Synthesis and Planning

Dagstuhl Seminar 05241, June 12 – 17, 2005
Executive Summary

H. Kautz[1], W. Thomas[2] and M. Y. Vardi[3]

[1] Univ. of Washington, US
kautz@cs.washington.edu
[2] RWTH Aachen, DE
thomas@informatik.rwth-aachen.de
[3] Rice University, US
vardi@cs.rice.edu

This meeting has brought together researchers working in two complementary fields: automatic synthesis of (control) programs, and methods for devising planning algorithms in artifical intelligence (AI). Thus, the seminar combines a strong thread of current research in automata theory with an area of possible but so far unexplored applications.

The idea of organizing such a seminar arose during IJCAI 2003, where Vardi gave an invited talk on the automata-theoretic approach to design verification. In discussions between Kautz and Vardi after the talk it became clear that methods of synthesizing strategies for reactive systems is an issue of common interest to automata theory and artificial intelligence.

Automatic Synthesis

The first results on automatic synthesis of control programs go back to the 1960's when Büchi, McNaughton, Rabin, and others showed how to realize specifications for non-terminating reactive computations by finite automata. These results extend the standard equivalence results connecting automata with logic; they are concerned with a specification of an open system (reacting to moves of its environment) and realizations by automata with output, providing the moves of the program component of a system.

Today these results have been recast in the terminology of infinite two-player games. Such a game is played on a directed graph which is the "arena" of the game. Each vertex is associated to one of the two players. A play starts in a given vertex and proceeds along the graph edges; in each step that player to whom the current vertex belongs moves via an edge to a new vertex. The winner is determined by a "winning condition" on the resulting finite or infinite path. Two fundamental algorithmic problems arise in this context: Given a graph and a winning condition, from which start vertices does the first player have a winning strategy, and - if yes - how can one construct a program which realizes auch a strategy?

Building on the classical work, much progress was achieved during the last decade. While in the early papers it was shown that in principle the automatic

Dagstuhl Seminar Proceedings 05241
Synthesis and Planning
http://drops.dagstuhl.de/opus/volltexte/2006/452

synthesis of winning strategies is possible (which opens a perspective for automatic controller synthesis), the focus has now shifted to refined and extended questions:

- problems of complexity and efficiency in the construction of strategies (usually in the form of finite automata)
- applications in model-checking (where the games are used in their connection with logic, reflecting the duality of existential and universal logical connectives)
- applications to the synthesis of reactive programs
- expansion of the techniques to further types of games, e.g. with infinite state spaces, and
- games involving continuous parameters, for example stochastic and timed games

A recent GI-Dagstuhl seminar volume (LNCS 2500, edited by Erich Grädel, Wolfgang Thomas, and Thomas Wilke) with survey contributions by young researchers gives an overview of the state of the art.

In this situation, where a solid body of constructions and nontrivial results is available and the further development is somewhat open, it is essential to expose, try, and adjust the methods in application areas. Planning in AI is one of them, another (not excluded for the seminar) is the connection with researchers on discrete event systems.

AI Planning

Planning is a sub-field of artificial intelligence which is concerned with the generation of a rational course of action given a declarative specification of the environment, the goals, and the possible actions. The field can be further subdivided by the kinds of problems considered:

Classical planning considers single-agent deterministic domains where the initial and goal states are specified by sets of logical formulas. Classical planning corresponds to reachability analysis in large state spaces. Research thus focus on algorithms that can perform such an analysis without actually enumerating the state space. Classical planning is closely connected to the area of verification called model checking, and in recent years there have been fruitful exchanges of techniques and algorithms between the two fields.

Universal planning involves synthesizing a reactive control program that can direct a agent toward a goal state from any possible situation. It is thus universal in that no fixed initial state is assumed. Furthermore, universal planning problems often include non-deterministic actions, which can be used to model action failure and/or changes in the world induced by nature. Universal planning can be viewed as control program synthesis where every computation terminates in a goal (or failure) state.

Decision-theoretic planning adds two features to universal planning: first, non-determinism (or nature) is modeled by a probability distribution over the result of each action; and second, a positive or negative reward is associated with each state. The goal of the agent is to maximize the sum of rewards that the agent receives over its lifetime (or in the case of an infinite lifetime the discounted or average reward) rather than to reach a particular goal state. If the agent is able to observe all variables in the domain the problem becomes that of solving a Markov Decision Process (MDP); if part of the state is hidden from the agent, the problem is that of solving a partially-observed MDP (POMDP). As with classical planning much research on decision theoretic planning focuses on techniques for handling large state spaces in a factored form, thus avoiding enumeration of all states.

Game-theoretic planning is a recent and fertile area of activity in AI research. While the previous approaches model the actions of other agents or other natural events simply as sources of uncertainty, a game-theoretic planner explicitly reasons about the choices other agents make in order to maximize their own utility. In terms of the synthesis of control programs as games against nature, this line of work allows us to consider cases where nature is actively hostile or actively helpful to purposes of the system.

Finally, any of the preceeding areas can be generalized to consider the case where the full specification of the problem in terms of a world model, an action model, and a reward function (or goal specification) is not known to the system in advance. The planner must act while learning about the environment on the basis of the feedback it receives from (possibly infrequent) rewards. This research, called reinforcement learning, has deep roots in both control theory and models of animal behavior.

Links

This description of the field of AI planning should make clear that it is closely linked to the problem of synthesizing reactive control programs: in fact, one can argue that the two fields have the same subject matter, and are distinct only because historic conditions. As we have noted, the strongest connection in terms of scientific dialog between different communities has occurred between classical planning in AI and model checking in formal methods. For example, researchers in AI have found uses for BDD (Boolean decision diagram) algorithms from model checking, and techniques for reducing planning to satisfiability testing that were originally developed in the AI planning world are now used for hardware verification.

The workshop helped to increase the awareness of the researchers working in one field of the problems and methods in the other one, and thus to increase the interaction and collaboration of the two research communities, and the transfer of methodologies from one field to another.

Participants

Piergiorgio Bertoli (IRST Trento, Italy)
Dietmar Berwanger (LaBRI Bordeaux, France)
Alexis Bouquet (LaBRI Bordeaux, France)
Xavier Briand (LaBRI Bordeaux, France)
Diego Calvanese (Free University of Bolzano, Italy)
Giuseppe De Giacomo (Università di Roma "La Sapienza", Italy)
Jacques Duparc (Université de Lausanne, Switzerland)
Stefan Edelkamp (Universität Dortmund, Germany)
Kousha Etessami (University of Edinburgh, GB)
Alfonso Gerevini (University of Brescia, Italy)
Enrico Giunchiglia (Università di Genova, Italy)
Robert Givan (Purdue University, USA)
Robert P. Goldman (Smart Information Flow Technologies, Minneapolis, USA)
Erich Grädel (RWTH Aachen, Germany)
David Janin (LaBRI Bordeaux, France)
Marcin Jurdzinski (University of Warwick, GB)
Henry Kautz (University of Washington, USA)
Vahur Kotkas (Tallinn University of Technology, Estonia)
Stephan Kreutzer (Humboldt Universität, Germany)
Stéphane Lafortune (University of Michigan, USA)
Martin Lange (Ludwig-Maximilians-Universität München, Germany)
Martin Leucker (TU München, Germany)
Derek Long (The University of Strathclyde, Glasgow, GB)
Christof Löding (RWTH Aachen, Germany)
Rupak Majumdar (Univ. of California at Los Angeles, USA)
Oded Maler (VERIMAG, France)
Anca Muscholl (LIAFA Paris, France)
Bernhard Nebel (Universität Freiburg, Germany)
Damian Niwinski (University of Warsaw, Poland)
Jan Obdrzalek (University of Edinburgh, GB)
Madhusudan Parthasarathy (University of Illinois at Urbana-Champaign, USA)
Sophie Pinchinat (Irisa Rennes, France)
Marco Pistore (Università degli Studi di Trento, Italy)
Nir Piterman (EPFL, Lausanne, Switzerland)
Amir Pnueli (New York University, USA)
William Rounds (University of Michigan, USA)
Pierre-Yves Schobbens (Université Notre-Dame de la Paix, Namur, Belgium)
Bart Selman (Cornell University, USA)
John G. Thistle (University of Waterloo, Canada)
Wolfgang Thomas (RWHT Aachen, Germany)
Moshe Y. Vardi (Rice University, Houston, USA)
Sergei Vorobyov (Uppsala University, Sweden)
Igor Walukiewicz (LaBRI Bordeaux, France)
Thomas Wilke (Universität Kiel)