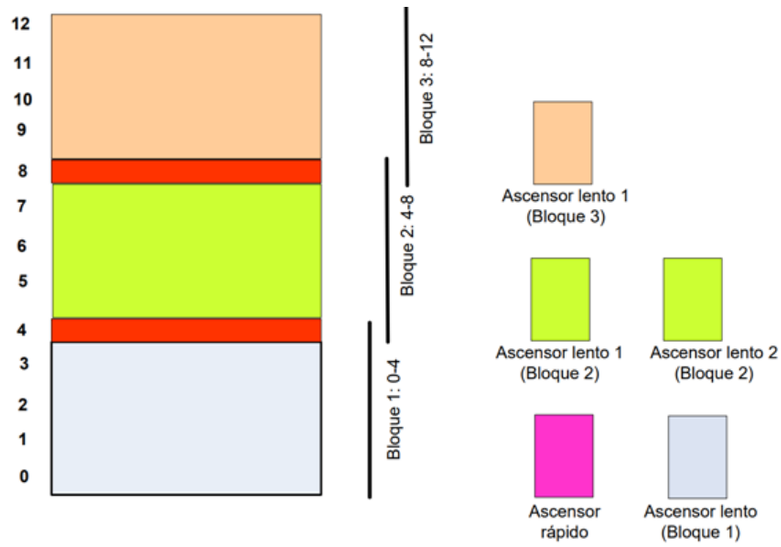
Ascensores

Tommaso Sacchetti, Fabian Sparbrodt

14/02/2022

1 Introduction of the Domain and Sample Instances



1.1 Elevators - A PDDL Sample Domain

The planning task shall be rolled out in an examplary domain were elevators take passengers from initial locations to their destination floors.

So the sample domain consists of the following:

```
person
floor
boarded passangers
slow-elevator fast-elevator
```

Elevators are divided into ”fast” elevators and ”slow” elevators. In the domain, only one fast elevator exists which can read all even floors. The slow elevators on the other hand have a certain block range where they are able to move a single floor, up or down. Further, slow elevators have a specific range. The first block reaches from floors (1-4) and is served by one slow elevator, the second from (4-8) and is served by two, and ranges (8-12) are served by another such elevator.

Consequently, the actions of the domain are:

```
move-up-fast
move-down-fast
move-up-slow
move-up-fast
get-inside
get-outside
```

A set of predicates, associated with the defined actions, is needed to check elevator and passanger locations, the capacities of elevators, the feasability of moves, etc.

The first planning exercise is unconstrained with repect to time or resources and only initial states and the set of actions to reach a certain goal state are regarded.

For this, sample instantiations of the domain have been implemented.

1.2 Planning

Given the described PDDL domain file, sample instantions have been tried out to check the behavior of different planners. All of the planners FF, LPG and the optic planner returned a valid plan while there were some qualitative differents in the obtained plans. Especially in this unconstrained domain, the FF planner came out on top, regarding the median amount of actions needed to reach the goal state of the instantiations provided. The FF planner calculates the shortest plan (least actions) in the fastest time while the LPG with no-timesteps was measurably slower and involved more actions than FF before reaching the goal state. The Optic planner on the other hand was more than 5 times slower than the LPG planner but the resulting plans were slightly shorter.

1.3 Various Instantiations

Different combinations of passangers and elevator configurations have been instantiated for the domain. Please refer to the appended files to see the different instantiations.

2 Introducing Time Constraints

2.1 The Domain

No major changes were applied to the structure of the previous domain in order to define the temporal domain. The only modfication executed was transforming the actions into durative-actions, adding a duration and adjusting the preconditions and effects to happen at start, at end or over all the duration of the durative-action.

2.2 Planning

The temporal domain and instances were tested with both LPG and OPTIC planners. Only LPG could return a valid plan while OPTIC returns the following output:

```
; Goals unreachable from the initial state
;; Problem unsolvable!
```

The LPG planner instead found the plan presented in the appendix A.

3 Limiting Resources

As a resource it was decided to use the electrical energy consumed by the elevator and is calculated as one over the duration of the action.

The two variations of the domain are created and in the limited version each elevator has its own battery that can be recharged simply by staying still in a floor and using the durative-action recharge-elevator.

It was observed that unless some tight constraints over the starting battery of the elevators are inserted, the planner will always to try to create a solution without never recharging the elevators, and if necessary, it recharges the battery as few times as possible, also at the cost of creating a solution that is longer.

The domains were ran with two instances: the first is the base instance of the given problem while the second is an instance with eleven passengers, two fast elevators and where slow elevators can carry up to four people while the fast have a capacity of five people.

Both the instances of the domains were tested minimizing the total energy used and later, the total plan length.

Two files with resulting plans are attached in the folder "3.2 - plans".

LPG behaved in an odd way, since some of the plans obtained minimizing time require more time than the plans obtained minimizing the energy. In particular the plan for the domain with limited resources, with the second instance of the problem (the bigger one), takes double the time with the minimized-time version. Only the domain with unlimited resources ran with the second instance had a drastically better plan with the minimize-time (considering only the duration).

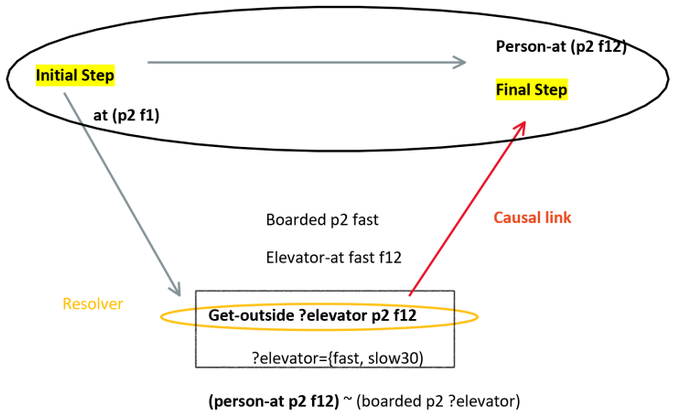
4 Partially Ordered Plan

Partially orderdered Plan Tree (POP) results in the following:

Let us assume that there is only one passenger, initially located at the first floor. Said passenger wants to move to the 12th floor.

4.1 First Iteration

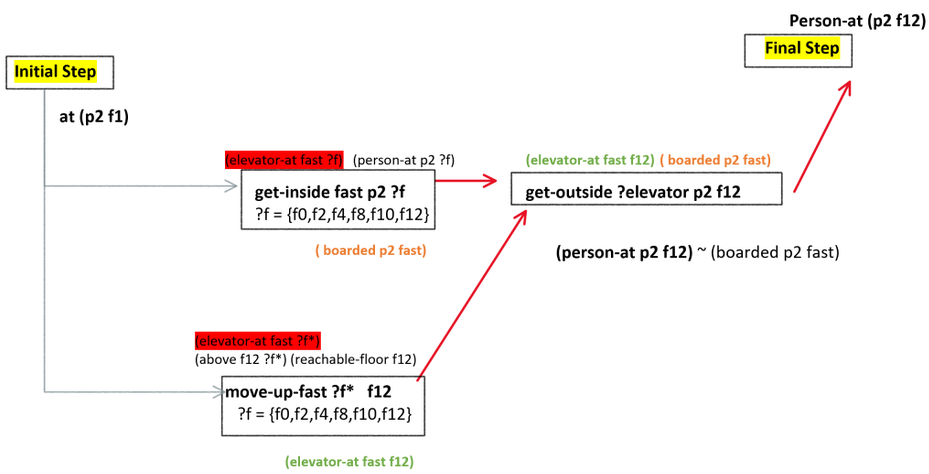
The POP starts from this goal state, where in the first iteration, the only flaw is the open goal state: `person-at p2 f12`. Resolving this flaw results in the action `get-outside ?elevator p2 f12`, where the variable elevator is still undefined. A causal link exists betwen this action and the final step (goal state). Through random choice, the `elevator` variable is set to the fast elevator. Now, the flaws consist of the two new open goals. The first concerns how the elevator would get to the final destination floor: `elevator-at fast f12` and the second, how (where) Person 2 would get inside this elevator `boarded p2 fast`.



4.2 Second Iteration

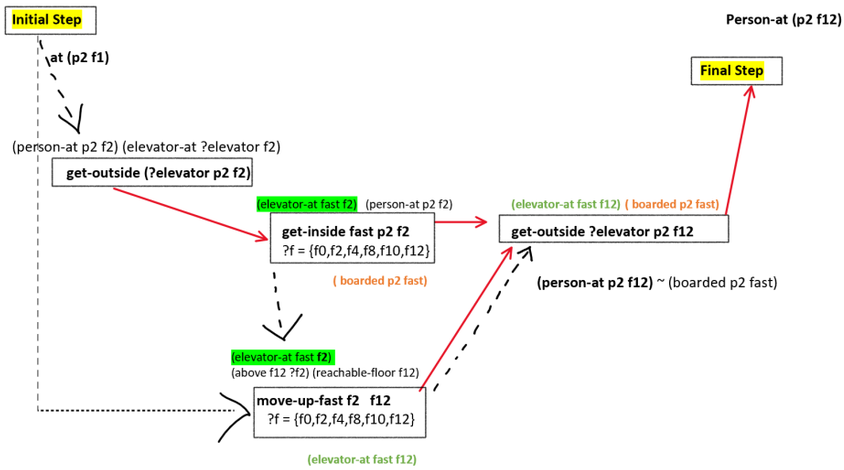
In this next iteration, the algorithm would resolve these open goals via the following actions. Let us suppose the algorithm chooses to resolve the unsolved variable flaw **boarded p2 fast**. `get-inside fast p2 ?f` would be a resolving method, to resolve the green marked precondition, the algorithm would choose `move-up-fast ?f* f12`. As can be seen marked in red, these two resolving actions have a precondition involving the same yet undetermined variable of type `?floor?`. These two actions can therefore not be parallelized, and an ordering constraint needs to be introduced.

`(elevator-at fast ?f)`



4.3 Third Iteration

Resolving the next flaw of the undetermined floor variable by setting it to the second floor `f2` implies an ordering constraint, where `get-inside` needs to happen before `move-up-fast`. The newly picked action `get-outside?elevator p2 f2` has been picked to resolve the open goal of `person-at fast f2`. The ordering constraints can be seen marked as dotted lines while causal links are marked in red.


Initial Step
Person-at (p2 f12)
at (p2 f1)
Final Step
(person-at p2 f2) (elevator-at ?elevator f2)
get-outside (?elevator p2 f2)
(elevator-at fast f2) (person-at p2 f2)
(elevator-at fast f12) (boarded p2 fast)
get-inside fast p2 f2
?f = {f0,f2,f4,f8,f10,f12}
get-outside ?elevator p2 f12
(boarded p2 fast)
(person-at p2 f12) ~ (boarded p2 fast)
(elevator-at fast f2)
(above f12 ?f2) (reachable-floor f12)
move-up-fast f2 f12
?f = {f0,f2,f4,f8,f10,f12}
(elevator-at fast f12)

5 Graphplan

The calculated values of the heuristics are: h-max = max(4,7)=7 and h-sum = 4+7 = 11.

The extracted plan is:

```
move-up-slow slow10 f0 f1
move-up-fast fast f0 f4
get-inside slow10 p0 f1
get-inside fast p1 f4
move-up-fast fast f4 f10
move-up-slow slow10 f1 f3
get-outside slow10 p0 f3 [GOAL-1 REACHED]
get-outside fast p1 f10
get-inside slow30 p1 f10
move-up-slow slow30 f10 f11
get-outside slow30 p1 f11 [GOAL-2 REACHED]
```

It's possible to extract in polynomial time from the planning graph since at every stage it was possible to choose non conflicting actions and it was possible to build a plan with no mutex, hence there was no necessity to backtrack the plan.

Out of the three heuristics, the most informed one for the given problem is the relaxed plan, since the problem is not particularly prone to mutex relations and the ones that appear are easily fixable (in this particular instance, the elevators are five and the passengers are two so it's unlikely that the plan will be particularly hard to extract from the relaxed plan).

Appendices

A LPG temporal plan

```
      Time: (ACTION) [action Duration; action Cost]
0.0000: (MOVE-UP-SLOW SLOW10 F0 F2) [D:0.00; C:0.10]
0.0000: (GET-INSIDE SLOW10 P0 F2 I0 I1) [D:2.00; C:0.10]
0.0000: (MOVE-UP-SLOW SLOW10 F2 F3) [D:0.00; C:0.10]
0.0000: (GET-INSIDE SLOW21 P1 F4 I0 I1) [D:2.00; C:0.10]
0.0000: (MOVE-UP-SLOW SLOW20 F4 F8) [D:0.00; C:0.10]
0.0000: (GET-INSIDE SLOW20 P3 F8 I0 I1) [D:2.00; C:0.10]
0.0000: (MOVE-DOWN-SLOW SLOW20 F8 F4) [D:40.00; C:0.10]
0.0000: (MOVE-UP-SLOW SLOW21 F4 F7) [D:0.00; C:0.10]
0.0000: (MOVE-UP-SLOW SLOW21 F7 F8) [D:0.00; C:0.10]
2.0000: (GET-OUTSIDE SLOW10 P0 F3 I1 I0) [D:2.00; C:0.10]
2.0000: (GET-OUTSIDE SLOW21 P1 F8 I1 I0) [D:2.00; C:0.10]
2.0000: (MOVE-UP-SLOW SLOW10 F3 F4) [D:0.00; C:0.10]
4.0000: (GET-INSIDE SLOW30 P1 F8 I0 I1) [D:2.00; C:0.10]
4.0000: (MOVE-UP-SLOW SLOW30 F8 F12) [D:0.00; C:0.10]
4.0000: (MOVE-DOWN-SLOW SLOW30 F12 F11) [D:12.00; C:0.10]
16.0000: (GET-OUTSIDE SLOW30 P1 F11 I1 I0) [D:2.00; C:0.10]
16.0000: (MOVE-DOWN-SLOW SLOW30 F11 F10) [D:12.00; C:0.10]
40.0000: (GET-OUTSIDE SLOW20 P3 F4 I1 I0) [D:2.00; C:0.10]
42.0000: (GET-INSIDE SLOW10 P3 F4 I0 I1) [D:2.00; C:0.10]
42.0000: (MOVE-DOWN-SLOW SLOW10 F4 F1) [D:30.00; C:0.10]
72.0000: (GET-OUTSIDE SLOW10 P3 F1 I1 I0) [D:2.00; C:0.10]
74.0000: (GET-INSIDE SLOW10 P4 F1 I0 I1) [D:2.00; C:0.10]
76.0000: (GET-INSIDE SLOW10 P2 F1 I1 I2) [D:2.00; C:0.10]
76.0000: (MOVE-UP-SLOW SLOW10 F1 F3) [D:0.00; C:0.10]
78.0000: (GET-OUTSIDE SLOW10 P4 F3 I2 I1) [D:2.00; C:0.10]
80.0000: (GET-INSIDE SLOW10 P4 F3 I1 I2) [D:2.00; C:0.10]
80.0000: (MOVE-DOWN-SLOW SLOW10 F3 F0) [D:30.00; C:0.10]
110.0000: (GET-OUTSIDE SLOW10 P4 F0 I2 I1) [D:2.00; C:0.10]
112.0000: (GET-INSIDE FAST P4 F0 I0 I1) [D:2.00; C:0.10]
112.0000: (MOVE-UP-FAST FAST F0 F4) [D:13.00; C:0.10]
112.0000: (GET-OUTSIDE SLOW10 P2 F0 I1 I0) [D:2.00; C:0.10]
125.0000: (MOVE-UP-FAST FAST F4 F10) [D:15.00; C:0.10]
140.0000: (GET-OUTSIDE FAST P4 F10 I1 I0) [D:2.00; C:0.10]
140.0000: (MOVE-DOWN-FAST FAST F10 F0) [D:18.00; C:0.10]
142.0000: (GET-INSIDE SLOW30 P4 F10 I0 I1) [D:2.00; C:0.10]
142.0000: (MOVE-DOWN-SLOW SLOW30 F10 F9) [D:12.00; C:0.10]
154.0000: (GET-OUTSIDE SLOW30 P4 F9 I1 I0) [D:2.00; C:0.10]
158.0000: (GET-INSIDE FAST P2 F0 I0 I1) [D:2.00; C:0.10]
158.0000: (MOVE-UP-FAST FAST F0 F12) [D:21.00; C:0.10]
179.0000: (GET-OUTSIDE FAST P2 F12 I1 I0) [D:2.00; C:0.10]
```