Planning for Temporally Extended Goals as Propositional Satisfiability

Robert Mattmüller (Univ. Freiburg), Jussi Rintanen (NICTA/ANU, Canberra)

Abstract

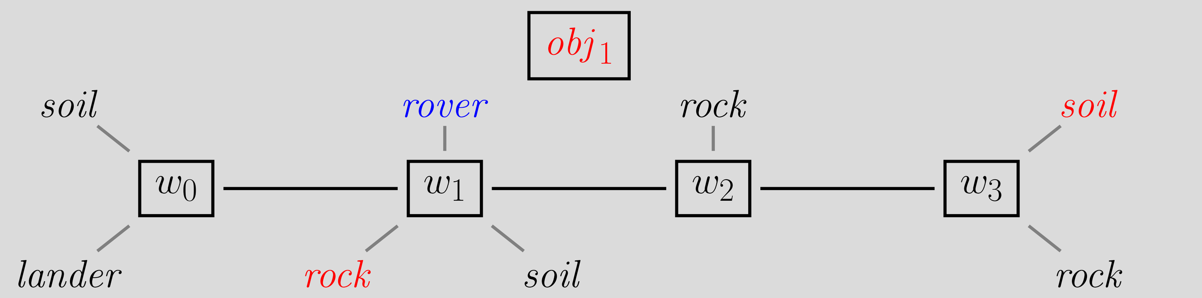
Temporally extended goals (TEGs) expressed as formulae of Linear-time Temporal Logic (LTL) can be used to express trajectory constraints. We present a satisfiability based encoding of planning for TEGs which allows for parallel plans, thus significantly increasing planning efficiency compared to purely sequential SAT planning. The results extend the practical applicability of satisfiability based planning to a wider class of planning problems.

1 Introduction

Motivation

In Classical Planning: reachability goals.
Higher Expressiveness: temporally extended goals
(maintenance goals, successive subgoals, safety goals, ...)
Representation: Linear-time Temporal Logic (LTL).

Example: Rovers Problem



The rover is equipped for soil and rock analysis and can take images in any mode.
All waypoints and objectives are mutually visible.

Reachability Goal:

```
(:goal (and (communicated_rock_data w1)
            (communicated_soil_data w3)
            (communicated_image_data obj1 high_res))
```

Additional Trajectory Constraints:

```
(:constraints (and (at-most-once (at rover w1))
                   (sometime-before (have_image rover obj1 high_res)
                                    (full store))
                   (at-most-once (empty store))))
```

Translation of Constraints to LTL$_{-X}$:
$\mathbf{G}(a \rightarrow (a\mathbf{U}\mathbf{G}\neg a)) \;\wedge\; (((\neg h \wedge \neg f)\mathbf{U}(\neg h \wedge f)) \vee \mathbf{G}(\neg h \wedge \neg f)) \;\wedge\; \mathbf{G}(e \rightarrow (e\mathbf{U}\mathbf{G}\neg e))$

Solving the Problem

Basic Idea: Use bounded LTL model checking for trajectory constraints.
Technique: Planning and LTL model checking as satisfiablity testing.
Contribution: Efficient parallel encoding.

2 Reduction to Satisfiability

Base Encoding

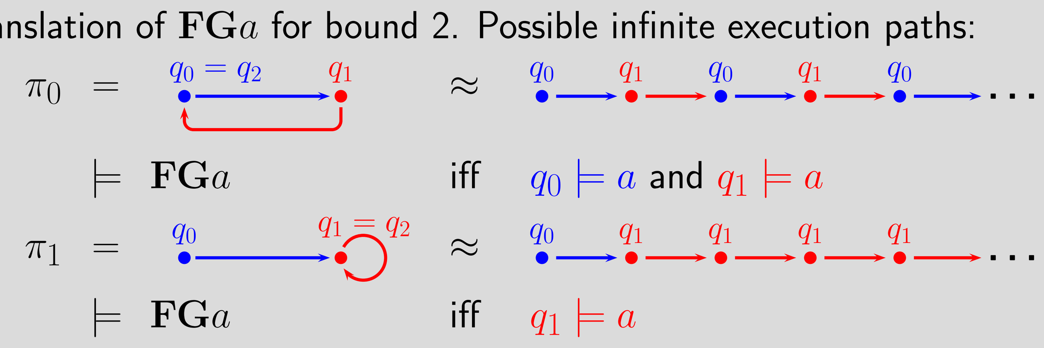
For all operators o with precondition p and effect e, state variables a and time points t:

Precondition axioms: $o_t \rightarrow p_t$
Effect axioms: $o_t \rightarrow \bigwedge e_{t+1}$
Frame axioms: $(a_t \wedge \neg a_{t+1}) \rightarrow \bigvee\{o_t \mid \neg a \in e\}$ and
$(\neg a_t \wedge a_{t+1}) \rightarrow \bigvee\{o_t \mid a \in e\}$

See [KAUTZ and SELMAN, 1992].

LTL Formulae

Example: Translation of $\mathbf{F}\mathbf{G}a$ for bound 2. Possible infinite execution paths:

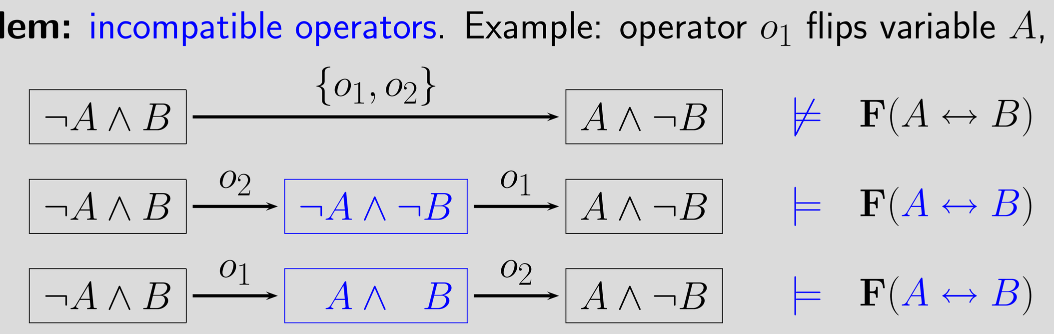


Thus $\mathbf{FG}a$ translates to $(loopto_0 \wedge a_0 \wedge a_1) \vee (loopto_1 \wedge a_1)$.
For details see [LATVALA et al., 2004].

Parallelism

Higher Efficiency through Parallel Plans: For n operators there are $n!$ possible orderings. Orderings may be equivalent or completely irrelevant. Therefore ignore ordering if possible. Leads to shorter plans, faster planning.

Problem: incompatible operators. Example: operator o_1 flips variable A, o_2 flips B.



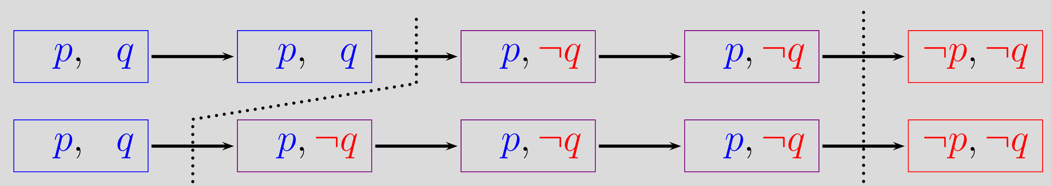
If only parallel execution is considered, a plan for $\mathbf{F}(A \leftrightarrow B)$ is overlooked.
Therefore: Make sure that parallel execution $\models \phi$ iff at least one serialized execution $\models \phi$.

Reduction to Satisfiability (continued)

Stuttering Equivalence

Definition: Two sequences π and $\tilde{\pi}$ of labeled states are stuttering equivalent ($\pi \sim \tilde{\pi}$) if they can be split into corresponding blocks of states with equal labels.

Example: Two stuttering equivalent paths

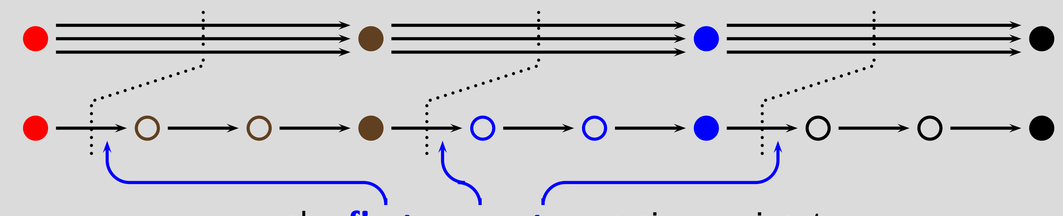


Theorem: [LAMPORT, 1983]
Let ϕ be an LTL$_{-X}$ formula and $\pi, \tilde{\pi}$ two state sequences labelled with the variables from ϕ. Then $\pi \sim \tilde{\pi}$ implies that $\pi \models \phi$ iff $\tilde{\pi} \models \phi$.

Consequence: It is sufficient to make sure that there is at least one serialized plan execution such that serialized execution $\sim$ parallel execution wrt the variables in ϕ.

Restriction of Parallelism

Make sure that there is a serialization such that for all time points t ...



... the **first operator** at time point t ...

... causes all effects relevant to ϕ at t.

Restrictions on Operators: Operator o may only precede operator o' at time point t if o causes all effects relevant to ϕ caused by o' at time point t. Thus o disables o' if o' might have some effect relevant to ϕ which o does not have. (Additionally, o disables o' if o falsifies a precondition of o' or affects the set of active effects of o'.)

Encoding via Disabling Graph

Definition: A Disabling Graph is a graph on set of operators with an edge from o to $o' \neq o$ if o and o' are simultaneously applicable in a reachable state and o disables o'.

Encoding: Encode acyclicity of subgraph of Disabling Graph induced by applied operators. If encoding is satisfied, there must be a serialization whose execution is stuttering equivalent to the parallel execution. Encoding has linear size.

For details see [RINTANEN et al., 2006].

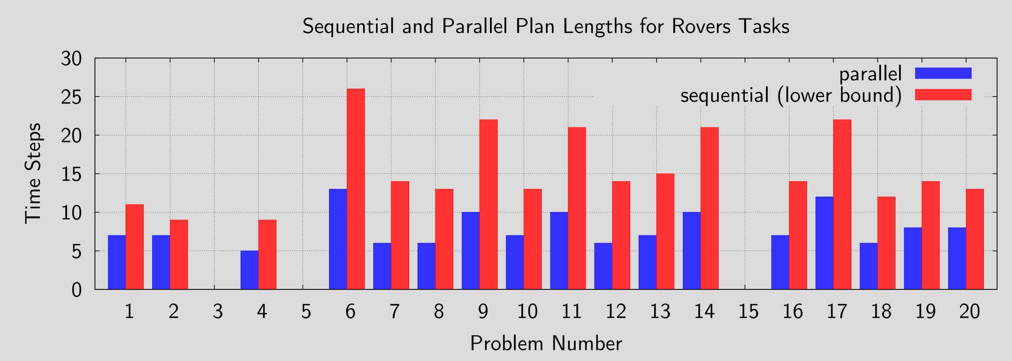
3 Experiments and Results

Experiments

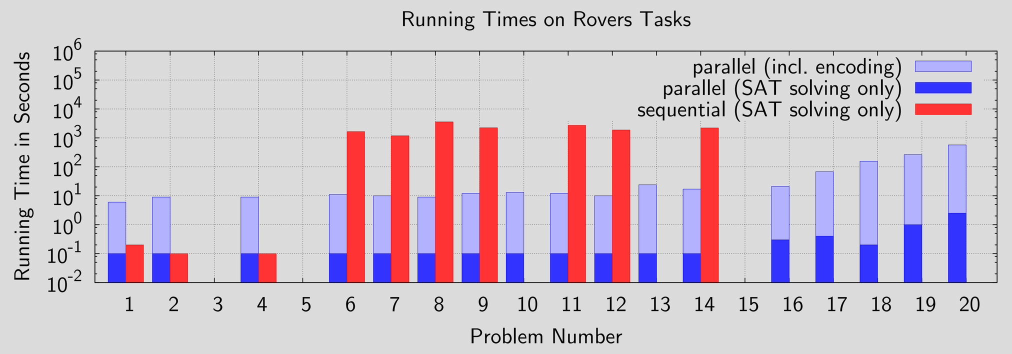
Comparison: parallel vs. purely sequential encoding
Benchmarks problems: qualitative preferences Rovers tasks from IPC 2006 with soft constraints turned into hard constraints, no metric function, randomly dropped constraints to keep problems solvable (retained three constraints per problem)
System: planner implementation in SML, SAT solver Siege V4 [RYAN, 2004],
1.8 GHz AMD Athlon 64, 768 MB RAM, Linux.

Results

Plan lengths/Time steps:
Sequential: lower bound.
Problems 3, 5, and 15: unsolvable.



Running times:
Dark: SAT solving only
Light: Preprocessing, encoding, SAT solving, decoding.
Missing values: unsolvable or 1h timeout exceeded.



4 Conclusion

Combining existing techniques for SAT planning, bounded LTL model-checking, and partial order reduction results in a reasonably efficient method of planning for TEGs. The experimental results show that, like in classical SAT based planning and in Graphplan, admitting parallelism can noticeably speed up SAT based planning for TEGs.

References

– KAUTZ, SELMAN, *Planning as Satisfiability.* In: Proc. ECAI, 1992, pp. 359–363.
– LAMPORT, *What Good is Temporal Logic?* In: Inform. Processing, 1983, pp. 657–668.
– LATVALA et al. *Simple Bounded LTL Model Checking.* In: LNCS 3312, 2004, pp. 186–200.
– RINTANEN et al. *Planning as Satisfiability: Parallel Plans and Algorithms for Plan Search.* Artif. Intell. 180(12–13):1031–1080, 2006.

Contact: Robert Mattmüller <mattmuel@informatik.uni-freiburg.de>, Research Group on the Foundations of Artificial Intelligence, Institute for Computer Science, University of Freiburg, Georges-Köhler-Allee, Bldg. 052, 79110 Freiburg, Germany
Jussi Rintanen <Jussi.Rintanen@nicta.com.au>, National ICT Australia and Australian National University, Canberra Research Laboratory, Locked Bag 8001, Canberra ACT 2601, Australia