
University of Belgrade
Faculty of Mathematics
Automated Reasoning GroUp
Fifth Workshop on Formal and Automated Theorem Proving and Applications
Belgrade, February 03-04, 2012.
argo.matf.bg.ac.rs/events/2012/fatpa2012
Participants:
Organization:
Automated Reasoning GroUp
University of Belgrade, Serbia
argo.matf.bg.ac.rs

Fifth Workshop on Formal and Automated Theorem Proving and Applications

http://argo.matf.bg.ac.rs/fatpa2012

Book of Abstracts and Discussions

and

Little Belgrade City Guide for Workshop Participants

February 3-4, 2012, Belgrade, Serbia

Preface

This booklet contains abstracts of the talks given at the:

Fifth Workshop on Formal and Automated Theorem Proving and Applications

held at the University of Belgrade on February 3-4, 2012. The meeting was attended by 25 participants coming from 9 research institutions from 6 European countries (Austria (1), Croatia (1), France (3), Serbia (18), Sweden (1), United Kingdom (1)).

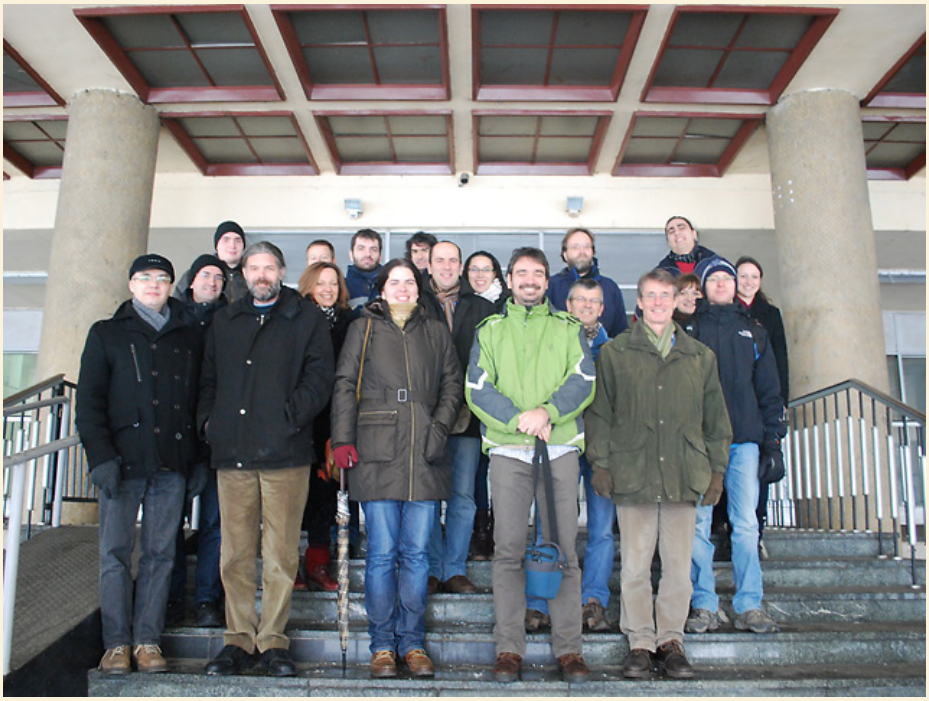
The program consisted of 18 talks, divided (rather loosely) into the four categories: SAT and SMT, Formal theorem proving and foundations, Geometry reasoning, and Applications of theorem proving.

More details about the meeting can be found online: `http://argo.matf.bg.ac.rs/fatpa2012`

The meeting was organized by the ARGO group (`http://argo.matf.bg.ac.rs`). For the success of the meeting, we are grateful to all speakers and all participants. We are also grateful to the Faculty of Mathematics, University of Belgrade which was the host institution of the meeting.

Predrag Janičić,
Faculty of Mathematics, University of Belgrade, Serbia

Participants



1. Milan Banković (University of Belgrade, Serbia)
   `http://www.matf.bg.ac.rs/~milan`
2. Jelena Čolić (University of Novi Sad, Serbia)
3. Silvia Ghilezan (University of Novi Sad, Serbia)
   `http://imft.ftn.ns.ac.rs/~silvia`
4. Hugo Herbelin (INRIA — PPS, Paris, France)
   `http://pauillac.inria.fr/~herbelin/index-eng.html`
5. Predrag Janičić (University of Belgrade, Serbia)
   `http://www.matf.bg.ac.rs/~janicic`
6. Oliver Kullmann (Swansea University, United Kingdom)
   `http://www.cs.swan.ac.uk/~csoliver/`
7. Petar Maksimović (Mathematical Institute, Belgrade, Serbia)
8. Marko Maliković (University of Rijeka, Croatia)
   `http://www.ffri.uniri.hr/~marko/`

9. Filip Marić (University of Belgrade, Serbia)
   http://www.matf.bg.ac.rs/~filip
10. Julien Narboux (University of Strasbourg, France)
    http://dpt-info.u-strasbg.fr/~narboux/
11. Walther Neuper (Graz University of Technology, Austria)
    http://www.ist.tugraz.at/neuper
12. Mladen Nikolić (University of Belgrade, Serbia)
    http://www.matf.bg.ac.rs/~nikolic
13. Zoran Petrić (Mathematical Institute, Belgrade, Serbia)
    http://www.mi.sanu.ac.rs/~zpetric/
14. Danijela Petrović (University of Belgrade, Serbia)
    http://www.matf.bg.ac.rs/~danijela/
15. Ivan Petrović (University of Belgrade, Serbia)
16. Nina Radojičić (University of Belgrade, Serbia)
    http://www.matf.bg.ac.rs/~nina/
17. Philipp Rümmer (University of Uppsala, Sweden)
    http://www.philipp.ruemmer.org
18. Pascal Schreck (University of Strasbourg, France)
    http://sites.google.com/site/pascalschreck/
19. Ana Spasić (University of Belgrade, Serbia)
    http://www.matf.bg.ac.rs/~aspasic/
20. Mirko Spasić (University of Belgrade, Serbia)
    http://www.matf.bg.ac.rs/~mirko
21. Mirko Stojadinović (University of Belgrade, Serbia)
    http://www.matf.bg.ac.rs/~mirkos
22. Sana Stojanović (University of Belgrade, Serbia)
    http://www.matf.bg.ac.rs/~sana
23. Milan Todorović (University of Belgrade, Serbia)
24. Milena Vujošević-Janičić (University of Belgrade, Serbia)
    http://www.matf.bg.ac.rs/~milena
25. Aleksandar Zeljić (Mathematical Institute, Belgrade, Serbia)

Program

Friday, February 3, 2012.	
09:30—09:55	Registration
09:55—10:00	Opening Remarks
Session *SAT and SMT*; Session Chair: Filip Marić	
10:00—10:30	Oliver Kullmann (Swansea University, United Kingdom): *Cube and Conquer: Guiding CDCL SAT Solvers by Lookaheads*
10:30—11:00	Philipp Rümmer (Uppsala University, Sweden): *E-Matching with Free Variables*
11:00—11:30	Coffee break
11:30—12:00	Mladen Nikolić (University of Belgrade, Serbia): *CDCL-Based Abstract State Transition System for Coherent Logic*
12:00—12:15	Milan Banković (University of Belgrade, Serbia): *pArgoSAT – Parallelization of Boolean Constraint Propagation in DPLL-based SAT solvers*
12:15—12:30	Aleksandar Zeljić (University of Belgrade, Serbia): *Instance Features for Non CNF Solver Portfolios*
12:30—15:00	Lunch break (Restaurant "Teatroteka")
Session *Formal Theorem Proving*; Session Chair: Predrag Janičić	
15:00—15:30	Hugo Herbelin (INRIA — PPS, Paris, France): *An enumeration-free proof of Gödel's completeness theorem with side effects*
15:30—15:45	Petar Maksimović (Mathematical Institute, Belgrade, Serbia): *A Logical Framework with External Predicates*
15:45—16:15	Coffee break
16:15—16:45	Marko Maliković (University of Rijeka, Croatia): *Formal Analysis of Correctness of a Strategy for the KRK Chess Endgame*
16:45—17:15	Filip Marić (University of Belgrade, Serbia): *Formalized Search for FC Families within Isabelle/HOL*
17:15—19:00	Informal discussions and individual meetings
19:30—22:00	Dinner at (Restaurant "Klub Književnika")

Saturday, February 4, 2012.	
Session *Geometry Reasoning*; Session Chair: Filip Marić	
10:00—10:30	Julien Narboux (University of Strasbourg, France): *Formalization of Simple Wu's Method in Coq*
10:30—11:00	Pascal Schreck (University of Strasbourg, France): *Geometric Constructions and First Order Logic*
11:00—11:30	Coffee break
11:30—11:45	Predrag Janičić (University of Belgrade, Serbia): *Automated Solving of Triangle Construction Problems*
11:45—12:00	Sana Stojanović (University of Belgrade, Serbia): *Exploiting symmetries and axiom reformulation in automated generation of formal proofs*
12:00—12:15	Ivan Petrović (University of Belgrade, Serbia): *Integration of OpenGeoProver with GeoGebra*
12:30—15:00	Lunch break (Restaurant "Teatroteka")
Session *Applications of Theorem Proving*; Session Chair: Predrag Janičić	
15:00—15:30	Walther Neuper (Graz University of Technology, Austria): *How Theorem-Prover Technology Advances Educational Math Software —* *Lessons Learned from Preparation of a Grant Proposal*
15:30—15:45	Milena Vujošević-Janičić (University of Belgrade, Serbia): *Automated Evaluation of Students' Programs: Testing, Verification and Similarity*
15:45—16:15	Coffee break
16:15—16:30	Mladen Nikolić (University of Belgrade, Serbia): *Program Similarity Measurement for Evaluation of Students' Programs*
16:30—16:45	Mirko Stojadinović (University of Belgrade, Serbia): *Reduction of finite linear CSPs to SAT using different encodings*
16:45—19:00	Informal discussions and individual meetings
19:30—22:30	Dinner (Restaurant "Little Bay")

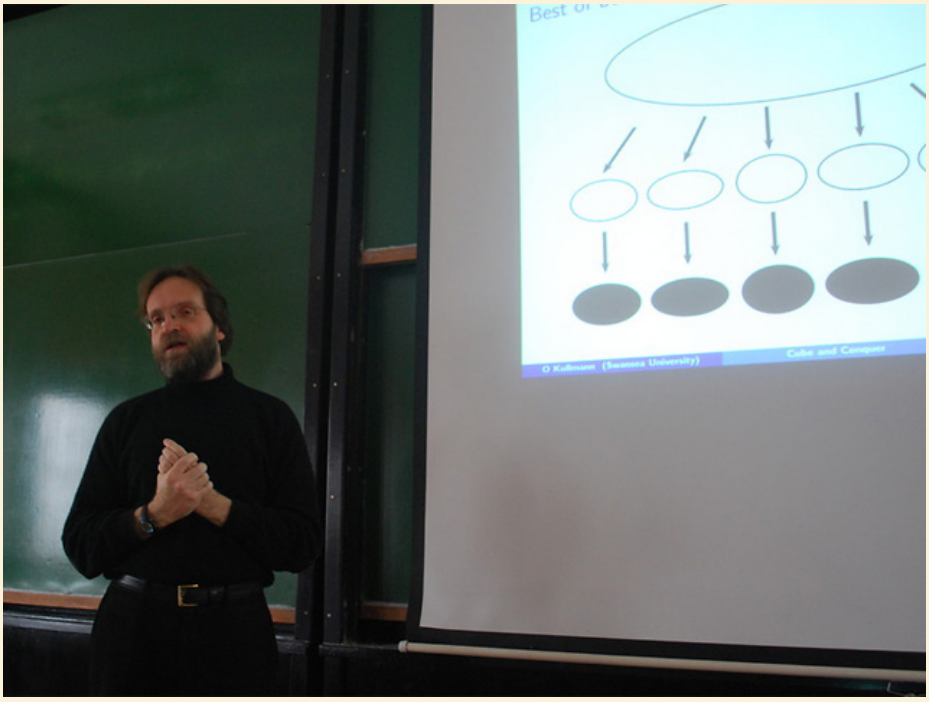
Opening

Predrag Janičić

SAT and SMT

Session Chair: Filip Marić

1 Cube and Conquer: Guiding CDCL SAT Solvers by Lookaheads



Oliver Kullmann
Swansea University, United Kingdom

— *Joint work with Marijn J.H. Heule, Siert Wieringa and Armin Biere* —

Abstract

We present a new SAT approach, called "cube-and-conquer", targeted at reducing solving time on hard instances. This two-phase approach partitions a problem into many thousands (or millions) of cubes using lookahead techniques. Afterwards, a conflict-driven solver tackles the problem, using the cubes to guide the search. On several hard competition benchmarks, our hybrid approach outperforms both lookahead and conflict-driven solvers. Moreover, because cube-and-conquer is natural to parallelise, it is a competitive alternative for solving SAT problems in parallel.

This approach was originally developed for solving hard (unsatisfiable) van-der-Waerden problems, and here we do not only achieve optimal speed-up for hundreds of processors, but also the total run-time (in comparison to all other available methods, parallel or sequential) is always improved, often by orders of magnitude.

The corresponding paper has been accepted at HVC 2011; best-paper-award at the conference. Preprint available at: `http://www.cs.swan.ac.uk/~csoliver/papers.html#CuCo2011`

Slides:

`http://argo.matf.bg.ac.rs/events/2012/fatpa2012/slides/OliverKullmann.pdf`

Discussion

Predrag: You mentioned cryptanalysis problems. Do you have a feeling that this approach can be suitable for cryptanalysis SAT instances?
Re: Yes, we believe the approach is suitable for cryptanalysis. The main challenge here is to handle satisfiable instances. Initially the cube-and-conquer approach focussed on unsatisfiable instances, so more research is needed.

Predrag: Can you somehow generally describe SAT instances for which this approach is suitable? Can one have some deeper argument about that suitability or only an experimental evidence?
Re: The cube-and-conquer method has a tree-like part at the root, and then (some approximation of) a dag-like part at the leaves, when taking into account that lookahead SAT-solvers are more related to tree-resolution, while conflict-driven SAT-solvers are more related to dag-resolution. So it seems a description of instances where the approach is suitable could be: it must be possible to split the resolution refutation into independent parts. This applies for unsatisfiable instances, while for satisfiable instances more research is still needed to get some kind of feeling (see above).

Filip: You can use off-the-shelf SAT solvers within your system. What is needed for that, what modification of the solvers?
Re: If using a solver in the conquer-phase, then no modifications are needed (just SAT-solving is used). For the cube-phase, according to our general understanding only lookahead-solvers are considered, and then it is a rather minor modification to cut off the processing according to the cube-heuristics.

Filip: What is the relation between solving times of cube and conquer parts?
Re: Say it is at most 10% for the cube-part. And the harder the problem, the less the cube-part.

---

2 E-Matching with Free Variables

Philipp Rümmer
Uppsala University, Sweden

Abstract

E-matching is the most commonly used technique to handle quantifiers in SMT solvers. It works by identifying characteristic sub-expressions of quantified formulae, named triggers, which are matched during proof search on ground terms to discover relevant instantiations of the quantified formula. E-matching has proven to be an efficient and practical approach to handle quantifiers, in particular because triggers can be provided by the user to guide proof search; however, as it is heuristic in nature, e-matching alone is typically insufficient to establish a complete proof procedure. In contrast, free variable methods in tableau-like calculi are more robust and give rise to complete procedures, e.g., for first-order logic, but are not comparable to e-matching in terms of scalability. This talk discusses how e-matching can be combined with free variable approaches, leading to calculi that enjoy similar completeness properties as pure free variable procedures, but in which it is still possible for a user to provide domain-specific triggers to improve performance.

Slides:

`http://argo.matf.bg.ac.rs/events/2012/fatpa2012/slides/PhilippRuemmer.pdf`

Discussion

Predrag: When constructing C in the case of a pure Presburger arithmetic input formula, is C the formula itself?
Re: It might be the formula itself, but generally it is obtained from it by elimination of inner quantifiers.

Filip: Are triggers provided by the user or computed?

Re: As in SMT solvers — there are two categories of benchmarks: one with user supplied triggers, and the other without them. In the second case, the system automatically tries to find good triggers.

Predrag: Can your system be used not only for linear arithmetic, but for any theory that admits quantifier elimination?
Re: Integers were present in the system from the beginning, but generally you can have other theories that admit quantifier elimination.

Silvia: It is appealing that you use sequent calculus for your system, how you made that choice?
Re: There is no a special reason for it, I used it earlier a lot so it was a natural choice for me.

---

3 CDCL-Based Abstract State Transition System for Coherent Logic

Mladen Nikolić
University of Belgrade, Serbia

— *Joint work with Predrag Janičić* —

Abstract

We present a new, CDCL-based approach for automated theorem proving in coherent logic - an expressive fragment of first-order logic that provides potential for obtaining readable and machine verifiable proofs. The approach is described by means of an abstract transition system, inspired by existing transition systems for SAT. The presented transition system serves as a base for our semi-decision procedure for coherent logic, for which we prove key properties.

Slides:

http://argo.matf.bg.ac.rs/events/2012/fatpa2012/slides/MladenNikolic1.pdf

Discussion

Oliver: Could you explain the notion of negative quantified literals?
Re: An important property of positive and negative literals in SAT is that they clash, and then we know we should backtrack. So, what simple formulae have such property in our context? For atom l the formulae that signal the need for backtrack are formulae of form $l \Rightarrow \bot$ or $l \Rightarrow goal$.

Philipp: Do you use an existing SMT solver in your implementation?

Re: No, we build our own solver.

Marko: Do you expect some constraints coming from the structure of geometry axioms?
Re: In Hilbert's system, the continuity axiom does not have coherent form, but other axioms do.

Predrag: Maybe it is worth mentioning what is real expressive power of coherent logic?
Re: Yes, any first-order formula can be translated to a coherent formula.

---

4 pArgoSAT – Parallelization of Boolean Constraint Propagation in DPLL-based SAT solvers

Milan Banković
University of Belgrade, Serbia

— *Joint work with Filip Marić* —

Abstract

In modern DPLL-based SAT/SMT solvers, most of the processing time is spent in boolean constraint propagation/theory propagation. Parallelizing the implementation of such procedures could take advantage of the modern multi-core processors, since such processors can run more than one thread of execution at the same time. In this talk, our preliminary parallel implementation of the boolean constraint propagation procedure based on the well-known two-watched-literals scheme is presented. The overall implementation design is discussed, and some experimental results are given. We also discuss the future work, including: further implementation optimizing, implementation of missing features, and extending the implementation to support SMT, which is the final goal of this work.

Slides:

`http://argo.matf.bg.ac.rs/events/2012/fatpa2012/slides/MilanBankovic.pdf`

Discussion

Oliver: For parallelization you have only low level parallelization. Why don't you consider splitting at high level? Have you considered other sorts of parallelization within your architecture, or only BCP parallelization?

Re: For now, we consider only parallelization of decision procedures, not parallelization of the search process. Still, since our approach is general ("job" can be anything), there is a room for experiments with other sorts of parallelization.

Filip: What speed-up can we expect from the parallelization? In particular, how much speed up is expected with 2 threads?
Re: Since this is still an early phase work, we still don't have an estimate on the speed-up. In related work speed up factor was 1.57 and this is near the theoretical optimum. We expect it to be little less then this. In related approaches, additional speed-up was not gained by using 4 threads, and it would be interesting to see if our approach has speedup on 4 threads.

Philipp: Many SMT theories use split-on-demand, so there is a room for this sort of parallelization.
Re: Yes, indeed, this actually was also our idea.

---

5 Instance Features for Non CNF Solver Portfolios

Aleksandar Zeljić
University of Belgrade, Serbia

— Joint work with Mladen Nikolić, Milan Todorović, and Predrag Janičić —

Abstract

In recent years, a new approach for solving propositional satisfiability problems has appeared in form of non-CNF solvers. Also, portfolio approach for CNF solvers has shown to be very successful, with the most important and influential portfolio SATzilla. One of key contributions of the SATzilla portfolio is the set of features for CNF instances that was later successfully used by other portfolios. Having in mind the importance of these features for portfolios of CNF solvers, we propose a set of instance features for non-CNF instances. The proposed instance features are easy to implement and are computable in linear time with respect to the input formula size. Based on them, we developed a non-CNF solver portfolio that uses a simple k-nearest neighbors classification. Our experimental results show that the proposed features enable our portfolio to make best ranked choices much more often than the approach in which the best single solver is chosen in advance.

Slides:

`http://argo.matf.bg.ac.rs/events/2012/fatpa2012/slides/AleksandarZeljic.pdf`

Discussion

Oliver: By portfolio, you think of using algorithm that runs in the beginning and decides what solver to use?
Re: Yes.

Oliver: Did you consider some attributes based on short runs of the solver?
Re: (Mladen) There are such attributes for CNF instances used by SATzilla. We have not tried such features yet, but it is an interesting direction to proceed.

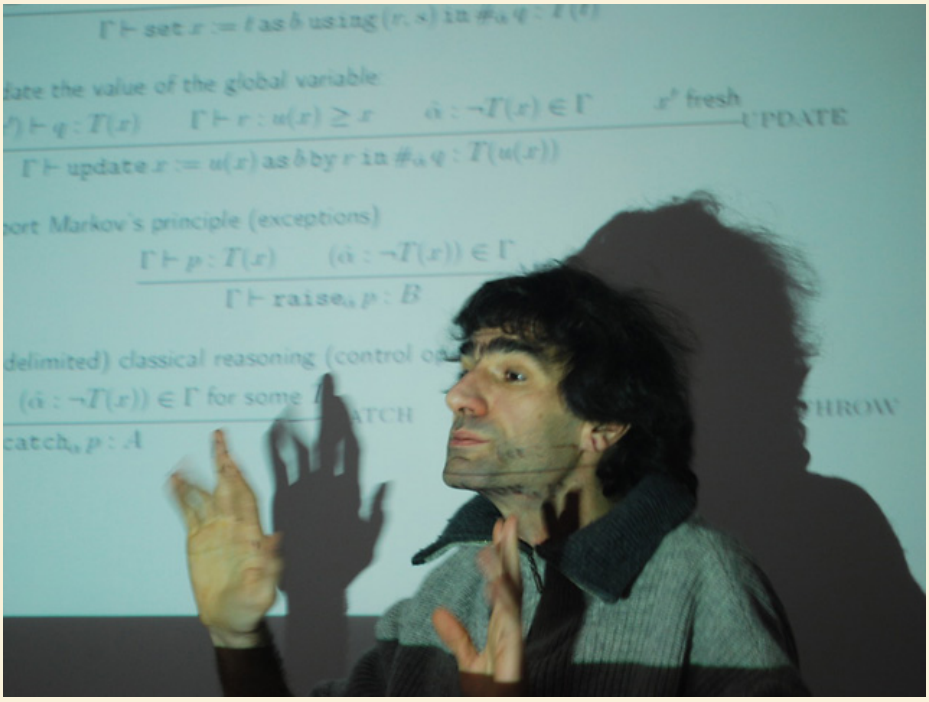
Predrag: How your attributes relate to SATzilla attributes for CNF SAT?
Re: We were inspired by some SATzilla attributes, but they could not be used directly.

Oliver: It would be good that machine learning decides which features are more important.
Re: Using different measures, from other, already solved instances, we decide which solver to use.

Formal Theorem Proving

Session Chair: Predrag Janičić

6 An enumeration-free proof of Gödel's completeness theorem with side effects



Hugo Herbelin
INRIA — PPS, Paris, France, France

Abstract

Completeness proofs are related to type-directed partial evaluation. Danvy showed that type-directed partial evaluation in the presence of sums conveniently takes advantage of delimited control. Along the proof-as-program correspondence, Danvy's program can in turn be seen as a proof formulated in a logical system extended with delimited control and monotone memory assignments. We deduce from this an original enumeration-free proof of Gödel's completeness theorem where the definition of the model depends on a global state.

Slides:

http://argo.matf.bg.ac.rs/events/2012/fatpa2012/slides/HugoHerbelin.pdf

Discussion

Silvia: How do you use CPs translations?
Re: You get proof in usual intuitionistic logic.

Predrag: In Isabelle, there is support for monads, is there support for monads in Coq?
Re: No, but you can make it yourself.

---

7 A Logical Framework with External Predicates

Petar Maksimović
Mathematical Institute Belgrade, Serbia

Abstract

The $\mathsf{LF}_{\mathcal{P}}$ Framework is an extension of the Harper-Honsell-Plotkin's Edinburgh Logical Framework LF with *external logical predicates*. This is accomplished by defining *lock type constructors*, which are a sort of $\diamond$ modality constructors, releasing their argument under the condition that a (possibly) external predicate is satisfied on an appropriate typed judgement. Lock types are defined using the standard pattern of Constructive Type Theory, i. e. via introduction and elimination rules. Using $\mathsf{LF}_{\mathcal{P}}$, one can factor out the complexity of encoding specific features of logical systems which are awkwardly encoded in plain LF, such as side-conditions in the application of rules in Modal Logics, substructural logics including *non-commutative Linear Logic*, or pre- and post-conditions in programming languages. Once these conditions have been isolated, their *verification* can be delegated to an external proof engine.

Slides:

http://argo.matf.bg.ac.rs/events/2012/fatpa2012/slides/PetarMaksimovic.pdf

Discussion

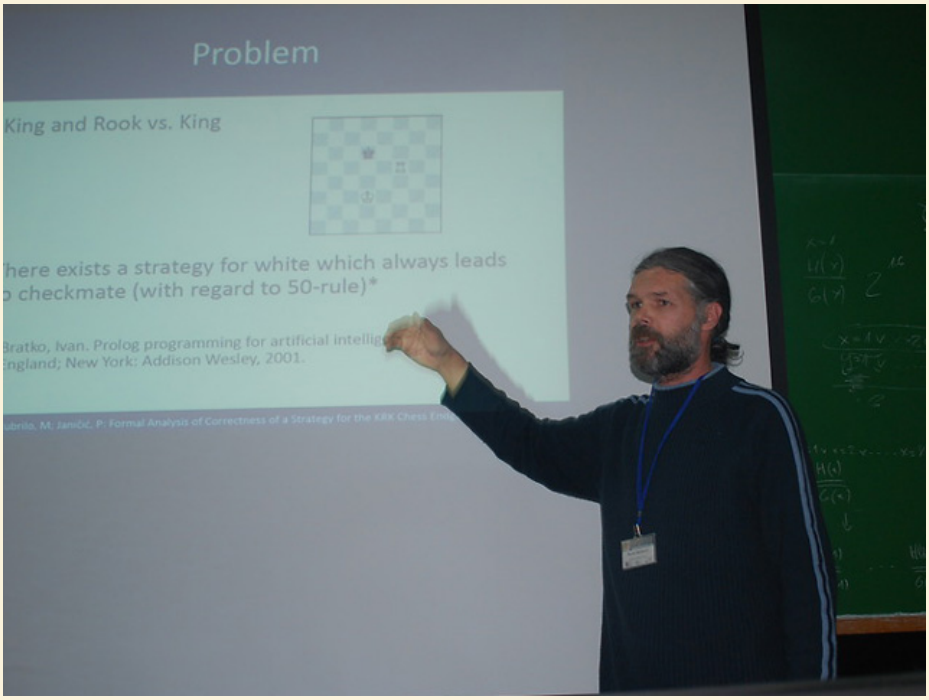
Silvia: Are confluence and termination gained immediately?
Re: This is the extension of LF, so these two come from strong normalization of LF.

Silvia: Modularity, connection for external systems?
Re: You can express basically everything using this and then optimize for that system.

Predrag: You told me that this logical framework could serve as a base for a new proof assistant. What would be most significant benefits of such proof assistant?
Re: Such proof assistant could easily use various external tools.

---

8 Formal Analysis of Correctness of a Strategy for the KRK Chess Endgame



Marko Maliković
University of Rijeka, Croatia

— Joint work with Mirko Čubrilo and Predrag Janičić —

Abstract

We present our ongoing formalization in Coq of one chess endgame (KRK), including a correctness proof for one strategy for this endgame. The motivation for this formalization was making first steps in formalization of the whole of the chess game and first formal, rigorous analysis of chess problems. We show that most of the considered notions and conjectures can be expressed in a simple theory of linear arithmetic. Also, the considered problem serves as a test case for exploring limits of automation for linear arithmetic in Coq. It turned out that even simple decision procedures can be very useful, with additional techniques, in reasoning about non-trivial problems.

Slides:

`http://argo.matf.bg.ac.rs/events/2012/fatpa2012/slides/MarkoMalikovic.pdf`

Discussion

Julien: Within Coq, there are other decision procedures for linear arithmetic — romega and micromega, there should be more efficient, have you tried them?
Re: Yes, we are aware of them but we are still not happy with them. First, they are procedures for linear arithmetic over integers, so our formalization should be somewhat changed. Second, although more efficient,

they are still not much more efficient than omega.

Oliver: When you finish this, will you be able to solve other endings, some more or less complicated. There are enumeration tables for endings with 6 pieces, could you use your approach to prove correctness of strategies for such endgames?
Re: In principle yes, but even KRK ending is very complex for formal analysis.

Filip: Some of your definitions are problem-specific, they assume there are only three pieces on the board. Could you generalize these definition, so this formalization can be used for other endings, for instance.
Re: Yes, we could but we want to deal with the simple case first. We should definitely use more general definitions later.

Predrag: Actually, we started with a general definition, but then switched to a specific one, for efficiency. Anyway, this specifics appear only in a very few places in the core of our system.

Oliver: This strategy does not ensure optimality, does it? But there are complete enumerations, even for 6-piece endings that ensure enumerations?
Re: (Predrag) No, the strategy does not ensure optimality. But, what we have is a formal, machine verifiable proof of correctness. Even it is very complex, a proof for an optimal strategy (if it exists) would be much more complex. Concerning the enumeration tables, what they give is not a strategy, not a human-understandable strategy, but only a list of optimal moves. There are some machine-learning-based approaches for extracting strategies from such complete tables.

Filip: There is symmetric cases, how do you deal with symmetry?
Re: We use symmetry to some extent, but we plan is to use it in a more systematic way.

---

9 Formalized Search for FC Families within Isabelle/HOL

Filip Marić
University of Belgrade, Serbia

— *Joint work with Bojan Vučković and Miodrag Živković* —

Abstract

Frankl's conjecture, states that for every family of sets closed under unions, there is an element is contained in at least half of the sets. FC-families are families of sets such that every union closed set that contains them satisfies the Frankl's condition (for example, every union-closed set that contains a one-element set is Frankl's, so one-element sets form the basis of simplest FC-families). In this talk we present a verified computer assisted approach for discovering FC-families. *Proof-by-evaluation* paradigm is used and proof assistant Isabelle/HOL is used both to check mathematical content given in the paper, and to perform (verified) combinatorial computations on which the proofs rely. All known FC-families are confirmed, including a new FC-family, discovered recently using unverified computer assisted approach.

Slides:

`http://argo.matf.bg.ac.rs/events/2012/fatpa2012/slides/FilipMaric.pdf`

Discussion

Oliver: Your formalization makes the mathematical content precise and clear, but what about the algorithmic content?
Re: The algorithmic content is also made very precise. Algorithms are given in an abstract form, their correctness is proved, and then they are refined to efficiently executable versions.

Oliver: I mean the abstract content is lost?
Re: I think it is covered by the is algorithm (shows the slide), there is a lemma proving this is correct.

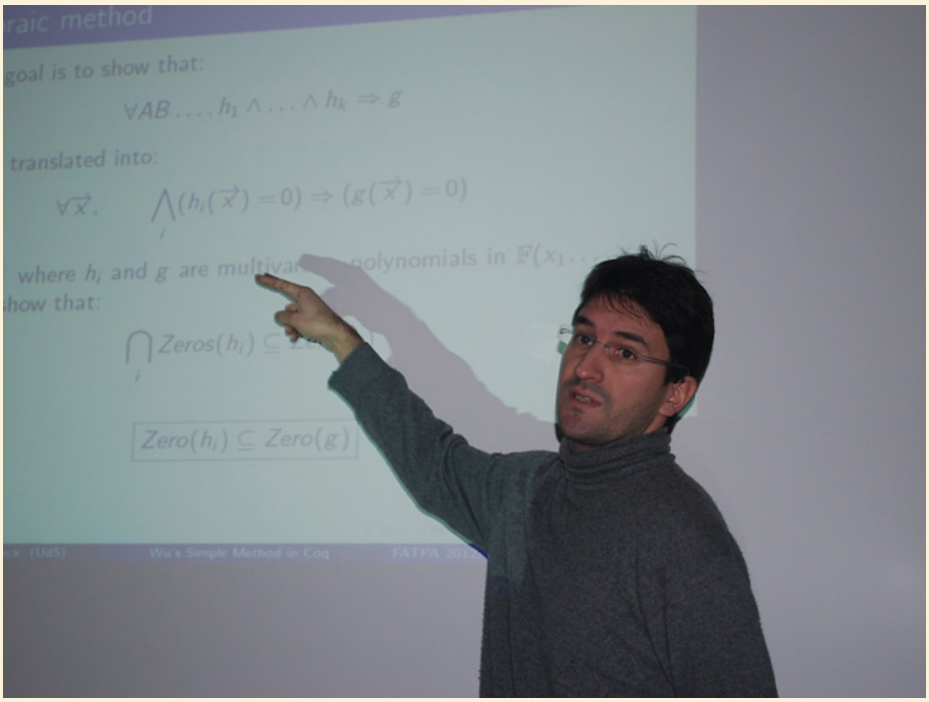
Julien: How far can you go with computations done within a theorem prover (without code generation)?
Re: I am not sure that Isabelle supports doing computations without code generation (but that is an Isabelle question). For computations I use "by eval" method for which I think that it employs the code generator and performs the computations on the ML level.

Predrag: For finite cases, maybe the problem could be reduced to SAT. Then, with your verified SAT solvers, you would have verified answer. Have anyone used reduction to SAT so far? And, given your verified SAT solvers, would you accept this kind of a proof?
Re: For analyses for up to 11, computers were not used at all. In order to use reduction to SAT, there must be a simple reduction mechanism, so one can believe there is no error in the problem encoding.

Geometry Reasoning

Session Chair: Filip Marić

10 Formalization of Simple Wu's Method in Coq



Julien Narboux
University of Strasbourg, France

Abstract

We present the integration within the Coq proof assistant of a method for automatic theorem proving in geometry. We use an approach based on the validation of a certificate. The certificate is generated by an implementation in Ocaml of a simple version of Wu's method.

Slides:

http://argo.matf.bg.ac.rs/events/2012/fatpa2012/slides/JulienNarboux.pdf

Discussion

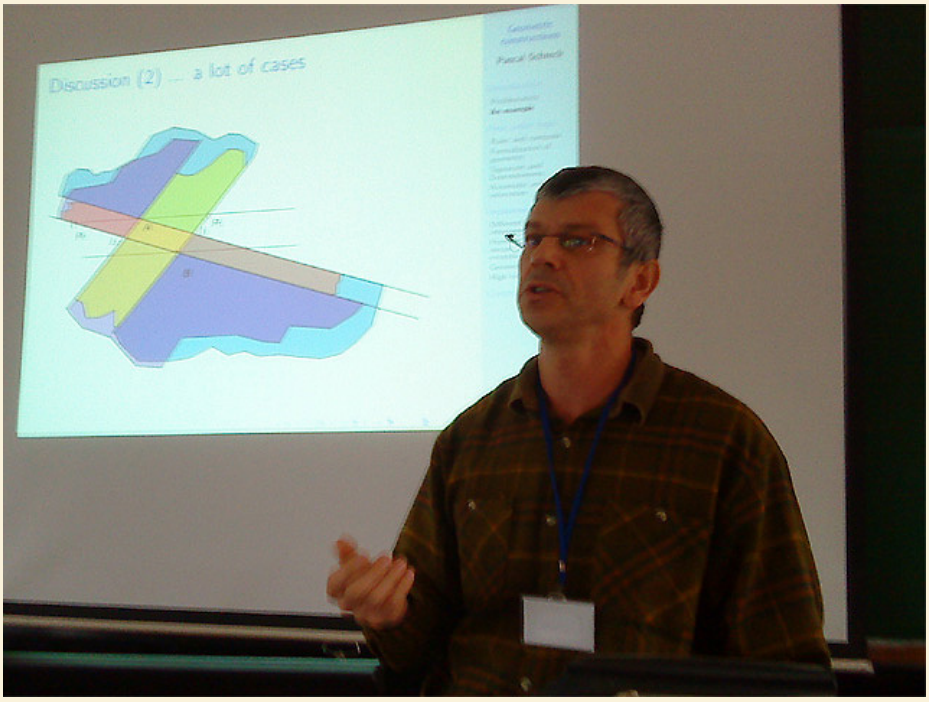
Predrag: Your geometry primitives are translated into algebraic form. Are all these translations individual, or there is some relationship between them? Also, some of these translation could be redundant if we know some geometry relationships between the geometry primitives. For instance, the segment bisector is perpendicular on the segment, and passes through the segment midpoint.
Re: All geometry primitives are translated individually. We could derive them from one another, but then we may get some more complex algebrizations.

Predrag: Your NDGs have algebraic, not geometric form. But, if an external tool can translate NDGs into geometric form, then you could use that translated form and only check, within Coq, if it is ok. That would be in the same spirit as using certificates.
Re: Yes, it is possible. The difficulty is in finding a geometric form of NDG conditions.

Filip: You use the function `line P1 P2` that constructs a line through two points. Is this function total? What does it return when `P1=P2`?
Re: That is why here we use `line_parallel` instead of `parallel`. The semantics of `line_parallel (line P1 P2) (line P3 P4)` is such that it corresponds to "either lines are parallel or points are equal".

---

11 Geometric Constructions and First Order Logic



Pascal Schreck
University of Strasbourg, France

Abstract

This talk is about an ad hoc first order formalization of geometric constructions in the style of the well known "ruler and compass" constructions. Considering some examples, we propose a multi-sorted framework and a syntax which allows to define the notion of program of constructions. Then, we propose a first order formalism to automatically synthesize such programs from a geometric statement. We describe then an implementation of these ideas in Prolog.

Slides:

http://argo.matf.bg.ac.rs/events/2012/fatpa2012/slides/PascalSchreck.pdf

Discussion

Predrag: You construction programs are somehow ,,correct by construction"?
Re: Yes.

Predrag: Can they be verified and accompanied by a correctness proof? And, actually, there are two issues: (i) that the constructed objects exist — this cannot be solved algebraically; (ii) that the constructed objects meet the specification — for this we can use algebraic methods.
Re: We can have verification after the construction. That is something we want to do with Narboux.

Filip: You have ad-hoc axiomatizations — what is the relationship with Hilbert and Tarski's axiom systems?

Re: This is an effective system of axioms that give explanations, but it is not minimalistic like Tarski's one.

Predrag: Concerning axiom systems for construction problems, there are systems by Victor Pambuccian.

---

12 Automated Solving of Triangle Construction Problems

Predrag Janičić
University of Belgrade, Serbia

— joint work with Vesna Marinković —

Abstract

We present a new approach for automated solving of triangle construction problems. The approach relies on: (*i*) off-line, manual analysis of one set of problems leading to the set of relevant geometry knowledge; (*ii*) on-line simple, forward search over automatically generated set of primitive constructions steps (obtained by the first phase); (*iii*) post festum verification by a geometry theorem prover. The approach leads to a small search space (contrary to other relevant approaches) and efficient problem solving, but also to better understanding of the relevant mathematical knowledge and its suitable representation. We implemented the approach and it can solve a number of triangle construction problems appearing in the literature.

Slides:

`http://argo.matf.bg.ac.rs/events/2012/fatpa2012/slides/PredragJanicic.pdf`

Discussion

Mladen: I remember that in earlier stages you combined backward and forward search. What happened with that?
Re: Yes, indeed, we used some combination, but et the end it turned out that a simple forward search gives equally good results. Now we think that the main problem is not in the search mechanism, but in a suitable knowledge representation.

Julien: You say that you need to prove that the constructed points really exist, what you exactly need to prove?
Re: We need to detect conditions under which these points really exists. These conditions are actually close to non-degeneracy conditions that can be provided by an algebraic prover. And then, we can prove that under these conditions, the required points really exist.

Walther: Your approach concentrates on triangles — what is needed to extend it for four points etc?
Re: We focus on triangle construction problems in order to try to locate the main problems and define a solving methodology. Once we have that, we should be able to easily apply it to some wider ranges of problems. Probably not as wide as general problems considered by Pascal, but still rather wide.

Pascal: How do you deal with symmetries?
Re: We don't deal with symmetries in the search phase at all. We derive all ground rules, all instantiations in the preprocessing phase. It seems inefficient but the problem isn't in the search.

---

13 Exploiting symmetries and axiom reformulation in automated generation of formal proofs

Sana Stojanović
University of Belgrade, Serbia

Abstract

We present our current work on ArgoCLP, a theorem prover based on coherent logic that produces formal proofs in Isabelle and readable proofs in English. We focus on dealing with symmetric predicates and with axioms that introduce more than one witness. The used techniques improved efficiency of the prover.

Slides:

http://argo.matf.bg.ac.rs/events/2012/fatpa2012/slides/SanaStojanovic.pdf

Discussion

Oliver: Do you consider only full symmetries or also symmetries on certain argument positions?
Re: Currently, I consider only full symmetries, but the approach can be used of other sorts of symmetries. For instance, the predicate between is symmetric only on its first and third argument, but this follows directly from the axioms. We are more interested in proving symmetries in other cases.

Hugo: Many theorem provers deal with symmetries and work modulo symmetry. Does your prover do that?
Re: (Predrag) Yes, in our prover ArgoCLP, canonical form of symmetrical predicates are considered. This Sana's work addresses the problem of checking if some predicate is symmetrical.

Filip: So you perform unification modulo in your system?
Re: Yes

14 Integration of OpenGeoProver with GeoGebra

Ivan Petrović
University of Belgrade, Serbia

— *Joint work with Predrag Janičić* —

Abstract

We present our project on integration of OpenGeoProver, our Java implementation of simple Wu's method for automated theorem proving in geometry, and GeoGebra, a widely used dynamic geometry software. The integration is flexible, and the developed interface could also be used for linking OpenGeoProver to other dynamic geometry tools, or linking GeoGebra to other theorem provers. OpenGeoProver accepts an XML representation of a construction and a conjecture described in GeoGebra, decides (if it can) whether the conjecture is valid and returns a report, including NDG conditions in geometry form.

Slides:

`http://argo.matf.bg.ac.rs/events/2012/fatpa2012/slides/IvanPetrovic.pdf`

Discussion

Walther: Do you have contacts with Zoltan and other members of GeoGebra team?
Re: Yes, we are in contact, and this integration should be made in collaboration with them.

Julien: Your experimental comparison is not perfect if JGEX implements the full Wu's method and OpenGeoProver implements Wu's simple method.
Re: Yes, the comparison is not perfect. Still, we wanted to make a rough comparison with the simplest available system.

Pascal: Is it possible to modify the prover so it accepts other types of objects (e.g. ellipse). Does one have to implement that?
Re: Yes, it is possible since the implementation is modular. For additional objects, support must be implemented.

Pascal: Constructions in dynamic geometry tools are performed by mouse. Does your prover have some graphical interface?
Re: No, it is intended to be integrated with other sorts of tools.

Predrag: The prover is self-contained but still it uses some libraries, like for XML. If you eliminate such libraries, the prover will be small?
Re: Yes, it will be very small.

Applications of Theorem Proving

Session Chair: Predrag Janičić

15 How Theorem-Prover Technology Advances Educational Math Software — Lessons Learned from Preparation of a Grant Proposal

Walther Neuper
Graz University of Technology, Austria

Abstract

In a few years every student will have powerful math tools in some kind of standard hand-held device. The ubiquitous availability of those tools will intensify the controversial discussion about their benefits/drawbacks for math education.

If accepted, a submitted project will deliver prototypes of a new generation of educational math software based on Theorem-Prover (TP) technology: What are the novel aspects this generation will contribute to the discussion mentioned?

The talk will approach this question from an educational and from a technical point of view, drawing from experiences gained from the contacts with practitioners in education and with academic TP-experts during preparation of the proposal.

Slides:

http://argo.matf.bg.ac.rs/events/2012/fatpa2012/slides/WaltherNeuper.pdf

Discussion

Oliver: I think students should not learn mathematics by clicking, but by writing by hand. Students have problem with concentration because of plenty information that they can find on Internet.

Re: Interested students can go down and change level of materials. Interesting questions are not answered on classes and this software can answer these questions. A majority of students, 80%, are those who are not interested.

Predrag: You have a rich experience — are students in high-school ready for interactive theorem proving and at what level?
Re: Never! There is no way to present interactive theorem proving to all students. At least 80% are not ready for that.

---

16 Automated Evaluation of Students' Programs: Testing, Verification and Similarity

Milena Vujošević-Janičić
University of Belgrade, Serbia

— joint work with Mladen Nikolić and Dušan Tošić —

Abstract

We present our idea for automated evaluation of students' programs. A tool that could help students and teachers to evaluate quality of students' programs would be mutually beneficial. For students, such tool would be helpful when there is no teacher to check the solution (which is, most of the time, the case). For teachers, such tool would be helpful in marking exams and for pointing to standard errors. The approach integrates three features: testing, verification and measuring similarity of programs. Testing indicates functional correctness of a program. Verification points to errors such as buffer overflows, division by zero, and null pointer dereferencing. Similarity of student's program to the teacher's solution should give information on program' modularity and structural simplicity.

Slides:

`http://argo.matf.bg.ac.rs/events/2012/fatpa2012/slides/MilenaVujosevicJanicic.pdf`

Discussion

Danijela: Your program for evaluating student's programs is not user friendly. Is it possible to write an application such that other teachers would easily use it?
Re: Yes, hopefully within in the future work.

Oliver: My experience is that it is extremely difficult to make students obey the specification and required outputs. Once I did some web-based system for assessment and it had around 100000 commits by around 100 students, truing to get the output syntax correct. Eventually, I gave it up.
Re: I agree. Student were not trained for this sort of evaluation, so if the formulation was not precise, there was a lot of various errors, too difficult to predict them all. But if students can use this program, on a final exam they would make less errors.

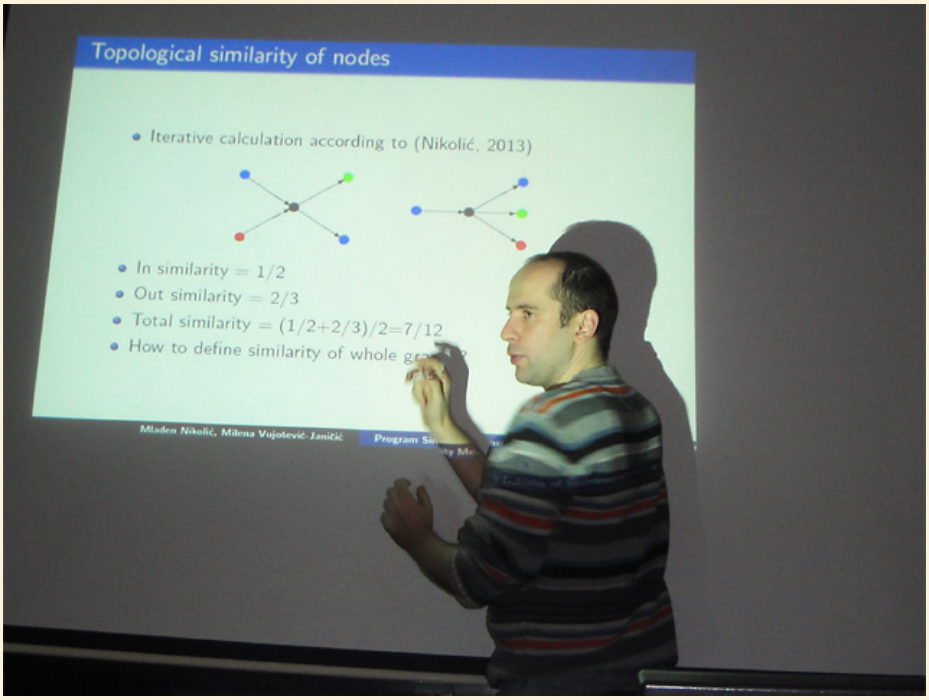
Julien: I think it is not good to use this system for evaluation, but rather for training. Do you think everything should be available to the students?
Re: Not everything, some options could be hidden. For example, some test cases could be hidden and then used when evaluating.

Filip: I think that training and evaluation environment should be the same. Btw, concerning your example, I find the first one substantially different than the remaining ones. The first one is about modularity, and the next two about efficiency. Issues of modularity and the style should be evaluated by a human. I am very skeptical about computer evaluating the style. Efficiency issues can be measured with a profiler.
Re: In the first example, it is not only a matter of style, but rather checking if they know what we taught them — modularity. This is elementary course, no many solutions, maybe 4 or 5 exist, I think we can evaluate efficiency.

Danijela: Is it possible to use your tool for other programming languages?
Re: Yes, the system uses LLVM code representation and there is LLVM support for C++, Fortran, Python, etc. We tested it for Fortran and it was successful. Still, for C++, some support in my tools should still be implemented.

---

17 Program Similarity Measurement for Evaluation of Students' Programs



Mladen Nikolić
University of Belgrade, Serbia

— joint work with Milena Vujošević-Janičić —

Abstract

We present our ongoing work on measuring program similarity based on their control flow graphs using known graph similarity measurement techniques. We hope to tackle the problem of evaluating program modularity and structural simplicity. A possible application is automated evaluation of students' solutions to programming problems by comparison with a teacher provided solution. The first evaluation results for our approach are encouraging.

Slides:

http://argo.matf.bg.ac.rs/events/2012/fatpa2012/slides/MladenNikolic2.pdf

Discussion

Filip: Did you try manual checking, were all of the considered programs correct?
Re: No, we have not made a manual checking.

Filip: My experience is that students sometimes provide programs completely unrelated to the given specification. So, your results are maybe even better on the set of programs that do something meaningful.

Oliver: How do you measure similarity, I would like to have some axiomatic approach?
Re: I had that as a motivation. There are some natural properties that the existing methods do not possess. I took those as the requirements for my method before it was developed.

Oliver: Did you look at programs for checking plagiarism? I think they do something simpler.
Re: I have to look at them for more details.

Filip: One remark, I tested plagiarism between student works, and simple comparing of size of programs and frequencies of keywords gave good results, because the students only change the names of variables.

Predrag: What if student has long but efficient solution? Maybe you could penalize different solution only if it is less efficient?
Re: Yes, we could consider something like that.

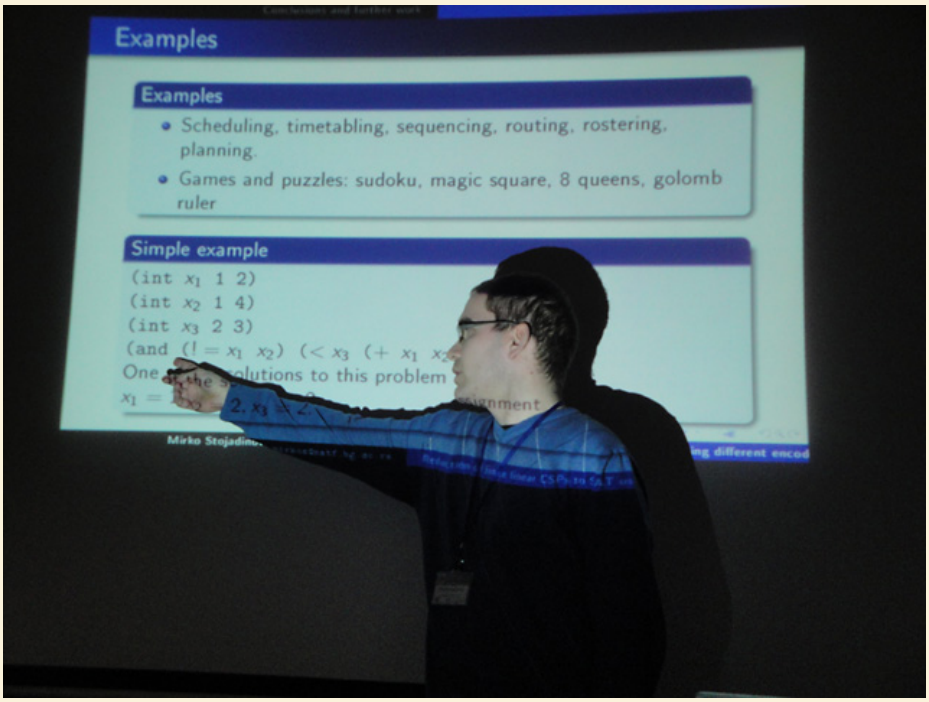
Milena: I disagree, the important aim of the courses is to teach students to split programs in functions and organize code. And efficiency can be improved by simply putting everything in the main function. We have to check what we taught them.

Mladen: I think that this kind of evaluation is simply not suitable for considering efficiency. It deals with modularity, and if you want, you can assess the efficiency separately, and then decide what you want to do with these two measures.

Oliver: You could use some threshold for speed-up.

---

18 Reduction of finite linear CSPs to SAT using different encodings



Mirko Stojadinović
University of Belgrade, Serbia

Abstract

One approach in solving Constraint Satisfaction Problems (CSP) and Constraint Optimization Problems (COP) is reduction to propositional satisfiability (SAT). A number of encoding methods (e.g., direct, log, support, order) for this purpose exist, but there is no single encoding that performs the best on all classes of problems. We present a system that translates specifications of finite linear CSP or COP problems into SAT instances using several well-known encodings. Encoding into Satisfiability Modulo Theory (SMT) is also supported as well as specific encodings for some global constraints (cardinality constraints, alldifferent). The tool can be used for experimental comparison of different SAT encoding schemes within a single platform and it can enable choosing the most efficient SAT representation for practical problems that are being solved.

Slides:

http://argo.matf.bg.ac.rs/events/2012/fatpa2012/slides/MirkoStojadinovic.pdf

Discussion

Predrag: What solvers were used in comparisons? You used the same solver both for your reductions and for sugar's reduction?
Re: Yes, I used minisat in all cases. I used Yices for reduction to SMT.

Marko: What was the largest magical square that you constructed? It would be interesting to see the behavior on larger dimensions?

Re: The largest dimension that we have is 12. For larger dimensions, programs were not able to finish in the given timeout.

Workshop Photos

Working session, February 3, 2012.

Working session, February 3, 2012.

Working session, February 3, 2012.

Lunch at Teatroteka, February 3, 2012.

Dinner at Klub književnika, February 3, 2012.

Dinner at Klub književnika, February 3, 2012.

Lunch at Teatroteka, February 4, 2012.

Lunch at Teatroteka, February 4, 2012.

Dinner at Little Bay, February 4, 2012.

Dinner at Little Bay, February 4, 2012.

Little Belgrade City Guide for Workshop Participants

Workshop Site

The workshop site is the building of the faculties of sciences of the University of Belgrade. It is located in the very city centre and close to Kalemegdan fortress, and the rivers Danube and Sava. The workshop site

is just 300m from the Knez Mihajlova street and the surrounding pedestrian zone, with a large number of impressive buildings and mansions built in XIX and XX century in the style of neoclassicism, academism, secession, and art-deco. Just 200m from the workshop site are remains of large Roman termes (built in III centrury) and 100m away is Sheikh Mustapha's turbeh (Turkish mausoleum; erected in XVIII century over the tomb of this religious figure), to name just a few intersting sights that are nearby.

Brief History of Belgrade

Belgrade, a city of very turbulent history, is one of the oldest cities in Europe. Its history lasts full 7000 years. The area around two great rivers, the Sava and the Danube has been inhabited as early as palaeolithic period. Remains of human bones and skulls of Neanderthals, found in Belgrade date back to the early Stone Age. The founding of Singidunum (the ancient name of Belgrade) is attributed to the Celtic tribe, the Scordiscs. Singidunum was mentioned for the first time in 279 B.C. The first part of the word - Singi - means "round" and dunum means "fortress" or "town". The Romans conquered Belgrade in the beginning of the I century A.D. and it has been under their rule for full four centuries. The Huns captured the town and completely destroyed it in 441. After the fall of the Huns, the town became a part of the Byzantine Empire in 454, but it was soon conquered by the Sarmatians, and later the Eastern Goths. In 488, it became a Byzantine town again. Around 630, the Serbian settlers come to this area. The town was first mentioned under the Slavic name Beograd (White Town - probably because of the walls made of white limestone) in 878. The Serbian rule over Belgrade began in 1284. but during some periods it was under Hungarians again. After almost a century of resisted sieges and attacks, Belgrade fell to Turks's rule in 1521. The town, getting more and more oriental look, counted in XVII population of 100000 and was the second-largest town in the Empire, right after Istanbul. The Austrians conquered Belgrade in 1688. When in 1739 it was captured again by the Turks, it was exposed to a heavy destruction. After two Serbian insurrections (started in 1804 in 1815) and the period of weakening of their power in Serbia, the Turks left Belgrade for good in 1867. In World War I, the Austrian army conquered the city in October 1915. The Serbian army and parts of the Allies' army liberated Belgrade in 1918. During WWI, Serbia lost 28% of its whole population, while Belgrade was the most destroyed town in Serbia. After the liberation, Belgrade became the capital of the newly-created Kingdom of the Serbs, Croats and Slovenes (later called Yugoslavia). In April 1941, Belgrade became the target of a terrible destruction by German air force. Belgrade also had to undergo losses in the Allies' bombing, especially in 1944. During World War II Belgrade lost about 50000 citizens and suffered inestimable damage. Belgrade was liberated by the units of the National Liberation Army of Yugoslavia and the Red Army on October 20, 1944. The monarchy in Yugoslavia was abolished in 1945 when the communist rule of Josip Broz Tito started. Thanks to a specific policy of Yugoslavia, Belgrade became an important international, political, cultural, sports, and economic center, linking East and West, North and South. Many unsolved national problems led to disintegration of Yugoslavia in 1991 and since 2006, the Republic of Serbia is independent state with Belgrade as its capital.

Briefly About Modern Belgrade

Belgrade is the capital and the largest city of Serbia. The city lies at the confluence of Sava and Danube rivers. With a population of almost two million, Belgrade is the third largest city in Southeastern Europe. The architecture of Belgrade is a mirror of different cultural and historical periods, influences and styles: from old Oriental influences, across baroque architecture, secession, academism and neoclassicism, socialist and industrial features from post WW2 period, to modern architecture and layout of New Belgrade with wide boulevards. Knez Mihajlova Street is the main walking street in Belgrade. It is a pedestrian zone, protected by law as one of the most valuable monumental complexes of the city. Belgrade has many beautiful parks and the biggest one is Kalemegdan, with an old fortress, comprising remains from Ancient and Byzantine times to Turkish and Austro-Ugrian periods. Belgrade has more than 20 theaters and two opera houses and it is home to a number of film, theater, and music festivals. There are many excellent restaurants, cafés and pubs, and British Times proclaimed Belgrade as Europe's best nightlife city.

Knez Mihajlova Street

Knez Mihajlova Street, pedestrian precinct and main city street, now protected by law, is one of the oldest and most valuable city environments, with a whole range of impressive buildings and town houses which sprung up at the end of the 1880's. It is generally believed that as early as Roman times this was the centre of the settlement of Singidunum, while during Turkish rule the streets went through the gardens, fountains and mosques that stood in this part of town. Today it is the main business area of Belgrade and the headquarters of many national institutions (such as the Serbian Academy of Science and Arts, Belgrade City Library and the Belgrade Cultural Centre).

Kalemegdan Park and Fortress

Kalemegdan is the core and the oldest section of the urban area of Belgrade and for centuries the city population was concentrated only within the walls of the fortress, thus its history, until most recent history, equals the history of Belgrade itself. The name Kalemegdan derives from two Turkish words, kale (fortress) and megdan (battleground) (literally, "battlefield fortress"). Kalemegdan fortress is the most important cultural-historical complex in the city, standing above the Sava-Danube confluence. Since its construction the Belgrade fortress has been constantly attacked and defended, destroyed and renovated. Chronicles trace a history of about 40 to 60 devastations of the fortress. The landscaping of the wide plateau around the fortress was begun on the order of Prince Mihailo Obrenovic after the fortress had been handed over from the Turks to the Serbs in 1867. and it was converted into a park in the 1880's. Today, Kalemegdan park is the largest and loveliest park in Belgrade with an area of 52 hectares. There is a number of monuments, Sahat Tower, the Military Museum, the statue of Belgrade Victor, the Zoo.

Serbian Alphabet

Serbian is a South Slavic language. Both Latin and Cyrillic alphabets are used to write Serbian. Serbian is an example of synchronic digraphia. The orthography, introduced by the language reform led by Vuk Karadžić in mid XIX century, is very consistent: it is an approximation of the principle "one letter per sound". The following table gives 30 letters used in Serbian, both in Cyrillic and in Latin alphabet.

А а	Б б	В в	Г г	Д д	Ђ ђ	Е е	Ж ж	З з	И и	Ј ј	К к	Л л	Љ љ	М м
A a	B b	V v	G g	D d	Đ đ	E e	Ž ž	Z z	I i	J j	K k	L l	Lj lj	M m
Н н	Њ њ	О о	П п	Р р	С с	Т т	Ћ ћ	У у	Ф ф	Х х	Ц ц	Ч ч	Џ џ	Ш ш
N n	Nj nj	O o	P p	R r	S s	T t	Ć ć	U u	F f	H h	C c	Č č	Dž dž	Š š