Critical Sections Lab Assignment

Diaconescu Bogdan-Florin
CR 3.2 A, Anul 3

November 2019

1 Problem statement

Trebuie sa implementam in Promela un algoritm pentru numararea concurenta si sa analizam ce valoare poate lua n la sfarsitul executiei programului.

2 Implementation/Solution

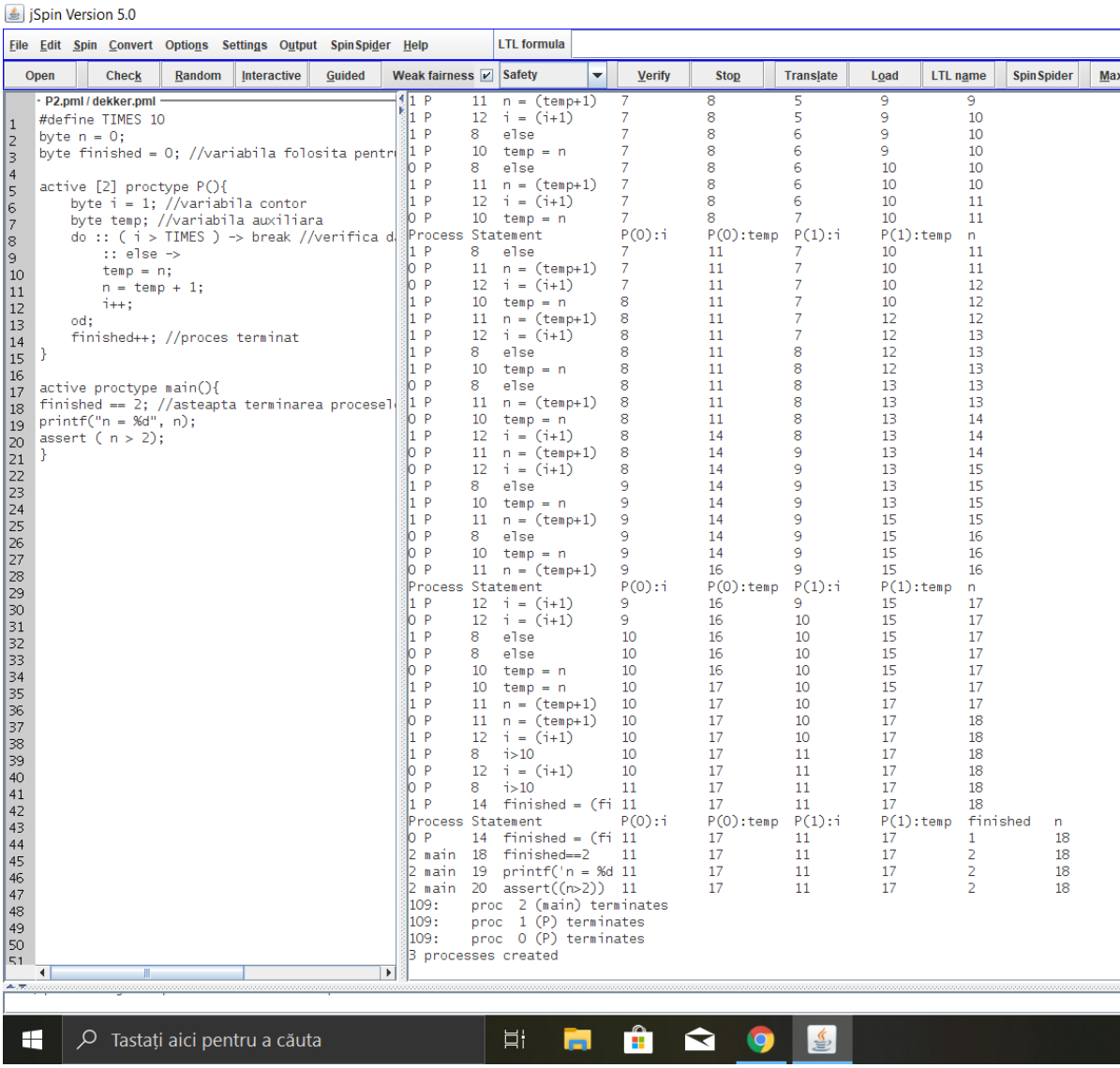
Am solosit un MACRO TIMES 10 deoarece doresc ca fiecare proces sa incrementeze variabila n de 10 ori. Am definit 2 variabile "n" si "finished" de tipul byte, echivalent cu uchar in C, ce pot lua valori cuprinde intre 1 si 255. Variabila n este cea care trebuie incrementata, iar variabila finished este folosita pentru a marca terminarea procesului(cand se termina un proces variabila finished se incrementeaza). Acestea sunt variabile globale ce pot fi utilizate in toate procesele.

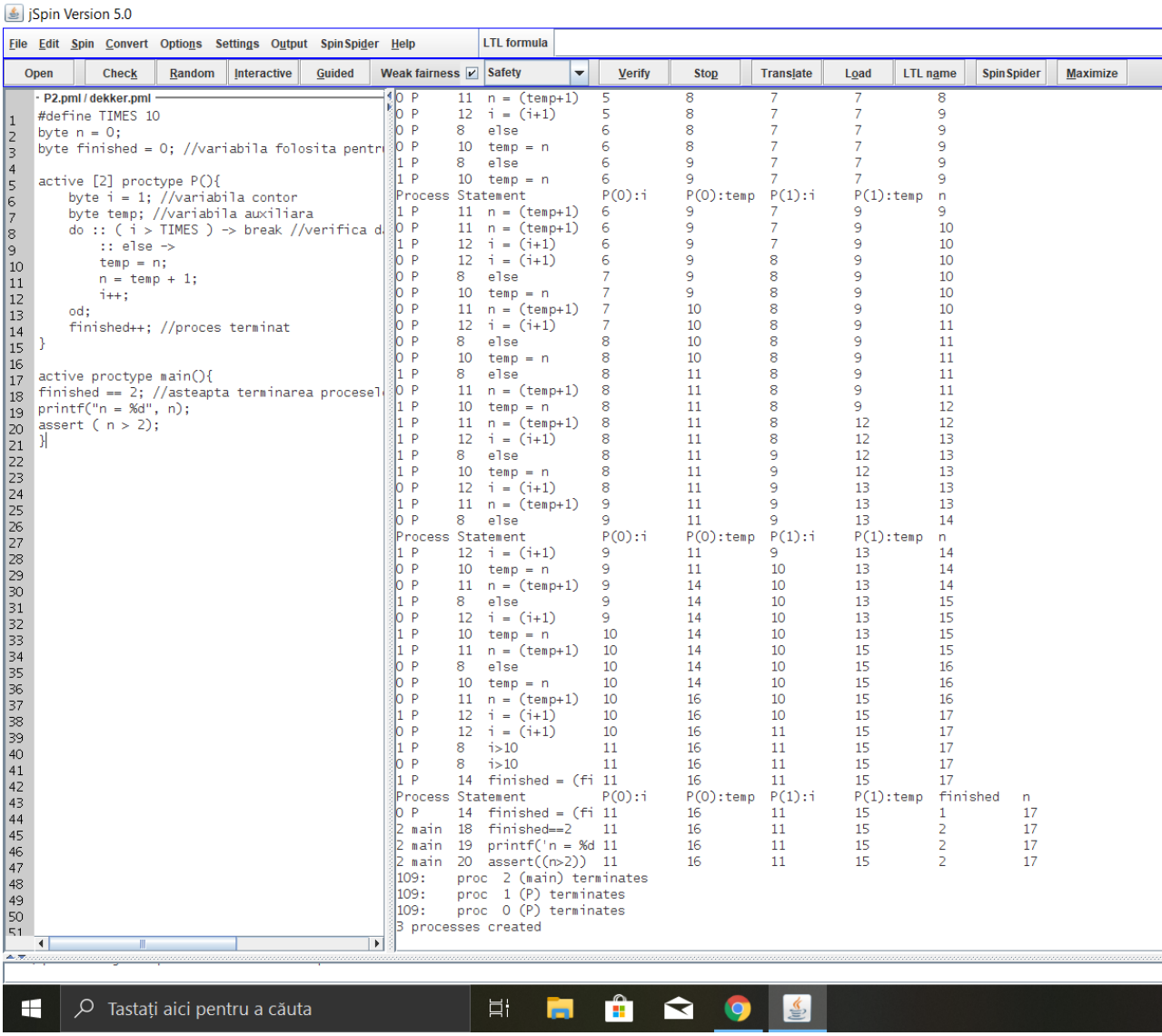
La linia 5 am creat 2 procese P in interiorul carora am declarat 2 variabile de tipul byte "i" si "temp" utilizate la incrementarea numarului n. La linia 8 am folosit o bucla "do" urmata de 2 conditii. Daca i este mai mare decat TIMES inseamna ca numarul s-a incrementat deja de 10 ori si se iese din bucla. In caz contrat, se efectueaza incrementarea numarului. Dupa iesirea din bucla se incrementeaza si variabila finished.

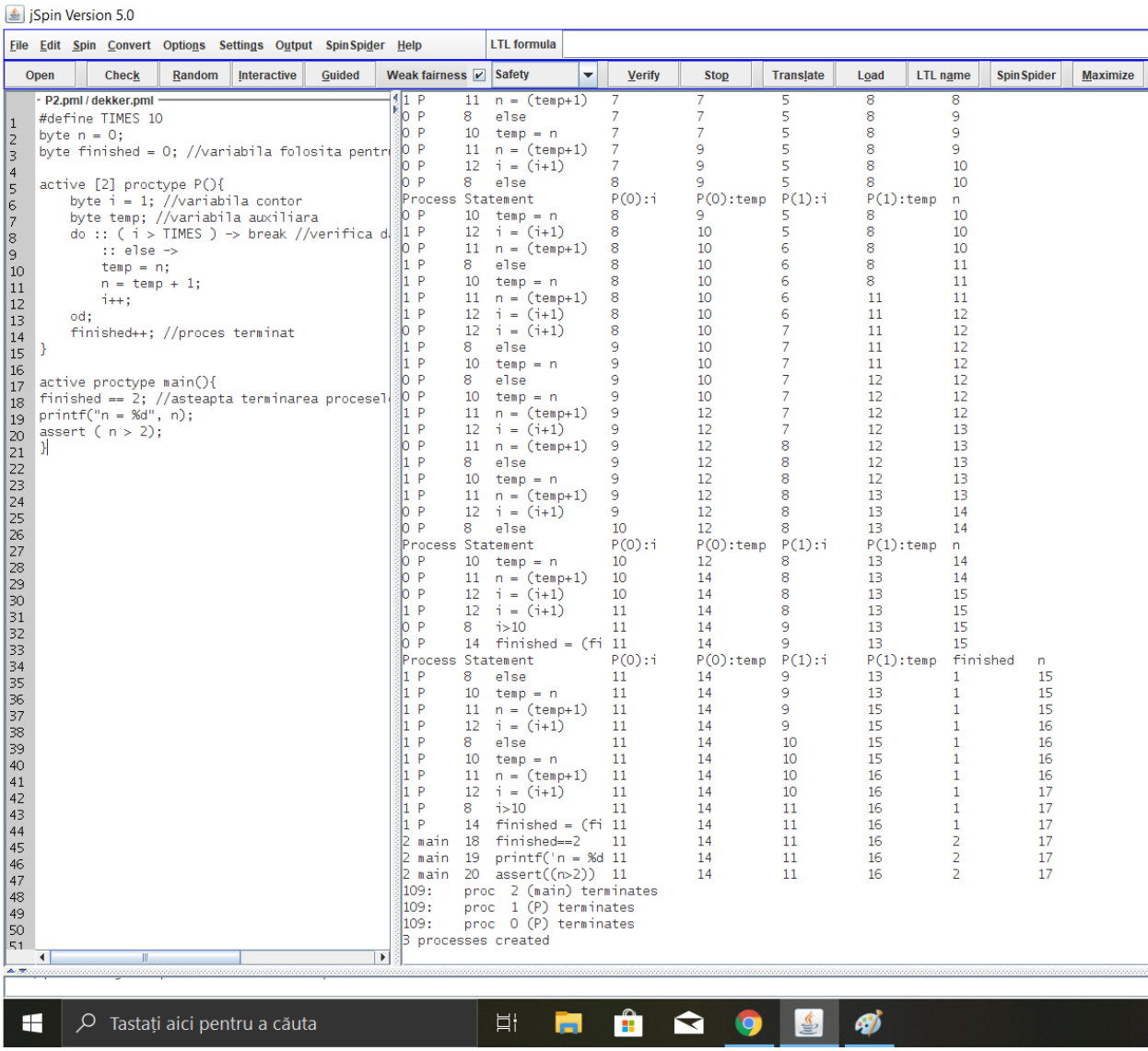
La linia 17 am creat un nou proces numit "main". In acesta se verifica daca finished are valoarea 2, adica daca cele 2 procese s-au incheiat. In caz contrar, se asteapta terminarea acestora. Apoi se afiseaza valoarea finala a lui n.

3 Experimental data

Am rulat algoritmul de mai multe ori folosind optiunea Random. De fiecare data la iesire n avea valori apropiate de 20, dar niciodata valoarea 20. Se observa ca valoarea finala a variabilei finished este 2, ceea ce inseamna ca ambele procese s-au incheiat.


jSpin Version 5.0
File Edit Spin Convert Options Settings Output SpinSpider Help
LTL formula
Open Check Random Interactive Guided Weak fairness Safety Verify Stop Translate Load LTL name SpinSpider
P2.pml / dekker.pml
#define TIMES 10
byte n = 0;
byte finished = 0; //variabila folosita pentr
active [2] proctype P(){
byte i = 1; //variabila contor
byte temp; //variabila auxiliara
do :: (i > TIMES) -> break //verifica d
:: else ->
temp = n;
n = temp + 1;
i++;
od;
finished++; //proces terminat
}
active proctype main(){
finished == 2; //asteapta terminarea procesel
printf("n = %d", n);
assert (n > 2);
}
1 P 11 n = (temp+1) 7 8 5 9 9
1 P 12 i = (i+1) 7 8 5 9 10
1 P 8 else 7 8 6 9 10
1 P 10 temp = n 7 8 6 9 10
0 P 8 else 7 8 6 10 10
1 P 11 n = (temp+1) 7 8 6 10 10
1 P 12 i = (i+1) 7 8 6 10 11
0 P 10 temp = n 7 8 7 10 11
Process Statement P(0):i P(0):temp P(1):i P(1):temp n
1 P 8 else 7 11 7 10 11
0 P 11 n = (temp+1) 7 11 7 10 11
0 P 12 i = (i+1) 7 11 7 10 12
1 P 10 temp = n 8 11 7 10 12
1 P 11 n = (temp+1) 8 11 7 12 12
1 P 12 i = (i+1) 8 11 7 12 13
1 P 8 else 8 11 8 12 13
1 P 10 temp = n 8 11 8 12 13
0 P 8 else 8 11 8 13 13
1 P 11 n = (temp+1) 8 11 8 13 13
0 P 10 temp = n 8 11 8 13 14
1 P 12 i = (i+1) 8 14 8 13 14
0 P 11 n = (temp+1) 8 14 9 13 14
0 P 12 i = (i+1) 8 14 9 13 15
1 P 8 else 9 14 9 13 15
1 P 10 temp = n 9 14 9 13 15
1 P 11 n = (temp+1) 9 14 9 15 15
0 P 8 else 9 14 9 15 16
0 P 10 temp = n 9 14 9 15 16
0 P 11 n = (temp+1) 9 16 9 15 16
Process Statement P(0):i P(0):temp P(1):i P(1):temp n
1 P 12 i = (i+1) 9 16 9 15 17
0 P 12 i = (i+1) 9 16 10 15 17
1 P 8 else 10 16 10 15 17
0 P 8 else 10 16 10 15 17
0 P 10 temp = n 10 16 10 15 17
1 P 10 temp = n 10 17 10 15 17
1 P 11 n = (temp+1) 10 17 10 17 17
0 P 11 n = (temp+1) 10 17 10 17 18
1 P 12 i = (i+1) 10 17 10 17 18
1 P 8 i>10 10 17 11 17 18
0 P 12 i = (i+1) 10 17 11 17 18
0 P 8 i>10 11 17 11 17 18
1 P 14 finished = (fi 11 17 11 17 18
Process Statement P(0):i P(0):temp P(1):i P(1):temp finished n
0 P 14 finished = (fi 11 17 11 17 1 18
2 main 18 finished==2 11 17 11 17 2 18
2 main 19 printf('n = %d 11 17 11 17 2 18
2 main 20 assert((n>2)) 11 17 11 17 2 18
109: proc 2 (main) terminates
109: proc 1 (P) terminates
109: proc 0 (P) terminates
3 processes created
Tastați aici pentru a căuta


jSpin Version 5.0
File Edit Spin Convert Options Settings Output SpinSpider Help
LTL formula
Open Check Random Interactive Guided Weak fairness Safety Verify Stop Translate Load LTL name SpinSpider Maximize
P2.pml / dekker.pml
#define TIMES 10
byte n = 0;
byte finished = 0; //variabila folosita pentr
active [2] proctype P(){
byte i = 1; //variabila contor
byte temp; //variabila auxiliara
do :: (i > TIMES) -> break //verifica d
:: else ->
temp = n;
n = temp + 1;
i++;
od;
finished++; //proces terminat
}
active proctype main(){
finished == 2; //asteapta terminarea procesel
printf("n = %d", n);
assert (n > 2);
}
0 P 11 n = (temp+1) 5 8 7 7 8
0 P 12 i = (i+1) 5 8 7 7 9
0 P 8 else 6 8 7 7 9
0 P 10 temp = n 6 8 7 7 9
1 P 8 else 6 9 7 7 9
1 P 10 temp = n 6 9 7 7 9
Process Statement P(0):i P(0):temp P(1):i P(1):temp n
1 P 11 n = (temp+1) 6 9 7 9 9
0 P 11 n = (temp+1) 6 9 7 9 10
1 P 12 i = (i+1) 6 9 7 9 10
0 P 12 i = (i+1) 6 9 8 9 10
0 P 8 else 7 9 8 9 10
0 P 10 temp = n 7 9 8 9 10
0 P 11 n = (temp+1) 7 10 8 9 10
0 P 12 i = (i+1) 7 10 8 9 11
0 P 8 else 8 10 8 9 11
0 P 10 temp = n 8 10 8 9 11
1 P 8 else 8 11 8 9 11
0 P 11 n = (temp+1) 8 11 8 9 11
1 P 10 temp = n 8 11 8 9 12
1 P 11 n = (temp+1) 8 11 8 12 12
1 P 12 i = (i+1) 8 11 8 12 13
1 P 8 else 8 11 9 12 13
1 P 10 temp = n 8 11 9 12 13
0 P 12 i = (i+1) 8 11 9 13 13
1 P 11 n = (temp+1) 9 11 9 13 13
0 P 8 else 9 11 9 13 14
Process Statement P(0):i P(0):temp P(1):i P(1):temp n
1 P 12 i = (i+1) 9 11 9 13 14
0 P 10 temp = n 9 11 10 13 14
0 P 11 n = (temp+1) 9 14 10 13 14
1 P 8 else 9 14 10 13 15
0 P 12 i = (i+1) 9 14 10 13 15
1 P 10 temp = n 10 14 10 13 15
1 P 11 n = (temp+1) 10 14 10 15 15
0 P 8 else 10 14 10 15 16
0 P 10 temp = n 10 14 10 15 16
0 P 11 n = (temp+1) 10 16 10 15 16
1 P 12 i = (i+1) 10 16 10 15 17
0 P 12 i = (i+1) 10 16 11 15 17
1 P 8 i>10 11 16 11 15 17
0 P 8 i>10 11 16 11 15 17
1 P 14 finished = (fi 11 16 11 15 17
Process Statement P(0):i P(0):temp P(1):i P(1):temp finished n
0 P 14 finished = (fi 11 16 11 15 1 17
2 main 18 finished==2 11 16 11 15 2 17
2 main 19 printf('n = %d 11 16 11 15 2 17
2 main 20 assert((n>2)) 11 16 11 15 2 17
109: proc 2 (main) terminates
109: proc 1 (P) terminates
109: proc 0 (P) terminates
3 processes created
Tastați aici pentru a căuta



4 Results & Conclusions

Am implementat in Promela numararea concurenta utilizand o variabila n si 2 procese care incrementeaza fiecare variabila n de 10 ori. Am implementat instructiunea "for" utilizand bucla "do" si explicand ce trebuie sa faca algoritmul in cazul in care o conditie nu este indeplinita. Valoarea finala a lui n a fost apropiata de 20, dar niciodata 20, desi am rulat programul de mai multe ori pentru a-l testa.