eg. in c:

int main()

{

while(1)

f1();

return 0;

}

int f1()

{

int i;

for(i = 1; i < 10; i++)

f2();

return 0;

}

int f2()

{

f3();

return 0;

}

int f3()

{

f1();

return 0;

}

now i’m gonna use modex to translate it to promela:

- while using %x -xe or without prx

promela code:

int res\_p\_f3;

bool lck\_p\_f3\_ret;

bool lck\_p\_f3;

int res\_p\_f2;

bool lck\_p\_f2\_ret;

bool lck\_p\_f2;

int res\_p\_f1;

bool lck\_p\_f1\_ret;

bool lck\_p\_f1;

int res\_p\_main;

bool lck\_p\_main\_ret;

bool lck\_p\_main;

chan ret\_p\_f3 = [1] of { pid };

chan exc\_cll\_p\_f3 = [0] of { pid };

chan req\_cll\_p\_f3 = [1] of { pid };

chan ret\_p\_f2 = [1] of { pid };

chan exc\_cll\_p\_f2 = [0] of { pid };

chan req\_cll\_p\_f2 = [1] of { pid };

chan ret\_p\_f1 = [1] of { pid };

chan exc\_cll\_p\_f1 = [0] of { pid };

chan req\_cll\_p\_f1 = [1] of { pid };

chan ret\_p\_main = [1] of { pid };

chan exc\_cll\_p\_main = [0] of { pid };

chan req\_cll\_p\_main = [1] of { pid };

active proctype p\_main()

{

pid lck\_id;

L\_0:

do

:: true;

atomic {

lck\_p\_f1 == 0 && empty(req\_cll\_p\_f1) -> req\_cll\_p\_f1!\_pid;

exc\_cll\_p\_f1!\_pid;

}

ret\_p\_f1?eval(\_pid);

c\_code { ; now.lck\_p\_f1\_ret = 0; };

goto L\_0;

:: c\_expr { !1 }; -> break

od;

atomic { !lck\_p\_main\_ret -> lck\_p\_main\_ret = 1 };

c\_code { now.res\_p\_main = (int ) 0; }; goto Return;

Return: skip;

}

active proctype p\_f1()

{

int i;

pid lck\_id;

endRestart:

atomic {

nempty(req\_cll\_p\_f1) && !lck\_p\_f1 -> lck\_p\_f1 = 1;

req\_cll\_p\_f1?lck\_id; exc\_cll\_p\_f1?eval(lck\_id);

lck\_p\_f1 = 0;

};

c\_code { Pp\_f1->i=1; };

L\_1:

do

:: c\_expr { (Pp\_f1->i<10) };

atomic {

lck\_p\_f2 == 0 && empty(req\_cll\_p\_f2) -> req\_cll\_p\_f2!\_pid;

exc\_cll\_p\_f2!\_pid;

}

ret\_p\_f2?eval(\_pid);

c\_code { ; now.lck\_p\_f2\_ret = 0; };

c\_code { Pp\_f1->i++; };

goto L\_1;

c\_code { Pp\_f1->i++; };

:: c\_expr { !(Pp\_f1->i<10) }; -> break

od;

atomic { !lck\_p\_f1\_ret -> lck\_p\_f1\_ret = 1 };

c\_code { now.res\_p\_f1 = (int ) 0; }; goto Return;

Return: skip;

ret\_p\_f1!lck\_id;

goto endRestart

}

active proctype p\_f2()

{

pid lck\_id;

endRestart:

atomic {

nempty(req\_cll\_p\_f2) && !lck\_p\_f2 -> lck\_p\_f2 = 1;

req\_cll\_p\_f2?lck\_id; exc\_cll\_p\_f2?eval(lck\_id);

lck\_p\_f2 = 0;

};

atomic {

lck\_p\_f3 == 0 && empty(req\_cll\_p\_f3) -> req\_cll\_p\_f3!\_pid;

exc\_cll\_p\_f3!\_pid;

}

ret\_p\_f3?eval(\_pid);

c\_code { ; now.lck\_p\_f3\_ret = 0; };

atomic { !lck\_p\_f2\_ret -> lck\_p\_f2\_ret = 1 };

c\_code { now.res\_p\_f2 = (int ) 0; }; goto Return;

Return: skip;

ret\_p\_f2!lck\_id;

goto endRestart

}

active proctype p\_f3()

{

pid lck\_id;

endRestart:

atomic {

nempty(req\_cll\_p\_f3) && !lck\_p\_f3 -> lck\_p\_f3 = 1;

req\_cll\_p\_f3?lck\_id; exc\_cll\_p\_f3?eval(lck\_id);

lck\_p\_f3 = 0;

};

atomic {

lck\_p\_f1 == 0 && empty(req\_cll\_p\_f1) -> req\_cll\_p\_f1!\_pid;

exc\_cll\_p\_f1!\_pid;

}

ret\_p\_f1?eval(\_pid);

c\_code { ; now.lck\_p\_f1\_ret = 0; };

atomic { !lck\_p\_f3\_ret -> lck\_p\_f3\_ret = 1 };

c\_code { now.res\_p\_f3 = (int ) 0; }; goto Return;

Return: skip;

ret\_p\_f3!lck\_id;

goto endRestart

}

5 different ways extracted froom spin:

1)→ main – f1 – f2 – f3

Selected: 4

1: proc 0 (p\_main:1) model-noprx.pml:33 (state 1) [(1)]

Selected: 4

2: proc 0 (p\_main:1) model-noprx.pml:35 (state 2) [(((lck\_p\_f1==0)&&empty(req\_cll\_p\_f1)))]

3: proc 0 (p\_main:1) model-noprx.pml:35 (state 3) [req\_cll\_p\_f1!\_pid]

Selected: 3

4: proc 1 (p\_f1:1) model-noprx.pml:53 (state 1) [((nempty(req\_cll\_p\_f1)&&!(lck\_p\_f1)))]

5: proc 1 (p\_f1:1) model-noprx.pml:53 (state 2) [lck\_p\_f1 = 1]

6: proc 1 (p\_f1:1) model-noprx.pml:54 (state 3) [req\_cll\_p\_f1?lck\_id]

Selected: 4

7: proc 0 (p\_main:1) model-noprx.pml:36 (state 4) [exc\_cll\_p\_f1!\_pid]

7: proc 1 (p\_f1:1) model-noprx.pml:54 (state 4) [exc\_cll\_p\_f1?eval(lck\_id)]

8: proc 1 (p\_f1:1) model-noprx.pml:55 (state 5) [lck\_p\_f1 = 0]

Selected: 3

c\_code4: { /\* line 57 model-noprx.pml \*/

Pp\_f1->i=1; }

9: proc 1 (p\_f1:1) model-noprx.pml:57 (state 7) [{c\_code4}]

c\_code5: /\* line 60 model-noprx.pml \*/

(Pp\_f1->i<10)

c\_code9: /\* line 70 model-noprx.pml \*/

!(Pp\_f1->i<10)

Selected: 3

c\_code5: /\* line 60 model-noprx.pml \*/

(Pp\_f1->i<10)

10: proc 1 (p\_f1:1) model-noprx.pml:60 (state 8) [({c\_code5})]

Selected: 3

11: proc 1 (p\_f1:1) model-noprx.pml:62 (state 9) [(((lck\_p\_f2==0)&&empty(req\_cll\_p\_f2)))]

12: proc 1 (p\_f1:1) model-noprx.pml:62 (state 10) [req\_cll\_p\_f2!\_pid]

Selected: 2

13: proc 2 (p\_f2:1) model-noprx.pml:83 (state 1) [((nempty(req\_cll\_p\_f2)&&!(lck\_p\_f2)))]

14: proc 2 (p\_f2:1) model-noprx.pml:83 (state 2) [lck\_p\_f2 = 1]

15: proc 2 (p\_f2:1) model-noprx.pml:84 (state 3) [req\_cll\_p\_f2?lck\_id]

Selected: 3

16: proc 1 (p\_f1:1) model-noprx.pml:63 (state 11) [exc\_cll\_p\_f2!\_pid]

16: proc 2 (p\_f2:1) model-noprx.pml:84 (state 4) [exc\_cll\_p\_f2?eval(lck\_id)]

17: proc 2 (p\_f2:1) model-noprx.pml:85 (state 5) [lck\_p\_f2 = 0]

Selected: 2

18: proc 2 (p\_f2:1) model-noprx.pml:88 (state 7) [(((lck\_p\_f3==0)&&empty(req\_cll\_p\_f3)))]

19: proc 2 (p\_f2:1) model-noprx.pml:88 (state 8) [req\_cll\_p\_f3!\_pid]

Selected: 1

20: proc 3 (p\_f3:1) model-noprx.pml:104 (state 1) [((nempty(req\_cll\_p\_f3)&&!(lck\_p\_f3)))]

21: proc 3 (p\_f3:1) model-noprx.pml:104 (state 2) [lck\_p\_f3 = 1]

22: proc 3 (p\_f3:1) model-noprx.pml:105 (state 3) [req\_cll\_p\_f3?lck\_id]

Selected: 2

23: proc 2 (p\_f2:1) model-noprx.pml:89 (state 9) [exc\_cll\_p\_f3!\_pid]

23: proc 3 (p\_f3:1) model-noprx.pml:105 (state 4) [exc\_cll\_p\_f3?eval(lck\_id)]

24: proc 3 (p\_f3:1) model-noprx.pml:106 (state 5) [lck\_p\_f3 = 0]

Selected: 1

25: proc 3 (p\_f3:1) model-noprx.pml:109 (state 7) [(((lck\_p\_f1==0)&&empty(req\_cll\_p\_f1)))]

26: proc 3 (p\_f3:1) model-noprx.pml:109 (state 8) [req\_cll\_p\_f1!\_pid]

2)→ main – f1 – main – f1 – f2 – f3

Selected: 4

1: proc 0 (p\_main:1) model-noprx.pml:33 (state 1) [(1)]

Selected: 4

2: proc 0 (p\_main:1) model-noprx.pml:35 (state 2) [(((lck\_p\_f1==0)&&empty(req\_cll\_p\_f1)))]

3: proc 0 (p\_main:1) model-noprx.pml:35 (state 3) [req\_cll\_p\_f1!\_pid]

Selected: 3

4: proc 1 (p\_f1:1) model-noprx.pml:53 (state 1) [((nempty(req\_cll\_p\_f1)&&!(lck\_p\_f1)))]

5: proc 1 (p\_f1:1) model-noprx.pml:53 (state 2) [lck\_p\_f1 = 1]

6: proc 1 (p\_f1:1) model-noprx.pml:54 (state 3) [req\_cll\_p\_f1?lck\_id]

Selected: 4

7: proc 0 (p\_main:1) model-noprx.pml:36 (state 4) [exc\_cll\_p\_f1!\_pid]

7: proc 1 (p\_f1:1) model-noprx.pml:54 (state 4) [exc\_cll\_p\_f1?eval(lck\_id)]

8: proc 1 (p\_f1:1) model-noprx.pml:55 (state 5) [lck\_p\_f1 = 0]

Selected: 3

c\_code4: { /\* line 57 model-noprx.pml \*/

Pp\_f1->i=1; }

9: proc 1 (p\_f1:1) model-noprx.pml:57 (state 7) [{c\_code4}]

c\_code5: /\* line 60 model-noprx.pml \*/

(Pp\_f1->i<10)

c\_code9: /\* line 70 model-noprx.pml \*/

!(Pp\_f1->i<10)

Selected: 4

c\_code9: /\* line 70 model-noprx.pml \*/

!(Pp\_f1->i<10)

10: proc 1 (p\_f1:1) model-noprx.pml:70 (state 18) [({c\_code9})]

Selected: 3

11: proc 1 (p\_f1:1) model-noprx.pml:59 (state 22) [break]

Selected: 3

12: proc 1 (p\_f1:1) model-noprx.pml:72 (state 23) [(!(lck\_p\_f1\_ret))]

13: proc 1 (p\_f1:1) model-noprx.pml:72 (state 24) [lck\_p\_f1\_ret = 1]

Selected: 3

c\_code10: { /\* line 73 model-noprx.pml \*/

14: proc 1 (p\_f1:1) model-noprx.pml:73 (state 26) [{c\_code10}]

Selected: 3

15: proc 1 (p\_f1:1) model-noprx.pml:74 (state 28) [(1)]

Selected: 3

16: proc 1 (p\_f1:1) model-noprx.pml:75 (state 29) [ret\_p\_f1!lck\_id]

Selected: 4

17: proc 0 (p\_main:1) model-noprx.pml:38 (state 6) [ret\_p\_f1?eval(\_pid)]

Selected: 4

c\_code1: { /\* line 39 model-noprx.pml \*/

18: proc 0 (p\_main:1) model-noprx.pml:39 (state 7) [{c\_code1}]

c\_code2: /\* line 41 model-noprx.pml \*/

!1

Selected: 4

19: proc 0 (p\_main:1) model-noprx.pml:33 (state 1) [(1)]

Selected: 4

20: proc 0 (p\_main:1) model-noprx.pml:35 (state 2) [(((lck\_p\_f1==0)&&empty(req\_cll\_p\_f1)))]

21: proc 0 (p\_main:1) model-noprx.pml:35 (state 3) [req\_cll\_p\_f1!\_pid]

Selected: 3

22: proc 1 (p\_f1:1) model-noprx.pml:53 (state 1) [((nempty(req\_cll\_p\_f1)&&!(lck\_p\_f1)))]

23: proc 1 (p\_f1:1) model-noprx.pml:53 (state 2) [lck\_p\_f1 = 1]

24: proc 1 (p\_f1:1) model-noprx.pml:54 (state 3) [req\_cll\_p\_f1?lck\_id]

Selected: 4

25: proc 0 (p\_main:1) model-noprx.pml:36 (state 4) [exc\_cll\_p\_f1!\_pid]

25: proc 1 (p\_f1:1) model-noprx.pml:54 (state 4) [exc\_cll\_p\_f1?eval(lck\_id)]

26: proc 1 (p\_f1:1) model-noprx.pml:55 (state 5) [lck\_p\_f1 = 0]

Selected: 3

c\_code4: { /\* line 57 model-noprx.pml \*/

Pp\_f1->i=1; }

27: proc 1 (p\_f1:1) model-noprx.pml:57 (state 7) [{c\_code4}]

c\_code5: /\* line 60 model-noprx.pml \*/

(Pp\_f1->i<10)

c\_code9: /\* line 70 model-noprx.pml \*/

!(Pp\_f1->i<10)

Selected: 3

c\_code5: /\* line 60 model-noprx.pml \*/

(Pp\_f1->i<10)

28: proc 1 (p\_f1:1) model-noprx.pml:60 (state 8) [({c\_code5})]

Selected: 3

29: proc 1 (p\_f1:1) model-noprx.pml:62 (state 9) [(((lck\_p\_f2==0)&&empty(req\_cll\_p\_f2)))]

30: proc 1 (p\_f1:1) model-noprx.pml:62 (state 10) [req\_cll\_p\_f2!\_pid]

Selected: 2

31: proc 2 (p\_f2:1) model-noprx.pml:83 (state 1) [((nempty(req\_cll\_p\_f2)&&!(lck\_p\_f2)))]

32: proc 2 (p\_f2:1) model-noprx.pml:83 (state 2) [lck\_p\_f2 = 1]

33: proc 2 (p\_f2:1) model-noprx.pml:84 (state 3) [req\_cll\_p\_f2?lck\_id]

Selected: 3

34: proc 1 (p\_f1:1) model-noprx.pml:63 (state 11) [exc\_cll\_p\_f2!\_pid]

34: proc 2 (p\_f2:1) model-noprx.pml:84 (state 4) [exc\_cll\_p\_f2?eval(lck\_id)]

35: proc 2 (p\_f2:1) model-noprx.pml:85 (state 5) [lck\_p\_f2 = 0]

Selected: 2

36: proc 2 (p\_f2:1) model-noprx.pml:88 (state 7) [(((lck\_p\_f3==0)&&empty(req\_cll\_p\_f3)))]

37: proc 2 (p\_f2:1) model-noprx.pml:88 (state 8) [req\_cll\_p\_f3!\_pid]

Selected: 1

38: proc 3 (p\_f3:1) model-noprx.pml:104 (state 1) [((nempty(req\_cll\_p\_f3)&&!(lck\_p\_f3)))]

39: proc 3 (p\_f3:1) model-noprx.pml:104 (state 2) [lck\_p\_f3 = 1]

40: proc 3 (p\_f3:1) model-noprx.pml:105 (state 3) [req\_cll\_p\_f3?lck\_id]

Selected: 2

41: proc 2 (p\_f2:1) model-noprx.pml:89 (state 9) [exc\_cll\_p\_f3!\_pid]

41: proc 3 (p\_f3:1) model-noprx.pml:105 (state 4) [exc\_cll\_p\_f3?eval(lck\_id)]

42: proc 3 (p\_f3:1) model-noprx.pml:106 (state 5) [lck\_p\_f3 = 0]

Selected: 1

43: proc 3 (p\_f3:1) model-noprx.pml:109 (state 7) [(((lck\_p\_f1==0)&&empty(req\_cll\_p\_f1)))]

44: proc 3 (p\_f3:1) model-noprx.pml:109 (state 8) [req\_cll\_p\_f1!\_pid]

3)→ main – f1 – main – f1

Selected: 4

1: proc 0 (p\_main:1) model-noprx.pml:33 (state 1) [(1)]

Selected: 4

2: proc 0 (p\_main:1) model-noprx.pml:35 (state 2) [(((lck\_p\_f1==0)&&empty(req\_cll\_p\_f1)))]

3: proc 0 (p\_main:1) model-noprx.pml:35 (state 3) [req\_cll\_p\_f1!\_pid]

Selected: 3

4: proc 1 (p\_f1:1) model-noprx.pml:53 (state 1) [((nempty(req\_cll\_p\_f1)&&!(lck\_p\_f1)))]

5: proc 1 (p\_f1:1) model-noprx.pml:53 (state 2) [lck\_p\_f1 = 1]

6: proc 1 (p\_f1:1) model-noprx.pml:54 (state 3) [req\_cll\_p\_f1?lck\_id]

Selected: 4

7: proc 0 (p\_main:1) model-noprx.pml:36 (state 4) [exc\_cll\_p\_f1!\_pid]

7: proc 1 (p\_f1:1) model-noprx.pml:54 (state 4) [exc\_cll\_p\_f1?eval(lck\_id)]

8: proc 1 (p\_f1:1) model-noprx.pml:55 (state 5) [lck\_p\_f1 = 0]

Selected: 3

c\_code4: { /\* line 57 model-noprx.pml \*/

Pp\_f1->i=1; }

9: proc 1 (p\_f1:1) model-noprx.pml:57 (state 7) [{c\_code4}]

c\_code5: /\* line 60 model-noprx.pml \*/

(Pp\_f1->i<10)

c\_code9: /\* line 70 model-noprx.pml \*/

!(Pp\_f1->i<10)

Selected: 4

c\_code9: /\* line 70 model-noprx.pml \*/

!(Pp\_f1->i<10)

10: proc 1 (p\_f1:1) model-noprx.pml:70 (state 18) [({c\_code9})]

Selected: 3

11: proc 1 (p\_f1:1) model-noprx.pml:59 (state 22) [break]

Selected: 3

12: proc 1 (p\_f1:1) model-noprx.pml:72 (state 23) [(!(lck\_p\_f1\_ret))]

13: proc 1 (p\_f1:1) model-noprx.pml:72 (state 24) [lck\_p\_f1\_ret = 1]

Selected: 3

c\_code10: { /\* line 73 model-noprx.pml \*/

14: proc 1 (p\_f1:1) model-noprx.pml:73 (state 26) [{c\_code10}]

Selected: 3

15: proc 1 (p\_f1:1) model-noprx.pml:74 (state 28) [(1)]

Selected: 3

16: proc 1 (p\_f1:1) model-noprx.pml:75 (state 29) [ret\_p\_f1!lck\_id]

Selected: 4

17: proc 0 (p\_main:1) model-noprx.pml:38 (state 6) [ret\_p\_f1?eval(\_pid)]

Selected: 4

c\_code1: { /\* line 39 model-noprx.pml \*/

18: proc 0 (p\_main:1) model-noprx.pml:39 (state 7) [{c\_code1}]

c\_code2: /\* line 41 model-noprx.pml \*/

!1

Selected: 4

19: proc 0 (p\_main:1) model-noprx.pml:33 (state 1) [(1)]

Selected: 4

20: proc 0 (p\_main:1) model-noprx.pml:35 (state 2) [(((lck\_p\_f1==0)&&empty(req\_cll\_p\_f1)))]

21: proc 0 (p\_main:1) model-noprx.pml:35 (state 3) [req\_cll\_p\_f1!\_pid]

Selected: 3

22: proc 1 (p\_f1:1) model-noprx.pml:53 (state 1) [((nempty(req\_cll\_p\_f1)&&!(lck\_p\_f1)))]

23: proc 1 (p\_f1:1) model-noprx.pml:53 (state 2) [lck\_p\_f1 = 1]

24: proc 1 (p\_f1:1) model-noprx.pml:54 (state 3) [req\_cll\_p\_f1?lck\_id]

Selected: 4

25: proc 0 (p\_main:1) model-noprx.pml:36 (state 4) [exc\_cll\_p\_f1!\_pid]

25: proc 1 (p\_f1:1) model-noprx.pml:54 (state 4) [exc\_cll\_p\_f1?eval(lck\_id)]

26: proc 1 (p\_f1:1) model-noprx.pml:55 (state 5) [lck\_p\_f1 = 0]

Selected: 3

c\_code4: { /\* line 57 model-noprx.pml \*/

Pp\_f1->i=1; }

27: proc 1 (p\_f1:1) model-noprx.pml:57 (state 7) [{c\_code4}]

c\_code5: /\* line 60 model-noprx.pml \*/

(Pp\_f1->i<10)

c\_code9: /\* line 70 model-noprx.pml \*/

!(Pp\_f1->i<10)

Selected: 4

c\_code9: /\* line 70 model-noprx.pml \*/

!(Pp\_f1->i<10)

28: proc 1 (p\_f1:1) model-noprx.pml:70 (state 18) [({c\_code9})]

Selected: 3

29: proc 1 (p\_f1:1) model-noprx.pml:59 (state 22) [break]

4) → main – f1 – main

Selected: 4

1: proc 0 (p\_main:1) model-noprx.pml:33 (state 1) [(1)]

Selected: 4

2: proc 0 (p\_main:1) model-noprx.pml:35 (state 2) [(((lck\_p\_f1==0)&&empty(req\_cll\_p\_f1)))]

3: proc 0 (p\_main:1) model-noprx.pml:35 (state 3) [req\_cll\_p\_f1!\_pid]

Selected: 3

4: proc 1 (p\_f1:1) model-noprx.pml:53 (state 1) [((nempty(req\_cll\_p\_f1)&&!(lck\_p\_f1)))]

5: proc 1 (p\_f1:1) model-noprx.pml:53 (state 2) [lck\_p\_f1 = 1]

6: proc 1 (p\_f1:1) model-noprx.pml:54 (state 3) [req\_cll\_p\_f1?lck\_id]

Selected: 4

7: proc 0 (p\_main:1) model-noprx.pml:36 (state 4) [exc\_cll\_p\_f1!\_pid]

7: proc 1 (p\_f1:1) model-noprx.pml:54 (state 4) [exc\_cll\_p\_f1?eval(lck\_id)]

8: proc 1 (p\_f1:1) model-noprx.pml:55 (state 5) [lck\_p\_f1 = 0]

Selected: 3

c\_code4: { /\* line 57 model-noprx.pml \*/

Pp\_f1->i=1; }

9: proc 1 (p\_f1:1) model-noprx.pml:57 (state 7) [{c\_code4}]

c\_code5: /\* line 60 model-noprx.pml \*/

(Pp\_f1->i<10)

c\_code9: /\* line 70 model-noprx.pml \*/

!(Pp\_f1->i<10)

Selected: 4

c\_code9: /\* line 70 model-noprx.pml \*/

!(Pp\_f1->i<10)

10: proc 1 (p\_f1:1) model-noprx.pml:70 (state 18) [({c\_code9})]

Selected: 3

11: proc 1 (p\_f1:1) model-noprx.pml:59 (state 22) [break]

Selected: 3

12: proc 1 (p\_f1:1) model-noprx.pml:72 (state 23) [(!(lck\_p\_f1\_ret))]

13: proc 1 (p\_f1:1) model-noprx.pml:72 (state 24) [lck\_p\_f1\_ret = 1]

Selected: 3

c\_code10: { /\* line 73 model-noprx.pml \*/

14: proc 1 (p\_f1:1) model-noprx.pml:73 (state 26) [{c\_code10}]

Selected: 3

15: proc 1 (p\_f1:1) model-noprx.pml:74 (state 28) [(1)]

Selected: 3

16: proc 1 (p\_f1:1) model-noprx.pml:75 (state 29) [ret\_p\_f1!lck\_id]

Selected: 4

17: proc 0 (p\_main:1) model-noprx.pml:38 (state 6) [ret\_p\_f1?eval(\_pid)]

Selected: 4

c\_code1: { /\* line 39 model-noprx.pml \*/

18: proc 0 (p\_main:1) model-noprx.pml:39 (state 7) [{c\_code1}]

c\_code2: /\* line 41 model-noprx.pml \*/

!1

Selected: 5

c\_code2: /\* line 41 model-noprx.pml \*/

!1

19: proc 0 (p\_main:1) model-noprx.pml:41 (state 9) [({c\_code2})]

Selected: 4

20: proc 0 (p\_main:1) model-noprx.pml:32 (state 13) [break]

Selected: 4

21: proc 0 (p\_main:1) model-noprx.pml:43 (state 14) [(!(lck\_p\_main\_ret))]

22: proc 0 (p\_main:1) model-noprx.pml:43 (state 15) [lck\_p\_main\_ret = 1]

Selected: 4

c\_code3: { /\* line 44 model-noprx.pml \*/

23: proc 0 (p\_main:1) model-noprx.pml:44 (state 17) [{c\_code3}]

Selected: 4

24: proc 0 (p\_main:1) model-noprx.pml:45 (state 19) [(1)]

5)→ main

Selected: 5

c\_code2: /\* line 41 model-noprx.pml \*/

!1

1: proc 0 (p\_main:1) model-noprx.pml:41 (state 9) [({c\_code2})]

Selected: 4

2: proc 0 (p\_main:1) model-noprx.pml:32 (state 13) [break]

Selected: 4

3: proc 0 (p\_main:1) model-noprx.pml:43 (state 14) [(!(lck\_p\_main\_ret))]

4: proc 0 (p\_main:1) model-noprx.pml:43 (state 15) [lck\_p\_main\_ret = 1]

Selected: 4

c\_code3: { /\* line 44 model-noprx.pml \*/

5: proc 0 (p\_main:1) model-noprx.pml:44 (state 17) [{c\_code3}]

Selected: 4

6: proc 0 (p\_main:1) model-noprx.pml:45 (state 19) [(1)]

- while using %x -x

promela:

active proctype p\_main()

{

L\_0:

do

:: true;

c\_code { f1(); };

goto L\_0;

:: c\_expr { !1 }; -> break

od;

goto Return;

Return: skip;

}

active proctype p\_f1()

{

int i;

c\_code { Pp\_f1->i=1; };

L\_1:

do

:: c\_expr { (Pp\_f1->i<10) };

c\_code { f2(); };

c\_code { Pp\_f1->i++; };

goto L\_1;

c\_code { Pp\_f1->i++; };

:: c\_expr { !(Pp\_f1->i<10) }; -> break

od;

goto Return;

Return: skip;

}

active proctype p\_f2()

{

c\_code { f3(); };

goto Return;

Return: skip;

}

active proctype p\_f3()

{

c\_code { f1(); };

goto Return;

Return: skip;

}

spin extracted:

the function verification turns are so complicated in this version. there are many ways.

Modifying the simpler code (-x):

active proctype p\_main()

{

L\_0:

do

:: true;

run p\_f1();

goto L\_0;

:: !1 -> break

od;

goto Return;

Return: skip;

}

proctype p\_f1()

{

int i;

i=1;

L\_1:

do

:: i<10;

run p\_f2();

i++;

goto L\_1;

i++;

:: !(i<10) -> break

od;

goto Return;

Return: skip;

}

proctype p\_f2()

{

run p\_f3();

goto Return;

Return: skip;

}

proctype p\_f3()

{

run p\_f1();

goto Return;

Return: skip;

}

(detail of modifying)

the spin result:

- 1 of the ways is what we want(false negative):

0: proc - (:root:) creates proc 0 (p\_main)

1: proc 0 (p\_main:1) model-prefer.pml:8 (state 1) [(1)]

Starting p\_f1 with pid 1

2: proc 0 (p\_main:1) creates proc 1 (p\_f1)

2: proc 0 (p\_main:1) model-prefer.pml:9 (state 2) [(run p\_f1())]

Selected: 1

3: proc 1 (p\_f1:1) model-prefer.pml:19 (state 1) [i = 1]

Selected: 1

4: proc 1 (p\_f1:1) model-prefer.pml:22 (state 2) [((i<10))]

Selected: 1

Starting p\_f2 with pid 2

5: proc 1 (p\_f1:1) creates proc 2 (p\_f2)

5: proc 1 (p\_f1:1) model-prefer.pml:23 (state 3) [(run p\_f2())]

Selected: 1

Starting p\_f3 with pid 3

6: proc 2 (p\_f2:1) creates proc 3 (p\_f3)

6: proc 2 (p\_f2:1) model-prefer.pml:34 (state 1) [(run p\_f3())]

Selected: 1

Starting p\_f1 with pid 4

7: proc 3 (p\_f3:1) creates proc 4 (p\_f1)

7: proc 3 (p\_f3:1) model-prefer.pml:40 (state 1) [(run p\_f1())]

Selected: 1

8: proc 4 (p\_f1:1) model-prefer.pml:19 (state 1) [i = 1]

Now the problem is the procedure threads!

Another change

int function\_turn;

int function\_prev;

active proctype p\_main()

{

function\_turn = 0;

function\_prev = -1;

L\_0:

do

:: true;

run p\_f1();

function\_prev = 0;

function\_turn = 1;

L\_turn\_0:

do

:: !(function\_turn == 0);

goto L\_turn\_0;

:: function\_turn == 0 -> break

od;

goto L\_0;

:: !1 -> break

od;

goto Return;

function\_turn = function\_prev;

Return: skip;

}

proctype p\_f1()

{

int i;

i=1;

L\_1:

do

:: i<10;

run p\_f2();

function\_prev = 1;

function\_turn = 2;

L\_turn\_1:

do

:: !(function\_turn == 1);

goto L\_turn\_1;

:: function\_turn == 1 -> break

od;

i++;

goto L\_1;

i++;

:: !(i<10) -> break

od;

goto Return;

function\_turn = function\_prev;

Return: skip;

}

proctype p\_f2()

{

run p\_f3();

function\_prev = 2;

function\_turn = 3;

L\_turn\_2:

do

:: !(function\_turn == 2);

goto L\_turn\_2;

:: function\_turn == 2 -> break

od;

goto Return;

function\_turn = function\_prev;

Return: skip;

}

proctype p\_f3()

{

run p\_f1();

function\_prev = 3;

function\_turn = 1;

L\_turn\_3:

do

:: !(function\_turn == 3);

goto L\_turn\_3;

:: function\_turn == 3 -> break

od;

goto Return;

function\_turn = function\_prev;

Return: skip;

}

inja momkene tamame cpu too hamoon main bemoone masalan

with local synch chan:

--code:

active proctype p\_main()

{

L\_0:

do

:: true;

int flag;

chan out\_p\_f1 = [0] of {int};

run p\_f1(out\_p\_f1);

out\_p\_f1 ? flag;

goto L\_0;

:: !1 -> break

od;

goto Return;

Return: skip;

}

proctype p\_f1(chan out\_p\_f1)

{

int i;

i=1;

L\_1:

do

:: i<10;

int flag;

chan out\_p\_f2 = [0] of {int};

run p\_f2(out\_p\_f2);

out\_p\_f2 ? flag;

i++;

goto L\_1;

i++;

:: !(i<10) -> break

od;

goto Return;

Return: skip;

out\_p\_f1 ! 1;

}

proctype p\_f2(chan out\_p\_f2)

{

int flag;

chan out\_p\_f3 = [0] of {int};

run p\_f3(out\_p\_f3);

out\_p\_f3 ? flag;

goto Return;

Return: skip;

out\_p\_f2 ! 1;

}

proctype p\_f3(chan out\_p\_f3)

{

int flag;

chan out\_p\_f1 = [0] of {int};

run p\_f1(out\_p\_f1);

out\_p\_f1 ? flag;

goto Return;

Return: skip;

out\_p\_f3 ! 1;

}

--spin result: (one path)

0: proc - (:root:) creates proc 0 (p\_main)

spin: model-prefer.pml:27, Error: place initialized chan decl of 'out\_p\_f2' at start of proctype

1: proc 0 (p\_main:1) model-prefer.pml:8 (state 1) [(1)]

2: proc 0 (p\_main:1) model-prefer.pml:10 (state 2) [flag = 0]

Starting p\_f1 with pid 1

3: proc 0 (p\_main:1) creates proc 1 (p\_f1)

3: proc 0 (p\_main:1) model-prefer.pml:11 (state 3) [(run p\_f1(out\_p\_f1))]

Selected: 1

4: proc 1 (p\_f1:1) model-prefer.pml:22 (state 1) [i = 1]

Selected: 1

5: proc 1 (p\_f1:1) model-prefer.pml:25 (state 2) [((i<10))]

Selected: 1

6: proc 1 (p\_f1:1) model-prefer.pml:27 (state 3) [flag = 0]

Selected: 1

Starting p\_f2 with pid 2

7: proc 1 (p\_f1:1) creates proc 2 (p\_f2)

7: proc 1 (p\_f1:1) model-prefer.pml:28 (state 4) [(run p\_f2(out\_p\_f2))]

Selected: 1

Starting p\_f3 with pid 3

8: proc 2 (p\_f2:1) creates proc 3 (p\_f3)

8: proc 2 (p\_f2:1) model-prefer.pml:43 (state 1) [(run p\_f3(out\_p\_f3))]

Selected: 1

Starting p\_f1 with pid 4

9: proc 3 (p\_f3:1) creates proc 4 (p\_f1)

9: proc 3 (p\_f3:1) model-prefer.pml:53 (state 1) [(run p\_f1(out\_p\_f1))]

Selected: 1

10: proc 4 (p\_f1:1) model-prefer.pml:22 (state 1) [i = 1]

Selected: 1

11: proc 4 (p\_f1:1) model-prefer.pml:25 (state 2) [((i<10))]

Selected: 1

12: proc 4 (p\_f1:1) model-prefer.pml:27 (state 3) [flag = 0]

Selected: 1

Starting p\_f2 with pid 5

13: proc 4 (p\_f1:1) creates proc 5 (p\_f2)

13: proc 4 (p\_f1:1) model-prefer.pml:28 (state 4) [(run p\_f2(out\_p\_f2))]

Selected: 1

Starting p\_f3 with pid 6

14: proc 5 (p\_f2:1) creates proc 6 (p\_f3)

14: proc 5 (p\_f2:1) model-prefer.pml:43 (state 1) [(run p\_f3(out\_p\_f3))]

Selected: 1

Starting p\_f1 with pid 7

15: proc 6 (p\_f3:1) creates proc 7 (p\_f1)

15: proc 6 (p\_f3:1) model-prefer.pml:53 (state 1) [(run p\_f1(out\_p\_f1))]

Selected: 1

16: proc 7 (p\_f1:1) model-prefer.pml:22 (state 1) [i = 1]

Selected: 1

17: proc 7 (p\_f1:1) model-prefer.pml:25 (state 2) [((i<10))]

Selected: 1

18: proc 7 (p\_f1:1) model-prefer.pml:27 (state 3) [flag = 0]

Selected: 1

Starting p\_f2 with pid 8

19: proc 7 (p\_f1:1) creates proc 8 (p\_f2)

19: proc 7 (p\_f1:1) model-prefer.pml:28 (state 4) [(run p\_f2(out\_p\_f2))]

kham kamel:

active proctype p\_main()

{

L\_0:

do

:: true;

run p\_f1();

// ?????????????

goto L\_0;

:: !1 -> break

od;

goto Return;

Return: skip;

}

proctype p\_f1()

{

int i;

i=1;

L\_1:

do

:: i<10;

run p\_f2();

// ?????????????

i++;

goto L\_1;

i++;

:: !(i<10) -> break

od;

goto Return;

Return: skip;

}

proctype p\_f2()

{

run p\_f3();

// ?????????????

goto Return;

Return: skip;

}

proctype p\_f3()

{

run p\_f1();

// ?????????????

goto Return;

Return: skip;

}

kham sade:

active proctype p\_main()

{

L\_0:

do

:: true;

run p\_f1();

// ?????????????

goto L\_0;

:: !1 -> break

od;

goto Return;

Return: skip;

}

proctype p\_f1()

{

int i;

i=1;

L\_1:

do

:: i<10;

run p\_f2();

// ?????????????

i++;

goto L\_1;

i++;

:: !(i<10) -> break

od;

goto Return;

Return: skip;

}

proctype p\_f2()

{

goto Return;

Return: skip;

}