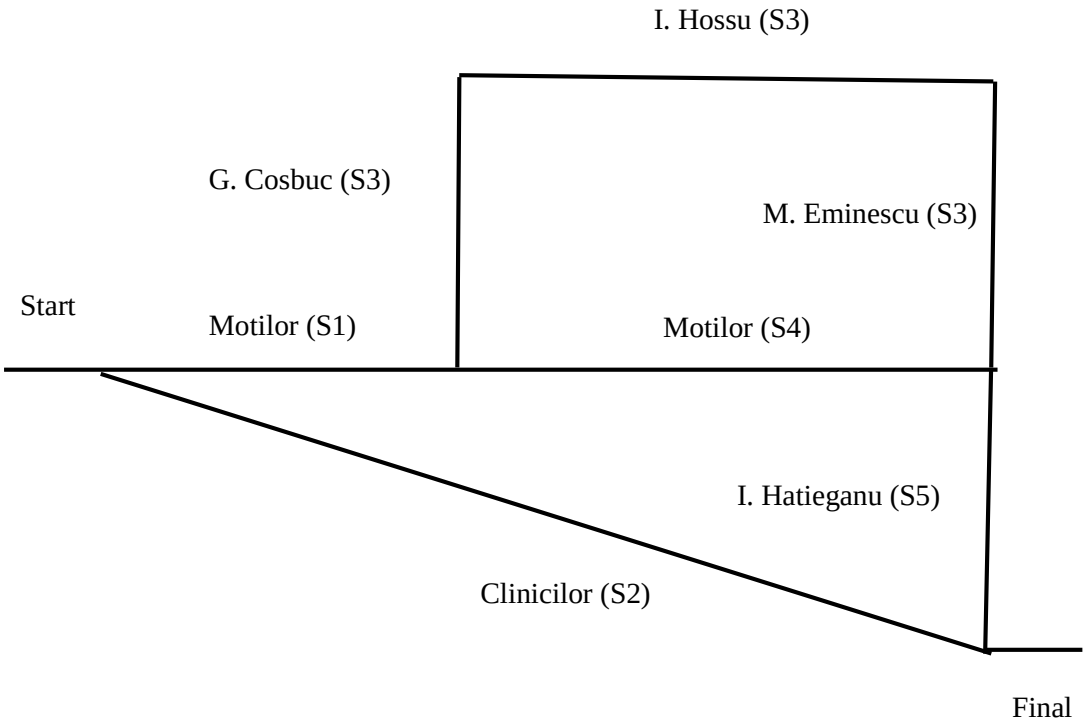
Model design pentru componenta IT inovativa de rutare inteligenta in traficul rutier urban, in context de Smart City - calcul ruta optima si analiza congestie

Aceasta componenta utilizeaza date statistice colectate offline pe parcursul unei lungi perioade de timp. Modelul de design a fost realizat ca masina de stare (masina cu stari finite) utilizand instrumentul PRISM versiunea 4.2. In cele ce urmeaza se presupune ca este cunoscuta durata medie de parcugere de catre un automobil a fiecarui segment de strada la fiecare interval de timp din fiecare zi, tinand cont si de factori aleatori, in particular de conditiile meteorologice. Se stie ca pe timp de ploaie traficul este mai aglomerat, fapt care poate reduce in mod semnificativ viteza medie pe fiecare segment de drum. De exemplu, se presupune cunoscuta durata de parcurgere a fiecarui segment de drum, in fiecare zi din saptamana, pe intervale de timp discrete pentru care viteza medie pe un segment de drum, si implicit durata medie de parcurgere a unui segment de drum se pot considera ca fiind stabile. In practica, aceste valori se pot modifica frecvent la orele de varf, de exemplu la fiecare 5 minute, sau se pot modifica mai rar in restul unei zile.

Durata medie de parcurgere a unui segment de trafic intr-un anumit interval poate fi calculata inmultind viteza medie cu lungimea segmentului, la care se adauga timpul de transfer de la segmentul curent la un segment conex; acest timp de transfer intre segmente conexe include timpul (eventual) de asteptare la semafor. Se presupune ca duratele medii sunt estimate in intervale de timp discrete. Pentru a ilustra utilitatea modelului de design realizat, codul PRISM prezentat mai jos are la baza o retea de strazi din orasul Cluj-Napoca. Modelul pemite utilizarea oricaror date concrete pentru duratele de arcugere din fiecare interval de timp. Duratele si intervalele de timp discrete utilizate in model pot fi alese in functie de necesitatile concrete.

Pentru ilustrare se considera un grup de strazi din Cluj-Napoca cu intersectiile lor, anume: Motilor, Clinicilor, George Cosbuc, Cardinal Iuliu Hossu, Mihai Eminescu, Iuliu Hatieganu. Aceste strazi sunt conectate cum se vede in figura de mai jos.



Se studiaza conexiunea intre un punct Start si un punct Final. Punctul Start este pe strada Motilor, anume la intersectia intre strada Motilor cu strada Clinicilor. Punctul Final este pe strada Clinicilor, anume la

intersectia intre strada Clinicilor cu strada Iuliu Hatieganu. Segmentele de strazi sunt etichetate cu numele: S1, S2, S3, S4 si S5. Astfel, strada Motilor este impartita in 2 segmente: S1 si S4. Deoarece strazile George Cosbuc, Cardinal Iuliu Hossu si Mihai Eminescu, nu au puncte de ramificare pe traseul intre Start si Final, acestea sunt toate etichetate cu numele S3. Strada Clinicilor este etichetata cu numele S2. Strada Iuliu Hatieganu este etichetata cu numele S5. In cele ce urmeaza ne vom referi la (segmentele de) strazi direct prin aceste nume: S1, S2, S3, S4 si S5.

Scenariul modelat este acela al unui automobil care se deplaseaza intre punctele Start si Final. Se calculeaza ruta optima si paramerii de (congestie in) trafic de interes in aceasta deplasare.

Se va utiliza instrumentul PRISM pentru a proiecta o masina de stare (cu un numar finit de stari) care descrie fluxul de trafic pe aceste strazi si permite elaborarea de interogari utilizand limbajul PRISM de specificare a proprietatilor. Se vor elabora interogari si experimente de analiza statistica pentru rutare optima in trafic, modelarea si analiza efectului congestiilor.

Se elaboreaza doua modele, ambele de tip Markov Decision Processes (MDP). Primul model este nedeterminist, si permite calculul tuturor rutelor intre punctele Start si Final. Nedeterminismul este specific modelelor de tip MDP. In cel de al doilea model, nedeterminismul este inlocuoit cu alegere probabilistica. In modelul initial nedeterminismul apare la alegerea intre segmentele S3 si S4 la iesirea sin segmentul S1, respectiv la alegerea intre segmentele S1 si S2 la iesirea din punctul de Start. Aceste alegeri nedeterminiiste sunt apoi inlocuite in cel de a doilea model cu alegeri probabilistice. In fiecare caz se considera ca exista informatii statistice care permit stabilirea acestor probabilitati, dar in experimentele PRISM aceste probabilitati pot sa ia valori din intervalul de valori reale [0,1] (in PRISM valor de tip double).

Modelul PRISM nedeterminist

S-a proiectat un model PRISM de tip MDP (Markov Decision Process). Pentru ilustrare s-au ales valori concrete pentru valorile de ceas si durata medie (timpul mediu) de parcurgere a fiecarui segment de drum in fiecare interval de timp. Codul PRISM care specifica aceste date este dat mai jos.

```
mdp

const int maxc = 24; // valoarea maxima a ceasului global
const int maxlc = 10; // valoarea maxima a ceasului local (pe fiecare segment de strada)

const int tmed11 = 1; // durata medie necesara parcurgerii segmentului S1 in intervalul de timp 1
const int tmed12 = 3; // durata medie necesara parcurgerii segmentului S1 in intervalul de timp 2
const int tmed13 = 1; // durata medie necesara parcurgerii segmentului S1 in intervalul de timp 3

const int tmed21 = 7; // durata medie necesara parcurgerii segmentului S2 in intervalul de timp 1
const int tmed22 = 1; // durata medie necesara parcurgerii segmentului S2 in intervalul de timp 2
const int tmed23 = 3; // durata medie necesara parcurgerii segmentului S2 in intervalul de timp 3

const int tmed31 = 1; // durata medie necesara parcurgerii segmentului S3 in intervalul de timp 1
const int tmed32 = 3; // durata medie necesara parcurgerii segmentului S3 in intervalul de timp 2
const int tmed33 = 2; // durata medie necesara parcurgerii segmentului S3 in intervalul de timp 3

const int tmed41 = 2; // durata medie necesara parcurgerii segmentului S4 in intervalul de timp 1
const int tmed42 = 3; // durata medie necesara parcurgerii segmentului S4 in intervalul de timp 2
const int tmed43 = 2; // durata medie necesara parcurgerii segmentului S4 in intervalul de timp 3

const int tmed51 = 1; // durata medie necesara parcurgerii segmentului S5 in intervalul de timp 1
const int tmed52 = 2; // durata medie necesara parcurgerii segmentului S5 in intervalul de timp 2
const int tmed53 = 3; // durata medie necesara parcurgerii segmentului S5 in intervalul de timp 3
```

Exista un ceas global, care masoara un timp de referinta pentru fiecare activitate. Exista si ceasuri locale care masoara activitatile pe fiecare segment de strada. Duratele activitatilor depind insa de valoarea ceasului global, asa cum s-a specificat mai sus. Pentru simplitate, timpul global s-a impartit in 24 valori discrete (si intervale corespunzatoare). Numai primele 12 unitati de timp sunt alocate deplasarii nnei masini intre punctele Start si Final. Dupa ce ceasul global trece de valoarea 12 sistemul este intr-o stare idle. Cele 12 unitati de timp sunt impartite in 3 intervale discrete, corespunzatoare unor viteze medii (respectiv durate medii de deplasare) in general diferite pe fiecare segment de strada considerat: c>=0 & c<4, c>=4 & c<8 si c>=8 & c<12, unde c este o variabila a modulului Ceas (global) care masoara timpul global de referinta.

```
module Ceas
  c : [0..maxc] init 0;
  [transfercontrol1] true -> (c'=(c+1>=maxc)?0:c+1);
  [transfercontrol2] true -> (c'=(c+1>=maxc)?0:c+1);
  [transfer2control] true -> (c'=(c+1>=maxc)?0:c+1);
  [transfer5control] true -> (c'=(c+1>=maxc)?0:c+1);
  [idle1] true -> (c'=(c+1>=maxc)?0:c+1);
  [idle2] true -> (c'=(c+1>=maxc)?0:c+1);
  [idle3] true -> (c'=(c+1>=maxc)?0:c+1);
  [idle4] true -> (c'=(c+1>=maxc)?0:c+1);
  [idle5] true -> (c'=(c+1>=maxc)?0:c+1);
  [run1] true -> (c'=(c+1>=maxc)?0:c+1);
  [run2] true -> (c'=(c+1>=maxc)?0:c+1);
  [run3] true -> (c'=(c+1>=maxc)?0:c+1);
  [run4] true -> (c'=(c+1>=maxc)?0:c+1);
  [run5] true -> (c'=(c+1>=maxc)?0:c+1);
  [transfer13] true -> (c'=(c+1>=maxc)?0:c+1);
  [transfer35] true -> (c'=(c+1>=maxc)?0:c+1);
  [transfer14] true -> (c'=(c+1>=maxc)?0:c+1);
  [transfer45] true -> (c'=(c+1>=maxc)?0:c+1);
endmodule
```

Ceasul global este incrementat sincron cu efecturarea fiecarei tranzitii discrete de la o stare la alta. In MDP toate aceste tranzitii se efectueaza in pasi discreti. Tranzitiile idle1,...,idle5 se efectueza numai dupa ce ceasul global a depasit valoarea 12. Celealte tranzitii corespund deplasarii masinii pe fiecare segment de drum (run1,...,run5), respectiv transferurilor de la un segment de drum la alt segment de drum.

```
module Control // start si intrare spre S1 sau S2, iesire din S2 sau S5 si punctul final
  run : [0..2] init 0; // 0 = start, 1=run, 2=final
  [transfercontrol1] run=0 -> (run'=1);
  [transfercontrol2] run=0 -> (run'=1);
  [transfer2control] run=1 -> (run'=2);
  [transfer5control] run=1 -> (run'=2);
endmodule
```

Modulul Control porneste si opreste deplasarea masinii. Pornirea este realizata in punctul Start. Oprirea este realizata in punctul Final. Corespunzator, variabila run are 3 stari: start, run si final. De remarcat ca la pornire, in punctul Start, modelul implica un comportament nedeterminist, comenzile PRISM etichetate cu [transfercontrol1] si [transfercontrol2] fiind ambele executabile in starea run=0. Acest nedeterminism ne va permite sa calculam toate rutele intre Start si Final, si sa identificam ruta optima (de timp minim), utilizand comanda PRISM filter. Intrarea in punctul Final este posibila dinspre segmentul S2 si dinspre segmentul S5, dar in acest caz nu exista nedeterminism, segmentele de strada

fiind mutual disjuncte/ Cu alte cuvinte, automobilul pentru care se face estimarea parametrilor de trafic se poate afla la un moment dat pe (cel mult) un singur segment de strada.

```
module S1 // intrare din Control, iesire spre S3 sau S4
  t1 : bool init false; // tranzitare segment
  s1 : bool init false; // false = idle, true = run
  lc1 : [1..maxlc] init 1;
  [idle1] !s1 & c>=12 -> true;
  [transfercontrol1] !s1 & c>=0 & c<4 -> (s1'=true) & (lc1'=tmed11);
  [transfercontrol1] !s1 & c>=4 & c<8 -> (s1'=true) & (lc1'=tmed12);
  [transfercontrol1] !s1 & c>=8 & c<12 -> (s1'=true) & (lc1'=tmed13);
  [run1] s1 & !s2 & !s3 & !s4 & !s5 & lc1>1 & c<12 -> (lc1'=lc1-1);
  [transfer13] s1 & !s2 & !s3 & !s4 & !s5 & lc1=1 & c<12 -> (s1'=false) & (t1'=true);
  [transfer14] s1 & !s2 & !s3 & !s4 & !s5 & lc1=1 & c<12 -> (s1'=false) & (t1'=true);
endmodule
```

Modulul S1 descrie comportamentul masinii (automobilului) pe segmentul de strada S1. Fiecare dintre modulele S1,..,S5 au o variabila locala corespunzatoare t1,...,t5, care indica la sfarsitul executiei intre punctele Start si Final daca respectivul segment de strada a fost traversat de automobil. s1 este o variabila booleana de stare, care poate fi false (idle) sau true (run). lc1 este ceasul local, care se initializeaza conforma datelor statice prezentate mai sus cu valori specifice pe fiecare interval de timp. Automobilul poate traversa un intreg segment intr-un interval de timp, sau poate traversa numai o parte dintr-un segment. Din nou, se remarca faptul ca segmentul S1 are o iesire aleasa nedeterminist catre segmentele S3 si S4.

Celelalte module segment de strada sunt proiectate in mod asemanator.

```
module S2 // intrare din Control, iesire spre Control
  t2 : bool init false; // tranzitare segment
  s2 : bool init false; // false = idle, true = run
  lc2 : [1..maxlc] init 1;
  [idle2] !s2 & c>=12 -> true;
  [transfercontrol2] !s2 & c>=0 & c<4 -> (s2'=true) & (lc2'=tmed21);
  [transfercontrol2] !s2 & c>=4 & c<8 -> (s2'=true) & (lc2'=tmed22);
  [transfercontrol2] !s2 & c>=8 & c<12 -> (s2'=true) & (lc2'=tmed23);
  [run2] !s1 & s2 & !s3 & !s4 & !s5 & lc2>1 & c<12  -> (lc2'=lc2-1);
  [transfer2control] !s1 & s2 & !s3 & !s4 & !s5 & lc2=1 & c<12 -> (s2'=false) & (t2'=true);
endmodule

module S3 // intrare din S1, iesire spre S5
  t3 : bool init false; // tranzitare segment
  s3 : bool init false; // false = idle, true = run
  lc3 : [1..maxlc] init 1;
  [idle3] !s3 & c>=12 -> true;
  [transfer13] !s3 & c>=0 & c<4 -> (s3'=true) & (lc3'=tmed31);
  [transfer13] !s3 & c>=4 & c<8 -> (s3'=true) & (lc3'=tmed32);
  [transfer13] !s3 & c>=8 & c<12 -> (s3'=true) & (lc3'=tmed33);
  [run3] !s1 & !s2 & s3 & !s4 & !s5 &  lc3>1 & c<12  -> (lc3'=lc3-1);
  [transfer35] !s1 & !s2 & s3 & !s4 & !s5 & lc3=1 & c<12  -> (s3'=false) & (t3'=true);
endmodule

module S4 // intrare din S1, iesire spre S5
  t4 : bool init false; // tranzitare segment
  s4 : bool init false; // false = idle, true = run
```

```
  lc4 : [1..maxlc] init 1;
  [idle4] !s4 & c>=12 -> true;
  [transfer14] !s4 & c>=0 & c<4 -> (s4'=true) & (lc4'=tmed41);
  [transfer14] !s4 & c>=4 & c<8 -> (s4'=true) & (lc4'=tmed42);
  [transfer14] !s4 & c>=8 & c<12 -> (s4'=true) & (lc4'=tmed43);
  [run4] !s1 & !s2 & !s3 & s4 & !s5 & lc4>1 & c<12  -> (lc4'=lc4-1);
  [transfer45] !s1 & !s2 & !s3 & s4 & !s5 & lc4=1 & c<12  -> (s4'=false) & (t4'=true);
endmodule

module S5 // intrare din S3 sau S4, iesire spre Control
  t5 : bool init false; // tranzitare segment
  s5 : bool init false; // false = idle, true = run
  lc5 : [1..maxlc] init 1;
  [idle5] !s5 & c>=12 -> true;
  [transfer35] !s5 & c>=0 & c<4 -> (s5'=true) & (lc5'=tmed51);
  [transfer35] !s5 & c>=4 & c<8 -> (s5'=true) & (lc5'=tmed52);
  [transfer35] !s5 & c>=8 & c<12 -> (s5'=true) & (lc5'=tmed53);
  [transfer45] !s5 & c>=0 & c<4 -> (s5'=true) & (lc5'=tmed51);
  [transfer45] !s5 & c>=4 & c<8 -> (s5'=true) & (lc5'=tmed52);
  [transfer45] !s5 & c>=8 & c<12 -> (s5'=true) & (lc5'=tmed53);
  [run5] !s1 & !s2 & !s3 & !s4 & s5 & lc5>1 & c<12  -> (lc5'=lc5-1);
  [transfer5control] !s1 & !s2 & !s3 & !s4 & s5 & lc5=1 & c<12  -> (s5'=false) & (t5'=true);
endmodule
```

Utilizand instrumentul PRISM, se constata ca acest sistem este un automat finit cu 17 stari diferite.

```
(c,run,t1,s1,lc1,t2,s2,lc2,t3,s3,lc3,t4,s4,lc4,t5,s5,lc5)
0:(0,0,false,false,1,false,false,1,false,false,1,false,false,1,false,false,1)
1:(1,1,false,false,1,false,true,7,false,false,1,false,false,1,false,false,1)
2:(1,1,false,true,1,false,false,1,false,false,1,false,false,1,false,false,1)
3:(2,1,false,false,1,false,true,6,false,false,1,false,false,1,false,false,1)
4:(2,1,true,false,1,false,false,1,false,false,1,false,true,2,false,false,1)
5:(2,1,true,false,1,false,false,1,false,true,1,false,false,1,false,false,1)
6:(3,1,false,false,1,false,true,5,false,false,1,false,false,1,false,false,1)
7:(3,1,true,false,1,false,false,1,false,false,1,false,true,1,false,false,1)
8:(3,1,true,false,1,false,false,1,true,false,1,false,false,1,false,true,1)
9:(4,1,false,false,1,false,true,4,false,false,1,false,false,1,false,false,1)
10:(4,1,true,false,1,false,false,1,false,false,1,true,false,1,false,true,1)
11:(4,2,true,false,1,false,false,1,true,false,1,false,false,1,true,false,1)
12:(5,1,false,false,1,false,true,3,false,false,1,false,false,1,false,false,1)
13:(5,2,true,false,1,false,false,1,false,false,1,true,false,1,true,false,1)
14:(6,1,false,false,1,false,true,2,false,false,1,false,false,1,false,false,1)
15:(7,1,false,false,1,false,true,1,false,false,1,false,false,1,false,false,1)
16:(8,2,false,false,1,true,false,1,false,false,1,false,false,1,false,false,1)
```

De asemenea, matricea de tranzitie pentru automatul MDP generata de instrumentul PRISM este data mai jos. Se remarca existenta celor 2 puncte de alegere nedeterminista, anume in starea 0 (prin tranzitiile etichetate cu transfercontrol1 si transfercontrol2) si in starea 2 (prin tranzitiile etichetate cu transfer13 si transfer14). Ambele au in acest model probabilitatea 1. Se stie ca in cazul unul automat MDP nedeterminismul nu se elimina. De aceea, interogarile se refera la valor minime sau maxime pentru probabilitati, respectiv pentru recompense sau costuri. Vom putea astfel idetifica timpul minim care trebuie sa fie petrecut in trafic de automobil. In iteratia urmatoare a modelului, cele 2 puncte de alegere nedeterminista vor fi eliminate, prin specificarea unor probabilitati de alegere a rutelor alternative, ca indicatie a congestiilor existente statistic in trafic. Se va vedea ca in modelul cu alegee probabilistica

nedeterminismul este eliminat, prin urmare valorile minime si maxime calculate de PRISM coincid in acel model, reprezentand medii ale valorilor statistice. Deocamdata, in modelul nedeterminist curent valorile minime si maxime sunt diferite.

```
17 19 19
0 0 1 1 transfercontrol2
0 1 2 1 transfercontrol1
1 0 3 1 run2
2 0 4 1 transfer14
2 1 5 1 transfer13
3 0 6 1 run2
4 0 7 1 run4
5 0 8 1 transfer35
6 0 9 1 run2
7 0 10 1 transfer45
8 0 11 1 transfer5control
9 0 12 1 run2
10 0 13 1 transfer5control
11 0 11 1
12 0 14 1 run2
13 0 13 1
14 0 15 1 run2
15 0 16 1 transfer2control
16 0 16 1
```

Pentru a calcula timpul petrecut in trafic de automobilul nostru, se proiecteaza urmatarea structura PRISM de recompensa, sau mai bine zis de cost:

```
rewards "timp_consumat"
   [transfercontrol1] true:1;
   [transfercontrol2] true:1;
   [transfer2control] true:1;
   [transfer5control] true:1;
   [run1] true:1;
   [run2] true:1;
   [run3] true:1;
   [run4] true:1;
   [run5] true:1;
   [transfer13] true:1;
   [transfer35] true:1;
   [transfer14] true:1;
   [transfer45] true:1;
endrewards
```

Aceasta structura de recompense ataseaza costul 1 tuturor tranzitiilor corespunzatoare deplasarii automobilului nostru in traffic. Deoarece nu toate tranzitiile s vor realiza la orice deplasare, in general, timpul consumat in traffic va fi diferit, in functie de ruta aleasa si de perioada in care se face deplasarea.

Utilizand limbajul de specificare a proprietatilor din PRISM se pot proiecta diferite interogari. Se poate calcula timpul minim si maxim necesar deplasarii automobilului nostru intre punctele Start si Final ca mai jos. Timpul minim este calculate prin urmatoarea interogare, care cumuleaza costul de timp pana

cand sistemul ajunge intr-o stare in care variabila run=2, adica pana cand automobilul ajunge in la punctul Final.

R{"timp_consumat"}min=? [F (run=2)]

Instrumentul PRISM raspunde la aceasta interogare astfel:

Result (minimum expected tim_consumat):

4.0

Prin urmare, timpul minim necesar automobilului pentru deplasarea intre punctele Start si Final este 4 (unitati de timp). Se oate calcula si timpul maxim necesar autombilului pentru o deplasare intre punctele Start si Final astfel:

R{"timp_consumat"}max=? [F (run=2)]

Instrumentul PRISM raspunde la aceasta interogare astfel:

Result (maximum expected tim_consumat):

8.0

Instrumentul PRIM poate fi utilizat si pentru generarea rutelor corespunzatoare tuturor deplasarilor posibile intre Start si Final, utilizand urmatoarea comanda filter:

filter (print,run=2)

In acest caz se returneaza toate starile in care run=2, anume 11, 13 si 16:

(c,run,t1,s1,lc1,t2,s2,lc2,t3,s3,lc3,t4,s4,lc4,t5,s5,lc5)
11:(4,2,true,false,1,false,false,1,true,false,1,false,false,1,true,false,1)
13:(5,2,true,false,1,false,false,1,false,false,1,true,false,1,true,false,1)
16:(8,2,false,false,1,true,false,1,false,false,1,false,false,1,false,false,1)

Starea finala 11 corespunde rutei de cost (timp) minim, adica c=4. Se remarca fatul ca aceata ruta consta din traversarea urmatoarelor segmente de strada: S1, S3 si S5. Ruta S1, S4, S5 este traversata in 5 unitati de timp (in final c=5). Ruta S1, S2 este traversata in 8 unitati de timp (in final c=8). Segmentele parcurse in fiecare deplasare sunt cele pentru care variabila corespunzatoare ti este true in starea finala. De exemplu, in starea 11, avem t1=true, t2=false, t3=true,t4=false,t5=true; deducem ca au fost traversate segmentele S1, S3 si S5. Ruta S!, S2 are cost maxim, deoarece in intervalul initial de timp avem tmed21 = 7 (adica durata medie necesara parcurgerii segmentului S2 in intervalul de timp 1).

Modelul PRISM cu tranzitii probabilistice

Se poate elimina nedeterminismul din modelul MDP dat mai sus prin stabilirea unor probabilitati pentru tranzitiile nedeterministe. Acestea pot fi determinate practice tot prin experimente si estimari statistice. Pentru simplitate, se considera in continuare ca probabilitatea de alegere intre rutele alternative conectate la acelasi sergment este fixa. In acest scop, se modifica modulele Control si S1 astfel. Se inlocuiesc alegerile nedeterministe descries mai sus cu alegeri probabilistice. Pentru a vedea cum influenteaza statistic aceste probabilitati timpii petrecuti in traffic constantele p_alegere12 si p_alegere34 sunt lasate nedefinite. Utilizand instrumental PRISM se vor putea efectua experimente pentru intervale intregi de valori ale constantelor p_alegere12 si p_alegere34. Practic, in acest mod se simuleaza diferite grade de congestive in traffic pe ruele alternative. Noul cod PRISM care specifica modulele Control si S1 este dat mai jos. In rest, codul PRISM ramane ca mai sus.

const double p_alegere12;

```
module Control // start si intrare spre S1 sau S2, iesire din S2 sau S5 si stop
  run : [0..2] init 0; // 0 = start, 1=run, 2=stop
  choice : [0..2] init 0;
  [] run=0 & choice=0 -> p_alegere12 : (choice'=1) + (1-p_alegere12):(choice'=2);
  [transfercontrol1] run=0 & choice=1 -> (run'=1);
  [transfercontrol2] run=0 & choice=2 -> (run'=1);
  [transfer2control] run=1 -> (run'=2);
  [transfer5control] run=1 -> (run'=2);
endmodule

const double p_alegere34;

module S1 // intrare din Control, iesire spre S3 sau S4
  t1 : bool init false;
  s1 : bool init false; // false = idle, true = run
  lc1 : [1..maxlc] init 1;
  choice1 : [0..2] init 0;
  [idle1] !s1 & c>=12 -> true;
  [transfercontrol1] !s1 & c>=0 & c<4 -> (s1'=true) & (lc1'=tmed11);
  [transfercontrol1] !s1 & c>=4 & c<8 -> (s1'=true) & (lc1'=tmed12);
  [transfercontrol1] !s1 & c>=8 & c<12 -> (s1'=true) & (lc1'=tmed13);
  [run1] s1 & !s2 & !s3 & !s4 & !s5 & lc1>1 & c<12 -> (lc1'=lc1-1);
  [] s1 & !s2 & !s3 & !s4 & !s5 & lc1=1 & c<12 & choice1=0 -> p_alegere34:(choice1'=1) + (1-
p_alegere34):(choice1'=2);
  [transfer13] s1 & !s2 & !s3 & !s4 & !s5 & lc1=1 & c<12 & choice1=1 -> (s1'=false) & (t1'=true);
  [transfer14] s1 & !s2 & !s3 & !s4 & !s5 & lc1=1 & c<12 & choice1=2 -> (s1'=false) & (t1'=true);
endmodule
```

Acum daca dorim sa vedem care este numarul de stari ale noului automat finit instrumental PRISM ne cere sa introducem valori concrete pentru constantele p_alegere12 si p_alegere34.

Fiind un automat de tip MDP, putem specifica numai valori minime sau maxime pentru probabilitati si costuri. Deoarece alegerea nedeterminista a fost inlocuita cu alegere probabilitica, este posibil sa se obtina rezultate identice pentru valorile minime si maxime. Se poate cere instrumentului PRISM sa calculeze probabilitatea ca automobilul sa ajunga intr-o stare finala (in care run=2) in maxim 4 unitati de timp:

```
Pmin=?[F (c<=4 & run=2)]
```

Pentru a calcula aceasta probabilitate PRISM va cere sa se introduca valori concrete pentru constantele p_alegere12 si p_alegere34. Daca setam p_alegere12=0.5 si p_alegere34=0.5, raspunsul PRISM este:

```
Result (minimum probability)
0.25
```

Daca setam p_alegere12=0.2 si p_alegere34=0.8, raspunsul PRISM este:

```
Result (minimum probability)
0.16
```

Motivul pentry care probabilitatea ca automobilul sa se deplaseze intre punctul de Start si punctul Final a scazut de la 0.25 la 0.16 este dat de faptul ca ruta optima este S1, S3, S5. Deoarece p_alegere12 a scazut de la 0.5 la 0.2, inseamna ca si probabilitatea de alegere a rutei optime S1, S3, S5 a scazut, iar probabilitatea de alegere a rutei mai defavorabile S1, S2 a crescut de la 0.5 la 0.8.

Alegerile nedterministe fiind inlocuite cu alegeri probabilistice, se obtin aceleasi valori si pentru probabilitatile maxime. De exemplu, setand p_alegere12=0.5 si p_alegere34=0.5, raspunsul PRISM la interogarea

Pmin=?[F (c<=4 & run=2)]

va fi

Result (maximum probability)
0.25

Se poate declara o constanta de tip int T, si se poate cere instrumentului PRISM sa faca calculele parametrizat cu aceasta onstanta.

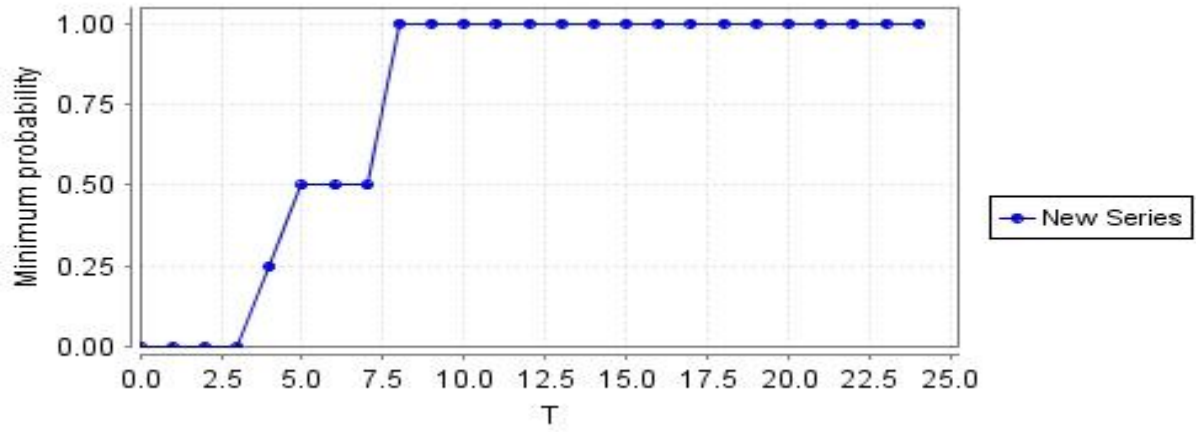
const int T;

Acum se poate formula urmatoarele interogari, si se pot efectua experimente PRISM pe baza unor asemenea interogari: Pmin=?[F (c<=T & run=2)], sau Pmax=?[F (c<=T & run=2)].

De exemplu, setand p_alegere12=0.5 si p_alegere34=0.5, instrumental PRISM permite efectuarea urmatorului experiment, in care se genereaza valoarea probabilitatii minime pentru toate valorile lui T intre 0 si 24. Astfel, la interogarea

Pmin=?[F (c<=T & run=2)]

Se genereaza urmatorul graphic de comportament, care arata probabilitatea minima ca automobilul sa ajunga din punctul de Start in punctul Final in T unitati de timp, unde T ia vaori intre 0 si 24



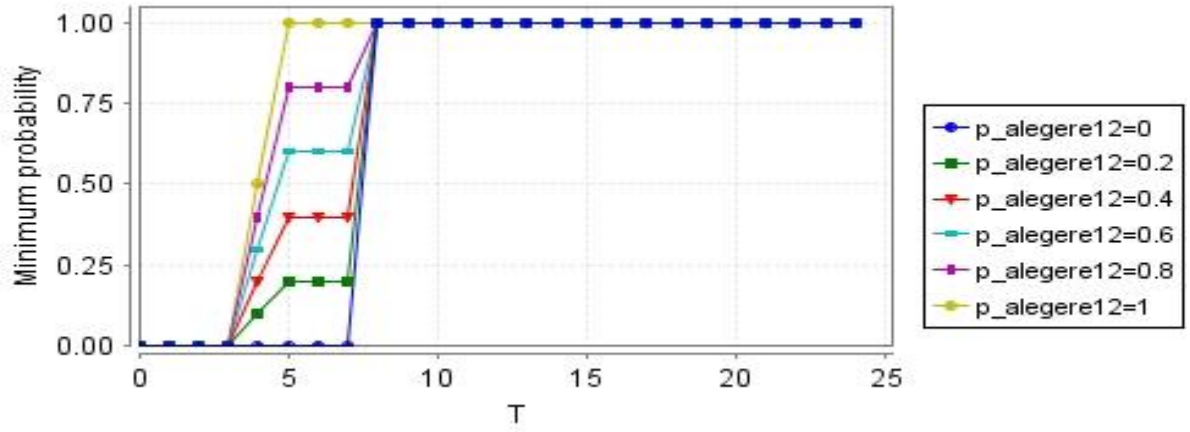
Se remarca faptul ca statistic, automobilul va parurge traseul intre punctul de Start sic el Final cu o probabilitate de minima 0.25 in 4 unitati de timp, cu o prababilitate minima de 0.5 va parcurge traseul in 5 pana la 7 unitati de timp, iar cu probabilitatea minima 1 (adica cu certitudine) va parcurge traseul in cel mult de 8 unitati de timp. Deoarece PRISM genereaza acelasi graphic si pentru interogarea

Pmax=?[F (c<=T & run=2)]

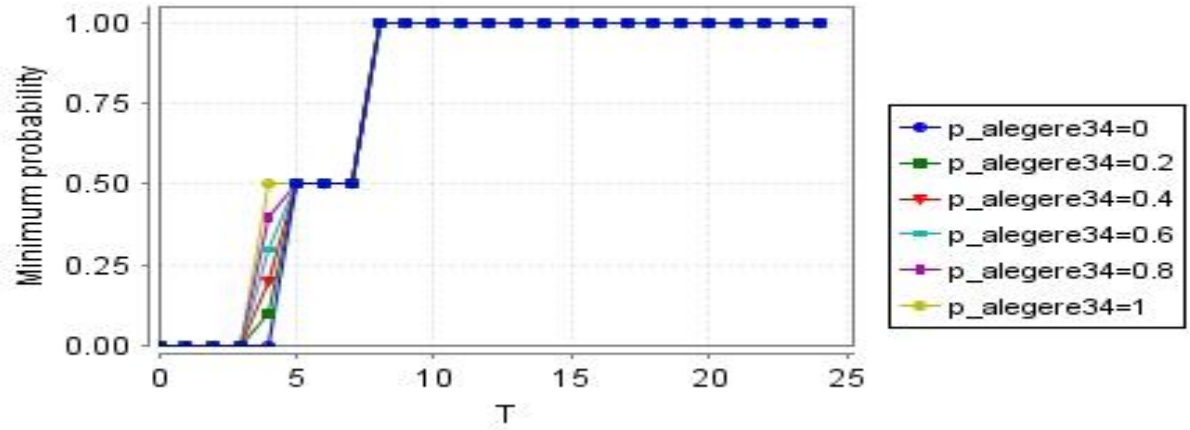
deducem ca acestea sunt probabilitati exacte.

Se poate cere instrumentului PRISM sa genereze un grafic mai complex in care sa tina cont de valori diferite ale probabilitatilor de alegere p_alegere12=0.5 si p_alegere34=0.5. Asemenea experimente vor vizualiza efectul diferitelor niveluri de congestie in traffic in calculul timpului minim si maxim (sau mediu, cand valorile minim si maxim coincid) necesar deplasarii automobilului nostru in traffic. Astfel,

fixand p_alegere34=0.5, si lasand p_alegere12 sa ia valori in intervalul [0,1] cu pasul 0.2, iar T lunad valori in intervalul [0,24] se obtine urmatorul graphic de comportament:



Daca acum fixam p_alegere12=0.5 si lasam si constanta p_alegere34 sa ia valori in intervalul [0,1] cu pasul 0.2 se obtine urmatorul grapfic de comportament:



Constantele p_alegere12 si p_alegere34 indica probabilitatile (obtinute tot experimental, prin observatii de lunga durata) ca la o anumita intersectie, un auomobil sa aleaga o ruta sau o alta ruta. In acest mod se poate monitoriza gradul dinamic de congestie in trafic pe fiecare ruta, si prin urmare inarcarea respectivului segment de drum sau a respectivei rute. Asemenea date statistice sunt foarte utile pentru a se cunoaste atat gradul de uzura pe rute diferite, cat si pentru a se obtine informatii statistice utile pentru soferi in alegrea rutelor algernative, prercum si in evidentierea necesitatii construirii de drumuri alternative, exact acolo unde gradul de congestie este maxim, statistic vorbind.

Daca utilizam acum operatorul PRISM pentru calcul recompense sau costuri, se obtin medii statistice. De exemplu, setand p_alegere12=0.1 si p_alegere34=0.7, la interogarea

R{"timp_consumat"}min=? [F (run=2)]

PRISM va raspunde cu o valoare medie statistica apropiata de timpul maxim 8, deoarece se aleg cu mare probabilitate rutle mai putin favorabil (anume S1, S2, sau, eventual, S1, S4, S5):

Result (minimum expected timp_consumat)
7.63