

Evaluating Temporal Planning Domains

Juraj Dzifcak

Arizona state university
School of Computing and Informatics
699 S. Mill Avenue
Tempe, AZ 85281-8809, U.S.A.

April 29th, 2008

Introduction and Motivation

- Many international planning competition domains do not require concurrency
- Some theoretical work already present
- Knowing if the domain does require concurrency might be very useful
- Various concurrency based planning results

Challenges

- Translating theoretical results into algorithms,
- Deciding if a domain requires concurrency or not is hard
- Combining the results
- Implementation
- Evaluation

Some interesting results

Results include

- Any repeatable action is either a weakly conditional action, a null action or a null effect action.
- If two instances of a simple durative action, can execute concurrently, then either a is either a deadlocking, pseudo-durative or purely state-preserving action, or else A_e is weakly conditional.
- If a domain forbids temporal gap, the domain is inherently sequential.
- If every action in the domain is AT START causally independent or AT END causally independent then the domain is inherently sequential.

Decidability?

- Deciding if the domain can be solved without concurrency seems to be most helpful
- Deciding if the domain cannot be solved without concurrency seems to be hard
- On which domains should we test it?

- Selected results

Test no	Domain	Sequential?	Time
1	Depots	Yes	N/C
2	DriverLog	Yes	N/C
3	ZenoTravel	Yes	N/C
4	Light and match 1	Unknown	N/C
5	Light and match 2	Unknown	N/C

- Results are in line with the theory
- Time variation does not yield conclusive evidence on usefulness, however for the given domains the required time was very short ($\leq 0.1s$)

High level implementation overview

- 1 Parse the domain
- 2 Check for temporal gaps and print out the actions with them. If no gaps are found, the domain must be inherently sequential.
- 3 Check for At Start and At End causal independence. If every action is found to be independent, the domain must be inherently sequential.
- 4 Check for action concurrency. If an action cannot occur concurrently with itself report this action.
- 5 Based on the above points, classify the domain as either inherently sequential or unknown.

- Translating results into algorithms
- First step in implementing the theoretical advances
- Several types of 'checkers'
- Optimization

Thank you for your attention.