Forward-Search Temporal Planning with Simultaneous Events

Daniel Furelos-Blanco[1], Anders Jonsson[1], Héctor Palacios[2] and Sergio Jiménez[3]

[1]Universitat Pompeu Fabra [2]Nuance Communications [3]Universitat Politècnica de València

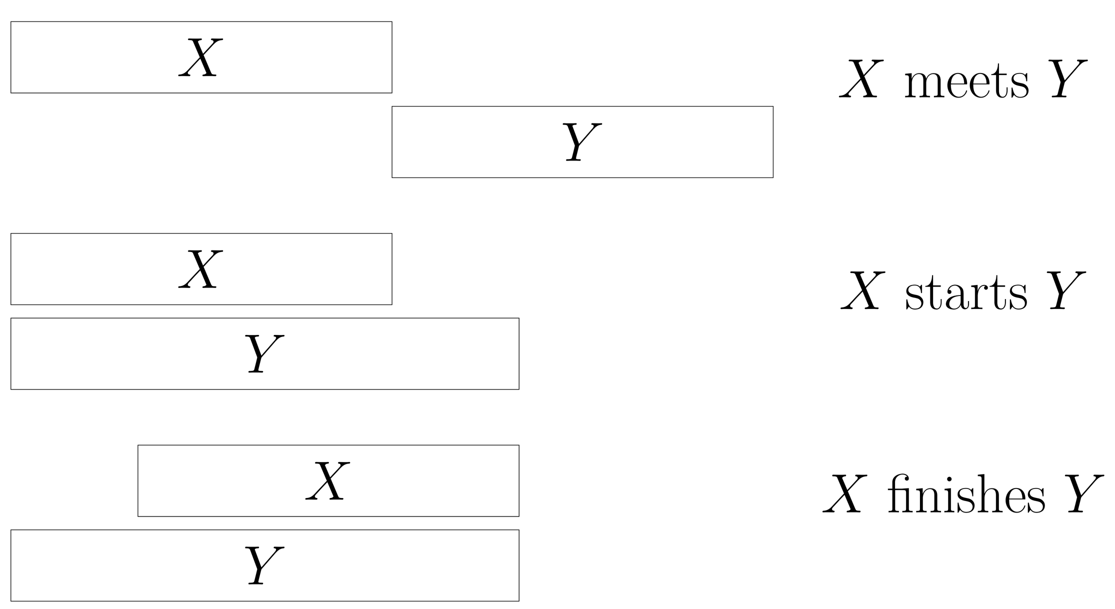
Motivations

1. Many situations in the real-world involve **simultaneous events**.

2. PDDL 2.1 induces **temporal gaps** [1]:

- State-of-the-art planners using PDDL do not solve problems with simultaneous events.
- Potentially, more decision points.

3. Allen's Interval Algebra (AIA) [2], a domain that requires simultaneous events.

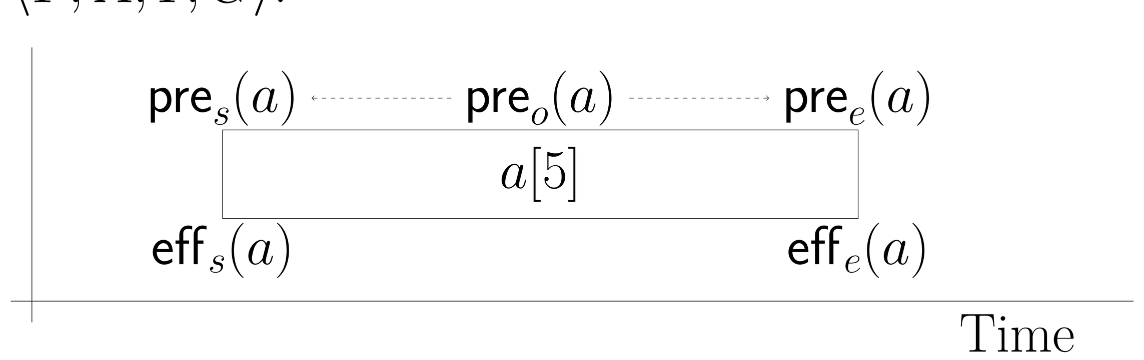


Proposed Approach

1. Compile the temporal problem into a classical problem.
2. Solve the classical problem using the Fast Downward planner + STNs to check temporal consistency.

Temporal Planning

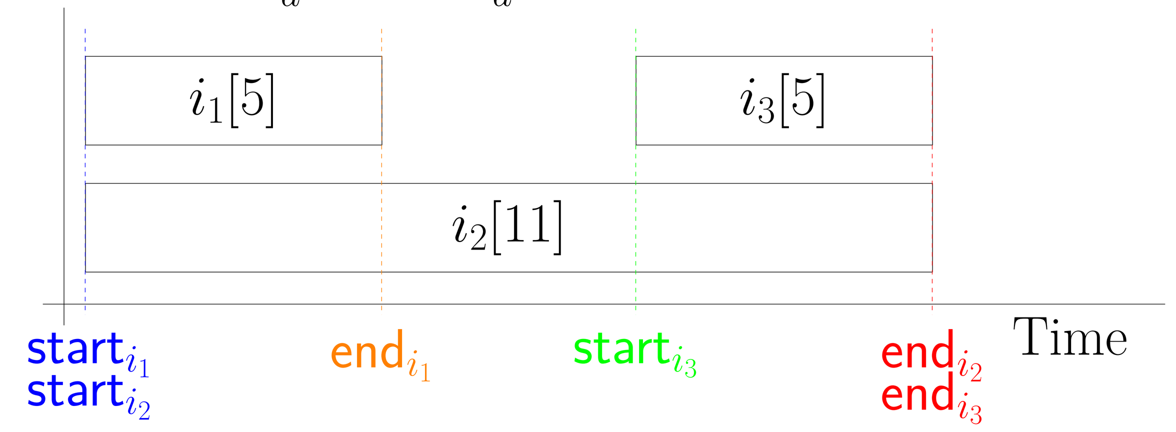
A temporal planning **problem** is a tuple $P = \langle F, A, I, G \rangle$.



A temporal **plan** is a list of time-action pairs. The quality of a plan is given by its **makespan**.

Simultaneous Events

A temporal action a can be defined in terms of two events: start$_a$ and end$_a$.



- Given an individual event e, no effect of e can be mentioned by another event simultaneous with e [3].

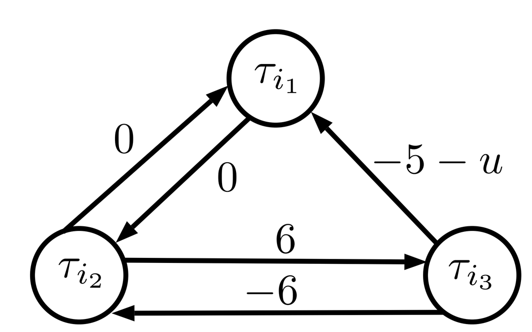
Simple Temporal Networks (STN)

They are used to represent **temporal constraints** on time variables using a directed graph:

- Nodes = time variable τ_i.
- Edges (τ_i, τ_j) with label c = constraints $\tau_j - \tau_i \leq c$.

Example

$\tau_{i_1} - \tau_{i_3} \leq -5 - u,$
$\tau_{i_1} - \tau_{i_2} \leq 0,$
$\tau_{i_2} - \tau_{i_1} \leq 0,$
$\tau_{i_3} - \tau_{i_2} \leq 6,$
$\tau_{i_2} - \tau_{i_3} \leq -6.$



$\tau_{i_1} = 0, \tau_{i_2} \in [-d_{21}, d_{12}] = [0, 0] \rightarrow \tau_{i_2} = 0$
$\tau_{i_3} \in [-d_{31}, d_{13}] = [6, 6] \rightarrow \tau_{i_3} = 6$

The STP Planner

Extension of the TP planner [2]:

- Add STNs to Fast Downward.
- Bound K on the number of active actions.
- Problems with fixed durations and no duration dependent effects.

Each joint event is divided in **3 phases**:

1. End phase: active actions are scheduled to end.
2. Event phase: simultaneous events take place.
3. Start phase: check that pre$_o$ of active actions are satisfied.

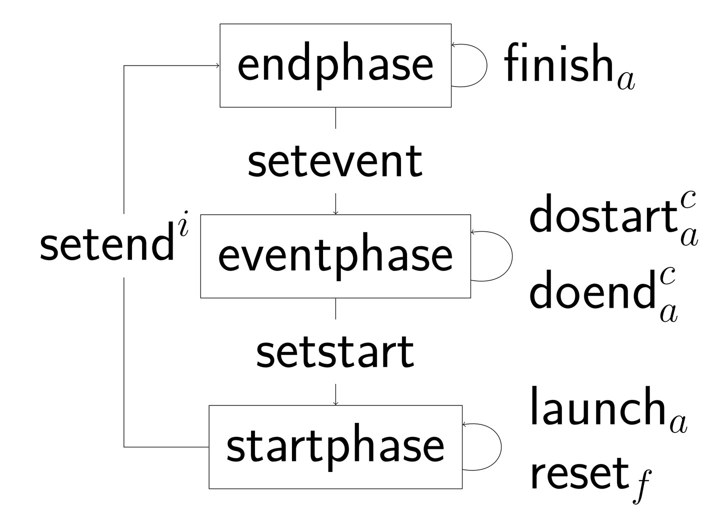


Figure 1:Interaction between the different actions introduced by the STP planner in the different phases.

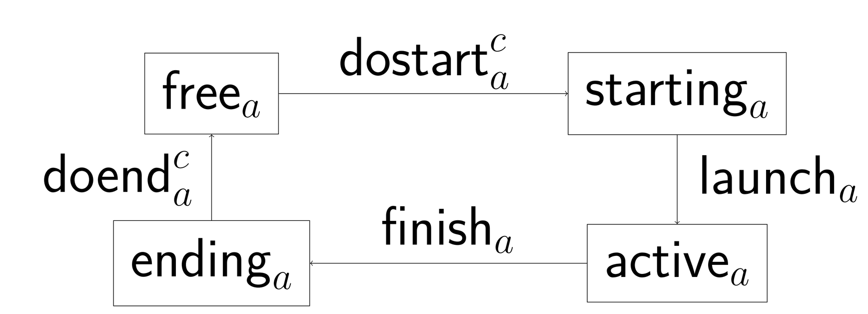


Figure 2:Fluents that are enabled each time an action in the compilation is executed.

Results

	TPSHE	TP(2)	TP(3)	TP(4)	STP(2)	STP(3)	STP(4)	POPF2	YAHSP3-MT	ITSAT
AIA[25]	3/3	6.5/8	7.5/9	8.5/10	17.17/22	19.51/24	**23.5/25**	10/10	3/3	3/3
CUSHING[20]	0/0	0/0	4.07/**20**	4.93/**20**	0/0	3.31/14	2.28/5	**20/20**	0/0	0/0
DRIVERLOG[20]	**14.78/15**	0.93/3	1.08/4	0.91/3	0/0	0/0	0/0	0/0	2.31/4	1/1
DLS[20]	9.37/11	10/10	7.7/9	8.06/9	3.78/4	3.9/4	3.49/4	7/7	0/0	**16.18/19**
FLOORTILE[20]	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	4.93/5	**19.7/20**
MAPANALYSER[20]	**17.38/20**	13.08/**20**	12.34/**20**	12.02/19	9.81/17	10.09/16	7.69/12	0/0	1/1	0/0
MATCHCELLAR[20]	15.72/**20**	15.71/**20**	15.71/**20**	15.71/**20**	15.71/**20**	15.71/**20**	15.71/**20**	**20/20**	0/0	18.91/19
PARKING[20]	6.73/**20**	5.79/17	5.67/17	5.33/16	1.79/6	1.93/6	1.93/6	12/13	**16.84/20**	0.96/6
RTAM[20]	**16/16**	2.45/6	2.73/6	2.79/6	0/0	0/0	0/0	0/0	0/0	0/0
SATELLITE[20]	**16.63**/18	4.97/13	5.04/13	4.67/12	0/0	0/0	0/0	2.92/3	13.82/**20**	1.68/7
STORAGE[20]	4.92/**9**	0/0	0/0	0/0	0/0	0/0	0/0	0/0	3.91/**9**	**9/9**
TMS[20]	0.06/9	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	**16/16**
TURN&OPEN[20]	**15.53/19**	5.05/10	5.03/10	5.19/10	0/0	0/0	0/0	7.31/8	0/0	5.88/6

Conclusions

- Method that returns sound temporal plans if used in a forward-search planner maintaining STNs.
- Good performance in a domain requiring simultaneous events.
- Not competitive in combinatorially challenging domains requiring simpler forms of concurrency.

Future work: Analyze problems before solving them.

References

[1] Jussi Rintanen.
Models of Action Concurrency in Temporal Planning.
In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015.*, pages 1659–1665, 2015.

[2] Sergio Jiménez, Anders Jonsson, and Héctor Palacios.
Temporal Planning With Required Concurrency Using Classical Planning.
In *Proceedings of the Twenty-Fifth International Conference on Automated Planning and Scheduling, ICAPS 2015.*, pages 129–137, 2015.

[3] Maria Fox and Derek Long.
PDDL2.1: An Extension to PDDL for Expressing Temporal Planning Domains.
J. Artif. Intell. Res. (JAIR), 20:61–124, 2003.

Acknowledgements

This work has been partially funded by the Maria de Maeztu Units of Excellence Programme (MDM-2015-0502).

Contact Information

- Software: https://github.com/aig-upf/temporal-planning
- Email: daniel.furelos@upf.edu

Universitat Pompeu Fabra Barcelona
NUANCE
UNIVERSITAT POLITÈCNICA DE VALÈNCIA
EXCELENCIA MARÍA DE MAEZTU