

Artificial intelligence - Project 3

- Planning -

Vesa Bianca

13/01/2021

1 Domain Definition

1.1 First Aid Protocol

In this section the solution for the following problem will be presented:

Planning Domain Definition Language (PDDL) is the input language of most planning tools. The PDDL language supports many different levels of expressivity starting with STRIPS (Stanford Research Institute Problem Solver) or ADL (Action Description Language). Planning tasks formalised in PDDL are separated into two files: domain and problem to solve.

Your task is to model a planning domain in PDDL. You can either extend a given domain, or write one from scratch..

1.1.1 Code implementation

Code:

```
1  ( define (domain FirstAid )
2
3      (:types PERSON BODYPART DEGREE WOUND)
4
5      (:predicates (dead ?p - PERSON)
6                   (adult ?p - PERSON)
7                   (child ?p - PERSON)
8                   (infant ?p - PERSON)
9                   (conscious ?p - PERSON)
10                  (encircle_abdomen ?p - PERSON)
11                  (breathing ?p - PERSON)
12                  (obese ?p - PERSON)
13                  (pregnant ?p - PERSON)
14                  (free_airway ?p - PERSON)
15                  (complete_obstruction ?p - PERSON)
16                  (see_foreign_body ?p - PERSON)
17                  (infected ?p - PERSON ?w - WOUND)
18                  (dress_wound ?p - PERSON ?w - WOUND)
19                  (treat_wounds ?p - PERSON)
20                  (tetanus ?p - PERSON)
21                  (bleeding ?p - PERSON ?w - WOUND)
22                  (hasWound ?p - PERSON ?w - WOUND)
23                  (pressure ?p - PERSON ?w - WOUND)
24                  (antiseptic ?p - PERSON ?w - WOUND)
25                  (hasBodyP ?p - PERSON ?b - BODYPART)
26                  (first_deg_burn ?p - PERSON ?b - BODYPART)
27                  (second_deg_burn ?p - PERSON ?b - BODYPART)
28                  (third_deg_burn ?p - PERSON ?b - BODYPART)
29                  (treat_burns ?p - PERSON ?d - DEGREE)
30                  (redness ?p - PERSON ?b - BODYPART)
31                  (inflammation ?p - PERSON ?b - BODYPART)
32                  (pain ?p - PERSON ?b - BODYPART)
33                  (blistering ?p - PERSON ?b - BODYPART)
34                  (swollen ?p - PERSON ?b - BODYPART)
35                  (first_degree_burn ?p - PERSON ?b - BODYPART)
36                  (ems_wounds ?p - PERSON) ;; emergency medical services
37                  (ems_burns ?p - PERSON)
38                  (ems_airwayObs ?p - PERSON)
```

```

39      )
40
41      ;; check the obstruction of the person's airway
42      (:action sense_complete_obstruction
43         :parameters (?p - PERSON)
44         :observe (complete_obstruction ?p)
45      )
46
47
48      ;; check the visibilty of the foreign body in person's airway
49      (:action sense_foreign_body
50         :parameters (?p - PERSON)
51         :observe (see_foreign_body ?p)
52      )
53
54
55      ;; check if the person is conscious
56      (:action sense_consciousness
57         :parameters (?p -PERSON)
58         :observe (conscious ?p)
59      )
60
61
62      ;; check if the person can breathing
63      (:action sense_breathing
64         :parameters (?p -PERSON)
65         :observe (breathing ?p)
66      )
67
68
69      ;; check if able to encircle the person's abdomen
70      (:action sense_encircle_abdomen
71         :parameters (?p -PERSON)
72         :observe (encircle_abdomen ?p)
73      )
74
75
76      ;; check if person's body parts have redness
77      (:action sense_redness
78         :parameters (?p - PERSON ?b - BODYPART)
79         :observe (redness ?p ?b)
80      )
81
82
83      ;; check if person's body parts have inflammation
84      (:action sense_inflammation
85         :parameters (?p - PERSON ?b - BODYPART)
86         :observe (inflammation ?p ?b)
87      )
88
89
90      ;; check if person's body parts have blistering
91      (:action sense_blistering
92         :parameters (?p - PERSON ?b - BODYPART)

```

```

93         :observe (blistering ?p ?b)
94     )
95
96
97     ;; check if person's body parts are painful
98     (:action sense_pain
99         :parameters (?p - PERSON ?b - BODYPART)
100         :observe (pain ?p ?b)
101     )
102
103
104     ;; check if person's wound is infected
105     (:action sense_infection
106         :parameters (?p - PERSON)
107         :observe (infected ?p)
108     )
109
110
111     ;; check if person's wound is bleeding
112     (:action sense_bleeding
113         :parameters (?p - PERSON)
114         :observe (bleeding ?p)
115     )
116
117
118     ;; check if the person has immunization to tetanus
119     (:action sense_tetanus
120         :parameters (?p - PERSON)
121         :observe (tetanus ?p)
122     )
123
124     ;; if the person shows signs of mild airway obstruction,
125     ;; encourage coughing
126     (:action encourage_coughing
127         :parameters (?p - PERSON)
128         :precondition (and (breathing ?p)
129                             (not (dead ?p))
130                             (or (adult ?p) (child ?p))
131                             (conscious ?p)
132                             (not (complete_obstruction ?p))
133                         )
134         :effect (free_airway ?p)
135     )
136
137
138     ;; if the person shows signs of complete airway obstruction, but can breathe weakly
139     ;; apply five back blows
140     (:action five_back_blows
141         :parameters (?p - PERSON)
142         :precondition (and (breathing ?p)
143                             (not (dead ?p))
144                             (or (adult ?p) (child ?p) (infant ?p))
145                             (conscious ?p)
146                             (complete_obstruction ?p)

```

```

147         )
148         :effect (free_airway ?p)
149     )
150
151
152     ;; manually remove the foreign object in the airway
153     ;; only if it can be seen
154     (:action finger_sweep
155         :parameters (?p - PERSON)
156         :precondition (and (breathing ?p)
157                             (or (adult ?p) (child ?p) (infant ?p))
158                             (see_foreign_body ?p)
159                             (conscious ?p)
160                             (not (dead ?p))
161                         )
162         :effect (free_airway ?p)
163     )
164
165
166     ;; if the person is not able to breathe,
167     ;; is not an infant, pregnant or obese and the person's abdomen cannot be encircled
168     ;; give five abdominal thrusts
169     (:action abdominal_thrusts
170         :parameters (?p - PERSON)
171         :precondition (and (not (breathing ?p))
172                             (or (not (obese ?p)) (and (encircle_abdomen ?p) (obese ?p)))
173                             (not (pregnant ?p))
174                             (not (infant ?p))
175                             (conscious ?p)
176                             (not (dead ?p))
177                         )
178         :effect (free_airway ?p)
179     )
180
181
182     ;; if the person is not able to breathe
183     ;; is an infant, pregnant or obese and the person's abdomen cannot be encircled
184     ;; give five chest thrusts
185     (:action chest_thrusts
186         :parameters (?p - PERSON)
187         :precondition (and (not (breathing ?p))
188                             (or (pregnant ?p) (and (not (encircle_abdomen ?p)) (obese ?p)) (infant ?p))
189                             (conscious ?p)
190                             (not (dead ?p))
191                         )
192         :effect (free_airway ?p)
193     )
194
195
196
197     ;; if at anytime, the person becomes or is found unconscious or in cardiac arrest
198     ;; begin cpr and call EMS
199     (:action begin_cpr
200         :parameters (?p - PERSON)

```

```

201         :precondition (and (not (dead ?p))
202                             (or (adult ?p) (child ?p) (infant ?p))
203                             (not (conscious ?p))
204                         )
205         :effect (ems_airway0bs ?p)
206     )
207
208
209 ;; check person's bodypart for first degree burn
210 (:action check_first_degree_burn
211     :parameters(?p - PERSON ?b - BODYPART)
212     :precondition (and
213         (not (dead ?p))
214         (redness ?p ?b)
215         (swollen ?p ?b)
216         (pain ?p ?b)
217         (hasBodyP ?p ?b)
218     )
219     :effect (and (first_degree_burn ?p ?b)
220                 (when (and (not (ems_burns ?p))
221                             (redness ?p palm)
222                             (swollen ?p palm)
223                             (pain ?p palm)
224                         )
225                     (ems_burns ?p)
226                 )
227                 (when (and (not (ems_burns ?p))
228                             (redness ?p face)
229                             (swollen ?p face)
230                             (pain ?p face)
231                         )
232                     (ems_burns ?p)
233                 )
234                 (when (and (not (ems_burns ?p))
235                             (redness ?p back_knee)
236                             (swollen ?p back_knee)
237                             (pain ?p back_knee)
238                         )
239                     (ems_burns ?p)
240                 )
241             )
242     )
243 )
244
245
246 ;; treat person for first degree burn
247 (:action clean_and_treat_first_degree_burn
248     :parameters (?p - PERSON ?b - BODYPART)
249     :precondition (and (not (ems_burns ?p))
250                       (first_degree_burn ?p ?b)
251                       (hasBodyP ?p ?b)
252                   )
253     :effect (treat_burns ?p first_degree)
254 )

```

255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308

```
;; check person's bodypart for second degree burn
(:action check_second_degree_burn
  :parameters(?p - PERSON ?b - BODYPART)
  :precondition (and (not (dead ?p))
                     (redness ?p ?b)
                     (inflammation ?p ?b)
                     (pain ?p ?b)
                     (blistering ?p ?b)
                     (hasBodyP ?p ?b)
                   )
  :effect (ems_burns ?p)
)

;; check person's bodypart for third degree burn
(:action check_third_degree_burn
  :parameters(?p - PERSON ?b - BODYPART)
  :precondition (and (not (dead ?p))
                     (redness ?p ?b)
                     (inflammation ?p ?b)
                     (not (pain ?p ?b))
                     (blistering ?p ?b)
                     (hasBodyP ?p ?b)
                   )
  :effect (ems_burns ?p)
)

;; apply bandage if wound is not infected
;; or is not a puncture and the person doesn't have immunization to tetanus
(:action bandage
  :parameters(?p - PERSON ?w - WOUND)
  :precondition (and (hasWound ?p ?w)
                     (dress_wound ?p ?w)
                     (not (dead ?p))
                   )
  :effect(and (when (infected ?p ?w) (ems_wounds ?p))
              (when (and (hasWound ?p puncture) (not (tetanus ?p))) (ems_wounds ?p))
              (when (and (not (infected ?p ?w)) (or (not (hasWound ?p puncture)) (and (has
)
)

;; apply pressure on bleeding wound
(:action apply_pressure
  :parameters (?p - PERSON ?w - WOUND)
  :precondition (pressure ?p ?w)
  :effect (dress_wound ?p ?w)
)
```

```

309
310 ;; apply antiseptic solution on puncture wound
311 (:action apply_antiseptic_solution
312       :parameters (?p - PERSON ?w - WOUND)
313       :precondition (antiseptic ?p ?w)
314       :effect (dress_wound ?p ?w)
315 )
316
317
318 ;; check and clean person's wound
319 (:action check_and_clean_wound
320       :parameters (?p -PERSON ?w - WOUND)
321       :precondition (and (hasWound ?p ?w)
322                          (not (dead ?p))
323                        )
324       :effect (and (when (bleeding ?p ?w) (pressure ?p ?w))
325                  (when (hasWound ?p puncture) (antiseptic ?p ?w))
326                  (when (not (bleeding ?p ?w)) (dress_wound ?p ?w))
327                )
328 )
329
330
331 (:action call_ems_for_airway_obstruction
332       :parameters (?p - PERSON)
333       :precondition (ems_airwayObs ?p)
334       :effect (and (free_airway ?p)
335                  (not (dead ?p))
336                )
337 )
338
339 (:action call_ems_for_burns
340       :parameters (?p - PERSON)
341       :precondition (ems_burns ?p)
342       :effect (and (treat_burns ?p first_degree)
343                  (treat_burns ?p second_degree)
344                  (treat_burns ?p third_degree)
345                  (not (dead ?p))
346                )
347 )
348
349
350 (:action call_ems_for_wounds
351       :parameters (?p - PERSON)
352       :precondition (ems_wounds ?p)
353       :effect (and (treat_wounds ?p)
354                  (not (dead ?p))
355                )
356 )
357 )

```

Explanation:

- For this project I chose to represent using planning domains and PDDL, a First Aid Protocol that includes guidelines for providing first aid in case of foreign body airway obstruction, burns as well as wounds.

- Line 3 describes the types of objects that I have used in order to reach a clear formalisation of the medical world. These objects include: a person, who is the victim that needs to be provided first aid care, his or her bodyparts, that have been injured by burning, the degree of the burning and the type of wound that the person has.
- Next, in lines 5 to 38 I went on and defined the predicates that I needed to use for defining the actions which the planner can use in searching for a suitable plan to reach the goals written in each problem file. The predicates use the types enumerated before and they are referring to certain attributes that they have, for example (**conscious ?p - PERSON**) which means that the person receiving first aid care is or is not conscious. Other predicates exemplify different actions that can be done : (**ems_wounds ?p - PERSON**) which means that Emergency Medical Services need to be called immediately for a wounded person.
- After initialising the types of objects and the predicates that are going to be used, I continued with implementing some actions that the objects defined in the problem can do.
- Because I found that the first aid domain is rather an uncertain one, I chose to describe some actions using Contingent Planning approach, which means that I included some observation actions like the one in lines 63 to 66. Action **sense_breathing** has the :observe field, which means that the first aid provider has to check the the victims breathing before taking any further steps.
- Other actions are fully described, with the exact effects that are going to take place before executing the certain action. A perfect example is action **check_second_degree_burn** which says that if the victim is not dead, presents redness, inflammation, pain and blistering on a specific body part, the Emergency Medical Services should be called for the person's second degree burn. All of the other actions are implemented in a similar way.

1.1.2 Personal observations and notes

Because I consider the medical field to be pretty uncertain when it comes to an environment like first aid assisting, I chose to represent the domain and the problems using nondeterministic and partially observable approaches, that the tool Contingent-FF and the PDDL language have to offer.

1.2 References

- Information regarding first aid guidelines for foreign body airway obstruction :https://www.ifrc.org/Global/Publications/Aid-2016-Guidelines_EN.pdf
- Information regarding first aid guidelines for burns : <https://www.medicinenet.com/burns/article.htm>
- Information regarding first aid guidelines for wounds : <https://www.skinsight.com/skin-conditions/first-aid/first-aid-wounds>

2 Problem 1

2.1 Foreign Body Airway Obstruction

In this section the solution for the following problem will be presented:

You should also create several problems for your domain, and verify that at least one planning engine can solve them. Then you will conduct empirical investigation of the performance of different planning engines and heuristics, using the domain you created in the first part. To do this, **you will create a set of problems of increasing size and complexity**, and you will measure runtime. You will identify some parameters that you can use to increase problem's complexity (like number of objects in the init state or number of predicates in the goal. Next, you will create problems for your domain by scaling up your parameters. Additionally, feel free to introduce some incomplete knowledge in the init state or nondeterministic effects.

2.1.1 Code implementation

Code:

```
1 (define (problem ForeignBodyAirwayObstruction)
2   (:domain FirstAid)
3   (:objects Robert Emily William - PERSON
4           first_degree second_degree third_degree - DEGREE
5           leg palm face back_knee - BODYPART
6           puncture - WOUND
7   )
8
9   (:init
10
11     (infant William)
12     (conscious William)
13     (breathing William)
14     (see_foreign_body William)
15
16     (adult Emily)
17     (conscious Emily)
18     (pregnant Emily)
19     (breathing Emily)
20     (complete_obstruction Emily)
21
22     (adult Robert)
23     (unknown (conscious Robert))
24     (unknown (complete_obstruction Robert))
25     (obese Robert)
26     (unknown (encircle_abdomen Robert))
27     (unknown (breathing Robert))
28     (unknown (see_foreign_body Robert))
29
30 )
31
32   (:goal (and (not (dead Robert))
33             (free_airway Robert)
34
35             (not (dead Emily))
36             (free_airway Emily)
37
```

```

38
39         (not (dead William))
40         (free_airway William)
41     )
42
43 )
44
45 )

```

Explanation:

- To illustrate how the planner can use the before described domain, I have written the first problem to test it. Each problem is referring to certain guidelines for providing first aid. This one is checking the foreign body airway obstruction guidelines. The objects are : three people, the three types of degrees that the burns can have, the body parts used to emphasize the scenarios, and a type of wound.
- Lines 11 to 14 present a simple example of providing first aid to a victim. The victim, William is an infant, conscious, is breathing, has complete airway obstruction and the foreign body causing the obstruction can be seen.
- The second victim is shown in lines 16 to 20. Emily is an adult, conscious, pregnant, is breathing and has complete airway obstruction.
- The third person is called Robert (lines 22 to 28), is an obese adult , but this time the planner has to come up with a strategy to figure out the status of the next facts: if Robert is conscious, if his airway is completely obstructed, if his abdomen can be encircled, if he is breathing or if the foreign body that it's obstructing his airway can be seen.
- The goals of this problem (lines 33 to 40) are, of course to keep all the victims alive and free their airways.

Commands:

- Contingent-FF -o domain.pddl -f p1.pddl

3 Problem 2

3.1 Skin Burns

In this section the solution for the following problem will be presented:

You should also create several problems for your domain, and verify that at least one planning engine can solve them. Then you will conduct empirical investigation of the performance of different planning engines and heuristics, using the domain you created in the first part. To do this, **you will create a set of problems of increasing size and complexity**, and you will measure runtime. You will identify some parameters that you can use to increase problem's complexity (like number of objects in the init state or number of predicates in the goal. Next, you will create problems for your domain by scaling up your parameters. Additionally, feel free to introduce some incomplete knowledge in the init state or nondeterministic effects.

3.1.1 Code implementation

Code:

```
1 (define (problem Burns)
2   (:domain FirstAid)
3   (:objects Oliver Jessica Joe Anna - PERSON
4           first_degree second_degree third_degree - DEGREE
5           back arm leg palm face back_knee - BODYPART
6           puncture - WOUND
7   )
8
9   (:init
10    (hasBodyP Jessica arm)
11    (swollen Jessica arm)
12    (redness Jessica arm)
13    (pain Jessica arm)
14
15
16    (hasBodyP Joe face)
17    (swollen Joe face)
18    (redness Joe face)
19    (pain Joe face)
20
21    (hasBodyP Oliver leg)
22    (swollen Oliver leg)
23    (redness Oliver leg)
24    (pain Oliver leg)
25
26
27    (hasBodyP Oliver arm)
28    (redness Oliver arm)
29    (inflammation Oliver arm)
30    (pain Oliver arm)
31    (blistering Oliver arm)
32
33    (hasBodyP Anna back)
34    (redness Anna back)
35    (inflammation Anna back)
36    (unknown (pain Anna back))
37    (blistering Anna back)
```

```

38 )
39
40 (:goal (and (not (dead Oliver))
41             (treat_burns Joe first_degree)
42
43             (treat_burns Jessica first_degree)
44
45             (treat_burns Oliver first_degree)
46             (treat_burns Oliver second_degree)
47
48             (treat_burns Anna second_degree)
49             (treat_burns Anna third_degree)
50         )
51     )
52 )
53
54 )

```

Explanation:

- This problem is showcasing the skin burns first aid guidelines. The objects are : four people, the three types of degrees that the burns can have, the body parts used to emphasize the scenarios, and a type of wound.
- The problem's initial state starts with the first victim, Jessica (lines 10 to 13) , who's arm presents swelling, redness and pain.
- The second person, Joe (lines 16 to 19) has a swollen face, redness and pain.
- Oliver (lines 21 to 24) presents a swollen leg, with redness and pain, as well as an inflammation on his arm, redness, pain and blistering.
- Anna (lines 33 to 37) has inflammation on her back, with redness, blistering, but this time, the body part's sense of pain is not known.
- The goals of this problem (lines 40 to 50) are to keep all the victims alive and treat their burns.

Commands:

- Contingent-FF -o domain.pddl -f p2.pddl

3.1.2 Solution Explanation and Followed Plan

This sub-section is dedicated to detailing the logical solution that the planner has found in order to reach the desired goal.

- For Oliver, in node 0|0, the type of burning is checked. Because is a second degree burn, in node 4|0, the EMS are called. In this case, 6|0 is the node that checks the other burn that Oliver has, but because the EMS have already been called, the burn won't be treated.
- Concerning Jessica, in node 2|0 her first degree burn is checked, according to her symptoms, and in node 5|0 is treated.
- On Joe's side, in node 1|0 , the symptoms that he presents correspond to a first degree burn. The difference between his and Jessica's case is that Joe's burn is on his face, which is considered to be a slow healing body part, so EMS are called, in node 3|0.
- When it comes to Anna, facts get less deterministic. In node 7|0 we determine if her nerves are intact and she feels pain in that area. If the effect is positive, it means she has a second degree burn (node 8|0), but if the effect is negative, she has a third degree burn (node 8|1). In either of the cases, EMS should be called (nodes 9|0 and 9|1), resulting in 10 layers and a total of 12 actions taken.

```

ff: found plan as follows
-----
0||0 --- CHECK_SECOND_DEGREE_BURN OLIVER ARM --- SON: 1||0
-----
1||0 --- CHECK_FIRST_DEGREE_BURN JOE FACE --- SON: 2||0
-----
2||0 --- CHECK_FIRST_DEGREE_BURN JESSICA ARM --- SON: 3||0
-----
3||0 --- CALL_EMS_FOR_BURNS JOE --- SON: 4||0
-----
4||0 --- CALL_EMS_FOR_BURNS OLIVER --- SON: 5||0
-----
5||0 --- CLEAN_AND_TREAT_FIRST_DEGREE_BURN JESSICA ARM --- SON: 6||0
-----
6||0 --- CHECK_FIRST_DEGREE_BURN OLIVER LEG --- SON: 7||0
-----
7||0 --- SENSE_PAIN ANNA BACK --- TRUESON: 8||0 --- FALSESON: 8||1
-----
8||0 --- CHECK_SECOND_DEGREE_BURN ANNA BACK --- SON: 9||0
8||1 --- CHECK_THIRD_DEGREE_BURN ANNA BACK --- SON: 9||1
-----
9||0 --- CALL_EMS_FOR_BURNS ANNA --- SON: 10||-1
9||1 --- CALL_EMS_FOR_BURNS ANNA --- SON: 10||-1
-----

tree layers: 10
total nr. actions: 12

```

Figure 1: The generated plan

4 Problem 3

4.1 Wounds

In this section the solution for the following problem will be presented:

You should also create several problems for your domain, and verify that at least one planning engine can solve them. Then you will conduct empirical investigation of the performance of different planning engines and heuristics, using the domain you created in the first part. To do this, **you will create a set of problems of increasing size and complexity**, and you will measure runtime. You will identify some parameters that you can use to increase problem's complexity (like number of objects in the init state or number of predicates in the goal. Next, you will create problems for your domain by scaling up your parameters. Additionally, feel free to introduce some incomplete knowledge in the init state or nondeterministic effects.

4.1.1 Code implementation

Code:

```
1 (define (problem Wounds)
2   (:domain FirstAid)
3   (:objects Lily Jake Mia - PERSON
4           first_degree second_degree third_degree - DEGREE
5           back arm leg palm face back_knee - BODYPART
6           puncture cut - WOUND
7   )
8
9   (:init
10
11     (hasWound Lily cut)
12     (tetanus Lily)
13     (infected Lily cut)
14     (unknown (bleeding Lily cut))
15
16     (hasWound Mia cut)
17     (tetanus Mia)
18     (unknown (infected Mia cut))
19     (unknown (bleeding Mia cut))
20
21     (hasWound Jake puncture)
22     (unknown (bleeding Jake puncture))
23     (unknown (tetanus Jake))
24
25   )
26
27   (:goal (and (not (dead Lily))
28             (treat_wounds Lily)
29
30             (not (dead Jake))
31             (treat_wounds Jake)
32
33             (not (dead Mia))
34             (treat_wounds Mia)
35           )
36
37   )
```

Explanation:

- This problem is showcasing the wounds first aid guidelines. The objects are : three people, the three types of degrees that the burns can have, the body parts used to emphasize the scenarios, and a types of wounds they have.
- The first person, Lily (lines 11 to 14) has an infected cut, has got the tetanus immunisation, but we don't know if she is bleeding.
- Mia (lines 16 to 19), has a cut, has got the tetanus immunisation, but the fact that her wound is infected or bleeding is not known.
- The goals of this problem (lines 27 to 35) are to keep all the victims alive and treat their burns.

Commands:

- `Contingent-FF -o domain.pddl -f p3.pddl`

4.2 Further Development

- This project could be improved by adding guidelines for other different types of first aid assisting.
- At the same time, the domain and problems can be extended with more predicates and more actions, to encompass situations in which a person suffers from multiple injuries at the same time.
- To test the validity of the domain, different configurations of the predicates can be illustrated in the problem files, or tested with a validation tool.