Relaxing the Relaxed Exist-Step Parallel Planning Semantics

Tomáš Balyo (biotomas@gmail.com)
Faculty of Mathematics and Physics,
Charles University in Prague, Czech Republic

What is Planning?

- World states are described as values of state variables
- Actions change the state of the world by changing the values of state variables by their effects
- Actions also have preconditions and are applicable only when their preconditions hold in the given state

Objective: given a set a of actions, an intial world state and the description of a goal state find a valid sequence of actions that transforms the world from the initial state to a goal state

State Variables and their domains:

- Truck location T, dom(T)={LA, SF, LV}
- Package locations P and Q

dom(P) = dom(Q) = {LA, SF, LV, Tr}

Initial State: T=LA, P=LA, Q=SF

Goal State: P=LV, Q=LV

Actions:

- move(x,y)=[prec: {T=x}, eff: {T=y}]
- loadP(x)=[prec: {T=x, P=x}, eff: {P=Tr}]
- loadQ(x)=[prec: {T=x, Q=x}, eff: {Q=Tr}]
- dropP(x)=[prec: {T=x, P=Tr}, eff: {P=x}]
- dropQ(x)=[prec: {T=x, Q=Tr}, eff: {Q=x}]

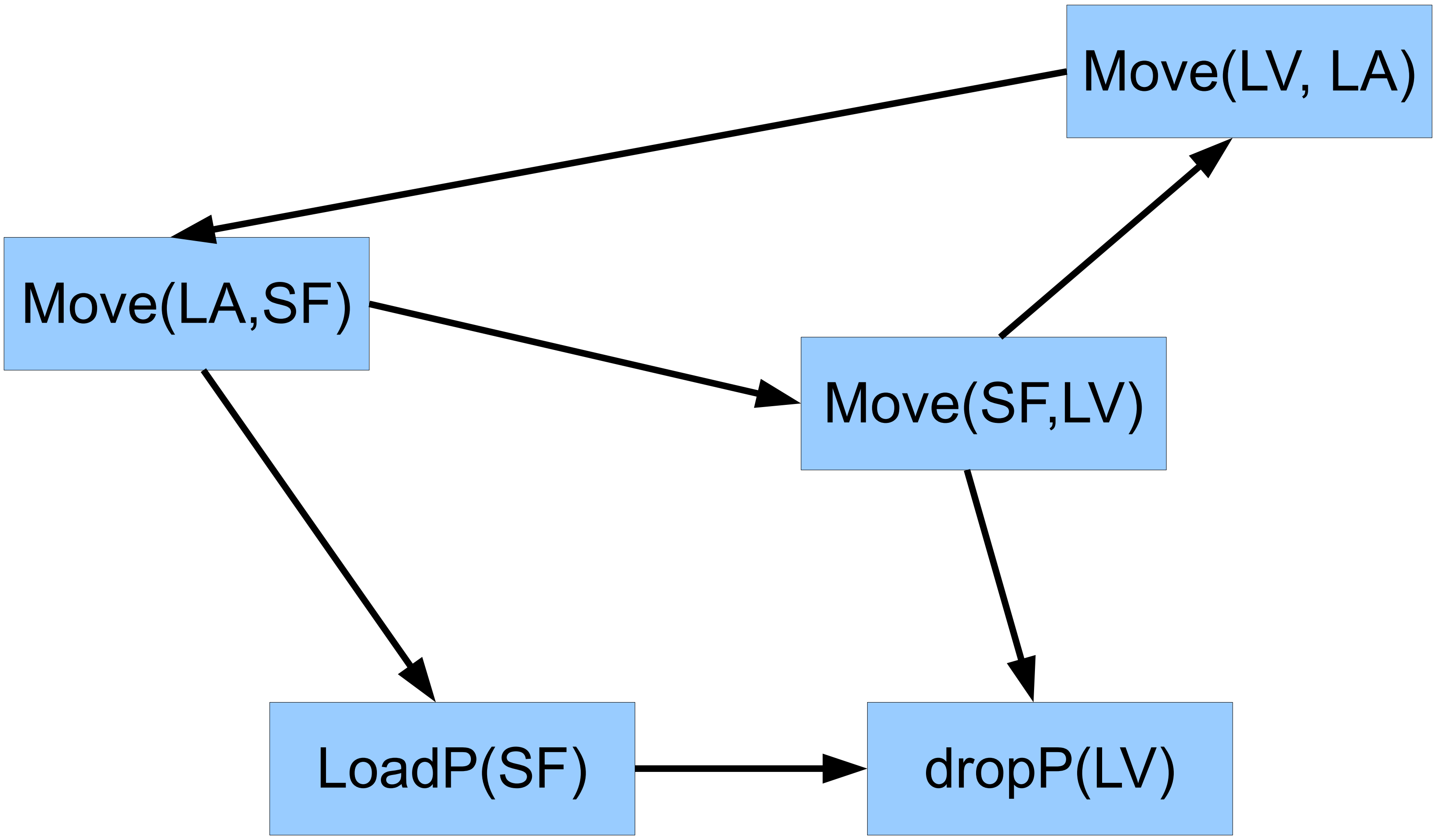
Where x,y are LA, SF, and LV

Plan: loadP(LA), move(LA, SF), loadQ(SF), move(SF, LV), dropP(LV), dropQ(LV)

Enabling Graph

$$V = \{ Actions \}$$

$$E = \{ (A_1 \rightarrow A_2), eff(A_1) \cap prec(A_2) \neq 0 \}$$



Planning as SATisfiability

- Construct a formula F_k such that it is satisfiable (if and) only if there is a plan of at most k steps
- Solve F_1, F_2, … using a SAT solver until you reach a satisfiable formula F_n
- Extract a plan from the satisfying assignment of F_n
- n is the called the makespan of the plan

- ***What actions can go inside a step together?***
  - If more action could be in a step then we would need fewer steps to find a plan

What actions can go inside a step together?

1. foreach step semantics

- The preconditions of all actions in a step must already hold in the beginning of the step
- The effects of all actions must hold at the end of this step
- The actions in a step do not interfere – they cannot destroy each others preconditions by their effects
- The actions in a step can be turned into a valid subplan sequence

- Plan: {loadP(LA)} ♦ {move(LA, SF)} ♦ {loadQ(SF)} ♦ {move(SF, LV)} ♦ {dropP(LV), dropQ(LV)} – **5 steps**

What actions can go inside a step together?

2. exist step semantics

- The preconditions of all actions in a step must already hold in the beginning of the step
- The effects of all actions must hold at the end of this step
- ~~The actions in a step do not interfere – they cannot destroy each others preconditions by their effects~~
- The actions in a step can be turned into a valid subplan sequence

- Plan: {loadP(LA), move(LA, SF)} ♦ {loadQ(SF), move(SF, LV)} ♦ {dropP(LV), dropQ(LV)} – **3 steps**

What actions can go inside a step together?

3. relaxed exist step semantics

- ~~The preconditions of all actions in a step must already hold in the beginning of the step~~
- The effects of all actions must hold at the end of this step
- ~~The actions in a step do not interfere – they cannot destroy each others preconditions by their effects~~
- The actions in a step can be turned into a valid subplan sequence

- Plan: {loadP(LA), move(LA, SF), loadQ(SF)} ♦ {move(SF, LV), dropP(LV), dropQ(LV)} – **2 steps**

What actions can go inside a step together?

4. relaxed relaxed exist step semantics

- ~~The preconditions of all actions in a step must already hold in the beginning of the step~~
- ~~The effects of all actions must hold at the end of this step~~
- ~~The actions in a step do not interfere – they cannot destroy each others preconditions by their effects~~
- The actions in a step can be turned into a valid subplan sequence

- Plan: {loadP(LA), move(LA, SF), loadQ(SF), move(SF, LV), dropP(LV), dropQ(LV)} – **1 step**

Basic ideas of the relaxed relaxed exist step SAT encoding

- The SAT encoding only approximates the semantics, i.e., the satisfiability of the constructed formula F_k implies the existence of a k-step plan (not vice versa)
- The actions are ranked using cycle–ignoring topological sorting on the enabling graph (action ranking can be arbitrary as long as it is injective)
- The encoding allows only lower ranking actions before higher ranking ones in a step
- The encoding uses implication chains similar to those used in the exist step and relaxed exist step encoding

Experimental Results

- IPC 2012 domains (20 problems each), time limit 30 minutes

Domain	Foreach Step		Exist Step		Relaxed Relaxed E.S.	
	Solved	Avg. Steps	Solved	Avg. Steps	Solved	Avg. Steps
Barman	8	46.3	8	36.6	14	14.8
Elevators	20	9.5	20	6.5	20	4.3
Parcprinter	20	13.5	20	13.5	20	1.5
Pegsol	7	22.8	13	24.0	19	8.6
Storage	15	9.2	19	7.9	19	4.3
Visitall	9	27.0	11	31.4	20	1.7
Woodwork	20	3.4	20	3.3	20	1.7
Zenotravel	16	5.9	16	4.5	15	2.7

Conclusion

- We have defined a novel parallel planning semantics and a SAT encoding which approximates it
- The results of the experiments show that the new encoding is successfull in solving IPC benchmark problems
- For the domains Pegsol, Barman, and Visitall we achieved a significant improvement in the number of solved instances
- The average number of required steps decreased for all domains, most significantly for the Visitall domain
- The encoding can be further improved to produce smaller formulas and to better approximate the defined semantics