Automata Theoretic Approach to the Theory of Rational Tree Relations
2008-06	Uwe Naumann, Laurent Hascoet, Chris Hill, Paul Hovland, Jan Riehme, Jean Utke: A Framework for Proving Correctness of Adjoint Message Passing Programs
2008-07	Alexander Nyßen, Horst Lichter: The MeDUSA Reference Manual, Second Edition
2008-08	George B. Mertzios, Stavros D. Nikolopoulos: The λ-cluster Problem on Parameterized Interval Graphs
2008-09	George B. Mertzios, Walter Unger: An optimal algorithm for the k-fixed-endpoint path cover on proper interval graphs
2008-10	George B. Mertzios, Walter Unger: Preemptive Scheduling of Equal-Length Jobs in Polynomial Time
2008-11	George B. Mertzios: Fast Convergence of Routing Games with Splittable Flows
2008-12	Joost-Pieter Katoen, Daniel Klink, Martin Leucker, Verena Wolf: Abstraction for stochastic systems by Erlang's method of stages
2008-13	Beatriz Alarcón, Fabian Emmes, Carsten Fuhs, Jürgen Giesl, Raúl Gutiérrez, Salvador Lucas, Peter Schneider-Kamp, René Thiemann: Improving Context-Sensitive Dependency Pairs
2008-14	Bastian Schlich: Model Checking of Software for Microcontrollers
2008-15	Joachim Kneis, Alexander Langer, Peter Rossmanith: A New Algorithm for Finding Trees with Many Leaves
2008-16	Hendrik vom Lehn, Elias Weingärtner and Klaus Wehrle: Comparing recent network simulators: A performance evaluation study
2008-17	Peter Schneider-Kamp: Static Termination Analysis for Prolog using Term Rewriting and SAT Solving
2008-18	Falk Salewski: Empirical Evaluations of Safety-Critical Embedded Systems

2008-19 Dirk Wilking: Empirical Studies for the Application of Agile Methods to Embedded Systems

2009-02 Taolue Chen, Tingting Han, Joost-Pieter Katoen, Alexandru Mereacre: Quantitative Model Checking of Continuous-Time Markov Chains Against Timed Automata Specifications

2009-03 Alexander Nyßen: Model-Based Construction of Embedded Real-Time Software - A Methodology for Small Devices

2009-04 Daniel Klünder: Entwurf eingebetteter Software mit abstrakten Zustandsmaschinen und Business Object Notation

2009-05 George B. Mertzios, Ignasi Sau, Shmuel Zaks: A New Intersection Model and Improved Algorithms for Tolerance Graphs

2009-06 George B. Mertzios, Ignasi Sau, Shmuel Zaks: The Recognition of Tolerance and Bounded Tolerance Graphs is NP-complete

2009-07 Joachim Kneis, Alexander Langer, Peter Rossmanith: Derandomizing Non-uniform Color-Coding I

2009-08 Joachim Kneis, Alexander Langer: Satellites and Mirrors for Solving Independent Set on Sparse Graphs

2009-09 Michael Nett: Implementation of an Automated Proof for an Algorithm Solving the Maximum Independent Set Problem

2009-10 Felix Reidl, Fernando Sánchez Villaamil: Automatic Verification of the Correctness of the Upper Bound of a Maximum Independent Set Algorithm

2009-11 Kyriaki Ioannidou, George B. Mertzios, Stavros D. Nikolopoulos: The Longest Path Problem is Polynomial on Interval Graphs

2009-12 Martin Neuhäußer, Lijun Zhang: Time-Bounded Reachability in Continuous-Time Markov Decision Processes

2009-13 Martin Zimmermann: Time-optimal Winning Strategies for Poset Games

2009-14 Ralf Huuck, Gerwin Klein, Bastian Schlich (eds.): Doctoral Symposium on Systems Software Verification (DS SSV'09)

2009-15 Joost-Pieter Katoen, Daniel Klink, Martin Neuhäußer: Compositional Abstraction for Stochastic Systems

2009-16 George B. Mertzios, Derek G. Corneil: Vertex Splitting and the Recognition of Trapezoid Graphs

2009-17 Carsten Kern: Learning Communicating and Nondeterministic Automata

2009-18 Paul Hänsch, Michaela Slaats, Wolfgang Thomas: Parametrized Regular Infinite Games and Higher-Order Pushdown Strategies

2010-02 Daniel Neider, Christof Löding: Learning Visibly One-Counter Automata in Polynomial Time

2010-03 Holger Krahn: MontiCore: Agile Entwicklung von domänenspezifischen Sprachen im Software-Engineering

2010-04 René Wörzberger: Management dynamischer Geschäftsprozesse auf Basis statischer Prozessmanagementsysteme

2010-05 Daniel Retkowitz: Softwareunterstützung für adaptive eHome-Systeme

2010-06 Taolue Chen, Tingting Han, Joost-Pieter Katoen, Alexandru Mereacre: Computing maximum reachability probabilities in Markovian timed automata

2010-07 George B. Mertzios: A New Intersection Model for Multitolerance Graphs, Hierarchy, and Efficient Algorithms

2010-08 Carsten Otto, Marc Brockschmidt, Christian von Essen, Jürgen Giesl: Automated Termination Analysis of Java Bytecode by Term Rewriting

2010-09 George B. Mertzios, Shmuel Zaks: The Structure of the Intersection of Tolerance and Cocomparability Graphs

2010-10 Peter Schneider-Kamp, Jürgen Giesl, Thomas Ströder, Alexander Serebrenik, René Thiemann: Automated Termination Analysis for Logic Programs with Cut

2010-11 Martin Zimmermann: Parametric LTL Games

2010-12 Thomas Ströder, Peter Schneider-Kamp, Jürgen Giesl: Dependency Triples for Improving Termination Analysis of Logic Programs with Cut

2010-13 Ashraf Armoush: Design Patterns for Safety-Critical Embedded Systems

2010-14 Michael Codish, Carsten Fuhs, Jürgen Giesl, Peter Schneider-Kamp: Lazy Abstraction for Size-Change Termination

2010-15 Marc Brockschmidt, Carsten Otto, Christian von Essen, Jürgen Giesl: Termination Graphs for Java Bytecode

2010-16 Christian Berger: Automating Acceptance Tests for Sensor- and Actuator-based Systems on the Example of Autonomous Vehicles

2010-17 Hans Grönniger: Systemmodell-basierte Definition objektbasierter Modellierungssprachen mit semantischen Variationspunkten

2010-18 Ibrahim Armaç: Personalisierte eHomes: Mobilität, Privatsphäre und Sicherheit

2010-19 Felix Reidl: Experimental Evaluation of an Independent Set Algorithm

2010-20 Wladimir Fridman, Christof Löding, Martin Zimmermann: Degrees of Lookahead in Context-free Infinite Games

2011-02 Marc Brockschmidt, Carsten Otto, Jürgen Giesl: Modular Termination Proofs of Recursive Java Bytecode Programs by Term Rewriting

2011-03 Lars Noschinski, Fabian Emmes, Jürgen Giesl: A Dependency Pair Framework for Innermost Complexity Analysis of Term Rewrite Systems

2011-04 Christina Jansen, Jonathan Heinen, Joost-Pieter Katoen, Thomas Noll: A Local Greibach Normal Form for Hyperedge Replacement Grammars

2011-07 Shahar Maoz, Jan Oliver Ringert, Bernhard Rumpe: An Operational Semantics for Activity Diagrams using SMV

2011-08 Thomas Ströder, Fabian Emmes, Peter Schneider-Kamp, Jürgen Giesl, Carsten Fuhs: A Linear Operational Semantics for Termination and Complexity Analysis of ISO Prolog

2011-11 Nils Jansen, Erika Ábrahám, Jens Katelaan, Ralf Wimmer, Joost-Pieter Katoen, Bernd Becker: Hierarchical Counterexamples for Discrete-Time Markov Chains

2011-12 Ingo Felscher, Wolfgang Thomas: On Compositional Failure Detection in Structured Transition Systems

2011-13 Michael Förster, Uwe Naumann, Jean Utke: Toward Adjoint OpenMP

2011-14 Daniel Neider, Roman Rabinovich, Martin Zimmermann: Solving Muller Games via Safety Games

2011-16 Niloofar Safiran, Uwe Naumann: Toward Adjoint OpenFOAM

2011-18 Kamal Barakat: Introducing Timers to pi-Calculus

2011-19 Marc Brockschmidt, Thomas Ströder, Carsten Otto, Jürgen Giesl: Automated Detection of Non-Termination and NullPointerExceptions for Java Bytecode

* These reports are only available as a printed version.
Please contact `biblio@informatik.rwth-aachen.de` to obtain copies.