Report from Dagstuhl Seminar 14482

Automated Planning and Model Checking

Edited by

Alessandro Cimatti[1], Stefan Edelkamp[2], Maria Fox[3], Daniele Magazzeni[4], and Erion Plaku[5]

1 **Bruno Kessler Foundation – Trento, IT,** `cimatti@fbk.eu`
2 **Universität Bremen, DE,** `edelkamp@tzi.de`
3 **King's College London, GB,** `maria.fox@kcl.ac.uk`
4 **King's College London, GB,** `daniele.magazzeni@kcl.ac.uk`
5 **CUA – Washington, US,** `plaku@cua.edu`

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 14482 "Automated Planning and Model Checking". There has been a lot of work on the exchanges between the areas of automated planning and model checking, based on the observation that a model-checking problem can be cast as a planning problem and vice-versa. The motivation for this seminar was to increase the synergy between the two research communities, and explore recent progress in the two areas in terms of techniques, tools and formalisms for describing planning and verification problems. The main outcomes were a greater common understanding of planning and model-checking issues and challenges, and greater appreciation of the crosswover between the modelling languages and methods. Different application domains were also explored, where planning and model-checking can be effectively integrated.

Seminar November 23–28, 2014 – http://www.dagstuhl.de/14482
1998 ACM Subject Classification I.2.8 Problem Solving, Control Methods, and Search, D.2.4 Software/Program Verification
Keywords and phrases planning via model checking, directed model checking, plan validation, falsification, GPU-based state space exploration, hybrid systems, heuristic search, SMT
Digital Object Identifier 10.4230/DagRep.4.11.227

1 Executive Summary

Alessandro Cimatti
Stefan Edelkamp
Maria Fox
Daniele Magazzeni
Erion Plaku

In the area of formal methods, model checking deals with the problem of fully-automated property validation and correctness verification. Given a formal model of a system and a property specification, the task is to explore the state space and verify whether or not the property is satisfied by the model. In artificial intelligence, automated planning deals with the automatic generation of plans for achieving a goal. Given the description of the initial state, the goal state, and the set of possible actions, a planner uses heuristic search to look for a sequence of actions that transforms the initial state into the goal state.

Automated Planning and Model Checking, *Dagstuhl Reports*, Vol. 4, Issue 11, pp. 227–245
Editors: Alessandro Cimatti, Stefan Edelkamp, Maria Fox, Daniele Magazzeni, and Erion Plaku
Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

There has been a lot of work on the exchanges between the two areas (automated planning and model checking), based on the observation that a model-checking problem can be cast as a planning problem, where the goal state is a state violating the property to be verified in the model-checking problem. Thus, if a plan is found by the planner, it corresponds to error trace that a model checker would return (this paradigm is called directed model checking). The link can also be exploited in the other way around, using a model checker to search the planning state space, stopping the search when a goal state is found (this paradigm is called planning via model checking).

The general aim of this Dagstuhl Seminar was to increase the synergy between the two research communities. This involved sharing views, thoughts and contributions across the following streams:

Techniques and Tools: During the seminar we considered the most recent advances in automated planning and model checking and explored the possibility of using recent planning tools (heuristic search, sampling-based motion-planning algorithms, symbolic search algorithms, ...) for system falsification (particularly in challenging domains such as hybrid systems) and for boosting the use of model checking systems for finding plans.

Modelling Languages: One of the goals of the seminar was to consider the family of PDDL languages and the formalisms for describing verification problems and discuss how to make the communication between the two areas easier, exploring the possibility of common languages or translation between existing formalisms.

There were a number of talks on "X" Modulo Theories, where "X" ranged from SAT and Search to Planning. These talks explored the relationship between generic solution methods and different proof systems, leading to discussions about the relative benefits of viewing problems from the perspectives of the different generic methods. This fostered an improved understanding of each other's perspectives and modelling approaches.

Discussions also focused on relationships between hybrid planning and hybrid model-checking. There were tutorials on the modelling languages used in the two paradigms, and their semantics, and the discrepancies between these led to lively discussion. In hybrid planning using PDDL, a key semantic issue is the use of epsilon time to separate inter-dependent actions, in order to prevent the planner from relying on synchronised activity. For example, if an action, A, achieves the precondition of another action, B, the validity of the plan depends on A being ordered strictly before B, by at least epsilon time. This is because the state following the co-occurrence of these two actions is indeterminate. The model-checking community does not require this epsilon. In hybrid model-checking a partial order on events is maintained, and there was an extended discussion about why planning forces an ordering using epsilon separation when this is not necessary in model-checking. From this discussion the following distinction emerged: model-checking simply requires there to be a single ordering of events that is consistent with the constraints, as this provides the required counter-example to the correctness of the model. In planning, by contrast, *all* orderings of events must be consistent with the constraints, requiring exponential work to check the validity of a partially ordered plan. Once this point was understood there was a greater common understanding between the planning and model-checking proponents, and greater appreciation of the crossover between modelling languages and methods.

Other topics covered by the contributed talks include:

- directed model checking and falsification
- plan validation
- heap and other data structures

- GPU-based state space exploration
- hybrid systems
- heuristic search
- planning and verification on real-world scenarios

The program featured the following components:

- On Monday we started with 7 tutorial-type introductory talks about plans validation, planning in hybrid systems through model checking, guided search for hybrid systems, falsification of hybrid systems through motion planning, planning via symbolic model checking, directed model checking of timed systems, heap implementations. The purpose of these tutorials was to familiarise members of the different communities with the basics of the other fields and with the existing synergies between the fields.
- From Tuesday, each day featured one or two long talks plus a number of short talks, with enough time for discussion after each talk.
- On Tuesday and Thursday afternoon we had two open discussion sessions.
- Wednesday afternoon featured a hike.

A feature of this seminar was the very high level of engagement and interaction between the participants, leading to a lively and productive week. The decision not to formalise discussions into panels or break-out sessions proved to be a good one, allowing more flexible response to topics as they arose. Similarly, the decision to leave some of the talk slots open allowed spontaneous pursuit of ideas that came out of discussions. The mixture of long and short talks also encouraged this. The workshop ended on a high note, with many new ideas for collaboration having been identified.

2 Table of Contents

3 Overview of Talks

3.1 Guided Search for Hybrid Systems

Sergiy Bogomolov (Universität Freiburg, DE)

Hybrid systems represent an important and powerful formalism for modeling real-world applications such as embedded systems. A verification tool like SpaceEx is based on the exploration of a symbolic search space (the region space). As a verification tool, it is typically optimized towards proving the absence of errors. In some settings, e. g., when the verification tool is employed in a feedback-directed design cycle, one would like to have the option to call a version that is optimized towards finding an error path in the region space. A recent approach in this direction is based on guided search. Guided search relies on a cost function that indicates which states are promising to be explored, and preferably explores more promising states first. In this talk, we present two approaches to define and compute efficient cost functions. We develop our approaches on the top of the symbolic hybrid model checker SpaceEx which uses regions as its basic data structures.

In the first part of the talk, we introduce a box-based distance measure which is based on the distance between regions in the concrete state space. In the second part of the talk, we discuss an abstraction-based cost function based on pattern databases for guiding the reachability analysis. For this purpose, a suitable abstraction technique that exploits the flexible granularity of modern reachability analysis algorithms is introduced. We illustrate the practical potential of our approaches in several case studies.

3.2 Transitive Reduction for Inference of Biological Networks on GPUs

Dragan Bošnački (TU Eindhoven, NL)

Joint work of Bošnačkii, Dragan; Odenbret, Maximilan R; Wijs, Anton; Ligtenberg, Willem; Hilbers, Peter;
Main reference D. Bošnački, M. R. Odenbrett, A. Wijs, W. Ligtenberg, P. Hilbers, "Efficient reconstruction of biological networks via transitive reduction on general purpose graphics processors," BMC Bioinformatics, 13:281, 2012.
URL http://dx.doi.org/10.1186/1471-2105-13-281

Transitive reduction [1] is a graph transformation which is inverse of transitive closure. A transitive reduction of a graph is a minimal graph (in the number of edges) that comprises the same nodes as the original graph while preserving the same connectivity of the original graph, i. e., there is a path between two nodes a and b in the original graph G if and only if there is a path between a and b in its transitive reduction G'. In bioinformatics transitive reduction arises in the problem of reconstruction of biological networks from so called perturbation experiments. The goal is to distinguish direct from indirect interactions, i. e., to remove each direct connection between two network nodes that can be explained as a chain of indirect interactions (path in the weighted graph). In this context we introduce transitive reduction for weighted graphs along the lines of [2]. The transitive reduction algorithm for weighted graphs is isomorphic with the Floyd-Warshall shortest path algorithm and as such it lends itself to parallelization. We present some efficient implementations of this algorithm and its variations on general purpose graphics processing units (GPUs). We also discuss how

transitive reduction can be possibly used in planning and model checking, e. g., in the context of computing canonical representations of zones in timed automata.

References

1 A. V. Aho, M. R. Garey, J. D. Ullman, *The transitive reduction of a directed graph*, SIAM J Comput 1972, 1(2):131–137, http://epubs.siam.org/doi/abs/10.1137/0201008

2 D. Bošnački, M. R. Odenbrett, A. Wijs, W. Ligtenberg, P. Hilbers, Efficient reconstruction of biological networks via transitive reduction on general purpose graphics processors, BMC Bioinformatics 2012, 13:281, http://dx.doi.org/10.1186/1471-2105-13-281

3.3 All You Want to Know About Heaps

Stefan Edelkamp (Universität Bremen, DE)

Joint work of Edelkamp, Stefan; Chen, Jingsen Elmasry, Amr; Katajainen; Jyrki; Larsson Träff, Jesper; Weiß, Armin

Main reference H. Kaplan, R. E. Tarjan, U. Zwick, "Fibonacci heaps revisited," arXiv:1407.5750v1 [cs.DS], 2014.

URL http://arxiv.org/abs/1407.5750v1

The talk surveys theoretical and practical results for state-of-the-art heap implementations.

Starting with introducing binary, strong, and weak heaps, we cover

- constant-factor-optimal sequential sorting with WeakHeapsort, QuickHeapsort and QuickXsort; the latter taking at most $n \lg n - 1.3999n + o(n)$ element comparisons on average,
- constant-factor-optimal adaptive heap sorting requiring at most $n \lg(Inv/n) + O(n)$ element comparisons and exploiting Cartesian trees as well as buffered heaps.

We observe (and prove) that – in contrast to a heap – repeated insertion in a weak heap requires at most $3.5n + o(n)$ element comparisons. There are other algorithm engineering aspects for heap construction (e. g. depth-first or bottom-up sift-downs) that reduce the number cache misses and branch mispredictions.

By modifying the heap construction algorithms of Gonnet and Munro and of McDiarmid and Reed, both requiring $O(n)$ extra space, via transforming weak heaps (or a navigation piles) into heaps, we contribute two *in-place* heap construction algorithms, where the extra space is limited to most $O(1)$ words. The algorithms match the best-known worst-case and average-case bounds, namely $\sim 1.625n$ and $\sim 1.52n$ element comparisons, respectively.

Strong heaps are used as a building block for *optimal in-place heaps*, which offer a constant-factor-optimal *delete-min* operation with at most $\lg n + O(1)$ element comparisons in the worst-case, while keeping the *insert* operation worst-case constant time. The history of this problem shows that it remained unresolved for more than 30 years. Strong heaps offer two *sift-down* operations: one rotates subtrees, the other stretches the heap. Allowing a limited number of heap violating nodes and by strengthening the heap condition in bottom trees this construction bypasses a well-known bound for heaps. The deamortization construction of such strengthened lazy heaps is tricky: in-case the insertion buffer becomes full, the approach relabels the insertion buffer into an area for *submersion*.

Next, we turn to algorithmic engineering aspects of *single-source shortest-paths* graph search (such as node labels, a joint graph and heap node representation and the proper use of C++-templates). We study the sequence of m *decrease-key*, n *insert*, and n *delete-min* operations known to be applied in Dijkstra's algorithm and A* equipped with a consistent heuristic. Rank-relaxed weak heaps (allowing a logarithmic number of heap-violating nodes)

are presented and shown to be state-of-the-art in theory calling for at most $2m + 1.5n \lg n$ element comparisons, while the best implementation of Fibonacci heaps requires about $2m + 2.88n \lg n + O(n)$ element comparisons. It is conjectured that for handling this sequence at most $2m + n \lg n + O(n)$ element comparisons are needed.

Last but not least, we will look at the latest developments of Fibonacci heaps, comparing *enhanced lazy Fibonacci heaps* that apply naive linkage in the consolidation with a 2014 proposal of *simple Fibonacci heaps* in terms of numbers of element comparisons, pointer assignments and running times for different sequences of operations. Variants with *cascading cuts* are experimentally compared with variants that apply *cascading rank decreases*, and put in to context with an implementation of 4-ary heaps.

All algorithms have been implemented for the CPHSTL library, which compares positively with LEDA.

References

1 A. Bruun, S. Edelkamp, J. Katajainen, and J. Rasmussen. *Policy-based benchmarking of weak heaps and their relatives.* In Int. Symp. on Exp. Algorithms (SEA), volume 6049 of Lecture Notes Comput. Sci., pages 459–435. Springer, 2010.

2 D. Cantone and G. Cinotti. *QuickHeapsort, an efficient mix of classical sorting algorithms.* Theoretical Computer Science, 285(1):25–42, 2002.

3 J. Chen, S. Edelkamp, A. Elmasry, and J. Katajainen. *In-place heap construction with optimized comparisons, moves, and cache misses.* In Symposium Mathematical Foundations of Computer Science (MFCS), volume 7464 of Lecture Notes Comput. Sci., pages 259–270. Springer, 2012.

4 T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms.* The MIT Press, 3nd edition, 2009.

5 V. Diekert and A. Weiss. *QuickHeapsort: Modifications and improved analysis.* In International Symposium on Computer Science in Russia (CSR), pages 24–35, 2013.

6 R. D. Dutton. *Weak-heap sort.* BIT, 33(3):372–381, 1993.

7 S. Edelkamp and I. Wegener. *On the performance of weak-heapsort.* In STACS, pages 254–266.Springer, 2000.

8 S. Edelkamp and P. Stiegeler. *Implementing Heapsort with $n \log n - 0.9n$ and Quicksort with $n \log n + 0.2n$ comparisons.* ACM Journal of Experimental Algorithmics, 7:5, 2002, 7: Article 5, 2002.

9 S. Edelkamp, A. Elmasry, and J. Katajainen. *Two constant-factor-optimal realizations of adaptive heapsort.* In Int. Workshop on Combinatorial Algorithms (IWOCA), volume 7056 of Lecture Notes in Computer Science, pages 195–208. Springer, 2011.

10 S. Edelkamp, A. Elmasry, and J. Katajainen. *A catalogue of algorithms for building weak heaps.* In Int. Workshop on Combinatorial Algorithms, pages 249–262, 2012.

11 S. Edelkamp, A. Elmasry, and J. Katajainen. *The weak-heap data structure: Variants and applications.* J. Discrete Algorithms, 16:187–205, 2012.

12 S. Edelkamp, A. Elmasry, and J. Katajainen. *The weak-heap family of priority queues in theory and praxis.* In Computing: the Australasian Theory Symposium (CATS), pages 103–112, 2012.

13 S. Edelkamp, A. Elmasry, and J. Katajainen. *Weak-heaps engineered.* J. Discrete Algorithms, 23:83–97, 2013.

14 S. Edelkamp and A.Weiss. *QuickXsort: Efficient sorting with $n \log n - 1.399n + o(n)$ comparisons on average.* CSR, 2014.

15 S. Edelkamp, A. Elmasry, and J. Katajainen. *An In-Place Priority Queue with $O(1)$ Time for Push and $\lg n + O(1)$ Comparisons for Pop.* CSR, Lake Balkal, 2015.

16 J. Katajainen. *The ultimate heapsort.* In Computing: CATS, pages 87–96, 1998.

17 D. E. Knuth. *Sorting and Searching,* volume 3 of The Art of Computer Programming. Addison Wesley Longman, 2nd edition, 1998.
18 C. J. H. McDiarmid and B. A. Reed. *Building heaps fast.* Journal of Algorithms, pages 352–365, 1989.
19 J. Vuillemin. *A data structure for manipulating priority queues.* Commun. ACM, 21(4):309–315, 1978.
20 I. Wegener. *The worst case complexity of McDiarmid and Reed's variant of bottom-up-heap sort is less than* $n \log n + 1.1n$. STACS, pages 137–147, 1991.
21 J. W. J. Williams. *Algorithm 232: Heapsort.* Commun. ACM, 7(6):347–348, 1964.

3.4 Specification guided testing and verification for Cyber-Physical Systems

Georgios Fainekos (Arizona State University – Tempe, US)

URL https://sites.google.com/a/asu.edu/s-taliro/

Every year the software which controls Cyber-Physical Systems (CPS) increases in complexity. Due to the increased complexity, numerous design and implementation errors are discovered while CPS are operational in the field. Such errors can have catastrophic effects to human life and to the economy. In this talk, we present how the error detection process in CPS can be accelerated using Model-Based Development (MBD) and formal specifications. Temporal logic is a formal specification language that can capture both state- space and real-time system requirements. For example, temporal logics can mathematically state requirements like "whenever the system switches to first gear, then it should not switch back to second gear within 2.5sec". Our approach in tackling this challenging problem is to convert the verification problem into an optimization problem through a notion of robustness for temporal logics. Through this transformation, any stochastic optimization algorithm can be used for automatic test case generation. In addition, we show how the basic underlying theory can be extended to provide a solution to the conformance problem and to the on-line monitoring problem. We have implemented our testing and verification framework into a Matlab (TM) toolbox called S-TaLiRo (System's TemporAl LogIc Robustness). Finally, in this talk, we demonstrate that S-TaLiRo can provide answers to challenge problems from the automotive and medical device industries.

3.5 VAL: The Plan Validator for PDDL+

Maria Fox (King's College London, GB)

Joint work of Fox, Maria; Long, Derek; Howey, Richard;
Main reference M. Fox. R. Howey, D. Long, "Validating Plans in the Context of Processes and Exogenous Events," in Proc. of the 20th National Conf. on Artificial Intelligence (AAAI'05), pp. 1151–1156, AAAI Press, 2005.
URL http://www.aaai.org/Library/AAAI/2005/aaai05-182.php

In recent years a significant part of the planning community has moved towards the application of planners to realistic problems. In the third International Planning Competition (held in 2002) planners addressed domains featuring temporal constraints and numeric resources. The modelling language PDDL2.1 was introduced for this competition, building on the previously

well established, but propositional, PDDL. A critical element in the use of PDDL2.1 has been the common understanding of the semantics of the language. The VAL system, described in this talk, was introduced as a reference semantics for PDDL2.1, enabling plan validity to be checked against domain and problem specifications. VAL has played this role since 2002, remaining a vital tool for many researchers. However, the competition did not explore the continuous modelling features of PDDL2.1, such as duration-dependent effects and continuously increasing and decreasing numeric values. Further modelling support for these features was provided in PDDL+, and VAL was extended to handle them but, with few exceptions, these features have not been much explored in planning since their introduction. The validation of plans using actions with continuous effects presents significant challenges. In this talk we review the need for continuous effects in planning, their semantics and the problems that arise in validation of plans that include them. We consider a motivating example and explain the exact methods exploited by VAL.

3.6 Heuristics for Classical Planning: We're More Rigorous Than You Thought!

Malte Helmert (Universität Basel, CH)

Joint work of Domshlak, Carmel; Helmert, Malte; Pommerening, Florian; Röger, Gabriele; Seipp, Jendrik

The talk discusses the problem of optimal classical planning and its solution by heuristic search. There are two main technical parts. The first part presents the four major concepts under which heuristic approaches until around 2009 have been subsumed: *delete relaxation*, *abstraction*, *critical paths* and *landmarks*. It then explores some formal connections between these approaches within the framework of polynomial heuristic compilation. The second part covers some more recent ideas in this area: the *state equation* heuristic, *negative cost partitioning* and *potential heuristics*. It turns out that these ideas are closely connected and help us better understand previous heuristics based on abstraction.

3.7 Scenario-based Online Planning under Uncertainty

David Hsu (National University of Singapore, SG)

Joint work of Ye, Nan; Somani, Adhiraj; Hsu, David; Lee, Wee Sun
Main reference A. Somani, N. Ye, D. Hsu, W. S. Lee, "DESPOT: Online POMDP planning with regularization," in Proc. of the 27th Annual Conf. on Neural Information Processing Systems (NIPS'13), pp. 1772–1780, 2013.
URL http://papers.nips.cc/paper/5189-despot-online-pomdp-planning-with-regularization
URL http://bigbird.comp.nus.edu.sg/pmwiki/farm/appl/

Partially observable Markov decision process (POMDPs) provide a principled framework for planning under uncertainty, but are computationally intractable, due to the "curse of dimensionality" and the "curse of history". To overcome these difficulties, our new online POMDP algorithm samples a small set of randomly sampled "scenarios". It captures the execution of all policies on these scenarios in a compact structure called a Determinized Sparse Partially Observable Tree (DESPOT) and focuses the search for an optimal policy

over the sampled scenarios. We show that while the DESPOT algorithm optimizes over the sampled scenarios only, it computes a near-optimal optimal policy, provided that an optimal policy admits a compact representation. It also compares favorably with the state-of-the-art online POMDP algorithm on a set of benchmark tests, but provides stronger theoretical guarantee. I will also present experimental results showing the DESPOT algorithm running online in real time on an autonomous robot vehicle navigating among many pedestrians. Source code is available for download at http://bigbird.comp.nus.edu.sg/pmwiki/farm/appl/.

3.8 SAT Modulo Monotonic Theories

Alan Hu (University of British Columbia – Vancouver, CA)

Joint work of Bayless, Sam; Bayless, Noah; Hoos, Holger; Hu, Alan;
Main reference S. Bayless, N. Bayless, H. H. Hoos, A. Hu, "SAT Modulo Monotonic Theories," AAAI 2015, to appear; pre-print available from author's webpage.
URL http://www.cs.ubc.ca/labs/isd/Projects/MonoSAT/

I will present the concept of a "monotonic theory" and show how to build efficient SMT (SAT Modulo Theory) solvers, including efficient theory propagation and clause learning, for such theories. Examples of monotonic theories include graph properties such as reachability, shortest path, connected components, minimum spanning tree, and max-flow/min-cut, as well as geometric properties like convex hulls, visibility, etc. We demonstrate our framework by building SMT solvers for each of these theories. We apply these solvers to procedural content generation problems, demonstrating major speed-ups over state-of-the-art approaches based on SAT or Answer Set Programming, and easily solving several instances that were previously impractical to solve. Because this method provides an easy way to extend SMT solving efficiently to "physical world" theories, I believe that this holds promise for planning problems as well (although we don't have any experimental results for planning yet).

3.9 Adapting a Policy On-line when the POMDP Model Changes

Hanna Kurniawati (The University of Queensland, AU)

Joint work of Kurniawati, Hanna; Klimenko, Dimitri; Song, Joshua; Yadav, Vinay
Main reference H. Kurniawati, V. Yadav, "An Online POMDP Solver for Uncertainty Planning in Dynamic Environment," in Proc. of the 16th Int'l Symp. on Robotics Research (ISRR'13), 2013; pre-print available from author's webpage.
URL http://robotics.itee.uq.edu.au/~hannakur/dokuwiki/papers/isrr13_abt.pdf

Over the past several years, POMDP-based planning has advanced tremendously. In 10 years, it has moved from planners that can only solve up to 12 states in days to planners that can solve problems with more than 5 dimensional continuous state space in seconds, making POMDP framework to start being practical for robotics problems. However, one issue remains: The POMDP model should not change. When a POMDP model may change during run-time, we either model all possible changes as part of the POMDP model, creating an unnecessarily large POMDP problem, or resort to replanning, which throws away all prior computation and plan from scratch. In this talk I will present my recent work that tries to mediate these two extreme approaches.

A software toolkit that implements the above algorithm can be downloaded from http://robotics.itee.uq.edu.au/~tapir. The algorithm and software can also be used as an on-line POMDP solver when there's no changes in the model.

3.10 PMT: Planning Modulo Theories

Derek Long (King's College London, GB)

Planning has traditionally focussed on states described by a finite set of propositional variables, or, equivalently, finite domain variables (SAS+). Over the past decade, there has also been interest in extending the more successful approaches to include treatment of states with number-valued state variables. In recent years, the SAT community has turned its attention to problems in which predicates, functions and constants can be "interpreted", modulo one or more theories (such as arithmetic, linear arithmetic constraints, arrays and so on) leading to SAT Modulo Theories (SMT). We explore the extension of Planning inspired by the ideas in SMT, to Planning Modulo Theories (PMT). PMT offers state variables with values drawn from new types, such as sets, arrays and so on. We show that the successful planning relaxations, based on reachability analysis and relaxed plan extraction, can be extended to PMT using abstract interpretations of the symbols in the corresponding theories. These abstractions must obey certain conditions that can be expressed in terms of semi-lattices. The talk illustrates these ideas and briefly outlines the behaviour of a PMT planner.

References

1 Peter Gregory, Derek Long, Maria Fox, J. Christopher Beck, *Planning Modulo Theories: Extending the Planning Paradigm.* In Proceedings of Twenty-Second International Conference on Automated Planning and Scheduling, ICAPS 2012.

3.11 Planning in Hybrid Domains through Model Checking

Daniele Magazzeni (King's College London, GB)

The talk presents two approaches for planning in hybrid domains, both based on model checking. The first approach is based on the *Discretise and Validate* method [1], where first the continuous model is discretised, then the discretised model is solved using explicit model checking, and finally the discretised solution is validated against the continuous model. If the solution is not valid, then the discretisation is refined and the process iterates. This approach has been implemented in the UPMurphi tool [3]. An application of the approach to the problem of efficient multiple battery load management is presented [2].

The second approach uses symbolic model checking, and is based on a translation from PDDL+ to the semantics of standard automata [4],[5]. This translation represents a first bridge between planning in hybrid domains and hybrid system model checking, as it allows

the use of existing tools in hybrid system verification for planning in hybrid domains. A case study based on SpaceEx is presented.

References

1 Giuseppe Della Penna, Daniele Magazzeni, Fabio Mercorio, *A universal planning system for hybrid domains.* In Applied Intelligence, 36(4), 932–959, 2012.

2 Maria Fox, Derek Long, Daniele Magazzeni, *Plan-based Policies for Efficient Multiple Battery Load Management.* In Journal of Artificial Intelligence Research (JAIR) 44:335–382. June 2012.

3 Giuseppe Della Penna, Benedetto Intrigila Daniele Magazzeni, Fabio Mercorio, *UPMurphi: a Tool for Universal Planning on PDDL+ Problems.* In Proceedings of the Nineteen International Conference on Automated Planning and Scheduling, ICAPS 2009.

4 Sergiy Bogomolov, Daniele Magazzeni, Andreas Podelski, Martin Wehrle, *Planning as Model Checking in Hybrid Domains.* In Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, 2014.

5 Sergiy Bogomolov, Daniele Magazzeni, Stefano Minopoli, Martin Wehrle, *Foundations of Must Semantics for PDDL+.* In Proceedings of the Twenty-Fifth International Conference on Automated Planning and Scheduling, ICAPS 2015.

3.12 LTL, Regular Expressions (Golog) & Automata: 101 things you can do with non-classical planning

Sheila McIlraith (University of Toronto, CA)

The field of Artificial Intelligence Automated Planning has seen significant advances in the last 15 years, largely as a result of innovations in planning-specific heuristic search and SAT techniques. Much of the focus has been on so-called classical planning systems which assume complete information about the initial state of a system at the outset of planning, deterministic actions, and a final state goal. Unfortunately, many interesting real-world problems violate classical planning assumptions. In this talk, I will discuss planning with temporally extended goals, constraints, and preferences, specified in Linear Temporal Logic (LTL) and via programs specified in an Algol-like language called Golog, which together capture the expressivity of finite state automata. A main focus of this talk will be on how to leverage these expressive languages to help guide heuristic search. We have explored these techniques in a diversity of problem settings from web service composition to verification and concurrent test generation. This talk should be of some interest to anyone who is concerned with reachability in dynamical systems. This is joint work with Jorge Baier, Christian Fritz, Meghyn Bienvenu, and Shirin Sohrabi.

3.13 Falsification of Safety Properties in Hybrid Systems through Motion Planning

Erion Plaku (CUA – Washington, US)

Hybrid systems often arise in embedded controllers used in the automotive industry, manufacturing, robotics, environmental- and health-monitoring devices whenever physical aspects of the system are modelled by combining discrete logic with continuous dynamics. This talk summarizes our research efforts in developing motion-planning approaches that can be used for the falsification of safety properties in hybrid systems. When the system is unsafe, these motion-planning approaches construct witness trajectories by expanding a search tree to explore the state space of the hybrid system. To improve the computational efficiency, these motion-planning approaches use discrete abstractions and discrete search to guide the expansion of the motion tree.

References

1 Erion Plaku and Lydia Kavraki and Moshe Vardi. *Falsification of LTL Safety Properties in Hybrid Systems.* In International Journal on Software Tools and Technology Transfer, 15(4):305–320, 2013.
2 James McMahon and Erion Plaku. *Sampling-based tree search with discrete abstractions for motion planning with dynamics and temporal logic.* In Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2014.
3 Erion Plaku. *Robot Motion Planning with Dynamics as Hybrid Search.* In Proceedings of AAAI Conference on Artificial Intelligence (AAAI), 2013.

3.14 Planning for Energy Systems

Sylvie Thiébaux (Australian National University, AU)

Public awareness of climate change, the rising costs of maintaining an ageing infrastructure, energy market volatility, and countries dependence on energy imports have motivated the development of new renewable and distributed technologies for generating, storing and managing energy. In turn, using these technologies to their full potential requires a fundamental paradigm shift in the way power systems are planned and operated. The talk describes a range of interesting planning problems that arise in the energy systems space, along with the challenges they raise for AI planners. Whilst these problems are currently solved using optimisation technologies, integrating ideas from AI planning has the potential to lead to improvement in scalability and solution quality. The talk illustrates in detail some of these challenges and potential benefits by comparing two approaches to the problem of power supply restoration in power distribution systems: one based on optimisation [15] and one based on an extension of AI planning [9].

References

1 Keith Bell, Amanda Jane Coles, Andrew Coles, Maria Fox, and Derek Long. The role of AI planning as a decision support tool in power substation management. *AI Commun.*, 22(1):37–57, 2009.

2 Joshua Campion, Chris Dent, Maria Fox, Derek Long, and Daniele Magazzeni. Challenge: Modelling unit commitment as a planning problem. In *Proc. 23rd International Conference on Automated Planning and Scheduling (ICAPS'13)*, June 2013.

3 Carleton Coffrin and Pascal Van Hentenryck. A linear-programming approximation of AC power flows. *INFORMS Journal on Computing*, 26(4):718–734, 2014.

4 Carleton Coffrin, Pascal Van Hentenryck, and Russell Bent. Last-mile restoration for multiple interdependent infrastructures. In *Proc. 26th AAAI Conference on Artificial Intelligence*, July 2012.

5 Giuseppe Della Penna, Daniele Magazzeni, Fabio Mercorio, and Benedetto Intrigila. Upmurphi: A tool for universal planning on PDDL+ problems. In *Proc. 19th International Conference on Automated Planning and Scheduling (ICAPS'09)*, September 2009.

6 Peter Gregory, Derek Long, Maria Fox, and J. Christopher Beck. Planning modulo theories: Extending the planning paradigm. In *Proc. 22nd International Conference on Automated Planning and Scheduling (ICAPS'12)*, June 2012.

7 C. Hijazi, C. Coffrin, and P. Van Hentenryck. Convex quadratic relaxations of nonlinear programs in power systems. *Optimization On-Line*, aug. 2013.

8 Hassan Hijazi and Sylvie Thiébaux. Optimal ac distribution systems reconfiguration. In *18th Power Systems Computation Conference (PSCC'14)*, Wroclaw, Poland, August 2014. IEEE.

9 Franc Ivankovic, Patrik Haslum, Sylvie Thiébaux, Vikas Shivashankar, and Dana Nau. Optimal planning with global numerical state constraints. In *24th International Conference on Automated Planning and Scheduling (ICAPS'12)*, pages 167–175, Portsmouth, USA, June 2014.

10 Boon Ping Lim, Menkes van den Briel, Sylvie Thiébaux, Scott Backhaus, and Russell Bent. Hvac-aware occupancy scheduling. In *Proc. 29th AAAI Conference on Artificial Intelligence (AAAI'15)*, January 2015.

11 Terrence Mak, Hassan Hijazi, and Pascal Van Hentenryck. Power system restoration with transient stability. In *Proc. 29th AAAI Conference on Artificial Intelligence (AAAI-15)*, January 2015.

12 Masahiro Ono, Wesley Graybill, and Brian C. Williams. Risk-sensitive plan execution for connected sustainable home. In *Proc. 4th ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings (BuildSys'12)*, pages 45–52, November 2012.

13 Chiara Piacentini, Varvara Alimisis, Maria Fox, and Derek Long. Combining a temporal planner with an external solver for the power balancing problem in an electricity network. In *Proc. 23rd International Conference on Automated Planning and Scheduling (ICAPS-13)*, June 2013.

14 Paul Scott, Sylvie Thiébaux, Menkes van den Briel, and Pascal Van Hentenryck. Residential demand response under uncertainty. In *International Conference on Principles and Practice of Constraint Programming (CP'13)*, September 2013.

15 Sylvie Thiébaux, Carleton Coffrin, Hassan Hijazi, and John Slaney. Planning with mip for supply restoration in power distribution systems. In *International Joint Conference on Artificial Intelligence (IJCAI'13)*, pages 2900–2907, August 2013.

16 Eric Timmons and Brian C. Williams. Enumerating preferred solutions to conditional simple temporal networks quickly using bounding conflicts. In *AAAI-15 Workshop on Planning, Search and Optimisation (PlanSOpt'15)*, January 2015.

3.15 Directed model checking for timed systems

Martin Wehrle (Universität Basel, CH)

Joint work of Wehrle, Martin; Kupferschmid, Sebastian

This talk provides an introduction to directed model checking for timed systems. Directed model checking is an established approach for detecting error states in concurrent systems. A popular variant to find shortest error traces is to apply the A* search algorithm with distance heuristics that never overestimate the real error distance. An important class of such distance heuristics is the class of pattern database heuristics, which are based on abstractions of the system under consideration. In this context, we propose downward pattern refinement, a systematic approach for the automatic construction of pattern database heuristics for timed automata. Our experiments show the practical potential of the resulting pattern database heuristic.

References

1 Sebastian Kupferschmid and Martin Wehrle. *Abstractions and Pattern Databases: The Quest for Succinctness and Accuracy.* In Proceedings of the 17th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2011), 2011.
2 Martin Wehrle and Sebastian Kupferschmid. *Downward Pattern Refinement for Timed Automata.* In International Journal on Software Tools for Technology Transfer (STTT), DOI: 10.1007/s10009-014-0346-x.

3.16 GPUexplore: Many-core On-the-fly State Space Exploration Using GPUs

Anton Wijs (RWTH Aachen, DE)

Joint work of Wijs, Anton; Bosnacki, Dragan
Main reference A. Wijs, D. Bošnački, "GPUexplore: Many-Core On-The-Fly State Space Exploration Using GPUs," in Proc. of the 20th Int'l Conf. on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'14), LNCS, Vol. 8413, pp. 233–247, Springer, 2014.
URL http://dx.doi.org/10.1007/978-3-642-54862-8_16

In recent years, General Purpose Graphics Processors (GPUs) have been successfully applied in multiple application domains to drastically speed up computations. Model checking is an automatic method to formally verify the correctness of a system specification. Such specifications can be viewed as implicit descriptions of a large directed graph or state space, and for most model checking operations, this graph must be analysed. Constructing it, or on-the-fly exploring it, however, is computationally intensive, so it makes sense to try to implement this for GPUs. In this talk, I explain the limitations involved, and how to overcome these. I discuss the possible approaches involving related work, and propose an alternative, using a new hash table approach for GPUs. Experimental results with our prototype implementations show significant speed-ups compared to the established sequential counterparts.

3.17 Optimal Temporal Plan Execution with Bounded Risk

Brian C. Williams (MIT – Cambridge, US)

Automated task planning and execution is key to creating robotic systems that perform tasks robustly based on intuitive goal descriptions. Automated activity planning and plan execution are both key elements. Plan executives must often map simple discrete activities specified in the plan to continuous control trajectories or motions. This involves automated planning, but within a continuous domain.

In addition, planners must ensure correctness despite uncertainty in the environment, for example, due to temporal delays, actuator disturbances and sensor noise. To adapt, plan executives should dynamically adjust the timing of activities and control trajectories that implement these activities, and activity plan descriptions should offer flexibility in temporal and state constraints, needed to perform these adaptations.

Finally, when uncertainty is unbounded, successful plan execution cannot be guaranteed, there is always some risk of failure. In this event, activity plans should include specifications of what level of risk of failure is acceptable, and plan executives should ensure that they operate within this risk bound.

The first half of this tutorial presents algorithms for dynamically executing temporal plans by scheduling activities and by mapping these activities to continuous control trajectories. The second half of this tutorial introduces temporal plan descriptions that specify risk bounds in the form of chance constraints, and then presents stochastic methods for scheduling and generating control trajectories that are guaranteed to operate within these risk bounds.

3.18 Statistical Model Checking for Markov Decision Processes

Paolo Zuliani (Newcastle University, GB)

Joint work of Henriques, David; Martins, Joao; Zuliani, Paolo; Platzer, Andre; Clarke, Edmund;
Main reference D. Henriques, J. Martins, P. Zuliani, A. Platzer, E. M. Clarke, "Statistical model checking for Markov decision processes," in Proc. of the 9th Int'l Conf. on Quantitative Evaluation of Systems (QEST'12), pp. 84–93, IEEE, 2012.
URL http://dx.doi.org/10.1109/QEST.2012.19

We present a statistical model checking approach for verifying temporal logic properties of Markov Decision Processes.

Statistical Model Checking (SMC) is a simulation-based verification technique for stochastic systems. It combines randomised sampling with statistical analysis. Basically, execution traces of the underlying (stochastic) model are first sampled and then checked against a linear-time bounded temporal property. The results of these checks are used to compute appropriate *statistical* guarantees on whether the given property is true or not. A disadvantage of SMC is that it might return wrong answers, albeit the probability of such occurences can be made arbitrarily low. An advantage of SMC is that it usually scales much better than traditional model checking techniques, since it does not perform an exhaustive state space search. As such, SMC can be useful for verifying extremely large systems.

Systems that combine probabilism and nondeterminism, *e. g.*, Markov Decision Processes (MDP), represent a hurdle for SMC. In particular, for MDPs it is not clear how the simulator

should resolve nondeterministic choices in order to generate sample executions of the model. Different resolutions of nondeterminism (*i. e.*, schedulers) may result in widely different system behaviours. In general, a finite-state MDP satisfies a given temporal property with a maximun and a minimun probability.

In this talk we give an overview of our recently proposed SMC-based approach for MDPs. In particular, our approach aims at finding a scheduler that maximises the probability that a given property is true. It does so by iteratively sampling over the model space and by building probabilistic candidate schedulers which cannot decrease the probability that the property is true. In particular, we use reinforcement learning to build schedulers which give more probability to those actions likely to lead to property satisfaction. We apply our approach to a number of case studies and compare the results with the leading probabilistic model checker, PRISM. The results show that our approach can scale to MDPs that cannot be handled with traditional techniques. We conclude by discussing possible extensions of our technique for schedulers with memory and for properties with unbounded temporal operators.

References

1 D. Henriques, J. Martins, P. Zuliani, A. Platzer, E. M. Clarke. *Statistical Model Checking for Markov Decision Processes.* In QEST 2012: IEEE, pages 84–93.

2 H. L. S. Younes and Reid G. Simmons. *Probabilistic Verification of Discrete Event Systems Using Acceptance Sampling.* In CAV 2002: LNCS vol. 2404, pages 223–235.

3 M. Kwiatkowska, G. Norman, D. Parker. *PRISM 4.0: Verification of Probabilistic Real-time Systems.* In CAV 2011: LNCS vol. 6806, pages 585–591.

4 H. L. S. Younes, E. M. Clarke, P. Zuliani. *Statistical Verification of Probabilistic Properties with Unbounded Until.* In SBMF 2010: LNCS vol. 6527, pages 144–160.

Participants

- Steven Andrews
Frederic Hutchinson Cancer Center – Seattle, US
- Sergiy Bogomolov
Universität Freiburg, DE
- Dragan Bosnacki
TU Eindhoven, NL
- Ronen I. Brafman
Ben Gurion University – Beer Sheva, IL
- Alessandro Cimatti
Bruno Kessler Foundation – Trento, IT
- Stefan Edelkamp
Universität Bremen, DE
- Georgios Fainekos
Arizona State University – Tempe, US
- Maria Fox
King's College London, GB
- Malte Helmert
Universität Basel, CH
- David Hsu
National Univ. of Singapore, SG
- Alan Hu
University of British Columbia – Vancouver, CA
- Hanna Kurniawati
The Univ. of Queensland, AU
- Derek Long
King's College London, GB
- Daniele Magazzeni
King's College London, GB
- Sheila McIlraith
University of Toronto, CA
- Bernhard Nebel
Universität Freiburg, DE
- Doron A. Peled
Bar-Ilan University – Ramat Gan, IL
- Erion Plaku
CUA – Washington, US
- Andreas Podelski
Universität Freiburg, DE
- Franco Raimondi
Middlesex University, GB
- Sylvie Thiébaux
Australian National Univ., AU
- Martin Wehrle
Universität Basel, CH
- Anton Wijs
RWTH Aachen, DE
- Brian C. Williams
MIT – Cambridge, US
- Paolo Zuliani
Newcastle University, GB