Towards Combination of Logic and Calculus for Near-Optimal Planning in Relational Hybrid Systems

Mikhail Soutchanski

Vardi Fest Workshop @FLOC-22: Sunday, July 31st, 2022

Towards Logic of Hybrid Systems

July 12 and 13, 2011: Moshe Vardi presented 2 lectures at the Fields Inst. for Research in Mathematical Sciences in Toronto: “From Philosophical to Industrial Logic", and “Logic, Automata, Games, and Algorithms".

Towards Logic of Hybrid Systems

July 12 and 13, 2011: Moshe Vardi presented 2 lectures at the Fields Inst. for Research in Mathematical Sciences in Toronto: “From Philosophical to Industrial Logic", and “Logic, Automata, Games, and Algorithms".

But he also did research on integration of Task and Motion Planning (TAMP) in robotics together with his colleagues from $\sim$2007.

Towards Logic of Hybrid Systems

July 12 and 13, 2011: Moshe Vardi presented 2 lectures at the Fields Inst. for Research in Mathematical Sciences in Toronto: “From Philosophical to Industrial Logic", and “Logic, Automata, Games, and Algorithms".

But he also did research on integration of Task and Motion Planning (TAMP) in robotics together with his colleagues from $\sim$2007.

Challenge in TAMP: planning and control of hybrid systems with discrete actions/events that can initiate or terminate continuous processes.

Towards Logic of Hybrid Systems

July 12 and 13, 2011: Moshe Vardi presented 2 lectures at the Fields Inst. for Research in Mathematical Sciences in Toronto: “From Philosophical to Industrial Logic", and “Logic, Automata, Games, and Algorithms".

But he also did research on integration of Task and Motion Planning (TAMP) in robotics together with his colleagues from $\sim$2007.

Challenge in TAMP: planning and control of hybrid systems with discrete actions/events that can initiate or terminate continuous processes.

Lesson to learn: if discreet symbolic reasoning is the driver for the system then it achieves its goals more efficiently and reliably.

Towards Logic of Hybrid Systems

July 12 and 13, 2011: Moshe Vardi presented 2 lectures at the Fields Inst. for Research in Mathematical Sciences in Toronto: “From Philosophical to Industrial Logic", and “Logic, Automata, Games, and Algorithms".

But he also did research on integration of Task and Motion Planning (TAMP) in robotics together with his colleagues from $\sim$2007.

Challenge in TAMP: planning and control of hybrid systems with discrete actions/events that can initiate or terminate continuous processes.

Lesson to learn: if discreet symbolic reasoning is the driver for the system then it achieves its goals more efficiently and reliably.

In general, the reachability problem for hybrid systems is undecidable (mid 1990s). However, the hybrid systems community develops different (incomplete but practically useful) algorithms and tools for reachability analysis.

Control of Hybrid Systems: 2 Camps

There are two main different research communities. First is inspired by logic: theorem proving and bounded model checking.

Control of Hybrid Systems: 2 Camps

There are two main different research communities. First is inspired by logic: theorem proving and bounded model checking.

From SAT solvers to Satisfiability Modulo Theories (SMT) tools.
From SMT to Optimization Modulo Theories (OMT) or Satisfiability Modulo Convex Optimization (SMC).

Control of Hybrid Systems: 2 Camps

There are two main different research communities. First is inspired by logic: theorem proving and bounded model checking.

From SAT solvers to Satisfiability Modulo Theories (SMT) tools.
From SMT to Optimization Modulo Theories (OMT) or Satisfiability Modulo Convex Optimization (SMC).

But SAT encoding can blow up to unmanageable size when grounding action schemas in planning over large domains. Counter-example: planning for a sequence of organic chemistry reactions to synthesize the target molecule.

Control of Hybrid Systems: 2 Camps

There are two main different research communities. First is inspired by logic: theorem proving and bounded model checking.

From SAT solvers to Satisfiability Modulo Theories (SMT) tools.
From SMT to Optimization Modulo Theories (OMT) or Satisfiability Modulo Convex Optimization (SMC).

But SAT encoding can blow up to unmanageable size when grounding action schemas in planning over large domains. Counter-example: planning for a sequence of organic chemistry reactions to synthesize the target molecule.

2nd: large OR community, Mixed Integer Non-Linear Programming (MI-NLP). But practical problems can be combinatorial, non-linear, and non-convex.

Control of Hybrid Systems: 2 Camps

There are two main different research communities. First is inspired by logic: theorem proving and bounded model checking.

From SAT solvers to Satisfiability Modulo Theories (SMT) tools.
From SMT to Optimization Modulo Theories (OMT) or Satisfiability Modulo Convex Optimization (SMC).

But SAT encoding can blow up to unmanageable size when grounding action schemas in planning over large domains. Counter-example: planning for a sequence of organic chemistry reactions to synthesize the target molecule.

2nd: large OR community, Mixed Integer Non-Linear Programming (MI-NLP). But practical problems can be combinatorial, non-linear, and non-convex.

Controls are represented as integer variables, but then too many are needed.

Discretisation of time, but what if an optimal action must be taken in between?

Linear (piece-wise) approximations: reduce MINLP to MILP to use efficient solvers, but can miss an optimal solution.

Direct experimental comparison:

SMC is better than MILP. But more studies are needed.

Use Constraint Logic Programming for Relational Hybrid Systems

From hybrid automata to modeling of *relational hybrid systems*. Lesson from data bases: relational structure is important. But the controller takes actions and the system evolves over time independently.

Use Constraint Logic Programming for Relational Hybrid Systems

From hybrid automata to modeling of *relational hybrid systems*. Lesson from data bases: relational structure is important. But the controller takes actions and the system evolves over time independently.

Hybrid Temporal Situation Calculus is a solution: joint work with Giuseppe De Giacomo, Vitaliy Batusov (PhD student), and Shaun Mathew (PhD student).

Use Constraint Logic Programming for Relational Hybrid Systems

From hybrid automata to modeling of *relational hybrid systems*. Lesson from data bases: relational structure is important. But the controller takes actions and the system evolves over time independently.

Hybrid Temporal Situation Calculus is a solution: joint work with Giuseppe De Giacomo, Vitaliy Batusov (PhD student), and Shaun Mathew (PhD student).

Develops situation calculus of J.McCarthy and sequential temporal situation calculus of R.Reiter. Novelty: temporal fluents vary over time according to the ODEs inside situation. Actions switch between different continuous evolutions

Use Constraint Logic Programming for Relational Hybrid Systems

From hybrid automata to modeling of *relational hybrid systems*. Lesson from data bases: relational structure is important. But the controller takes actions and the system evolves over time independently.

Hybrid Temporal Situation Calculus is a solution: joint work with Giuseppe De Giacomo, Vitaliy Batusov (PhD student), and Shaun Mathew (PhD student).

Develops situation calculus of J.McCarthy and sequential temporal situation calculus of R.Reiter. Novelty: temporal fluents vary over time according to the ODEs inside situation. Actions switch between different continuous evolutions

Lifted heuristic search planning: work with action schemas (no grounding up-front), delegate numerical constraints to an external NLP solver.

Use Constraint Logic Programming for Relational Hybrid Systems

From hybrid automata to modeling of *relational hybrid systems*. Lesson from data bases: relational structure is important. But the controller takes actions and the system evolves over time independently.

Hybrid Temporal Situation Calculus is a solution: joint work with Giuseppe De Giacomo, Vitaliy Batusov (PhD student), and Shaun Mathew (PhD student).

Develops situation calculus of J.McCarthy and sequential temporal situation calculus of R.Reiter. Novelty: temporal fluents vary over time according to the ODEs inside situation. Actions switch between different continuous evolutions

Lifted heuristic search planning: work with action schemas (no grounding up-front), delegate numerical constraints to an external NLP solver.

Build on the techniques from Constraint Logic Programming (CLP) proposed by Alain Colmerauer (PROLOG) and developed by a group of researchers.

Use Constraint Logic Programming for Relational Hybrid Systems

From hybrid automata to modeling of *relational hybrid systems*. Lesson from data bases: relational structure is important. But the controller takes actions and the system evolves over time independently.

Hybrid Temporal Situation Calculus is a solution: joint work with Giuseppe De Giacomo, Vitaliy Batusov (PhD student), and Shaun Mathew (PhD student).

Develops situation calculus of J.McCarthy and sequential temporal situation calculus of R.Reiter. Novelty: temporal fluents vary over time according to the ODEs inside situation. Actions switch between different continuous evolutions

Lifted heuristic search planning: work with action schemas (no grounding up-front), delegate numerical constraints to an external NLP solver.

Build on the techniques from Constraint Logic Programming (CLP) proposed by Alain Colmerauer (PROLOG) and developed by a group of researchers.

Hypothesis: reasoning in CLP (assisted by an external NLP solver) can compute near-optimal plans more efficiently than MINLP and for larger scale systems than SMT/OMT/SMC.

Use Constraint Logic Programming for Relational Hybrid Systems

From hybrid automata to modeling of *relational hybrid systems*. Lesson from data bases: relational structure is important. But the controller takes actions and the system evolves over time independently.

Hybrid Temporal Situation Calculus is a solution: joint work with Giuseppe De Giacomo, Vitaliy Batusov (PhD student), and Shaun Mathew (PhD student).

Develops situation calculus of J.McCarthy and sequential temporal situation calculus of R.Reiter. Novelty: temporal fluents vary over time according to the ODEs inside situation. Actions switch between different continuous evolutions

Lifted heuristic search planning: work with action schemas (no grounding up-front), delegate numerical constraints to an external NLP solver.

Build on the techniques from Constraint Logic Programming (CLP) proposed by Alain Colmerauer (PROLOG) and developed by a group of researchers.

Hypothesis: reasoning in CLP (assisted by an external NLP solver) can compute near-optimal plans more efficiently than MINLP and for larger scale systems than SMT/OMT/SMC.

Objective: contribute to a 2025 talk on the Unusual Effectiveness of Logic.