UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA

Proiect Sisteme Expert

Moral Machine

Student: Péntek Tamás

Grupa: 30642

Data: 01.12.2020

UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA

Domeniul aplicatiei

Masina morala (Moral Machine) a fost dezvoltata de cercetatorii de la Massachusetts Institute of Technology (MIT) si reprezinta o platforma pentru adunarea unei perspective umane asupra deciziilor morale luate de inteligenta masinii, cum ar fi masinile de tip self-driving. Pe aceasta platforma sunt create niste dileme morale, iar o masina self-driving trebuie sa aleaga intre cele doua situatii fatale, cum ar fi uciderea a 4 pasageri tineri sau a 4 pietoni batrani sau uciderea a 3 animale sau a 3 pasageri.

Fiindca nu exista o decizie corecta din punct de vedere moral, trebuie sa stabilim cateva criterii de selectie si bazand pe aceste criterii luam deciziile. Aceasta aplicatie simuleaza deciziile luate de masina morala in diferite situatii. Decizia este luata in functie de mai multe criterii, care sunt prezentate in continuare.

Critererii de selectie:

a. *Legislatia rutiera*
- Daca avem un drum cu doua benzi si pe o banda avem o bariera si pe cealalta banda avem o trecere de pietoni si semaforul trecerii este rosu, atunci masina va merge pe banda unde avem trecerea de pietoni si semaforul rosu.

b. *Varsta*
- Daca avem un drum cu doua benzi si pe ambele benzi sunt pietoni, masina va merge pe banda unde suma varstelor persoanelor este mai mare, adica vom merge pe banda pe care sunt mai multe persoane in varsta si salvam viata oamenilor tineri.

c. *Oameni vs animale*
- Daca avem un drum cu doua benzi si pe o banda avem pe trecere de pietoni oameni si pe cealalta banda animele (in acest caz caini si pisici), atunci masina se va indrepta pe banda unde avem animale, salvand viata oamenilor

d. *Numarul de persoane*
- Daca avem un drum cu doau benzi si pe o banda avem pe trecere de pietoni 5 persoane si pe cealalta banda sunt 3 persoane, atunci masina va merge pe banda pe care sunt 3 persoana, adica masina mereu se va indrepta pe banda care contine cele mai putine persoane, in acest fel salvand cele mai multe persoane

UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA

Descrierea scenariilor de test

Fiecare scenariu de test este descris prin niste perceptii care pot fi obtinute prin colectarea si prelucrarea datelor din senzorii si camera instalata pe masina.

Semafor: Detectarea se poate face folosind camera si image processing. Sursa [1] descrie o astfel de metoda: detectarea semaforului se face in real-time, fiindca prelucrarea unui cadru dureaza aproximativ 20 ms, iar sistemul are o acuratete de peste 95% pentru drumuri nationale si aproximativ 85% pentru zonele urbane. Pentru evaluarea calitativa sunt diferentiate doua tipuri de rezultate: true frame si false frame. True frame este numarat ca starea corecta a unui frame. False frame are doua subtipuri: noisy frame si defective frame. Un noisy frame este recunoscut atunci cand este detectata o stare activa intr-o stare inactiva. Atunci cand frame-ul A este in stare activa, dar sistemul a detectat ca frame-ul B este in stare activa, rezultatul este considerat a fi un defective frame.

Bariera: Detectarea acestui obstacol se face folosind camera instalata pe masina si prin tehnici de object detection. In lucrarea de cercetare [2] este prezentat un sistem care face detectie de obiecte in real-time, prelucrarea unui cadru dureaza 15 ms, iar mAP (mean Average Precision - este o masura care combina recall si precision) este 52.1%.

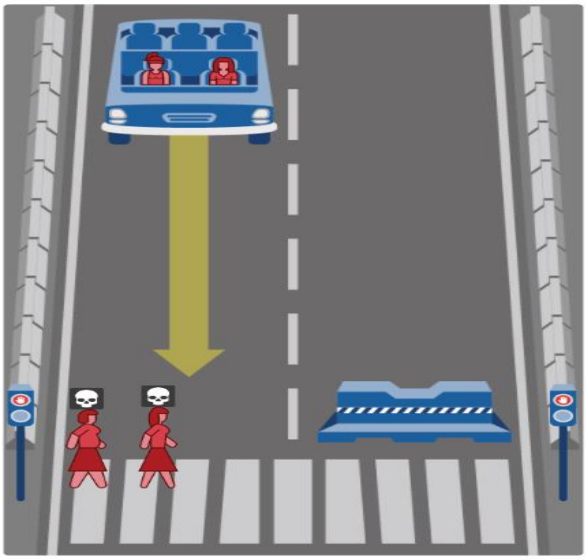
Varsta persoanelor: Se poate obtine prin tehnici de image processing si deep learning, prima data detectam fata persoanei, iar apoi pe baza fetei estimam varsta. In lucrarea [3] este descrisa o metoda de face detection care are ca si hit rate (sau true positive rate, este egal cu TP/(TP + FN)) 93.2%, ceea ce este o acuratete destul de buna, iar in lucrarea [4] este prezentat un estimator de varsta, care a obtinut o CCR (Correct Classification Rate - este raportul dintre obiecte clasificate corect si numarul total de obiecte din setul de test) de 72.53%.

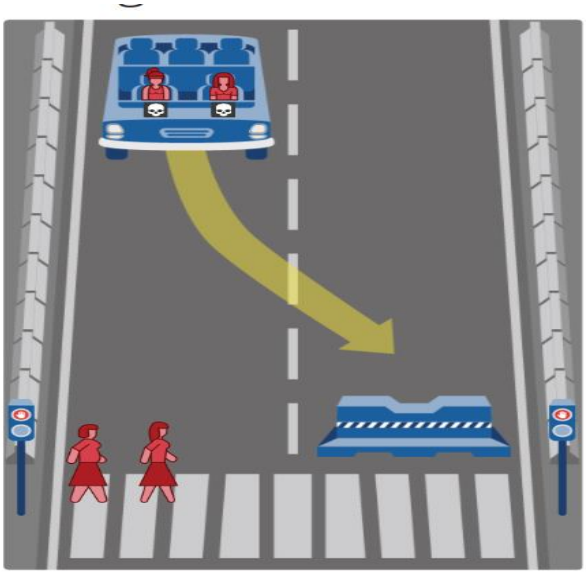
Oameni vs animale: Se poate detecta folosind camera si niste tehnici de object detection. In acest caz se poate folosi aceasi metoda care este prezentata in lucrarea [2].

Numarul de persoane: Prima data se face object detection, iar apoi se numara obiectele de acelasi tip. O abordare posibila este prezentat in lucrarea [2]. O alta metoda mai directa este prezentat in research paper [5]. Acest sistem (Real-time people counting system) functioneaza si in situatii mai complexe, de exemplu in diferite conditii de lumina. Viteza acestui sistem este aproximativ 25.9 ms / frame.

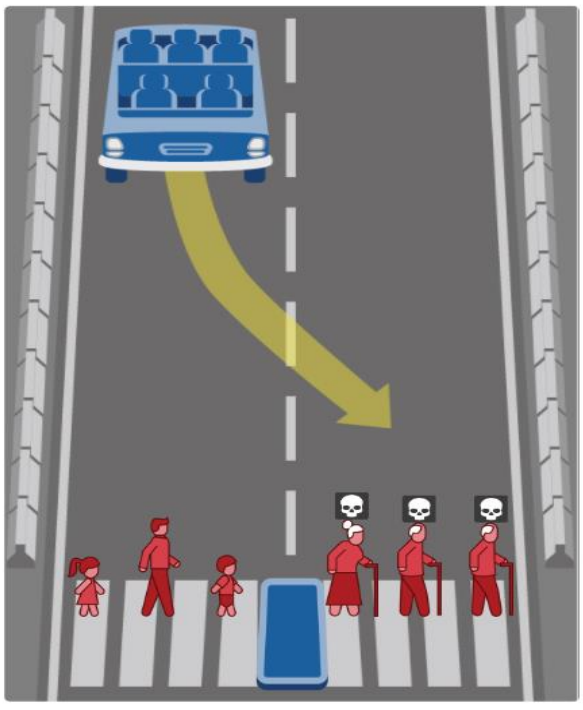
Scenariul 1

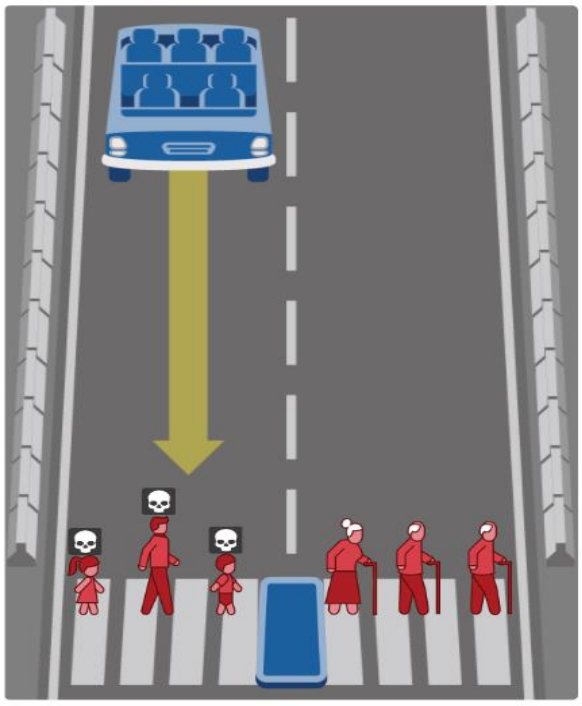
Inainte se afla doi pietoni pe trecerea de pietoni, dar semaforul trecerii este rosu, iar pe stanga este o bariera pe drum si in masina se afla doua persoane. In acest caz masina se va executa manevra de a merge **inainte**, bazand pe criteriul de legislatie rutiera si in acest fel salvand pasagerii masinii.





Scenariul 2

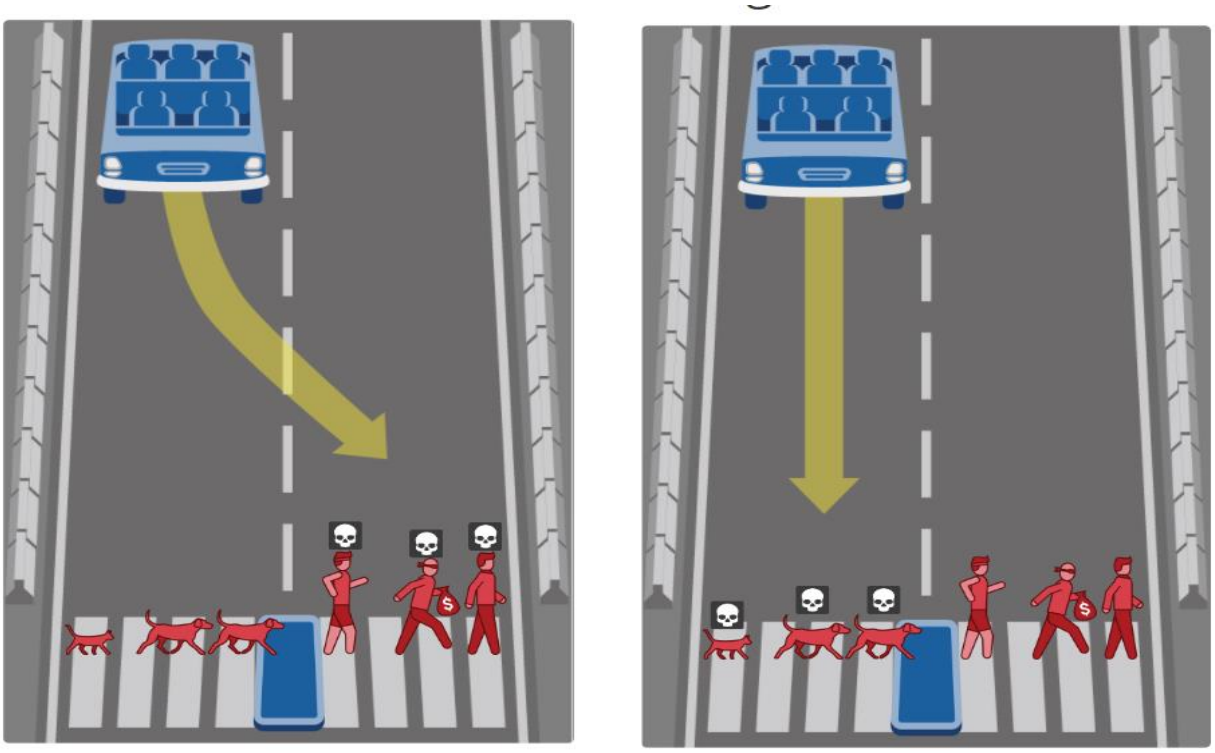




In masina nu se afla niciun pasager, presupunem ca este vorba despre un autovehicul de tip full self-driving. Pe ambele benzi avem o trecere de pietoni, iar inainte sunt doi copii si un

adult, iar pe partea stanga sunt trei persoane. Fiindca suma varstelor pe stanga este mai mare, masina se va executa manevra de a merge la **stanga**, bazand pe criteriul de varsta.

Scenariul 3



In masina nu se afla niciun pasager, presupunem ca este vorba despre un autovehicul de tip full self-driving. Pe ambele benzi avem o trecere de pietoni, inainte avem doi caini si o pisica, iar pe partea stanga avem 3 persoane. Din aceste fapte reiese ca numarul de pietoni este mai mare pe partea stanga. Masina se va executa manevra de a merge **inainte**, folosind doua criterii: criteriul oameni vs animale si criteriul de numarul de persoane.

Perceptii

Pentru a descrie perceptiile, a fost folosit template-ul numit ag_percept. Acest template are urmatoarele slot-uri si valori:

- percept_pobj: are ca si valori: e1, e2, ped1, ped2, sem1, pas1, pas2, bar1
- percept_pname: are ca si valori: isa, direction, age, color, partof
- percept_pval: are ca si valori: event, pedestrian, semaphore, passenger, barrier, ahead, left, 40, red, e1

Exemple de cadre din fiecare scenariu:

a. Scenariul 1 – manevra go ahead

(ag_percept (percept_pobj e1) (percept_pname isa) (percept_pval event)) => e1 reprezinta un eveniment
(ag_percept (percept_pobj ped1) (percept_pname isa) (percept_pval pedestrian)) => ped1 reprezinta un pieton
(ag_percept (percept_pobj ped2) (percept_pname isa) (percept_pval pedestrian)) => ped2 reprezinta un pieton
(ag_percept (percept_pobj sem1) (percept_pname isa) (percept_pval semaphore)) => sem1 reprezinta un semafor
(ag_percept (percept_pobj e1) (percept_pname direction) (percept_pval ahead)) => directia evenimentului e1 este inainte
(ag_percept (percept_pobj ped1) (percept_pname age) (percept_pval 40)) => pietonul ped1 are varsta de 40 de ani
(ag_percept (percept_pobj ped2) (percept_pname age) (percept_pval 45)) => pietonul ped1 are varsta de 45 de ani
(ag_percept (percept_pobj sem1) (percept_pname color) (percept_pval red)) => culoarea semaforului sem1 este rosu
(ag_percept (percept_pobj ped1) (percept_pname partof) (percept_pval e1)) => pietonul ped1 apartine evenimentului e1
(ag_percept (percept_pobj ped2) (percept_pname partof) (percept_pval e1)) => pietonul ped2 apartine evenimentului e1
(ag_percept (percept_pobj sem1) (percept_pname partof) (percept_pval e1)) => semaforul sem1 apartine evenimentului e1

b. Scenariul 2 – manevra go left

(ag_percept (percept_pobj e1) (percept_pname isa) (percept_pval event)) => e1 reprezinta un eveniment
(ag_percept (percept_pobj ped1) (percept_pname isa) (percept_pval pedestrian)) => ped1 reprezinta un pieton
(ag_percept (percept_pobj ped2) (percept_pname isa) (percept_pval pedestrian)) => ped2 reprezinta un pieton
(ag_percept (percept_pobj ped3) (percept_pname isa) (percept_pval pedestrian)) => ped3 reprezinta un pieton
(ag_percept (percept_pobj e1) (percept_pname direction) (percept_pval left)) => directia evenimentului e1 este inainte

(ag_percept (percept_pobj ped1) (percept_pname age) (percept_pval 70)) => pietonul ped1 are varsta de 70 de ani
(ag_percept (percept_pobj ped2) (percept_pname age) (percept_pval 75)) => pietonul ped2 are varsta de 75 de ani
(ag_percept (percept_pobj ped3) (percept_pname age) (percept_pval 73)) => pietonul ped3 are varsta de 73 de ani
(ag_percept (percept_pobj ped1) (percept_pname partof) (percept_pval e1)) => pietonul ped1 apartine evenimentului e1
(ag_percept (percept_pobj ped2) (percept_pname partof) (percept_pval e1)) => pietonul ped2 apartine evenimentului e1
(ag_percept (percept_pobj ped3) (percept_pname partof) (percept_pval e1)) => pietonul ped3 apartine evenimentului e1

c. Scenariul 3 – manevra go ahead

(ag_percept (percept_pobj e2) (percept_pname isa) (percept_pval event)) => e2 reprezinta un eveniment
(ag_percept (percept_pobj d1) (percept_pname isa) (percept_pval dog)) => d1 reprezinta un caine
(ag_percept (percept_pobj d2) (percept_pname isa) (percept_pval dog)) => d2 reprezinta un caine
(ag_percept (percept_pobj c1) (percept_pname isa) (percept_pval cat)) => c1 reprezinta o pisica
(ag_percept (percept_pobj e2) (percept_pname direction) (percept_pval ahead)) => directia evenimentului e2 este inainte
(ag_percept (percept_pobj d1) (percept_pname partof) (percept_pval e2)) => caine d1 apartine evenimentului e2
(ag_percept (percept_pobj d2) (percept_pname partof) (percept_pval e2)) => caine d2 apartine evenimentului e2
(ag_percept (percept_pobj c1) (percept_pname partof) (percept_pval e2)) => pisica c1 apartine evenimentului e2

Numele manevrei implementate

Numele manevrei implementate se poate gasi in fisierul maneuverValidityASK.clp. In cazul acestei aplicatii este definit un singur tip de manevra, si anume **move-on-maneuver**.

```
(deffacts AGENT::maneuvers-to-validate
   (ASK move-on-maneuver))
```

UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA

Detalii de implementare

Partile nou adaugate sau modificate se gasesc in fisierul DRIVER-AGENT.clp. Pentru fiecare scenariu de test a fost definit o noua regula. Aceste reguli sunt prezentate in continuare.

Regula **red_light_and_barrier** este folosita pentru primul scenariu de test, unde pe o banda avem o bariera si pe cealalta avem o trecere de pietoni cu semafor rosu. In functia de perceptii descrise, masina se ia o decizie facand un assert cu template-ul care contine directa de mers a manevrei move-on-maneuver.

```
; scene 1 - ahead red light, left barrier => ahead
(defrule AGENT::red_light_and_barrier
   (ag_percept (percept_pobj ?e1) (percept_pname isa) (percept_pval event))
   (ag_percept (percept_pobj ?sem) (percept_pname isa) (percept_pval semaphore))
   (ag_percept (percept_pobj ?sem) (percept_pname partof) (percept_pval ?e1))
   (ag_percept (percept_pobj ?sem) (percept_pname color) (percept_pval red))
   (ag_percept (percept_pobj ?e1) (percept_pname direction) (percept_pval ?dir))
   (ag_percept (percept_pobj ?e2) (percept_pname isa) (percept_pval event))
   (ag_percept (percept_pobj ?bar) (percept_pname isa) (percept_pval barrier))
   (ag_percept (percept_pobj ?bar) (percept_pname partof) (percept_pval ?e2))
=>
   (assert (ag_bel (bel_type moment) (bel_pname move-on-maneuver ) (bel_pval ?dir))))
```

De asemenea, avem inca doua reguli definite pentru cele doua scenarii:

- regula **young_old_people** pentru scenariul 2, unde pe o banda avem 3 tineri, iar pe cealalta banda avem 3 batrani
- regula **animal_people** definit pentru scenariul 3, unde pe o banda avem 3 animale si pe cealalta banda 3 persoane

```
   ;scene 2 - ahead youg people, left old people => left
   (defrule AGENT::young_old_people
   (ag_percept (percept_pobj ?e1) (percept_pname isa) (percept_pval event))
   (ag_percept (percept_pobj ?e1) (percept_pname direction) (percept_pval ?dir1))
   (ag_percept (percept_pobj ?ped1) (percept_pname isa) (percept_pval pedestrian))
   (ag_percept (percept_pobj ?ped1) (percept_pname partof) (percept_pval ?e1))
   (ag_percept (percept_pobj ?e2) (percept_pname isa) (percept_pval event))
   (ag_percept (percept_pobj ?e2) (percept_pname direction) (percept_pval ?dir2))
   (ag_percept (percept_pobj ?ped2) (percept_pname isa) (percept_pval pedestrian))
```

```
(ag_percept (percept_pobj ?ped2) (percept_pname partof) (percept_pval ?e2))
(test (and (not (eq ?e1 ?e2)) (not (eq (sum_of_ages ?e1) (sum_of_ages ?e2)))))
=>
(bind ?sum1 (sum_of_ages ?e1))
(bind ?sum2 (sum_of_ages ?e2))
(if (> ?sum1 ?sum2) then
   (assert (ag_bel (bel_type moment) (bel_pname move-on-maneuver ) (bel_pval ?dir1)))
else
   (assert (ag_bel (bel_type moment) (bel_pname move-on-maneuver ) (bel_pval ?dir2)))
))


;scene 3 - ahead animals, left people => ahead
(defrule AGENT::animal_people
(ag_percept (percept_pobj ?e1) (percept_pname isa) (percept_pval event))
(ag_percept (percept_pobj ?e1) (percept_pname direction) (percept_pval ?dir1))
(ag_percept (percept_pobj ?e2) (percept_pname isa) (percept_pval event))
(ag_percept (percept_pobj ?e2) (percept_pname direction) (percept_pval ?dir2))
(test (and (not (eq ?e1 ?e2)) (not (eq (nr_of_animals ?e1) (nr_of_pedestrians ?e2)))))
=>
(bind ?nr_a1 (nr_of_animals ?e1))
(bind ?nr_a2 (nr_of_animals ?e2))
(bind ?nr_p1 (nr_of_pedestrians ?e1))
(bind ?nr_p2 (nr_of_pedestrians ?e2))
 (if (and (> ?nr_a1 ?nr_a2) (< ?nr_p1 ?nr_p2)) then
   (assert (ag_bel (bel_type moment) (bel_pname move-on-maneuver ) (bel_pval ?dir1)))
else
   (if (and (> ?nr_a2 ?nr_a1) (< ?nr_p2 ?nr_p1)) then
      (assert (ag_bel (bel_type moment) (bel_pname move-on-maneuver ) (bel_pval ?dir2)
)))))
```

Pe langa regulile prezentate mai sus, sunt definite cateva functii, care sunt folosite in cadrul regulilor. Functia **sum_of_ages** calculeaza suma de varsta a persoanelor care sunt prezente la un eveniment, adica pietoni sau pasageri care sunt **“partof”** unui eveniment.

```
(deffunction AGENT::sum_of_ages (?e)
   (bind ?sum 0)
   (do-for-all-facts ((?fact ag_percept)) (and (eq ?fact:percept_pval ?e) (eq
?fact:percept_pname partof))
```

```
      (do-for-all-facts ((?selectedFact ag_percept)) (and (eq ?selectedFact:percept_pobj
?fact:percept_pobj) (eq ?selectedFact:percept_pname age))
         (bind ?sum (+ ?sum ?selectedFact:percept_pval))
      ))
   (return ?sum))
```

Functia **nr_of_animals** calculeaza numarul de animale care sunt prezente la un eveniment, acest eveniment este trimis ca si parametru la aceasta functie.

```
(deffunction AGENT::nr_of_animals (?e)
   (bind ?sum 0)
   (do-for-all-facts ((?fact ag_percept)) (and (eq ?fact:percept_pval ?e) (eq
?fact:percept_pname partof))
      (do-for-all-facts ((?selectedFact ag_percept)) (and (eq ?selectedFact:percept_pobj
?fact:percept_pobj) (eq ?selectedFact:percept_pname isa) (or (eq
?selectedFact:percept_pval dog) (eq ?selectedFact:percept_pval cat) )  )
         (bind ?sum (+ ?sum 1))))
   (return ?sum))
```

Ultima functie definita este **nr_of_pedestrians**, care are ca si scop calcularea numarului de pietoni care sunt implicati la un eveniment, trimis ca si parametru.

```
(deffunction AGENT::nr_of_pedestrians (?e)
   (bind ?sum 0)
   (do-for-all-facts ((?fact ag_percept)) (and (eq ?fact:percept_pval ?e) (eq
?fact:percept_pname partof))
      (do-for-all-facts ((?selectedFact ag_percept)) (and (eq ?selectedFact:percept_pobj
?fact:percept_pobj) (eq ?selectedFact:percept_pname isa) (eq ?selectedFact:percept_pval
pedestrian))
         (bind ?sum (+ ?sum 1))))
   (return ?sum))
```

Timpul de executie

Pentru masurarea timpului de executie se foloseste doua reguli care au fost definite in proiectul initial. Regula **time_start** salveaza timpul de inceput, iar regula **time_end** citeste timpul de inceput si calculeaza diferenta intre timpul de start si timpul de end, rezultand timpul de executie. Timpul de executie este calculata la nivelul fiecarui scenariu de test.

```
(defrule AGENT::time_start "Credit: Portik Annamaria & Stan Lavinia, CTI 2019"
  (declare (salience 92))
  (timp (valoare ?t))
=>
  (assert (tstart (time))))


(defrule AGENT::time_end
  (declare (salience -92))
  ?fsta <- (tstart ?time_start)
=>
  (bind ?time_end (time))
  (bind ?ex_time (- ?time_end ?time_start))
  (if (eq ?*ag-measure-time* TRUE) then (printout t "          <M> AGENT Decision time: "
?ex_time " sec." crlf))
  (retract ?fsta))
```

Rezultate obtinute

In cazul fiecarui scenariu de test este afisata directia manevrei move-on-maneuver si timpul de executie in secunde.

```
TRUE
CLIPS> (reset)
CLIPS> (run 100)
   PERCEPT-MANAGER: timp = 1
         AGENT move-on-maneuver ahead
            <M> AGENT Decision time: 0.0008828639984130 86 sec.
   PERCEPT-MANAGER: timp = 2
         AGENT move-on-maneuver left
            <M> AGENT Decision time: 0.0196309089660645 sec.
   PERCEPT-MANAGER: timp = 3
         AGENT move-on-maneuver ahead
            <M> AGENT Decision time: 0.0008888244628906 25 sec.
```

Aplicatia a fost rulata de 10 ori pe o masina virtuala, iar pe urma acestei rulari a fost calculata media si deviatia standard a timpului de executie in cazul fiecarui scenariu de test. Rezultatele obtinute dupa cele 10 rulari sunt prezentate in tabelul de mai jos.

Scenariu	Media	Deviatia standard
1	0.003081226348877	0.007056114709123
2	0.013755512237549	0.00871712448735
3	0.00214569568634	0.003821347638617

Limitele sistemului si posibilitati de imbunatatire

Limita aplicatiei este ca nu exista o decizie corecta din punct de vedere moral, deoarece pentru fiecare persoana, pentru fiecare comunitate sunt importante diferite criterii, de exemplu poate ca o persoana ar salva mai mult viata unui om batran, dar cealalta persoana ar salva pe viata unui tiner. Ambele decizii sunt acceptabile, nu putem spune ca e o decizie gresita.

Aplicatia poate fi imbunatatit prin adaugarea unor perceptii noi (de exemplu genul, ocupatia persoanelor, cat de sportiva este o persoana, etc.), prin declararea unor reguli noi sau prin declararea unor scenarii de test noi, astfel creand o aplicatie mai buna si complexa.

Referinte

[1] – T. H. Tran, C. C. Pham, T. P. Nguyen, T. T. Duong and J. W. Jeon, "Real-time traffic light detection using color density," 2016 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia), Seoul, 2016, pp. 1-4, doi: 10.1109/ICCE-Asia.2016.7804791.

https://www.researchgate.net/publication/312304505_Real-time_traffic_light_detection_using_color_density

[2] – Q. Mao, H. Sun, Y. Liu and R. Jia, "Mini-YOLOv3: Real-Time Object Detector for Embedded Applications," in IEEE Access, vol. 7, pp. 133529-133538, 2019, doi: 10.1109/ACCESS.2019.2941547.

https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8839032

[3] – M. SHARIF, K. AYUB, D. SATTAR, M. RAZA, „Real Time Face Detection”, Sindh Univ. Res. Jour. (Sci. Ser.) Vol.44 (4) 597-600 (2012)

https://www.researchgate.net/publication/257022696_Real_Time_Face_Detection

[4] – Jang-Hee Yoo, So-Hee Park, and Yongjin Lee, „Real-Time Age and Gender Estimation from Face Images”, ETRI (Electronics and Telecommunications Research Institute), Daejeon, S. Korea, Nov 2017

https://www.researchgate.net/publication/323868689_Real-Time_Age_and_Gender_Estimation_from_Face_Images

[5] – Damien Leflocha, Faouzi Alaya Cheikhb, Jon Yngve Hardebergb,

Pierre Goutona and Romain Picot-Clementea, „Real-time people counting system using a single video camera”, University of Burgundy, BP 47870, 21078 Dijon Cedex, France;Gjøvik University College, P.O. Box 191, N-2802 Gjøvik, Norway, March 2008

https://www.researchgate.net/publication/228437591_Real-time_people_counting_system_using_a_single_video_camera