

ProiectIA3

Balc Horia

January 2021

1 Prezentarea Temei

Proiectul creat doreste sa simuleze o berarie, in care daca este un client acesta poate sa isi aleaga tipul de bere dorita, modul de plata se poate efectua prin intermediul cardului bancar sau in numerar. Dupa ce clientul face aceste doua actiuni i se livreaza comanda online sau i se serveste in local, in functie de ce predicat se afla in goal(TipBere respectiv TipBereOnline). Dupa ce se livreaza comanda clientul primeste comanda dupa care se plateste comanda in modul ales mai inainte, se finalizeaza comanda, iar la final clientul se afla cu berea de tipul dorit in mana, deci este fericit. Dupa terminarea acestui procedeu, se poate relua pentru un alt client.

2 Implementarea proiectului

In aceasta sectiune se explica implementarea proiectului (codul).

Mai jos este definit domeniul si cerintele pentru implementarea costurilor. (define (domain berarie-domain) (:requirements :strips :action-costs))

Urmatoarea sectiune defineste toate predicatele folosite in proiect. Predicatul berarieDeschisa este True doar daca beraria este deschisa, altfel este False. Predicatul existaClient (este cu un singur parametru) daca este True arata faptul ca un client a intrat in magazin (online sau nu), altfel nu este nimeni in magazin dornic sa cumpere bere. Predicatul ocupat reprezinta faptul ca este ocupata sau nu casa de marcat a magazinului. Predicatul trimisComanda (este cu un singur parametru) daca este True inseamna ca s-a trimis comanda in local. Predicatul trimisComandaOnline (este cu un singur parametru) daca este True inseamna ca s-a trimis comanda Online. Predicatul primitComanda (este cu un singur parametru) daca este True inseamna ca s-a primit comanda in local. Predicatul primitComandaOnline (este cu un singur parametru) daca este True inseamna ca s-a primit comanda Online. Predicatul TestN (este cu un singur parametru) reprezinta ca clientul a ales berea NeneaIancu. Predicatul TestZ (este cu un singur parametru) reprezinta ca clientul a ales berea Zaganu. Predicatul TestB (este cu un singur parametru) reprezinta ca clientul a ales berea Bucur. Predicatul TestZI (este cu un singur parametru) reprezinta ca clientul a ales berea Zimbru. Predicatul TestA (este cu un singur parametru) reprezinta

ca clientul a ales berea Azuga. Predicatul TestS (este cu un singur parametru) reprezinta ca clientul a ales berea Strasnic. Predicatul TestCard (este un singur parametru) reprezinta faptul ca clientul alege plata cu cardul. Predicatul TestCash (este un singur parametru) reprezinta faptul ca clientul alege plata in numerar. ClientulAPrimitBere este true inseamna ca acesta isi savureaza berea in local. ClientulAPrimitBereOnline este true inseamna ca acesta isi savureaza berea acasa. Predicatul ModulCard (este cu un singur parametru) reprezinta faptul ca plata se efectueaza cu cardul. Predicatul ModulCash (este cu un singur parametru) reprezinta faptul ca plata se efectueaza in numerar. TipBere reprezinta faptul ca berea este savurata in local. TipBereOnline reprezinta faptul ca berea este savurata online. NeneaIancu reprezinta faptul ca clientul a ales acet tip de bere. Zimbru reprezinta faptul ca clientul a ales acet tip de bere. Zaganu reprezinta faptul ca clientul a ales acet tip de bere. Calimani reprezinta faptul ca clientul a ales acet tip de bere. Strasnic reprezinta faptul ca clientul a ales acet tip de bere. Azuga reprezinta faptul ca clientul a ales acet tip de bere.

```
(:predicates (berarieDeschisa)
(existaClient ?x)
(ocupat)
(trimisComanda ?x)
(trimisComandaOnline ?x)
(primitPlata ?x)
(primitPlataOnline ?x)
(testN ?x)
(testZ ?x)
(testA?x)
(testB ?x)
(testC ?x)
(testS ?x)
(testZI ?x)
(testCard ?x)
(testCash ?x)
(ClientulAPrimitBere ?x)
(ClientulAPrimitBereOnline ?x)
(modulcard ?x)
(modulcash ?x)
(TipBere ?x)
(TipBereOnline ?x)
(NeneaIancu ?x)
(Zaganu ?x)
(Azuga ?x)
(Bucur ?x)
(Calimani ?x)
(Zimbru ?x)
(Strasnic ?x)
)
```

(:functions (total-cost))

Mai jos sunt prezentate toate actiunile. Prima actiune este cea de deschidere a berariei in cazul i care aceasta este inchisa. (:action deschidere-Berarie
:parameters ()
:precondition (not (berarieDeschisa))
:effect (berarieDeschisa))

Urmatoarea actiune este cea care creeaza un nou client daca nu exista deja unul. (:action client-Nou
:parameters (?x)
:precondition (and (berarieDeschisa) (not(existaClient ?x))
(not(ocupat))))
:effect (and (existaClient ?x) (ocupat)))

Urmatoarea actiune este cea care alege modul cash de plata. (:action alege-Mod-Plata-Cash
:parameters (?x)
:precondition (and (berarieDeschisa) (testCash ?x) (existaClient ?x))
:effect (and (modulcash ?x)
(increase (total-cost) 1)))

Urmatoarea actiune este cea care alege modul card de plata. (:action alege-Mod-Plata-Card
:parameters (?x)
:precondition (and (berarieDeschisa) (testCard ?x) (existaClient ?x))
:effect (and(modulcard ?x)
(increase (total-cost) 1)))

Urmatoarea actiune este cea care alege berea Nenea-Iancu. (:action alege-Nenea-Iancu
:parameters (?x)
:precondition (and (berarieDeschisa) (testN ?x) (existaClient ?x))
:effect (NeneaIancu ?x))

Urmatoarea actiune este cea care alege berea Zaganu. (:action alege-Zaganu
:parameters (?x)
:precondition (and (berarieDeschisa) (testZ ?x) (existaClient ?x))
:effect (and (Zaganu ?x)))

Urmatoarea actiune este cea care alege berea Azuga. (:action alege-Azuga
:parameters (?x)
:precondition (and (berarieDeschisa) (testA ?x) (existaClient ?x))
:effect (Azuga ?x))

Urmatoarea actiune este cea care alege berea Calimani. (:action alege-

```

Calimani
:parameters (?x)
:precondition (and (berarieDeschisa) (testC ?x) (existaClient ?x))
:effect (Calimani ?x)

```

Urmatoarea actiune este cea care alege berea Zimbru. (:action alege-Zimbru

```

:parameters (?x)
:precondition (and (berarieDeschisa) (testZI ?x) (existaClient ?x))
:effect (Zimbru ?x))

```

Urmatoarea actiune este cea care alege berea Strasnic. (:action alege-Strasnic

```

:parameters (?x)
:precondition (and (berarieDeschisa) (testS ?x) (existaClient ?x))
:effect (Strasnic ?x) )

```

Urmatoarea actiune este cea in care comanda se livreaza online. (:action livrare-Comanda-Online

```

:parameters (?x)
:precondition (and (berarieDeschisa) (not (trimisComandaOnline ?x)
) (or (NeneaIancu ?x) (Strasnic ?x) (Azuga ?x) (Zimbru ?x) (Calimani ?x) (Za-
ganu ?x) (Bucur ?x)))
:effect (and (trimisComandaOnline ?x)
(increase (total-cost) 3)))

```

Urmatoarea actiune este cea in care comanda se livreaza in local. (:action livrare-Comanda

```

:parameters (?x)
:precondition (and (berarieDeschisa) (not (trimisComanda ?x))
(or (NeneaIancu ?x) (Strasnic ?x) (Azuga ?x) (Zimbru ?x) (Calimani ?x) (Za-
ganu ?x) (Bucur ?x)))
:effect (and (trimisComanda ?x) (not (comanda))
(increase (total-cost) 2)))

```

Urmatoarea actiune este cea in care plata se face la o comanda in local. (:action plata-Comanda

```

:parameters (?x)
:precondition (and (berarieDeschisa) (trimisComanda ?x))
:effect (and (primitPlata ?x) (not (trimisComanda ?x))))

```

Urmatoarea actiune este cea in care plata se face la o comanda Online. (:action plata-Comanda-Online

```

:parameters (?x)
:precondition (and (berarieDeschisa) (trimisComandaOnline ?x))
:effect (and (primitPlataOnline ?x) (not (trimisComandaOnline ?x))))
Urmatoarea actiune este cea care finalizeaza plata cu cardul in local. (:action finalizare-Plata-Card

```

```

:parameters (?x)
:precondition (and (berarieDeschisa) (primitPlata ?x) (modulcard ?x))
:effect (and (not (primitPlata?x)) (ClientulAPrimitBere ?x)))
Urmatoarea actiune este cea care finalizeaza plata in numerar in local. (:action
finalizare-Plata-Cash
:parameters (?x)
:precondition (and (berarieDeschisa) (primitPlata ?x) (modulcash ?x))
:effect (and (not (primitPlata ?x)) (ClientulAPrimitBere?x)))

```

Urmatoarea actiune este cea care finalizeaza plata cu cardul la o comanda online. (:action finalizare-Plata-Online-Card

```

:parameters (?x)
:precondition (and (berarieDeschisa) (primitPlataOnline ?x) (modulcard ?x))
:effect (and (not (primitPlataOnline ?x))
(ClientulAPrimitBereOnline?x) ))

```

Urmatoarea actiune este cea care finalizeaza plata cu cardul la o comanda efectuata in local. (:action finalizare-Plata-Online-Cash

```

:parameters (?x)
:precondition (and (berarieDeschisa) (primitPlataOnline ?x)
(modulcash ?x))
:effect (and (not (primitPlataOnline ?x))
(ClientulAPrimitBereOnline?x) ))

```

Urmatoarea actiune este cea care arata faptul ca un client si-a comandat o bere de tipul Nenea Iancu in local. (:action Clientul-A-Primit-Bere-De-Tipul-Nenea-Iancu

```

:parameters (?x)
:precondition (and (berarieDeschisa) (ocupat) (ClientulAPrimitBere ?x) (not
(TipBere ?x)) (NeneaIancu ?x))
:effect (and (TipBere ?x) (not (NeneaIancu ?x)) (not (ocupat))))

```

Urmatoarea actiune este cea care arata faptul ca un client si-a comandat o bere de tipul Nenea Iancu online. (:action Clientul-A-Primit-Online-Bere-De-Tipul-Nenea-Iancu

```

:parameters (?x)
:precondition (and (berarieDeschisa) (ClientulAPrimitBereOnline?x) (ocupat)
(not (TipBereOnline ?x)) (NeneaIancu ?x))
:effect (and (TipBereOnline ?x) (not (NeneaIancu ?x))(not (ocupat))))
Urmatoarea actiune este cea care arata faptul ca un client si-a comandat o bere
de tipul Zaganu in local. (:action Clientul-A-Primit-Bere-De-Tipul-Zaganu
:parameters (?x)
:precondition (and (berarieDeschisa) (ClientulAPrimitBere?x)
(ocupat) (not (TipBere ?x)) (Zaganu ?x))
:effect (and (TipBere ?x) (not (Zaganu ?x))(not (ocupat))))
Urmatoarea actiune este cea care arata faptul ca un client si-a comandat o

```

bere de tipul Zaganu online. (:action Clientul-A-Primit-Online-Bere-De-Tipul-Zaganu
:parameters (?x)
:precondition (and (berarieDeschisa) (ClientulAPrimitBereOnline?x) (ocupat)
(not (TipBereOnline ?x)) (Zaganu ?x))
:effect (and (TipBereOnline ?x) (not (Zaganu ?x))(not (ocupat))))
Urmatoarea actiune este cea care arata faptul ca un client si-a comandat o bere
de tipul Zimbru in local. (:action Clientul-A-Primit-Bere-De-Tipul-Zimbru
:parameters (?x)
:precondition (and (berarieDeschisa) (ClientulAPrimitBere?x) (ocupat) (not (Tip-
Bere ?x)) (Zimbru ?x))
:effect (and (TipBere ?x) (not (Zimbru ?x))(not (ocupat))))

Urmatoarea actiune este cea care arata faptul ca un client si-a comandat o
bere de tipul Zimbru online. (:action Clientul-A-Primit-Online-Bere-De-Tipul-
Zimbru
:parameters (?x)
:precondition (and (berarieDeschisa) (ClientulAPrimitBereOnline?x) (not (Tip-
BereOnline ?x)) (ocupat) (Zimbru ?x))
:effect (and (TipBereOnline ?x) (not (Zimbru ?x))(not (ocupat))))

Urmatoarea actiune este cea care arata faptul ca un client si-a comandat
o bere de tipul Strasic in local. (:action Clientul-A-Primit-Bere-De-Tipul-
Strasic
:parameters (?x)
:precondition (and (berarieDeschisa) (ClientulAPrimitBere?x) (not (TipBere
?x)) (ocupat) (Strasic ?x))
:effect (and (TipBere ?x) (not (Strasic ?x))(not (ocupat))))

Urmatoarea actiune este cea care arata faptul ca un client si-a comandat o
bere de tipul Strasic online. (:action Clientul-A-Primit-Online-Bere-De-Tipul-
Strasic
:parameters (?x)
:precondition (and (berarieDeschisa) (ClientulAPrimitBereOnline?x) (not (Tip-
BereOnline ?x)) (ocupat) (Strasic ?x))
:effect (and (TipBereOnline ?x) (not (Strasic ?x))(not (ocupat))))

Urmatoarea actiune este cea care arata faptul ca un client si-a comandat o
bere de tipul Azuga in local. (:action Clientul-A-Primit-Bere-De-Tipul-Azuga
:parameters (?x)
:precondition (and (berarieDeschisa) (ClientulAPrimitBere?x)
(ocupat) (not (TipBere ?x)) (Azuga ?x))
:effect (and (TipBere ?x) (not (Azuga ?x))(not (ocupat))))

Urmatoarea actiune este cea care arata faptul ca un client si-a comandat o
bere de tipul Azuga online. (:action Clientul-A-Primit-Online-Bere-De-Tipul-

```

Azuga
:parameters (?x)
:precondition (and (berarieDeschisa) (ClientulAPrimitBereOnline?x) (not (TipBereOnline ?x)) (ocupat)(Azuga ?x))
:effect (and (TipBereOnline ?x) (not (Azuga ?x)) (not (ocupat))))
Urmatoarea actiune este cea care arata faptul ca un client si-a comandat o bere de tipul Bucur in local. (:action Clientul-A-Primit-Bere-De-Tipul-Bucur
:parameters (?x)
:precondition (and (berarieDeschisa) (ClientulAPrimitBere?x) (not (TipBere?x)) (ocupat) (Bucur ?x))
:effect (and (TipBere ?x) (not (Bucur ?x)) (not (ocupat))))

```

Urmatoarea actiune este cea care arata faptul ca un client si-a comandat o bere de tipul Bucur online. (:action Clientul-A-Primit-Online-Bere-De-Tipul-Bucur

```

:parameters (?x)
:precondition (and (berarieDeschisa) (ClientulAPrimitBereOnline?x) (not (TipBereOnline ?x)) (ocupat) (Bucur ?x))
:effect (and (TipBereOnline ?x) (not (Bucur ?x)) (not (ocupat))))

```

Urmatoarea actiune este cea care arata faptul ca un client si-a comandat o bere de tipul Calimani in local. (:action Clientul-A-Primit-Bere-De-Tipul-Calimani

```

:parameters (?x)
:precondition (and (berarieDeschisa) (ClientulAPrimitBere?x) (not (TipBere?x)) (Calimani ?x) (ocupat))
:effect (and (TipBere ?x) (not (Calimani ?x)) (not (ocupat))))

```

Urmatoarea actiune este cea care arata faptul ca un client si-a comandat o bere de tipul Calimani online. (:action Clientul-A-Primit-Online-Bere-De-Tipul-Calimani

```

:parameters (?x)
:precondition (and (berarieDeschisa) (ClientulAPrimitBereOnline?x) (not (TipBereOnline ?x)) (Calimani ?x) (ocupat))
:effect (and (TipBereOnline ?x) (not (Calimani ?x)) (not (ocupat))))

```

)

3 Scenarii

Primul scenariu este cel mai simplu posibil, in care exista un singur client, iar acesta comanda in local o bere de tipul Nenea Iancu si o plateste numerar. Codul problemei date este redat mai jos.

```
define (problem berarie)
```

```

(:domain berarie-domain)
(:objects a)
(:init (testN a) (testCash a) (= (total-cost) 0))
(:goal (TipBere a))
(:metric minimize (total-cost))

```

In al doilea scenariu se afla doi clienti unul comanda in local o bere de tipul Zaganu si o plateste numerar, iar celalalt isi cumpara o bere Strasnic online si o plateste cu cardul. Codul problemei date este redat mai jos.

```

define (problem berarie)
  (:domain berarie-domain)
  (:objects a b)
  (:init (testZ a) (testCash a) (testS b) (testCard b) (= (total-cost) 0))
  (:goal (and (TipBere a) (TipBereOnline b)))
  (:metric minimize (total-cost))

```

In cel de-al treilea scenariu se afla trei clienti primul comanda in local o bere de tipul Zaganu si o plateste numerar, al doilea isi cumpara o bere Strasnic online si o plateste cu cardul, iar ultimul isi comanda in local o bere Zimbru si o plateste cu cardul. Codul problemei date este redat mai jos.

```

define (problem berarie)
  (:domain berarie-domain)
  (:objects a b)
  (:init (testZ a) (testCash a) (testS b) (testCard b) (testZI c) (testCard c)
    (= (total-cost) 0))
  (:goal (and (TipBere a) (TipBereOnline b) (TipBere c)))
  (:metric minimize (total-cost))

```


4 Rezultate

4.1 Primul rezultat

Rezultatul primei probleme este:

```
1 (deschidere-berarie )
2 (client-nou a)
3 (alege-mod-plata-cash a)
4 (alege-zaganu a)
5 (livrare-comanda a)
6 (plata-comanda a)
7 (finalizare-plata-cash a)
8 (clientul-a-primit-bere-de-tipul-zaganu a)
9 ; cost = 3 (general cost)
```

Figure 1: Scenariul1

4.2 Al doilea rezultat

Rezultatul celei de-a doua probleme este:

```
1 (deschidere-berarie )
2 (client-nou a)
3 (alege-mod-plata-cash a)
4 (alege-zaganu a)
5 (livrare-comanda a)
6 (plata-comanda a)
7 (finalizare-plata-cash a)
8 (clientul-a-primit-bere-de-tipul-zaganu a)
9 (client-nou b)
10 (alege-mod-plata-card b)
11 (alege-strasnic b)
12 (livrare-comanda b)
13 (plata-comanda b)
14 (finalizare-plata-card b)
15 (clientul-a-primit-bere-de-tipul-strasnic b)
16 ; cost = 6 (general cost)
```

Figure 2: Scenariul2

4.3 Al treilea rezultat

Rezultatul celei de-a treia probleme este:

```
1 (deschidere-berarie )
2 (client-nou a)
3 (alege-mod-plata-cash a)
4 (alege-zaganu a)
5 (livrare-comanda a)
6 (plata-comanda a)
7 (finalizare-plata-cash a)
8 (clientul-a-primit-bere-de-tipul-zaganu a)
9 (client-nou b)
10 (alege-mod-plata-card b)
11 (alege-strasnic b)
12 (livrare-comanda b)
13 (plata-comanda b)
14 (finalizare-plata-card b)
15 (clientul-a-primit-bere-de-tipul-strasnic b)
16 (client-nou c)
17 (alege-mod-plata-card c)
18 (alege-zimbru c)
19 (livrare-comanda-online c)
20 (plata-comanda-online c)
21 (finalizare-plata-online-card c)
22 (clientul-a-primit-online-bere-de-tipul-zimbru c)
23 ; cost = 10 (general cost)
```

Figure 3: Scenariul3

5 Euristici

5.1 Primul scenariu:

La primul scenariu daca apelam in terminal cu functia lazy-greedy cu 3 euristici:
In cazul ff vor fi 8 noduri expandate, 9 noduri evaluate si 24 noduri generate.
In cazul cg vor fi 8 noduri expandate, 9 noduri evaluate si 24 noduri generate.
In cazul add vor fi 8 noduri expandate, 9 noduri evaluate si 24 noduri generate.

La primul scenariu daca apelam in terminal cu functia lazy-wastar cu 3 euristici:

In cazul ff vor fi 11 noduri expandate, 12 noduri evaluate si 36 noduri generate.
In cazul cg vor fi 11 noduri expandate, 12 noduri evaluate si 36 noduri generate.
In cazul add vor fi 11 noduri expandate, 12 noduri evaluate si 36 noduri generate.

La primul scenariu daca apelam in terminal cu functia astar cu 3 euristici:
In cazul ff vor fi 9 noduri expandate, 16 noduri evaluate si 24 noduri generate.
In cazul cg vor fi 9 noduri expandate, 16 noduri evaluate si 24 noduri generate.
In cazul add vor fi 9 noduri expandate, 16 noduri evaluate si 24 noduri generate.

5.2 Al doilea scenariu:

La al doilea scenariu daca apelam in terminal cu functia lazy-greedy cu 3 euristici:

In cazul ff vor fi 7 noduri expandate, 16 noduri evaluate si de 64 noduri generate. In cazul cg vor fi 16 noduri expandate, 17 noduri evaluate si de 64 noduri generate. In cazul add vor fi 16 noduri expandate, 17 noduri evaluate si de 64 noduri generate.

La al doilea scenariu daca apelam in terminal cu functia lazy-wastar cu 3 euristici:

In cazul ff vor fi 22 noduri expandate, 23 noduri evaluate si de 94 noduri generate. In cazul cg vor fi 40 noduri expandate, 41 noduri evaluate si de 181 noduri generate. In cazul add vor fi 19 noduri expandate, 20 noduri evaluate si de 82 noduri generate.

La al doilea scenariu daca apelam in terminal cu functia astar cu 3 euristici:
In cazul ff vor fi 21 noduri expandate, 54 noduri evaluate si 99 noduri generate.
In cazul cg vor fi 21 noduri expandate, 54 noduri evaluate si 99 noduri generate.
In cazul add vor fi 21 noduri expandate, 54 noduri evaluate si 99 noduri generate.

5.3 Al treilea scenariu:

La al treilea scenariu daca apelam in terminal cu functia lazy-greedy cu 3 euristici:

In cazul ff vor fi 25 noduri expandate, 26 noduri evaluate si de 123 noduri generate. In cazul cg vor fi 25 noduri expandate, 26 noduri evaluate si de 123 noduri generate. In cazul add vor fi 25 noduri expandate, 26 noduri evaluate si de 123 noduri generate.

La al treilea scenariu daca apelam in terminal cu functia lazy-wastar cu 3 euristici:

In cazul ff vor fi 22 noduri expandate, 23 noduri evaluate si de 94 noduri generate.. In cazul cg vor fi 116 noduri expandate, 117 noduri evaluate si de 655 noduri generate. In cazul add vor fi 53 noduri expandate, 54 noduri evaluate si de 269 noduri generate.

La al treilea scenariu daca apelam in terminal cu functia astar cu 3 euristici: In cazul ff vor fi 40 noduri expandate, 132 noduri evaluate si 272 noduri generate. In cazul cg vor fi 40 noduri expandate, 132 noduri evaluate si 272 noduri generate. In cazul add vor fi 96 noduri expandate, 405 noduri evaluate si 854 noduri generate.

6 Concluzie

“In acest proiect am exersat limbajul pddl si in acelasi timp mi-am imbunatatit logica de gandire. (?)