

Artificial intelligence - Project 3
- Planning -

Birlutiu Claudiu-Andrei

9/1/2022

1 Introducere

1.1 Context

În această lucrare este descrisă implementare logicii de mișcare, în limbajul de planificare Planning Domain Definition Language (PDDL) a unui robot cu două brațe care se află într-un laborator de chimie și face diferite experimente cu substanțele pe care le are la dispoziție în laborator. Alegerea acestei teme de abordare este o pasiune veche pentru chimie.

1.2 Descrierea acțiunilor pe care le face robotul chimist

Rolul robotului chimist este de a realiza diferite reacții chimice având la dispoziție mai multe substanțe (săruri, acizi și baze) precum și ustensile. Robotul are două brațe identificate prin brațul stâng și brațul drept. Pentru a realiza diferitele experimente chimice, el trebuie să măsoare cantitățile de substanțe reactante cu ajutorul paharelor berzelius și erlenmeyer, iar după măsurare va introduce reactantul într-o eprubetă. Reacția e finalizată în momentul în care în eprubetă se află două substanțe reactante. La final se va verifica rezultatul reacției **reactant1 + reactant2 = result** astfel:

- acid + bază = sare
- baza + sare = bază insolubilă
- acid + sare = precipitat

Pentru realizarea măsurării, paharul berzelius sau erlenmeyer trebuie să fie curat înainte de a adăuga substanța pentru nu a apărea reacții nedorite cu cantitatea de substanță rămasă de la măsurătoarea anterioară. De aceea, roboțelul trebuie să spele recipientul dacă acesta nu e curat. De asemenea în timpul măsurării, robotul ține cu o mână paharul iar cu cealaltă va adăuga substanța de măsurat din recipientul ei (aici s-a considerat că trebuie doar să aibă acea mână liberă).

Eprubeta trebuie să fie goală și curată înainte de a se adăuga primul reactant. După adăugarea primului reactant se va aștepta adăugarea celui de-al doilea pentru realizarea reacției. În momentul în care în eprubetă sunt adăugate 2 reactanți aceasta se va considera plină și nu se vor mai adăuga alte substanțe.

Recipientele și eprubeta pot fi puse pe stativ, iar de acolo roboțelul chimist le poate lua cu brațele lui. Pentru măsurare, el trebuie să aibă în braț un recipient, iar celălalt să fie liber (pentru adăugarea substanței în recipient). Pentru curățarea recipientelor și a eprubetei ele trebuie ținute de un braț al robotului. În timpul acțiunii de adăugare a unui reactant din paharul de măsurat în eprubetă, într-unul din brațe robotul ține recipientul, iar în celălalt eprubeta.

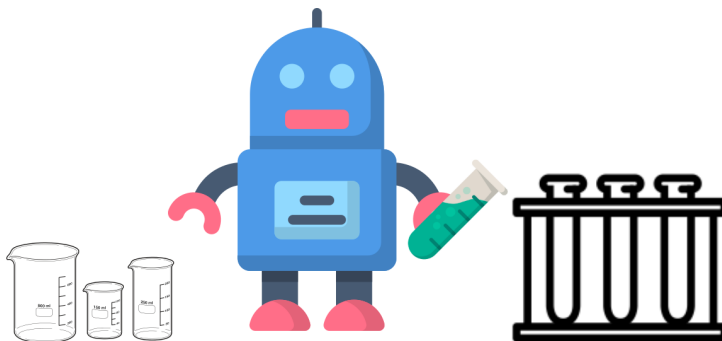


Figure 1: Robotul chimist

2 PDDL Clasic

2.1 Definirea domeniului

Code:

```
1 ;; planificarea unui robot de a face reactii chimice
2
3 (define (domain robchemistry)
4   (:requirements :strips :typing :negative-preconditions :adl :action-costs)
5
6   (:types
7     recipient reactant brat eprubeta          - object
8     left right                                - brat
9     berzelius erlenmeyer                       - recipient
10    acid baza sare                             - reactant
11
12  )
13  (:predicates
14    (free-arm ?a - brat )                      ;;bratul e liber
15    (empty-used ?r - recipient)                 ;;recipientul e gol si nu a fost spalat
16    (empty-clean ?r - recipient)                ;;recipientul e gol si curat
17    (used ?r - recipient)                       ;;recipientul e folosit si are ceva in el
18    (hold ?a - brat ?r - recipient)             ;;in bratul a se afla recipientul r
19    (on-stativ ?r - recipient)                  ;;recipientul se afla pe stativ
20    (contains-s-rec ?s - reactant ?r - recipient) ;;un recipient contine substanta
21    (contains-s-ep ?s - reactant ?e - eprubeta)  ;;eprubeta contine reactantul
22    (empty-e ?e - eprubeta)                     ;;eprubeta goala
23    (hold-e ?a - brat ?e - eprubeta)            ;;eprubeta e tinuta in brat
24    (on-stativ-e ?e - eprubeta)                 ;;eprubeta e pe stativ
25    (full-ep ?e - eprubeta)                     ;;eprubeta e plina
26    (result-sare ?e - eprubeta)                 ;;reactie ACID + BAZA
27    (result-baza-insolubila ?e - eprubeta)      ;;reactie BAZA INSOLUBILA + SARE
28    (result-precipitat ?e - eprubeta)           ;;reactie SARE +ACID
29
30  )
31
32  (:functions
33    (total-cost) - number
34  )
35
36  ;;ridicarea unui recipient in mana
37  (:action gripp-recipient
38    :parameters (?a - brat ?r - recipient)
39    :precondition (and (free-arm ?a)           ;;sa fie liber bratul
40                      (on-stativ ?r)          ;;recipientul sa fie pe stativ
41                      )
42    :effect (and (not (free-arm ?a))          ;;bratul nu va mai fi liber
43                (hold ?a ?r)                 ;;bratul a tine recipientul r
44                (not (on-stativ ?r))         ;;recipientul nu mai e pe stativ
45                (increase (total-cost) 1)
46              )
47  )
48
49
```

```

50 ;;lasarea unui recipient pe masa
51 (:action drop-recipient
52   :parameters (?a - brat ?r - recipient)
53   :precondition (and (hold ?a ?r) ;;implica faptul ca e mana ocupata si ca recipientul nu e pe s
54   )
55   :effect (and (not (hold ?a ?r))
56               (free-arm ?a)
57               (on-stativ ?r)
58               (increase (total-cost) 1)
59   )
60 )
61
62 ;;masurarea unei cantitati de substanta; (introducerea ei intr-un pahar Berzelius sau Erlenmeyer)
63 (:action measure-reactant ;;masurarea reactantului
64   :parameters (?r - recipient ?s - reactant ?a1 ?a2 - brat)
65   :precondition (and (hold ?a1 ?r) ;;in mana sa avem recipientul de masurare
66                     (free-arm ?a2) ;;sa fie o mana libera pentru adaugarea reactantului
67                     (empty-clean ?r) ;;sa fie curat si gol recipientul (sa nu apara reactii n
68   )
69   :effect (and (contains-s-rec ?s ?r) ;;substanta s se afla in recipientul r
70               (used ?r)
71               (not (empty-clean ?r)) ;;recipientul va contine ceva
72               (increase (total-cost) 2)
73   )
74 )
75
76 ;;adaugarea primului reactant in eprubeta
77 (:action first-reactant
78   :parameters (?r - recipient ?s - reactant ?a1 - brat ?a2 - brat ?e - eprubeta)
79   :precondition (and (contains-s-rec ?s ?r) ;;presupune ca e plin paharul de masurat
80                     (empty-e ?e) ;;eprubeta trebuie sa fie goala pentru primul re
81                     (hold ?a1 ?r) ;;recipientul de masurat e in bratul robotului
82                     (not (full-ep ?e)) ;; nu e plina eprubeta
83                     (hold-e ?a2 ?e)
84                     (used ?r) ;;recipientul e plin cu substanta respectiva; ad
85   )
86   :effect (and (contains-s-ep ?s ?e) ;;eprubeta contine reactantul s
87               (not (empty-e ?e)) ;;eprubeta nu mai e goala
88               (not (contains-s-rec ?s ?r)) ;;nu mai contine substanta recipientul
89               (empty-used ?r) ;;recipientul e empty-used (trebuie spalat daca s
90               (not (used ?r))
91               (increase (total-cost) 1)
92   )
93 )
94
95 ;;adaugarea celui de-al doilea reactant in eprubeta si ultimul
96 (:action second-reactant
97   :parameters (?r - recipient ?s - reactant ?a1 ?a2 - brat ?e - eprubeta)
98   :precondition (and (contains-s-rec ?s ?r) ;;e plin paharul de masurat
99                     (hold ?a1 ?r) ;;recipientul de masurat e in bratul robotului
100                     (exists (?s1 - reactant) (and (not (= ?s1 ?s)) (contains-s-ep ?s1 ?e) ));n
101                     (not (full-ep ?e)) ;; nu e plina eprubeta
102                     (hold-e ?a2 ?e) ;;eprubeta e tinuta in brat pentru a adauga subst
103                     (not (empty-e ?e)) ;;eprubeta contine primul reactant

```

```

104         (used ?r)                ;;recipientul e plin cu substanta respectiva; adia
105     )
106
107     :effect (and (contains-s-ep ?s ?e )           ;;eprubeta contine reactantul s
108                 (full-ep ?e)                     ;;eprubeta e plina
109                 (not (contains-s-rec ?s ?r))      ;;nu mai contine substanta recipientul
110                 (empty-used ?r)                  ;;recipientul e empty-used (trebuie spalata da
111                 (increase (total-cost) 1)
112             )
113 )
114
115 ;;curatarea recipientului
116 (:action clean-recipient
117   :parameters (?r - recipient ?a - brat)
118   :precondition (and ;;(empty-used ?r)
119                 (hold ?a ?r)                ;;trebuie sa fie in mana recipientul pt a fi spalat
120             )
121   :effect (and (empty-clean ?r)
122               (not (empty-used ?r))
123               (not (used ?r))
124               (forall (?s - reactant) (when (and (contains-s-rec ?s ?r)) (not (contains-s-rec ?s ?r)) )
125               (increase (total-cost) 3)    ;;e mai costisitor sa se spele paharul de masurat
126           )
127 )
128
129 ;;clean eprubeta
130 (:action clean-eprubeta
131   :parameters (?e - eprubeta ?a - brat)
132   :precondition (and (not (empty-e ?e))        ;;sa nu o spele degeaba
133                   (hold-e ?a ?e)              ;;sa aiba eprubeta in brat
134             )
135   :effect (and (empty-e ?e)
136               (not (full-ep ?e))
137               (forall (?s - reactant) (when (and (contains-s-ep ?s ?e)) (not (contains-s-ep ?s ?e)) )
138               (not (result-sare ?e));
139               (not (result-baza-islolubila ?e)) ;;golirea eprubetei
140               (not (result-precipitat ?e ))
141               (increase (total-cost) 2)
142           )
143 )
144 ;;ia eprubeta in brat
145 (:action gripp-eprubeta
146   :parameters (?e - eprubeta ?a - brat)
147   :precondition (and (on-stativ-e ?e)
148                   (free-arm ?a)
149             )
150   :effect (and (not (on-stativ-e ?e))
151               (hold-e ?a ?e)                ;;e in mana
152               (not (free-arm ?a))
153               (increase (total-cost) 1)
154           )
155 )
156
157 ;;lasa eprubeta pe stativ

```

```

158 (:action drop-eprubeta
159     :parameters (?e - eprubeta ?a - brat)
160     :precondition (and (hold-e ?a ?e)
161         )
162     :effect (and (on-stativ-e ?e)
163         (not (hold-e ?a ?e))      ;;e pe stativ
164         (free-arm ?a)
165         (increase (total-cost) 1)
166     )
167 )
168
169 ;;verificare rezultat reactie
170 (:action check-reactie-baza-acid
171     :parameters (?e - eprubeta)
172     :precondition (and (full-ep ?e)
173         (exists (?s - baza) (and (contains-s-ep ?s ?e) ))
174         (exists (?s - acid) (and (contains-s-ep ?s ?e) ))
175     )
176     :effect (and (result-sare ?e) (increase (total-cost) 1))
177 )
178
179
180 ;;verificare rezultat reactie
181 (:action check-reactie-baza-sare
182     :parameters (?e - eprubeta)
183     :precondition (and (full-ep ?e)
184         (exists (?s - baza) (and (contains-s-ep ?s ?e) ))
185         (exists (?s - sare) (and (contains-s-ep ?s ?e) ))
186     )
187     :effect (and (result-baza-issolubila ?e) (increase (total-cost) 1))
188 )
189
190
191 ;;verificare rezultat reactie
192 (:action check-reactie-acid-sare
193     :parameters (?e - eprubeta)
194     :precondition (and (full-ep ?e)
195         (exists (?s - sare) (and (contains-s-ep ?s ?e) ))
196         (exists (?s - acid) (and (contains-s-ep ?s ?e) ))
197     )
198     :effect (and (result-precipitat ?e) (increase (total-cost) 1))
199 )
200
201
202 )

```

Explicații:

- codul de mai sus descrie domeniul cu tipurile și acțiunile pe care le poate executa robotul
- în secțiunea de **requirements** au fost setate pe lângă **strips** următoarele
 - typing : pentru ideea de moștenire; astfel că prin reactant ne vom referi fie la o sare, acid sau fi la o bază
 - negative-preconditions : pentru folosirea negatiei în precondiții

- adl : s-au folosit **forall** și **exists**; cu ajutorul acestora s-a verificat existența substanțelor în eprubete sau recipiente, astfel că în momentul în care se va
- action-costs : acțiunile au costuri diferite
- * cele mai costisitoare acțiuni vor fi cele de spălare a recipientelor de măsurat (3), iar apoi spălarea eprubetelor; Efectul costului se va observa în momentul în care robotul decide pentru măsurarea celui de-al doilea reactant să ia un al recipient de măsurat în loc să îl spele pe cel folosit pentru măsurarea primului reactant
- pentru definirea tipurilor s-a ținut seama de ideea de moștenire astfel
 - reactantul poate fi acid, baza sau sare
 - bratul poate fi stang sau drept
 - recipientele pot fi berzelius si erlenmeyer
- pentru definirea predicatelor s-a ținut seama de toate posibilitățile în care se poate afla un recipient și eprubeta, dacă o substanță e într-un recipient sau eprubetă și de asemenea este descrisă și situația în care se pot afla brațele robotului (dacă acesta ține ceva în brațe sau nu)
- la linia 32 este definită funcția de cost ca număr; în cadrul definirii problemelor acest cost va trebui să fie inițializat
- acțiunile gripp-recipient și drop-recipient se referă la ridicarea și lăsarea unui pahar de măsurat de către robot și au costul de 1
- acțiunea measure-reactant descrie condițiile ce trebuie îndeplinite pentru măsurarea cantității de reactant și efectele măsurării. Aceasta va aduce un cost de 2.
- first-reactant și second-reactant descriu introducerea reactanților în eprubeta; Eprubeta va fi plină după adăugarea celui de-al doilea reactant.
- clean-recipient si clean-eprubeta sunt acțiunile de curățare a recipientului și respectiv a eprubetei. Se observă folosirea sintaxei **forall** pentru a elimina și substanțele existente în recipient, respectiv eprubetă. Un recipient de măsurat trebuie să fie gol și curat pentru măsurarea substanțelor. Se ține cont de faptul că dacă paharul e gol, dar înainte a conținut o substanță.
- grip-eprubeta și drop-eprubeta vor descrie acțiunile de ridicare în braț a eprubetei de pe stativ și respectiv punerea acesteia înapoi pe stativ
- check-reactie-baza-acid, check-reactie-baza-sare și check-reactie-acid-sare sunt acțiunile care verifică rezultatul reacției din eprubetă. Se remarcă folosirea lui **exists** pentru verificarea existenței substanțelor reactante.

2.1.1 Definirea problemei 1

În continuare e descrisă o problemă pentru robotul nostru. El trebuie să obțină în urma combinării reactanților puși la dispoziție (o bază, o sare și un acid) un precipitat. El are două recipiente curate pe stativ(masa): un pahar berzelius și unul erlenmeyer. Eprubeta în care se vrea să se facă reacția nu e curată și se află pe stativ.

Code:

```

1 (define (problem problemrob)
2   (:domain robchemy)
3   (:objects
4     ep1 - eprubeta
5     acid1 - acid
6     baza1 - baza
```

```

7      sare1 - sare
8      berzelius1 - berzelius
9      erlenmeyer1 - erlenmeyer
10     left1 - left
11     right1 - right
12     ep2 - eprubeta
13 )
14
15 (:init
16     ;paharele de masurat reactant
17     (on-stativ berzelius1)
18     (empty-clean berzelius1)
19     (on-stativ erlenmeyer1)
20     (empty-clean erlenmeyer1)
21
22     ;;eprubete
23     (on-stativ-e ep1)
24     (not (empty-e ep1))
25     (not (full-ep ep1))
26     (on-stativ-e ep2)
27     (empty-e ep2)
28     (not (full-ep ep2))
29
30     ;;bratele sunt libere
31     (free-arm left1)
32     (free-arm right1)
33     ;;costul total
34     (= (total-cost) 0)
35 )
36
37 (:goal (and
38
39     (full-ep ep1)
40     (result-precipitat ep1)
41
42     ;; (full-ep ep2)
43     ;; (result-sare ep2)
44 )
45
46 )
47 (:metric minimize ( total-cost) )
48
49 )

```

Explicații:

- sunt specificate condițiile atomice în secțiunea de init.
- la linia 34 se setează costul inițial la 0 și se dorește minimizarea acestuia. Cea mai concludentă acțiune în observarea acestei minimizări este decizia de a folosi pentru măsurarea substanțelor ambele pahare puse la dispoziție, în loc să folosească același pahar pe care să îl spele între măsurători (costul e mai ridicat pentru spălarea recipientelor).
- pentru rezolvarea problemei, robotul trebuie să combine acid1 cu baza1 pentru a obține un precipitat

S-a folosit fast-downward pentru a afla planul de execuție a acțiunilor în îndeplinirea scopului. Pentru aceasta s-au folosit mai multe euristici de căutare și s-au comparat rezultatele

Comenzi:

Cu euristica admisibila blind pentru astar:

- fast-downward.py robchemy.pddl problemrob.pddl -heuristic "h=blind" -search "astar(h)"
- (gripp-recipient left1 berzelius1)
(measure-reactant berzelius1 acid1 left1 right1)
(gripp-eprubeta ep1 right1)
(clean-eprubeta ep1 right1)
(first-reactant berzelius1 acid1 left1 right1 ep1)
(drop-recipient left1 berzelius1)
(gripp-recipient left1 erlenmeyer1)
(drop-eprubeta ep1 right1)
(measure-reactant erlenmeyer1 sare1 left1 right1)
(gripp-eprubeta ep1 right1)
(second-reactant erlenmeyer1 sare1 left1 right1 ep1)
(check-reactie-acid-sare ep1)
;cost = 15 (general cost)
- **S-au generat 14525 de stari; 4284 de stari au fost evaluate si au fost expandate 3228; costul e de 15 și este si cel minim deoarece cu euristica blind se obține un algoritm de căutare asemănător cu ucs**

Cu euristica hmax pentru astar:

- fast-downward.py robchemy.pddl problemrob.pddl -heuristic "h=hmax" -search "astar(h)"
- gripp-recipient left1 berzelius1 (1)
measure-reactant berzelius1 acid1 left1 right1 (2)
gripp-eprubeta ep1 right1 (1)
clean-eprubeta ep1 right1 (2)
first-reactant berzelius1 acid1 left1 right1 ep1 (1)
drop-recipient left1 berzelius1 (1)
gripp-recipient left1 erlenmeyer1 (1)
drop-eprubeta ep1 right1 (1)
measure-reactant erlenmeyer1 sare1 left1 right1 (2)
gripp-eprubeta ep1 right1 (1)
second-reactant erlenmeyer1 sare1 left1 right1 ep1 (1)
check-reactie-acid-sare ep1 (1)
;cost = 15 (general cost)
- **S-au generat 2587 de stari; 995 de stari au fost evaluate si au fost expandate 548; Se observă că aplicarea acestei euristici aduce un număr mai mic de stări generate față de cazul anterior. De asemenea și numărul stărilor evaluate și expandate a scăzut considerabil. Costul e cel minim. Această euristică se bazează pe maximul dintre subcosturi. Lungimea planului a rămas tot de 12.**

Cu inconsistent and not admissible heuristic ff pentru astar:

- fast-downward.py robchemy.pddl problemrob.pddl -heuristic "h=ff" -search "astar(h)"
- gripp-recipient left1 erlenmeyer1 (1)
measure-reactant erlenmeyer1 sare1 left1 right1 (2)
drop-recipient left1 erlenmeyer1 (1)
gripp-recipient left1 berzelius1 (1)
measure-reactant berzelius1 acid1 left1 right1 (2)
gripp-eprubeta ep1 right1 (1)
clean-eprubeta ep1 right1 (2)

first-reactant berzelius1 acid1 left1 right1 ep1 (1)
 drop-recipient left1 berzelius1 (1)
 gripp-recipient left1 erlenmeyer1 (1)
 second-reactant erlenmeyer1 sare1 left1 right1 ep1 (1)
 check-reactie-acid-sare ep1 (1)

- S-au generat 134 de stari, un numar mult mai mic decât cifrele obținute la evaluările anterioare; acesta se datorează faptului că problema s-a mai relaxat fiind aplicată o euristică care nu e admisibilă; 87 de stari au fost evaluate si au fost expandate 26; În ceea ce priveste costul si numarul de pasi nu au apărut modificări. costul e de 15; 12 pasi

Cu inconsistent and not admissible heuristic cg pentru astar:

- fast-downward.py robchem.pddl problemrob.pddl -heuristic "h=cg" -search "astar(h)"
- gripp-recipient left1 berzelius1 (1)
 measure-reactant berzelius1 acid1 left1 right1 (2)
 gripp-eprubeta ep1 right1 (1)
 drop-recipient left1 berzelius1 (1)
 gripp-recipient left1 erlenmeyer1 (1)
 drop-eprubeta ep1 right1 (1)
 measure-reactant erlenmeyer1 sare1 left1 right1 (2)
 gripp-eprubeta ep1 right1 (1)
 clean-eprubeta ep1 right1 (2)
 first-reactant erlenmeyer1 sare1 left1 right1 ep1 (1)
 drop-recipient left1 erlenmeyer1 (1)
 gripp-recipient left1 berzelius1 (1)
 second-reactant berzelius1 acid1 left1 right1 ep1 (1)
 check-reactie-acid-sare ep1 (1)

- S-au generat 317 de stari, problema de căutare s-a mai relaxat fiind aplicată o euristică care nu e admisibilă; 211 de stari au fost evaluate si au fost expandate 67. Aceste valori sunt mult mai mici față de ce am obținut pentru blind și hmax, dar observăm că nu am obținut valori mai bune față de evaluarea cu euristica "ff" ; Numărul de pași/ lungimea planului a crescut, din cauza faptului că se folosește o euristică ce nu e admisibilă, acesta a urcat la 14. Nici costul nu mai e cel minim, ci s-a ajuns o valoare de 17.

Cu inconsistent and not admissible heuristic ff pentru wighted astar (2):

- fast-downward.py robchem.pddl problemrob.pddl -heuristic "h=ff" -search "astar(weight(h,2))"
- gripp-recipient left1 erlenmeyer1 (1)
 measure-reactant erlenmeyer1 sare1 left1 right1 (2)
 drop-recipient left1 erlenmeyer1 (1)
 gripp-recipient left1 berzelius1 (1)
 measure-reactant berzelius1 acid1 left1 right1 (2)
 gripp-eprubeta ep1 right1 (1)
 clean-eprubeta ep1 right1 (2)
 first-reactant berzelius1 acid1 left1 right1 ep1 (1)
 drop-recipient left1 berzelius1 (1)
 gripp-recipient left1 erlenmeyer1 (1)
 second-reactant erlenmeyer1 sare1 left1 right1 ep1 (1)
 check-reactie-acid-sare ep1 (1)
 ; cost = 15 (general cost)
- Rezultatele obținute în urma acestei evaluări sunt foarte bune. Costul obținut e cel minim (15), iar numărul de pași este 12. Au fost generate 118 stari, un număr foarte bun. 83 de stari au fost evaluate si au fost expandate 22; Aceste rezultate sunt foarte bune având în vedere problema de căutare.

- Observăm scăderea numărului de stări expandate față de celelalte

Cu cu greedy și ff euristica:

- `fast-downward.py robchem.pddl problemrob.pddl -search "eager_greedy([ff])"`
- gripp-recipient left1 erlenmeyer1 (1)
measure-reactant erlenmeyer1 sare1 left1 right1 (2)
drop-recipient left1 erlenmeyer1 (1)
gripp-recipient left1 berzelius1 (1)
measure-reactant berzelius1 acid1 left1 right1 (2)
gripp-eprubeta ep1 right1 (1)
clean-eprubeta ep1 right1 (2)
first-reactant berzelius1 acid1 left1 right1 ep1 (1)
drop-recipient left1 berzelius1 (1)
gripp-recipient left1 erlenmeyer1 (1)
second-reactant erlenmeyer1 sare1 left1 right1 ep1 (1)
check-reactie-acid-sare ep1 (1)
; cost = 15 (general cost)
- **Au fost generate 118 stări. 83 de stări au fost evaluate și au fost expandate 22; costul e de 15; 12 pași**
- **Comparativ cu aplicarea euristicii ff pe astar, rezultatele obținute pentru greedy sunt puțin mai bune. Diferența nu e majoră, dar pentru probleme de căutare mai complexe se va simți această diferență;**

Cu cu greedy și cg euristica:

- `fast-downward.py robchem.pddl problemrob.pddl -search "eager_greedy([cg])"`
- gripp-recipient left1 berzelius1 (1)
measure-reactant berzelius1 acid1 left1 right1 (2)
gripp-eprubeta ep1 right1 (1)
drop-recipient left1 berzelius1 (1)
gripp-recipient left1 erlenmeyer1 (1)
drop-eprubeta ep1 right1 (1)
measure-reactant erlenmeyer1 sare1 left1 right1 (2)
gripp-eprubeta ep1 right1 (1)
clean-eprubeta ep1 right1 (2)
first-reactant erlenmeyer1 sare1 left1 right1 ep1 (1)
drop-recipient left1 erlenmeyer1 (1)
gripp-recipient left1 berzelius1 (1)
second-reactant berzelius1 acid1 left1 right1 ep1 (1)
check-reactie-acid-sare ep1 (1)
- **Numărul de stări generate este de 131. 94 de stări au fost evaluate și au fost expandate 28; costul e de 17; 14 pași**
- **Față de aplicarea euristicii cg pe algoritmul astar, observăm că rezultatele pentru greedy în ceea ce privește numărul de stări generate și evaluate e mai mic. În schimb, costul e mai mare și la fel numărul de pași.**

2.1.2 Definirea problemei 2

La problema 1 s-a adaugat sa se obtina si o sare in eprubeta 2

Code:

```
1 (define (problem problemrob)
2 (:domain robchemistry)
3 (:objects
4   ep1 - eprubeta
5   acid1 - acid
6   baza1 - baza
7   sare1 - sare
8   berzelius1 - berzelius
9   erlenmeyer1 - erlenmeyer
10  left1 - left
11  right1 - right
12  ep2 - eprubeta
13 )
14
15 (:init
16   ;paharele de masurat reactant
17   (on-stativ berzelius1)
18   (empty-clean berzelius1)
19   (on-stativ erlenmeyer1)
20   (empty-clean erlenmeyer1)
21
22   ;;eprubete
23   (on-stativ-e ep1)
24   (not(empty-e ep1))
25   (not (full-ep ep1))
26   (on-stativ-e ep2)
27   (empty-e ep2)
28   (not (full-ep ep2))
29
30   ;;bratele sunt libere
31   (free-arm left1)
32   (free-arm right1)
33   ;;costul total
34   (= (total-cost) 0)
35 )
36
37 (:goal (and
38
39   (full-ep ep1)
40   (result-precipitat ep1)
41
42   (full-ep ep2)
43   (result-sare ep2)
44 )
45
46 )
47 (:metric minimize ( total-cost) )
48
49 )
```

Solutia la problema 2 cu euristica ff la algoritmul de căutare astar weighted:
 fast-downward.py robchemistry.pddl problemrob2.pddl -heuristic "h=ff" -search "astar(weight(h,2))"

```
gripp-recipient left1 erlenmeyer1 (1)
measure-reactant erlenmeyer1 sare1 left1 right1 (2)
gripp-eprubeta ep1 right1 (1)
clean-eprubeta ep1 right1 (2)
first-reactant erlenmeyer1 sare1 left1 right1 ep1 (1)
clean-recipient erlenmeyer1 left1 (3)
drop-eprubeta ep1 right1 (1)
measure-reactant erlenmeyer1 acid1 left1 right1 (2)
drop-recipient left1 erlenmeyer1 (1)
gripp-recipient left1 berzelius1 (1)
measure-reactant berzelius1 baza1 left1 right1 (2)
gripp-eprubeta ep2 right1 (1)
first-reactant berzelius1 baza1 left1 right1 ep2 (1)
drop-eprubeta ep2 right1 (1)
clean-recipient berzelius1 left1 (3)
measure-reactant berzelius1 acid1 left1 right1 (2)
gripp-eprubeta ep1 right1 (1)
second-reactant berzelius1 acid1 left1 right1 ep1 (1)
drop-recipient left1 berzelius1 (1)
gripp-recipient left1 erlenmeyer1 (1)
drop-eprubeta ep1 right1 (1)
gripp-eprubeta ep2 right1 (1)
second-reactant erlenmeyer1 acid1 left1 right1 ep2 (1)
check-reactie-acid-sare ep1 (1)
check-reactie-baza-acid ep2 (1)
; cost = 34 (general cost)
```

Statistici: Lungimea planului: 25 pași Cost : 34 Stari generate: 1836 Stari expandate: 387 Stari evaluate: 894

Se observă o creștere a numărului de stări expandate cu mărirea problemei de căutare, dar rezultatele sunt tot mici față de rezultatele pe care le-am întâmpinat în cadrul unei probleme mai mici, dar cu căutarea pe o euristica slabă.

3 Conformant

Au fost necesare mici modificări pentru a trece domeniul definit în cadrul planificării clasice complet observabile, într-un mediu parțial observabil. Am fost nevoit să scot costurile și de asemenea o verificare cu exists în precondiția unei acțiuni pentru că se genera eroarea legată de faptul că nu se suportă disjuncțiile. În continuare se regăsește codul pentru domeniu.

3.1 Definirea domeniului

Code:

```
1 ;; planificarea unui robot de a face reactii chimice
2
3 (define (domain robchemistry)
4   (:requirements :strips :typing :adl)
5
6   (:types
7     recipient reactant brat eprubeta          - object
```

```

8         left right                                - brat
9         berzelius erlenmeyer                      - recipient
10        acid baza sare                             - reactant
11
12    )
13    (:predicates
14        (free-arm ?a - brat )                      ;;bratul e liber
15        (empty-used ?r - recipient)                 ;;recipientul e gol si nu a fost spalat
16        (empty-clean ?r - recipient)                ;;recipientul e gol si curat
17        (used ?r - recipient)                       ;;recipientul e folosit si are ceva in el
18        (hold ?a - brat ?r - recipient)             ;;in bratul a se afla recipientul r
19        (on-stativ ?r - recipient)                  ;;recipientul se afla pe stativ
20        (contains-s-rec ?s - reactant ?r - recipient) ;;un recipient contine substanta
21        (contains-s-ep ?s - reactant ?e - eprubeta) ;;eprubeta contine reactantul
22        (empty-e ?e - eprubeta)                     ;;eprubeta goala
23        (hold-e ?a - brat ?e - eprubeta)            ;;eprubeta e tinuta in brat
24        (on-stativ-e ?e - eprubeta)                 ;;eprubeta e pe stativ
25        (full-ep ?e - eprubeta)                     ;;eprubeta e plina
26        (result-sare ?e - eprubeta)                 ;;reactie ACID + BAZA
27        (result-baza-insolubila ?e - eprubeta)      ;;reactie BAZA INSOLUBILA + SARE
28        (result-precipitat ?e - eprubeta)           ;;reactie SARE +ACID
29
30    )
31
32
33
34    ;;ridicarea unui recipient in mana
35    (:action gripp-recipient
36        :parameters (?a - brat ?r - recipient)
37        :precondition (and (free-arm ?a)             ;;sa fie liber bratul
38                          (on-stativ ?r)            ;;recipientul sa fie pe stativ
39                          )
40        :effect (and (not (free-arm ?a))            ;;bratul nu va mai fi liber
41                    (hold ?a ?r)                   ;;bratul a tine recipientul r
42                    (not (on-stativ ?r))            ;;recipientul nu mai e pe stativ
43
44                    )
45    )
46
47
48    ;;lasarea unui recipient pe masa
49    (:action drop-recipient
50        :parameters (?a - brat ?r - recipient)
51        :precondition (and (hold ?a ?r)             ;;implica faptul ca e mana ocupata si ca recipientul nu e pe s
52                          )
53        :effect (and (not (hold ?a ?r))
54                    (free-arm ?a)
55                    (on-stativ ?r)
56
57                    )
58    )
59
60    ;;masurarea unei cantitati de substanta; (introducerea ei intr-un pahar Berzelius sau Erlenmeyer)
61    (:action measure-reactant                        ;;masurarea reactantului

```

```

62 :parameters (?r - recipient ?s - reactant ?a1 ?a2 - brat)
63 :precondition (and (hold ?a1 ?r) ;;in mana sa avem recipientul de masurare
64 (free-arm ?a2) ;;sa fie o mana libera pentru adaugarea reactantului
65 (empty-clean ?r) ;;sa fie curat si gol recipientul (sa nu apara reactii n
66 )
67 :effect (and (contains-s-rec ?s ?r) ;;substanta s se afla in recipientul r
68 (used ?r)
69 (not (empty-clean ?r)) ;;recipientul va contine ceva
70
71 )
72 )
73
74 ;;adaugarea primului reactant in eprubeta
75 (:action first-reactant
76 :parameters (?r - recipient ?s - reactant ?a1 - brat ?a2 - brat ?e - eprubeta)
77 :precondition (and ;;presupune ca e plin paharul de masurat
78 (empty-e ?e) ;;eprubeta trebuie sa fie goala pentru primul rea
79 (hold ?a1 ?r) ;;recipientul de masurat e in bratul robotului
80 (not (full-ep ?e)) ;; nu e plina eprubeta
81 (hold-e ?a2 ?e)
82 (used ?r) ;;recipientul e plin cu substanta respectiva; adica nu
83 )
84 :effect (and (when (and (contains-s-rec ?s ?r) )
85 (and (contains-s-ep ?s ?e) ;;eprubeta contine reactantul s
86 (not (empty-e ?e)) ;;eprubeta nu mai e goala
87 (not (contains-s-rec ?s ?r)) ;;nu mai contine substanta recipientul
88 (empty-used ?r) ;;recipientul e empty-used (trebuie
89 (not (used ?r))
90 )
91 )
92 )
93 )
94 )
95
96 ;;adaugarea celui de-al doilea reactant in eprubeta si ultimul
97 (:action second-reactant
98 :parameters (?r - recipient ?s - reactant ?a1 ?a2 - brat ?e - eprubeta)
99 :precondition (and
100 (hold ?a1 ?r) ;;recipientul de masurat e in bratul robotului
101 ;(exists (?s1 - reactant) (and (not (= ?s1 ?s)) (contains-s-ep ?s1 ?e) ));;
102 (not (full-ep ?e)) ;; nu e plina eprubeta
103 (hold-e ?a2 ?e) ;;eprubeta e tinuta in brat pentru a adauga subst
104 (not (empty-e ?e)) ;;eprubeta contine primul reactant
105 (used ?r)
106 )
107
108 :effect (and (when (and (contains-s-rec ?s ?r) )
109 (and (contains-s-ep ?s ?e) ;;eprubeta contine reactantul s
110 (full-ep ?e) ;;eprubeta e plina
111 (not (contains-s-rec ?s ?r)) ;;nu mai contine substanta recipientul
112 (empty-used ?r) ;;recipientul e empty-used (trebuie spa
113 )
114 )
115 )

```

```

116 )
117
118 ;;curatarea recipientului
119 (:action clean-recipient
120   :parameters (?r - recipient ?a - brat)
121   :precondition (and ;(empty-used ?r)
122                  (hold ?a ?r)      ;;trebuie sa fie in mana recipientul pt a fi spalat
123                  )
124   :effect (and (empty-clean ?r)
125               (not (empty-used ?r))
126               (not (used ?r))
127               (forall (?s - reactant) (when (and (contains-s-rec ?s ?r)) (not (contains-s-rec ?s ?r)) )
128               )
129 )
130
131 ;;clean eprubeta
132 (:action clean-eprubeta
133   :parameters (?e - eprubeta ?a - brat)
134   :precondition (and (not (empty-e ?e))      ;;sa nu o spele degeaba
135                     (hold-e ?a ?e)          ;;sa aiba eprubeta in brat
136                     )
137   :effect (and (empty-e ?e)
138               (not (full-ep ?e))
139               (forall (?s - reactant) (when (and (contains-s-ep ?s ?e)) (not (contains-s-ep ?s ?e)) )
140               (not (result-sare ?e));
141               (not (result-baza-isnolubila ?e)) ;;golirea eprubetei
142               (not (result-precipitat ?e ))
143               )
144 )
145
146 ;;ia eprubeta in brat
147 (:action gripp-eprubeta
148   :parameters (?e - eprubeta ?a - brat)
149   :precondition (and (on-stativ-e ?e)
150                     (free-arm ?a)
151                     )
152   :effect(and (not (on-stativ-e ?e))
153              (hold-e ?a ?e)      ;;e in mana
154              (not (free-arm ?a))
155              )
156 )
157
158
159 ;;lasa eprubeta pe stativ
160 (:action drop-eprubeta
161   :parameters (?e - eprubeta ?a - brat)
162   :precondition (and (hold-e ?a ?e)
163                     )
164   :effect(and (on-stativ-e ?e)
165              (not (hold-e ?a ?e))      ;;e pe stativ
166              (free-arm ?a)
167              )
168 )
169 )

```



```

170
171 ;;verificare rezultat reactie
172 (:action check-reactie-baza-acid
173   :parameters (?e - eprubeta)
174   :precondition (and (full-ep ?e)
175                       (exists (?s - baza) (and (contains-s-ep ?s ?e) ))
176                       (exists (?s - acid) (and (contains-s-ep ?s ?e) ))
177
178   )
179   :effect (and (result-sare ?e))
180 )
181
182 ;;verificare rezultat reactie
183 (:action check-reactie-baza-sare
184   :parameters (?e - eprubeta)
185   :precondition (and (full-ep ?e)
186                       (exists (?s - baza) (and (contains-s-ep ?s ?e) ))
187                       (exists (?s - sare) (and (contains-s-ep ?s ?e) ))
188
189   )
190   :effect (and (result-baza-isnolubila ?e))
191 )
192
193 ;;verificare rezultat reactie
194 (:action check-reactie-acid-sare
195   :parameters (?e - eprubeta)
196   :precondition (and (full-ep ?e)
197                       (exists (?s - sare) (and (contains-s-ep ?s ?e) ))
198                       (exists (?s - acid) (and (contains-s-ep ?s ?e) ))
199
200   )
201   :effect (and (result-precipitat ?e) )
202 )
203 )

```

Explicații:

- modificările aduse domeniului se pot observa în definirea acțiunilor *first-reactant* și *second-reactant*. S-au scos condițiile ca recipientul să conțină o substanță și s-au pus la effect cu **when** în cazul în care există substanța trimisă ca parametru în recipientul de măsurat, să se adauge în eprubetă.

3.2 Definirea problemei

În cadrul problemei s-au adăugat situații incerte/necunoscute prin UNKNOWN. **Code:**

```

1 (define (problem problemrob)
2   (:domain robchemistry)
3   (:objects
4     ep1 - eprubeta
5     acid1 - acid
6     baza1 - baza
7     sare1 - sare
8     berzelius1 - berzelius
9     erlenmeyer1 - erlenmeyer
10    left1 - left

```

```

11     right1 - right
12     ep2 - eprubeta
13 )
14
15 (:init
16     ;paharele de masurat reactant
17     (used berzelius1)
18     (on-stativ berzelius1)
19     (on-stativ erlenmeyer1)
20     (used erlenmeyer1)
21
22     ;;eprubete
23     (on-stativ-e ep1)
24     (empty-e ep1)
25     (on-stativ-e ep2)
26     (empty-e ep2)
27
28     (UNKNOWN (contains-s-rec acid1 berzelius1))
29     (UNKNOWN (contains-s-rec baza1 erlenmeyer1))
30     (UNKNOWN (contains-s-rec baza1 berzelius1))
31     (UNKNOWN (contains-s-rec acid1 erlenmeyer1))
32
33     (ONEOF (contains-s-rec acid1 berzelius1) (contains-s-rec baza1 berzelius1) )
34     (ONEOF (contains-s-rec acid1 erlenmeyer1) (contains-s-rec baza1 erlenmeyer1) )
35     (ONEOF (contains-s-rec acid1 berzelius1) (contains-s-rec acid1 erlenmeyer1) )
36     (ONEOF (contains-s-rec baza1 berzelius1) (contains-s-rec baza1 erlenmeyer1) )
37
38     ;;bratele sunt libere
39     (free-arm left1)
40     (free-arm right1)
41     ;;costul total
42
43 )
44
45 (:goal
46     (and
47         (result-sare ep1)
48     )
49 )
50 )

```

Explicații:

- starea inițială se definește ca fiind necunoscută. Stiim că cele două pahare sunt pline cu un acid1 sau baza1 și nu se știe exact ce substanță se află. Fie acid1 se află în paharul berzelius, iar baza1 în paharul erlenmeyer, fie acid1 se află în paharul erlenmeyer, iar baza1 se află în paharul berzelius.
- Rezultatul obținut prin apelul : *./Conformant-ff -o robchemistry.pddl -f problemrob.pddl*

- 0: GRIPP-RECIPIENT RIGHT1 BERZELIUS1
- 1: CLEAN-RECIPIENT BERZELIUS1 RIGHT1
- 2: MEASURE-REACTANT BERZELIUS1 ACID1 RIGHT1 LEFT1
- 3: GRIPP-EPRUBETA EP1 LEFT1
- 4: FIRST-REACTANT BERZELIUS1 ACID1 RIGHT1 LEFT1 EP1

- 5: CLEAN-RECIPIENT BERZELIUS1 RIGHT1
 - 6: DROP-EPRUBETA EP1 LEFT1
 - 7: MEASURE-REACTANT BERZELIUS1 BAZA1 RIGHT1 LEFT1
 - 8: GRIPP-EPRUBETA EP1 LEFT1
 - 9: SECOND-REACTANT BERZELIUS1 BAZA1 RIGHT1 LEFT1 EP1
 - 10: CHECK-REACTIE-BAZA-ACID EP1
- Cea mai simplă soluție pe care o va aborda roboțelul e de a spăla recipientul, deoarece nu cunoaște exact ce substanță se află în recipientele. Mă așteptam să introducă ambele substanțe din recipiente fiindcă se ajungea la rezultat, doar el a ales să spele recipientele și să adauge apoi reactanții necesari pentru obtinerea sării.
 - Cu hill-climbing nu s-a putut rezolva problema și s-a continuat cu Best-first Search
 - s-au realizat 159 de state transition bazate pe CNF
 -

4 Contingent

Au fost necesare mici modificări pentru a trece domeniul definit în cadrul planificării clasice complet observabile, într-un mediu parțial observabil. Am fost nevoit să scot costurile și de asemenea o verificare cu exists în precondiția unei acțiuni pentru că se genera eroarea legată de faptul că nu se suportă disjuncțiile. În continuare se regăsește codul pentru domeniu. S-a renunțat la when-urile din cadrul acțiunilor *first-reactant* și *second-reactant* și s-a revenit la adăugarea condiției în precondiții (față de codul pentru conformant). De asemenea s-a adăugat un observator.

4.1 Definirea domeniului

Code:

```

1 ;; planificarea unui robot de a face reactii chimice
2
3 (define (domain robchemistry)
4   (:requirements :strips :typing :adl)
5
6   (:types
7     recipient reactant brat eprubeta      - object
8     left right
9     berzelius erlenmeyer                  - brat
10    acid baza sare                         - recipient
11
12   )
13   (:predicates
14     (free-arm ?a - brat )                ;;bratul e liber
15     (empty-used ?r - recipient)           ;;recipientul e gol si nu a fost spalat
16     (empty-clean ?r - recipient)          ;;recipientul e gol si curat
17     (used ?r - recipient)                 ;;recipientul e folosit si are ceva in
18     (hold ?a - brat ?r - recipient)       ;;in bratul a se afla recipientul r
19     (on-stativ ?r - recipient)            ;;recipientul se afla pe stativ
20     (contains-s-rec ?s - reactant ?r - recipient) ;;un recipient contine substanta
21     (contains-s-ep ?s - reactant ?e - eprubeta) ;;eprubeta contine reactantul
22     (empty-e ?e - eprubeta)              ;;eprubeta goala
23     (hold-e ?a - brat ?e - eprubeta)     ;;eprubeta e tinuta in brat
24     (on-stativ-e ?e - eprubeta)          ;;eprubeta e pe stativ
25     (full-ep ?e - eprubeta)              ;;eprubeta e plina

```

```

26         (result-sare ?e - eprubeta)                ;;reactie ACID + BAZA
27         (result-baza-insolubila ?e - eprubeta)      ;;reactie BAZA INSOLUBILA + SARE
28         (result-precipitat ?e - eprubeta)           ;;reactie SARE +ACID
29
30     )
31
32
33
34     ;;ridicarea unui recipient in mana
35     (:action gripp-recipient
36         :parameters (?a - brat ?r - recipient)
37         :precondition (and (free-arm ?a)             ;;sa fie liber bratul
38                           (on-stativ ?r)             ;;recipientul sa fie pe stativ
39                           )
40         :effect (and (not (free-arm ?a))             ;;bratul nu va mai fi liber
41                     (hold ?a ?r)                     ;;bratul a tine recipientul r
42                     (not (on-stativ ?r))             ;;recipientul nu mai e pe stativ
43                     )
44     )
45
46
47     (:action senseContain
48         :parameters (?s - reactant ?rec - recipient)
49         :observe (contains-s-rec ?s ?rec)
50
51 )
52
53     ;;lasarea unui recipient pe masa
54     (:action drop-recipient
55         :parameters (?a - brat ?r - recipient)
56         :precondition (and (hold ?a ?r)             ;;implica faptul ca e mana ocupata si ca recipientul nu e pe s
57                           )
58         :effect (and (not (hold ?a ?r))
59                     (free-arm ?a)
60                     (on-stativ ?r)
61                 )
62     )
63
64     ;;masurarea unei cantitati de substanta; (introducerea ei intr-un pahar Berzelius sau Erlenmeyer)
65     (:action measure-reactant                        ;;masurarea reactantului
66         :parameters (?r - recipient ?s - reactant ?a1 ?a2 - brat)
67         :precondition (and (hold ?a1 ?r)             ;;in mana sa avem recipientul de masurare
68                           (free-arm ?a2)             ;;sa fie o mana libera pentru adaugarea reactantului
69                           (empty-clean ?r)           ;;sa fie curat si gol recipientul (sa nu apara reactii n
70                           )
71         :effect (and (contains-s-rec ?s ?r)          ;;substanta s se afla in recipientul r
72                     (used ?r)
73                     (not (empty-clean ?r))          ;;recipientul va contine ceva
74                 )
75     )
76
77
78     ;;adaugarea primului reactant in eprubeta
79     (:action first-reactant

```

```

80      :parameters (?r - recipient ?s - reactant ?a1 - brat ?a2 - brat ?e - eprubeta)
81      :precondition (and
82          (empty-e ?e)                ;;presupune ca e plin paharul de masurat
83          (hold ?a1 ?r)                ;;eprubeta trebuie sa fie goala pentru primul reactant
84          (not (full-ep ?e))           ;;recipientul de masurat e in bratul robotului
85          (hold-e ?a2 ?e)              ;; nu e plina eprubeta
86          (used ?r)
87          (contains-s-rec ?s ?r)        ;;recipientul e plin cu substanta respectiva
88      )
89      :effect (and (contains-s-ep ?s ?e) ;;eprubeta contine reactantul s
90          (not (empty-e ?e))            ;;eprubeta nu mai e goala
91          (not (contains-s-rec ?s ?r))   ;;nu mai contine substanta recipientului
92          (empty-used ?r)                ;;recipientul e empty-used (trebuie spalata daca s-a folosit)
93          (not (used ?r))
94      )
95  )
96
97  ;;adaugarea celui de-al doilea reactant in eprubeta si ultimul
98  (:action second-reactant
99      :parameters (?r - recipient ?s - reactant ?a1 ?a2 - brat ?e - eprubeta)
100      :precondition (and
101          (hold ?a1 ?r)                ;;recipientul de masurat e in bratul robotului
102          ;(exists (?s1 - reactant) (and (not (= ?s1 ?s)) (contains-s-ep ?s1 ?e) ));;
103          (not (full-ep ?e))           ;; nu e plina eprubeta
104          (hold-e ?a2 ?e)              ;;eprubeta e tinuta in brat pentru a adauga substanta
105          (not (empty-e ?e))           ;;eprubeta contine primul reactant
106          (used ?r)
107          (contains-s-rec ?s ?r)        ;;recipientul e plin cu substanta respectiva
108      )
109
110      :effect (and (contains-s-ep ?s ?e) ;;eprubeta contine reactantul s
111          (full-ep ?e)                 ;;eprubeta e plina
112          (not (contains-s-rec ?s ?r))  ;;nu mai contine substanta recipientului
113          (empty-used ?r)               ;;recipientul e empty-used (trebuie spalata daca s-a folosit)
114      )
115  )
116
117  ;;curatarea recipientului
118  (:action clean-recipient
119      :parameters (?r - recipient ?a - brat)
120      :precondition (and ;(empty-used ?r)
121          (hold ?a ?r)                ;;trebuie sa fie in mana recipientul pt a fi spalata
122      )
123      :effect (and (empty-clean ?r)
124          (not (empty-used ?r))
125          (not (used ?r))
126          (forall (?s - reactant) (when (and (contains-s-rec ?s ?r)) (not (contains-s-rec ?s ?r)) )
127      )
128  )
129
130  ;;clean eprubeta
131  (:action clean-eprubeta
132      :parameters (?e - eprubeta ?a - brat)
133      :precondition (and (not (empty-e ?e)) ;;sa nu o spele degeaba

```

```

134             (hold-e ?a ?e)           ;;sa aiba eprubeta in brat
135         )
136     :effect (and (empty-e ?e)
137                 (not (full-ep ?e))
138                 (forall (?s - reactant) (when (and (contains-s-ep ?s ?e)) (not (contains-s-ep ?s ?e) ))
139                 (not (result-sare ?e));
140                 (not (result-baza-isnolubila ?e)) ;;golirea eprubetei
141                 (not (result-precipitat ?e ))
142     )
143 )
144 )
145 ;;ia eprubeta in brat
146 (:action gripp-eprubeta
147     :parameters (?e - eprubeta ?a - brat)
148     :precondition (and (on-stativ-e ?e)
149                       (free-arm ?a)
150     )
151     :effect(and (not (on-stativ-e ?e))
152                (hold-e ?a ?e)           ;;e in mana
153                (not (free-arm ?a))
154    )
155 )
156 )
157
158 ;;lasa eprubeta pe stativ
159 (:action drop-eprubeta
160     :parameters (?e - eprubeta ?a - brat)
161     :precondition (and (hold-e ?a ?e)
162     )
163     :effect(and (on-stativ-e ?e)
164                 (not (hold-e ?a ?e))    ;;e pe stativ
165                 (free-arm ?a)
166    )
167 )
168 )
169
170 ;;verificare rezultat reactie
171 (:action check-reactie-baza-acid
172     :parameters (?e - eprubeta)
173     :precondition (and (full-ep ?e)
174                       (exists (?s - baza) (and (contains-s-ep ?s ?e) ))
175                       (exists (?s - acid) (and (contains-s-ep ?s ?e) ))
176    )
177     :effect (and (result-sare ?e))
178 )
179 )
180
181 ;;verificare rezultat reactie
182 (:action check-reactie-baza-sare
183     :parameters (?e - eprubeta)
184     :precondition (and (full-ep ?e)
185                       (exists (?s - baza) (and (contains-s-ep ?s ?e) ))
186                       (exists (?s - sare) (and (contains-s-ep ?s ?e) ))
187    )

```

```

188     )
189     :effect (and (result-baza-isnolubila ?e))
190 )
191
192 ;;verificare rezultat reactie
193 (:action check-reactie-acid-sare
194   :parameters (?e - eprubeta)
195   :precondition (and (full-ep ?e)
196                      (exists (?s - sare) (and (contains-s-ep ?s ?e) ))
197                      (exists (?s - acid) (and (contains-s-ep ?s ?e) ))
198   )
199   :effect (and (result-precipitat ?e) )
200 )
201
202
203
204
205
206 )

```

Explicații:

- ceea ce este de specificat este adăugarea observatorului **senseContain**; Prin adăugarea acestui observator, roboțelul poate observa ce substanțe se află în recipient. În acest caz, el poate să decidă dacă se poate folosi de ce are deja în recipient pentru a-și îndeplini scopul.

4.2 Definirea problemei

În cadrul problemei s-au adăugat situații incerte/necunoscute prin UNKNOWN. La fel ca și la conformant avem același scop.

Code:

```

1  (define (problem problemrob)
2  (:domain robchemistry)
3  (:objects
4    ep1 - eprubeta
5    acid1 - acid
6    baza1 - baza
7    sare1 - sare
8    berzelius1 - berzelius
9    erlenmeyer1 - erlenmeyer
10   left1 - left
11   right1 - right
12  )
13
14  (:init
15    ;paharele de masurat reactant
16    (used berzelius1)
17    (on-stativ berzelius1)
18    (on-stativ erlenmeyer1)
19    (used erlenmeyer1)
20
21    ;;eprubete
22    (on-stativ-e ep1)

```

```

23      (empty-e ep1)
24
25      (UNKNOWN (contains-s-rec acid1 berzelius1))
26      (UNKNOWN (contains-s-rec baza1 erlenmeyer1))
27      (UNKNOWN (contains-s-rec baza1 berzelius1))
28      (UNKNOWN (contains-s-rec acid1 erlenmeyer1))
29
30      (ONEOF (contains-s-rec acid1 berzelius1) (contains-s-rec baza1 berzelius1) )
31      (ONEOF (contains-s-rec acid1 erlenmeyer1) (contains-s-rec baza1 erlenmeyer1) )
32      (ONEOF (contains-s-rec acid1 berzelius1) (contains-s-rec acid1 erlenmeyer1) )
33      (ONEOF (contains-s-rec baza1 berzelius1) (contains-s-rec baza1 erlenmeyer1) )
34
35      ;;bratele sunt libere
36      (free-arm left1)
37      (free-arm right1)
38      ;;costul total
39
40 )
41
42 (:goal (and
43
44 ;; (full-ep ep1)
45 ;;(result-sare ep1)
46 ;; (hold-e left1 ep1)
47 ;; (full-ep ep2)
48 ;; (result-sare ep2)
49 ;;(contains-s-ep acid1 ep1);
50 (AND (result-sare ep1)
51 )
52 )
53
54 )
55 )

```

Explicații:

- starea inițială se definește ca fiind necunoscută. Este la fel ca la conformant. Știim că cele două pahare sunt pline cu un acid1 sau baza1 și nu se știe exact ce substanță se află. Fie acid1 se află în paharul berzelius, iar baza1 în paharul erlenmeyer, fie acid1 se află în paharul erlenmeyer, iar baza1 se află în paharul berzelius.
- Rezultatul obținut prin apelul : *./Contingent-ff -o robchemistry.pddl -f problemrob.pddl*

Rezultate

0||0 — GRIPP-EPRUBETA EP1 LEFT1 — SON: 1||0

1||0 — SENSECONTAIN BAZA1 ERLNMEYER1 — TRUESON: 2||0 — FALSESON: 2||1

2||0 — GRIPP-RECIPIENT RIGHT1 BERZELIUS1 — SON: 3||0

2||1 — GRIPP-RECIPIENT RIGHT1 BERZELIUS1 — SON: 3||1

3||0 — FIRST-REACTANT BERZELIUS1 ACID1 RIGHT1 LEFT1 EP1 — SON: 4||0

3||1 — FIRST-REACTANT BERZELIUS1 BAZA1 RIGHT1 LEFT1 EP1 — SON: 4||1

4||0 — DROP-RECIPIENT RIGHT1 BERZELIUS1 — SON: 5||0
4||1 — DROP-RECIPIENT RIGHT1 BERZELIUS1 — SON: 5||1

5||0 — GRIPP-RECIPIENT RIGHT1 ERLLENMEYER1 — SON: 6||0
5||1 — GRIPP-RECIPIENT RIGHT1 ERLLENMEYER1 — SON: 6||1

6||0 — SECOND-REACTANT ERLLENMEYER1 BAZA1 RIGHT1 LEFT1 EP1 — SON: 7||0
6||1 — SECOND-REACTANT ERLLENMEYER1 ACID1 RIGHT1 LEFT1 EP1 — SON: 7||1

7||0 — CHECK-REACTIE-BAZA-ACID EP1 — SON: 8||-1
7||1 — CHECK-REACTIE-BAZA-ACID EP1 — SON: 8||-1

tree layers: 8

total nr. actions: 14

- se observă că robotul primește o informație de la senzori; Aici problema se va ramifica în două în funcție de ce conține în paharul erlenmeyer; Dacă în erlenmeyer se găsește baza1, atunci se va merge pe ramura 0, iar în caz contrar se va continua pe ramura 1 (în care acid1 se va afla în paharul erlenmeyer).
- arborele format va avea 8 nivele, iar numărul total de acțiuni va fi de 14.
- mai jos se regăsește arborele de planificare în funcție de decizia pe care o să o ia robotul aflând informația că are substanțele necesare obținerii sării în cele două recipiente, scutindu-l astfel de a măsura cantitatea de substanțe.
-

4.3 References

R. Stuart, N. Peter, Artificial Intelligence: A Modern Approach, 4th US ed., capitol 3, [online]

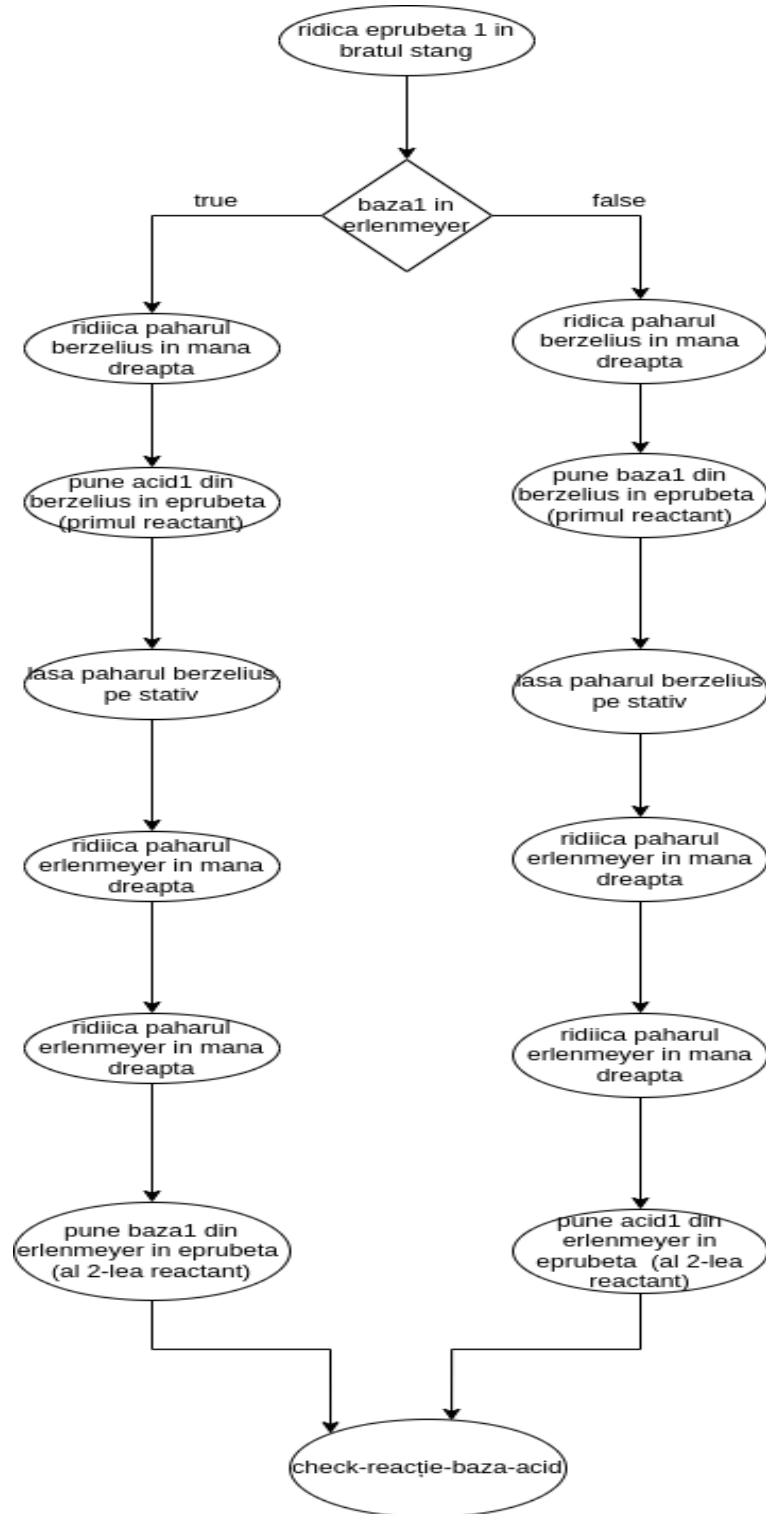


Figure 2: Arbore