Intersectie cu 4 benzi

Golban Ovidiu, Cengher Andrei

November 28, 2021

# 1 Introducere

Intersect,ia curena a lua nas,tere ca dezvoltare a uneia triviale, in forma de cruce, cu o singura banda pentru a o lua inainte/stanga/dreapta. Apoi o versiune a procesului de dezvoltare ne-a dus in punctul in care aveam o intersectie cu benzi separate de pe care se mergea inainte sau inapoi(se venea de pe sensul opus) sau se facea viraj la stanga.

In cele din urma prezentam intersect,ia finala cu 4 benzi separate pentru mersul inainte,ˆınapoi(sensul opus), viraj stanga, respectiv viraj dreapta.

# 2 Intersectia cu trecere de pietoni

**2.1 Cum a inceput totul?**

De la o intersectie simpla pe care poate o vedem zi de zi, ne-am gandit sa cream ceva mai complex, ca urmare, am creat o intersectie ca cea din figura de mai jos[Figura 1], pe care o mai dezvolta ulterior in proiect.

Codificarea stărilor:

bi – se trece pe banda i

pi – trecerea i este activată

## 2.2 Transformarea limbajului natural in logica propozitionala pentru intersectia intermediara

*%circula masinile de pe benzile paralele(aceeasi directie, sens opus)*

(b1 & b5) | (b2 & b6) | (b3 & b7) | (b4 & b8).

*%benzile care nu pot functiona cand o anumita banda e activa*

b1->-b3.

b1->-b6.

b1->-b7.

b1->-b8.

b2->-b4.

b2->-b5.

b2->-b7.

b2->-b8.

b3->-b5.

b3->-b6.

b3->-b8.

b4->-b5.

b4->-b6.

b4->-b7.

b5->-b7.

b6->-b7.

b6->-b8.

## 2.3 Spatiul starilor pentru intersectia intermediara

interpretation( 2, [number = 1,seconds = 0], [

relation(b1, [0]),

relation(b2, [0]),

relation(b3, [0]),

relation(b4, [1]),

relation(b5, [0]),

relation(b6, [0]),

relation(b7, [0]),

relation(b8, [1])]).

interpretation( 2, [number = 2,seconds = 0], [

relation(b1, [0]),

relation(b2, [0]),

relation(b3, [1]),

relation(b4, [0]),

relation(b5, [0]),

relation(b6, [0]),

relation(b7, [1]),

relation(b8, [0])]).

interpretation( 2, [number = 3,seconds = 0], [

relation(b1, [0]),

relation(b2, [1]),

relation(b3, [0]),

relation(b4, [0]),

relation(b5, [0]),

relation(b6, [1]),

relation(b7, [0]),

relation(b8, [0])]).

interpretation( 2, [number = 4,seconds = 0], [

relation(b1, [1]),

relation(b2, [0]),

relation(b3, [0]),

relation(b4, [0]),

relation(b5, [1]),

relation(b6, [0]),

relation(b7, [0]),

relation(b8, [0])]).

interpretation( 3, [number = 5,seconds = 0], [

relation(b1, [0]),

relation(b2, [0]),

relation(b3, [0]),

relation(b4, [1]),

relation(b5, [0]),

relation(b6, [0]),

relation(b7, [0]),

relation(b8, [1])]).

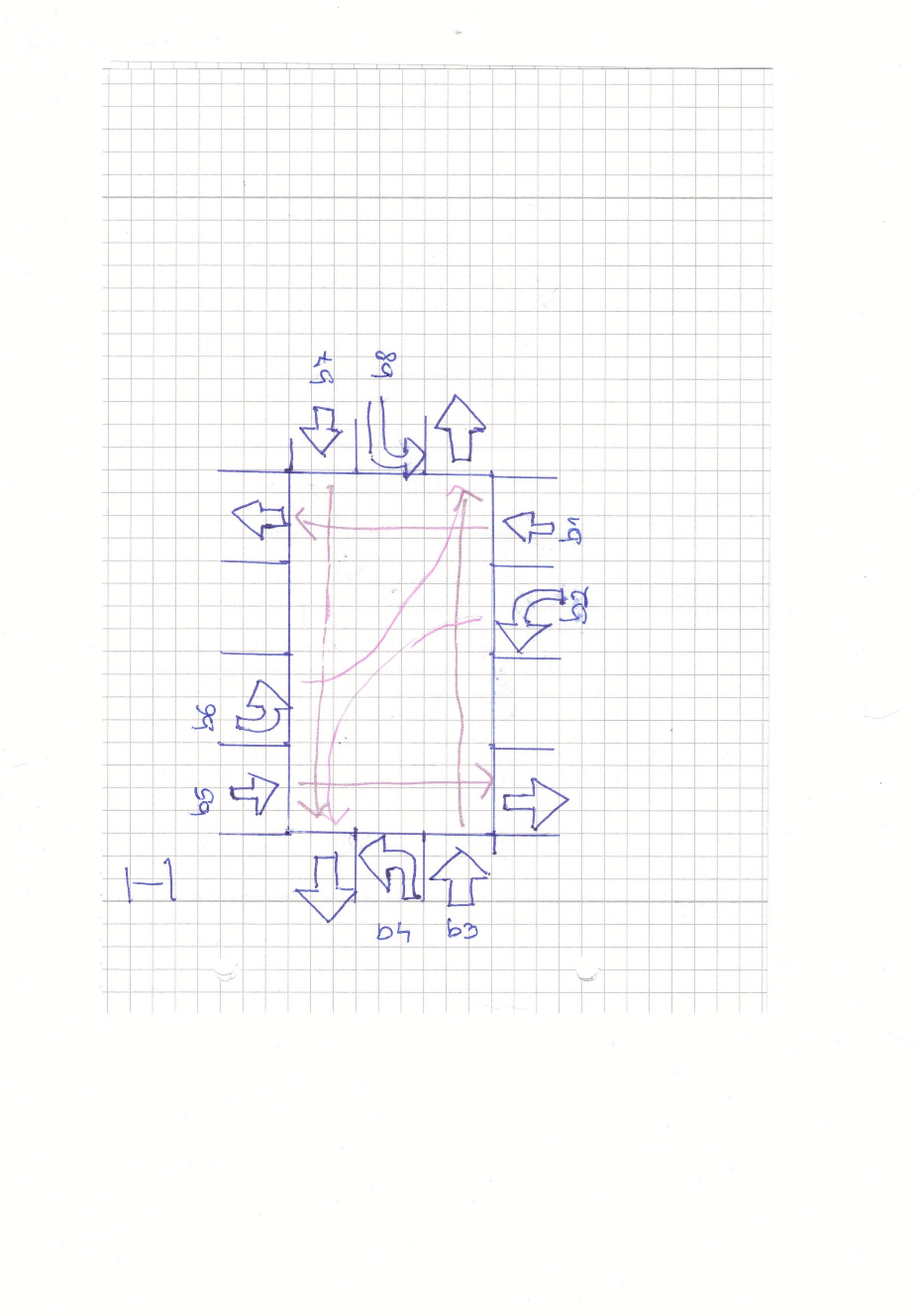


Figure 1: Intersectie intermediara.

## 2.4 Demonstratia din Prover9

%%in cazul intersectiei cu 3 benzi, dem ca daca b1 e activa, %%atunci e activa si b5

set(ignore\_option\_dependencies). % GUI handles dependencies

if(Prover9). % Options for Prover9

assign(max\_seconds, 60).

end\_if.

if(Mace4). % Options for Mace4

assign(max\_seconds, 60).

end\_if.

formulas(assumptions).

b1.

%circula masinile de pe benzile paralele(aceeasi directie, sens opus)

(b1 & b5) | (b2 & b6) | (b3 & b7) | (b4 & b8).

b1->-b3.

b1->-b6.

b1->-b7.

b1->-b8.

end\_of\_list.

formulas(goals).

b5.

end\_of\_list.

============================== PROOF =================================

% -------- Comments from original proof --------

% Proof 1 at 0.03 (+ 0.00) seconds.

% Length of proof is 15.

% Level of proof is 3.

% Maximum clause weight is 4.

% Given clauses 0.

1 b1 & b5 | b2 & b6 | b3 & b7 | b4 & b8 # label(non\_clause). [assumption].

2 b1 -> -b3 # label(non\_clause). [assumption].

3 b1 -> -b6 # label(non\_clause). [assumption].

5 b1 -> -b8 # label(non\_clause). [assumption].

6 b5 # label(non\_clause) # label(goal). [goal].

7 b1. [assumption].

13 b5 | b6 | b3 | b8. [clausify(1)].

16 -b1 | -b3. [clausify(2)].

17 -b3. [copy(16),unit\_del(a,7)].

18 -b1 | -b6. [clausify(3)].

19 -b6. [copy(18),unit\_del(a,7)].

22 -b1 | -b8. [clausify(5)].

23 -b8. [copy(22),unit\_del(a,7)].

24 -b5. [deny(6)].

25 $F. [back\_unit\_del(13),unit\_del(a,24),unit\_del(b,19),unit\_del(c,17),unit\_del(d,23)].

============================== end of proof ==========================

%%dem ca daca b2 e activa, atunci e si b6

set(ignore\_option\_dependencies). % GUI handles dependencies

if(Prover9). % Options for Prover9

assign(max\_seconds, 60).

end\_if.

if(Mace4). % Options for Mace4

assign(max\_seconds, 60).

end\_if.

formulas(assumptions).

b2.

%circula masinile de pe benzile paralele(aceeasi directie, sens opus)

(b1 & b5) | (b2 & b6) | (b3 & b7) | (b4 & b8).

b2->-b4.

b2->-b5.

b2->-b7.

b2->-b8.

end\_of\_list.

formulas(goals).

b6.

end\_of\_list.

============================== PROOF =================================

% -------- Comments from original proof --------

% Proof 1 at 0.01 (+ 0.00) seconds.

% Length of proof is 15.

% Level of proof is 3.

% Maximum clause weight is 4.

% Given clauses 0.

1 b1 & b5 | b2 & b6 | b3 & b7 | b4 & b8 # label(non\_clause). [assumption].

2 b2 -> -b4 # label(non\_clause). [assumption].

3 b2 -> -b5 # label(non\_clause). [assumption].

4 b2 -> -b7 # label(non\_clause). [assumption].

6 b6 # label(non\_clause) # label(goal). [goal].

7 b2. [assumption].

14 b5 | b6 | b7 | b4. [clausify(1)].

16 -b2 | -b4. [clausify(2)].

17 -b4. [copy(16),unit\_del(a,7)].

18 -b2 | -b5. [clausify(3)].

19 -b5. [copy(18),unit\_del(a,7)].

20 -b2 | -b7. [clausify(4)].

21 -b7. [copy(20),unit\_del(a,7)].

24 -b6. [deny(6)].

25 $F. [back\_unit\_del(14),unit\_del(a,19),unit\_del(b,24),unit\_del(c,21),unit\_del(d,17)].

============================== end of proof ==========================

## 2.5 Varianta finala de intersect,ie

Fata de cea prezentata anterior mai dispune de o banda pnetru virajul la dreapta, asa cum este prezentat in imaginea de mai jos[Figura 2] .

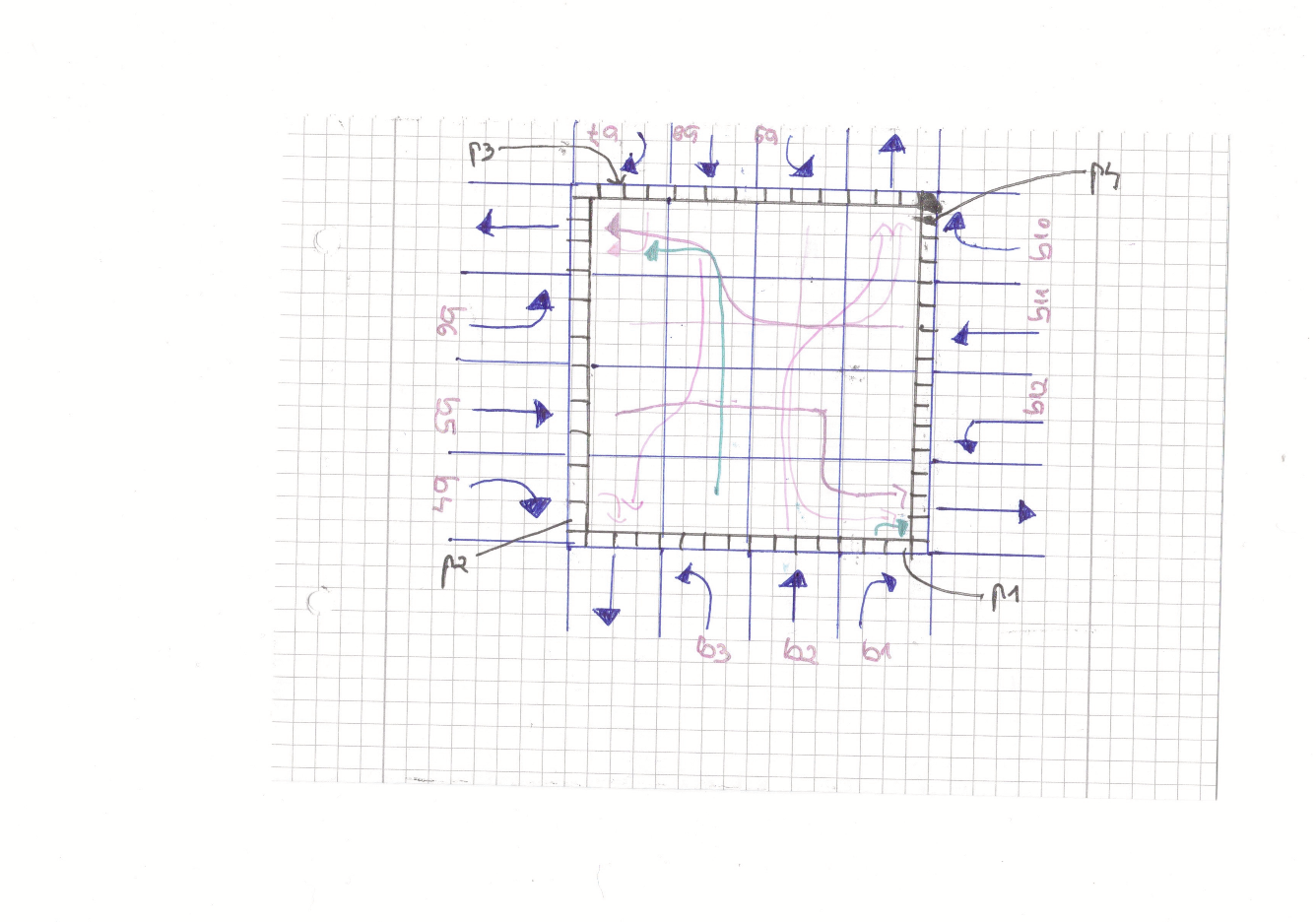


Figure 2: Intersectie finala.

## 2.6 Transformarea limbajului natural in logica propozitionala pentru intersectia finala

*%functionarea concomitenta a benzilor*

b1 & b4 & b7 & b10 | b2 & b8 | b3 & b9 | b5 & b11 | b6 & b12.

*%benzile care nu pot functiona cand o anumita banda e activa*

b1->-b5.

b1->-b9.

b2->-b5.

b2->-b6.

b2->-b9.

b2->-b10.

b2->-b11.

b2->-b12.

b3->-b5.

b3->-b6.

b3->-b7.

b3->-b8.

b3->-b11.

b3->-b12.

b4->-b8.

b4->-b12.

b5->-b8.

b5->-b9.

b5->-b12.

b6->-b8.

b6->-b9.

b6->-b10.

b6->-b11.

b7->-b11.

b8->-b10.

b8->-b11.

b8->-b12.

b9->-b11.

b9->-b12.

*%functionarea trecerilor de pietoni*

(p1&p3)|(p2&p4).

b5->p1.

b2->p2.

*%relatia dintre treceri si benzi*

*%cand e activa trecerea p -> nu pot functiona benzile*

p1->-b1.

p1->-b2.

p1->-b3.

p1->-b4.

p1->-b8.

p1->-b12.

p2->-b3.

p2->-b4.

p2->-b5.

p2->-b6.

p2->-b7.

p2->-b11.

p3->-b2.

p3->-b6.

p3->-b7.

p3->-b8.

p3->-b9.

p3->-b10.

p4->-b1.

p4->-b5.

p4->-b9.

p4->-b10.

p4->-b11.

p4->-b12.

*%cazul in care nu trec pietoni*

p1 & p3 | p2 & p4 | -p1&-p2&-p3&-p4.

## 

## 2.8 Spatiul starilor generat de Mace4

interpretation( 2, [number = 1,seconds = 0], [

relation(b1, [0]),

relation(b10, [0]),

relation(b11, [0]),

relation(b12, [0]),

relation(b2, [1]),

relation(b3, [0]),

relation(b4, [0]),

relation(b5, [0]),

relation(b6, [0]),

relation(b7, [0]),

relation(b8, [1]),

relation(b9, [0]),

relation(p1, [0]),

relation(p2, [1]),

relation(p3, [0]),

relation(p4, [1])]).

interpretation( 3, [number = 2,seconds = 0], [

relation(b1, [0]),

relation(b10, [0]),

relation(b11, [0]),

relation(b12, [0]),

relation(b2, [1]),

relation(b3, [0]),

relation(b4, [0]),

relation(b5, [0]),

relation(b6, [0]),

relation(b7, [0]),

relation(b8, [1]),

relation(b9, [0]),

relation(p1, [0]),

relation(p2, [1]),

relation(p3, [0]),

relation(p4, [1])]).

interpretation( 4, [number = 3,seconds = 0], [

relation(b1, [0]),

relation(b10, [0]),

relation(b11, [0]),

relation(b12, [0]),

relation(b2, [1]),

relation(b3, [0]),

relation(b4, [0]),

relation(b5, [0]),

relation(b6, [0]),

relation(b7, [0]),

relation(b8, [1]),

relation(b9, [0]),

relation(p1, [0]),

relation(p2, [1]),

relation(p3, [0]),

relation(p4, [1])]).

interpretation( 5, [number = 4,seconds = 0], [

relation(b1, [0]),

relation(b10, [0]),

relation(b11, [0]),

relation(b12, [0]),

relation(b2, [1]),

relation(b3, [0]),

relation(b4, [0]),

relation(b5, [0]),

relation(b6, [0]),

relation(b7, [0]),

relation(b8, [1]),

relation(b9, [0]),

relation(p1, [0]),

relation(p2, [1]),

relation(p3, [0]),

relation(p4, [1])]).

interpretation( 6, [number = 5,seconds = 0], [

relation(b1, [0]),

relation(b10, [0]),

relation(b11, [0]),

relation(b12, [0]),

relation(b2, [1]),

relation(b3, [0]),

relation(b4, [0]),

relation(b5, [0]),

relation(b6, [0]),

relation(b7, [0]),

relation(b8, [1]),

relation(b9, [0]),

relation(p1, [0]),

relation(p2, [1]),

relation(p3, [0]),

relation(p4, [1])]).

interpretation( 7, [number = 6,seconds = 0], [

relation(b1, [0]),

relation(b10, [0]),

relation(b11, [0]),

relation(b12, [0]),

relation(b2, [1]),

relation(b3, [0]),

relation(b4, [0]),

relation(b5, [0]),

relation(b6, [0]),

relation(b7, [0]),

relation(b8, [1]),

relation(b9, [0]),

relation(p1, [0]),

relation(p2, [1]),

relation(p3, [0]),

relation(p4, [1])]).

Asa cum ne asteptam obtinem 5 cazuri posibile.

## 2.9 Demonstrația în Prover9

%%intersectia finala, dem ca daca p1 e activa, atunci b1 nu e activa

*p1.*

*%functionarea trecerilor de pietoni*

(p1&p3)|(p2&p4).

*%functionarea concomitenta a benzilor*

b1 & b4 & b7 & b10 | b2 & b8 | b3 & b9 | b5 & b11 | b6 & b12.

*%relatia dintre treceri si benzi*

*%cand e activa trecerea p -> nu pot functiona benzile*

p1->-b1.

p1->-b2.

p1->-b3.

p1->-b4.

p1->-b8.

p1->-b12.

============================== PROOF =================================

% -------- Comments from original proof --------

% Proof 1 at 0.03 (+ 0.01) seconds.

% Length of proof is 7.

% Level of proof is 3.

% Maximum clause weight is 1.

% Given clauses 0.

3 p1 -> -b1 # label(non\_clause). [assumption].

9 -b1 # label(non\_clause) # label(goal). [goal].

10 p1. [assumption].

77 -p1 | -b1. [clausify(3)].

78 -b1. [copy(77),unit\_del(a,10)].

89 b1. [deny(9)].

90 $F. [copy(89),unit\_del(a,78)].

============================== end of proof ==========================

%%intersectia finala, dem ca daca p2 e activa, atunci b7 nu e activa

set(ignore\_option\_dependencies). % GUI handles dependencies

if(Prover9). % Options for Prover9

assign(max\_seconds, 60).

end\_if.

if(Mace4). % Options for Mace4

assign(max\_seconds, 60).

end\_if.

formulas(assumptions).

p2.

%functionarea trecerilor de pietoni

(p1&p3)|(p2&p4).

%functionarea concomitenta a benzilor

b1 & b4 & b7 & b10 | b2 & b8 | b3 & b9 | b5 & b11 | b6 & b12.

%relatia dintre treceri si benzi

%cand e activa trecerea p -> nu pot functiona benzile

p2->-b3.

p2->-b4.

p2->-b5.

p2->-b6.

p2->-b7.

p2->-b11.

end\_of\_list.

formulas(goals).

-b7.

end\_of\_list.

============================== PROOF =================================

% -------- Comments from original proof --------

% Proof 1 at 0.01 (+ 0.01) seconds.

% Length of proof is 7.

% Level of proof is 3.

% Maximum clause weight is 1.

% Given clauses 0.

7 p2 -> -b7 # label(non\_clause). [assumption].

9 -b7 # label(non\_clause) # label(goal). [goal].

10 p2. [assumption].

85 -p2 | -b7. [clausify(7)].

86 -b7. [copy(85),unit\_del(a,10)].

89 b7. [deny(9)].

90 $F. [copy(89),unit\_del(a,86)].

============================== end of proof ==========================

# References

1. Introduction to Artificial Intelligence by Adrian Groza, Radu Razvan Slavescu and Anca Marginean