

Dinning philosophers problems – SPIN

정세진
김재엽

1. Install
2. Dinning philosophers problem

INDEX

Install

- SPIN 설치
- Ubuntu 12.04 에 설치
- Spin 설치를 위해 필요한 추가 패키지
 - Bison, byacc-j, bison++, btyacc, byacc, perl-byacc, graphviz, tk, tcl8.4

```
* bison
* byacc-j
* bison++
* btyacc
* byacc
* perl-byacc
```

Install

The screenshot displays the Spin Version 6.4.2 interface. The main window shows a PMPL program for a dining philosophers problem with 5 philosophers. The program includes initialization, a process P, and a main loop. The Automata View on the right shows a state transition graph with nodes S1 through S18, representing different states of the system. The graph starts at state P and transitions through various states based on actions like 'eatings[i] = 1', 'thinking[i] = 1', and 'hungry[i] = 1'. The Automata View also includes a 'Select:' menu with options 'p_P' and 'init'. Below the main window, a terminal window shows the execution of the program, including verification and simulation commands.

```
1 #define NUM_PHIL 5
2 bool pthinking[NUM_PHIL], phungru[NUM_PHIL], eatings[NUM_PHIL] = false;
3 int forks[NUM_PHIL] = -1;
4
5 proctype P(int i){
6     int right = 1; int left = (i+1) % NUM_PHIL;
7
8     Think : atomic { eatings[i] = false; pthinking[i] = true; };
9     Hungru : atomic { eatings[i] = false; phungru[i] = true; };
10     if :: skip ;
11         atomic { forks[left] == -1 -> forks[left] = i };
12         atomic { forks[right] == -1 -> forks[right] = i };
13         :: skip;
14         atomic { forks[right] == -1 -> forks[right] = i };
15         atomic { forks[left] == -1 -> forks[left] = i };
16     fi;
17     Eatine : atomic { phungru[i] = false; eatings[i] = true; };
18     Done : forks[right] = -1; forks[left] = -1;
19 }
20
21 init{
22     run P(0);
23     run P(1);
24     run P(2);
25     run P(3);
26     run P(4);
27 }
28
29 }
```

```
35 ssibal2.pml:1
36 verification
37 /home/bkuh/spin/ssibal3.pml:1
38 verification
39 ssibal3.pml:1
40 verification
41 ssibal3.pml:1
42 verification
43 simulate/replau
44 spin -o3 -a ssibal3.pml
45 gcc -o pan pan.c
46 ./pan -D > dot.tmp
47 select init
48 spin -o3 -a ssibal3.pml
49 gcc -o pan pan.c
50 ./pan -D > dot.tmp
51 select p_P
52 verification
53 simulate/replau
54 verification
55 simulate/replau
```

Dinning philosophers problem – 5man

- modeling

```
1 #define MAX 5
2
3 mtype = {fork}
4
5 #define left forks[my_id]
6 #define right forks[(my_id + 1) %MAX]
7
8 bool pthinking[MAX], phungry[MAX], peating[MAX] = false;
9
10 chan forks[MAX] = [1] of {bit};
11
12 proctype phil(int my_id)
13 {
14     bit l_h = 0;
15     bit r_h = 0;
16
17     Think : atomic { peating[my_id] = false; pthinking[my_id] = true; };
18     Hungry : atomic { peating[my_id] = false; phungry[my_id] = true; };
19         if :: skip;
20             if
21                 :: l_h == 0 -> left?fork;
22                 l_h = 1;
23             fi
24             if
25                 :: r_h == 0 -> right?fork;
26                 r_h = 1;
27             fi
28         :: skip;
29             if
30                 :: r_h == 0 -> right?fork;
31                 r_h = 1;
32             fi
33             if
34                 :: l_h == 0 -> left?fork;
35                 l_h = 1;
36             fi
37         fi;
38     Eating : atomic { phungry[my_id] = false; peating[my_id] = true; };
39     Done : left!fork; l_h = 0; right!fork; r_h = 0; goto Think;
40 }
41
```

Dinning philosophers problem – 5man

- modeling

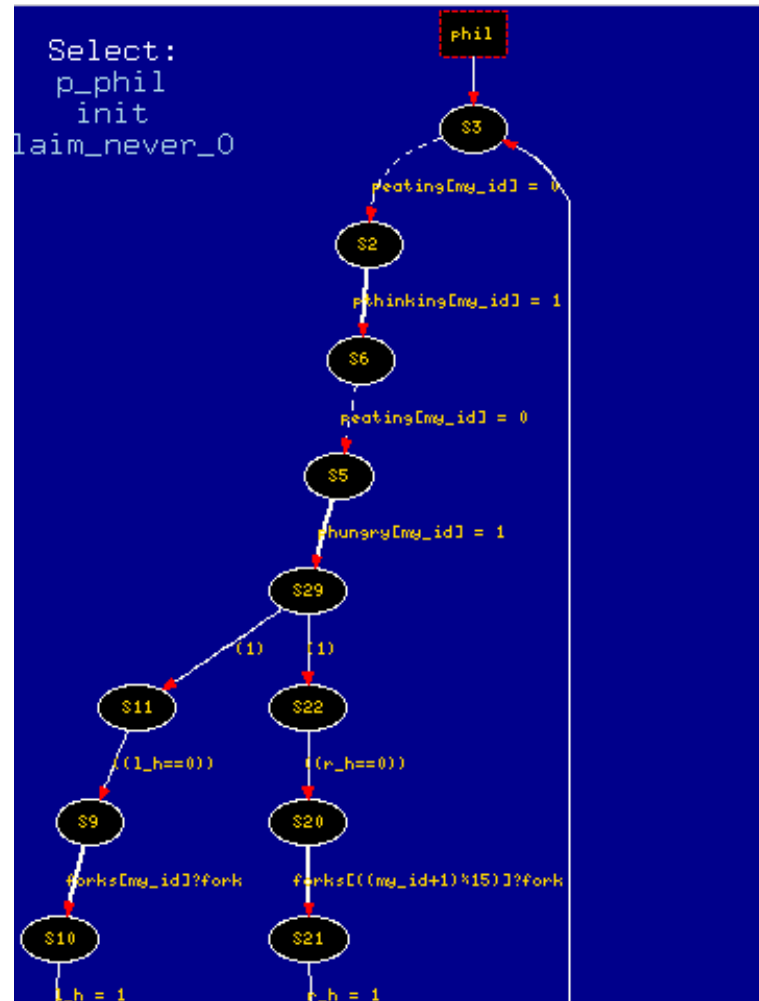
```
43 init
44 {
45     byte philosophers=0;
46     atomic{
47         do
48             ::philosophers<MAX ->
49                 forks[philosophers]!fork
50
51                 philosophers = philosophers+1;
52             ::philosophers==MAX->
53                 printf("start");
54                 break;
55         od
56         philosophers = 0;
57         do
58             ::philosophers<MAX ->
59                 run phil(philosophers);
60                 philosophers = philosophers+1;
61             ::philosophers==MAX->
62                 printf("start");
63                 break;
64         od
65     }
66 }
67
68
69 }
70
71 ltl p1{!<>(phil[1]:l_h != 0 && phil[2]:l_h != 0 && phil[3]:l_h != 0 && phil[4]:l_h != 0 && phil[0]:l_h != 0)};
72
```

Dinning philosophers problem – 5man

- Modeling
- `Lt1 p1{!<>(phil[1]:l_h != 0 && phil[2]:l_2 != 0 && phil[3]:l_h != 0 && phil[4]:l_h != 0 && phil[0]:l_h != 0)}`;

Dinning philosophers problem – 5man

- Model



Dinning philosophers problem – 5man

- Verification
 - Error 발생
 - 301 step

```
warning: only one claim defined, -N ignored
pan:1: invalid end state (at depth 300)
pan: wrote philo.trail

(Spin Version 6.4.2 -- 8 October 2014)
Warning: Search not completed
+ Partial Order Reduction

Full statespace search for:
  never claim      - (none specified)
  assertion violations +
  cycle checks    - (disabled by -DSAFETY)
  invalid end states +

State-vector 76 byte, depth reached 301, errors: 1
  275 states, stored
  40 states, matched
  315 transitions (= stored+matched)
  27 atomic steps
hash conflicts:    0 (resolved)

Stats on memory usage (in Megabytes):
  0.023 equivalent memory usage for states (stored*(State-vector + overhead))
  0.288 actual memory usage for states
  64.000 memory used for hash table (-w24)
  0.343 memory used for DFS stack (-m10000)
  64.539 total actual memory usage

pan: elapsed time 0.04 seconds
To replay the error-trail, goto Simulate/Replay and select "Run"
```

Dinning philosophers problem – 5man

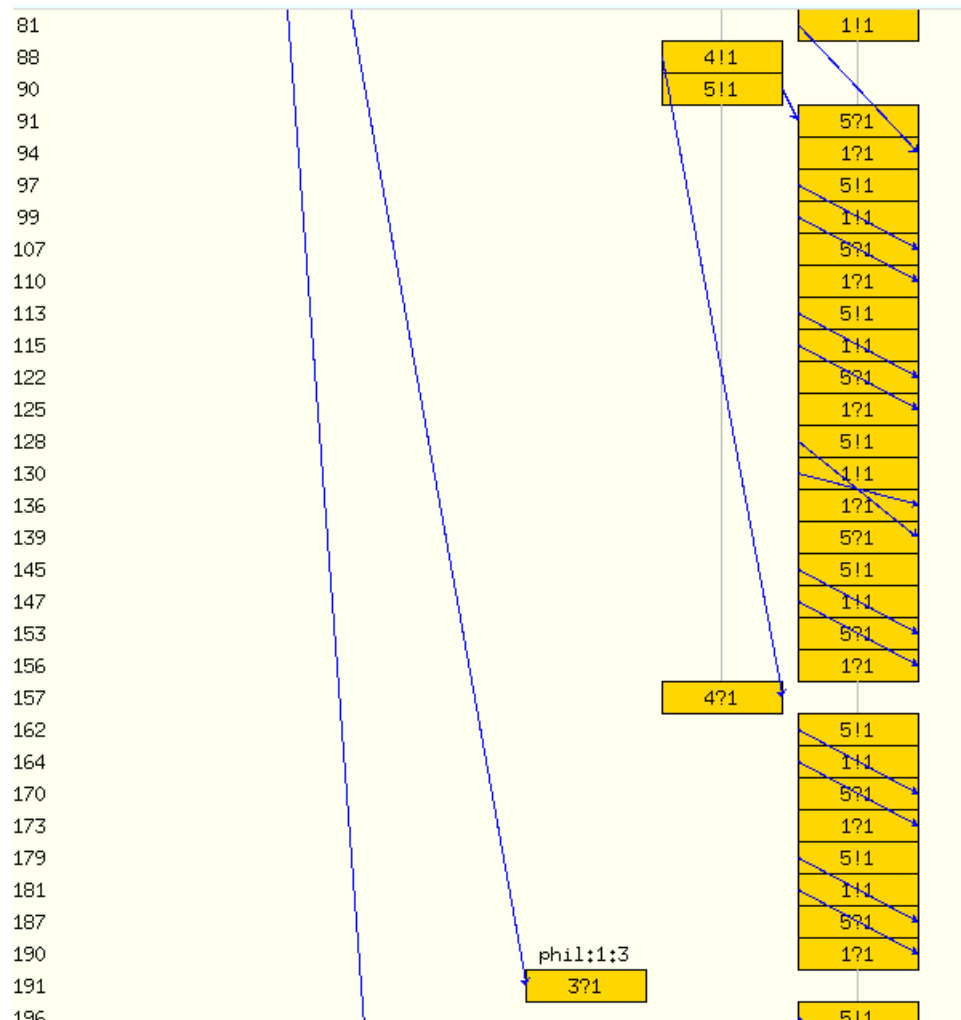
- Simulation
 - 변수의 상태 및 counter example

```
[variable values, step 301]
:init:(0):philosophers = 5
peating[0] = 0
peating[1] = 0
peating[2] = 0
peating[3] = 0
peating[4] = 0
phil(1):l_h = 1
phil(2):l_h = 1
phil(3):l_h = 1
phil(4):l_h = 1
phil(4):r_h = 0
phil(5):l_h = 1
phil(5):r_h = 0
phungry[0] = 1
phungry[1] = 1
phungry[2] = 1
phungry[3] = 1
phungry[4] = 1
pthinking[0] = 1
pthinking[1] = 1
pthinking[2] = 1
pthinking[3] = 1
pthinking[4] = 1

295: proc 5 (phil:1) philo:21 (state 8) [((l_n==0))]
296: proc 5 (phil:1) philo:21 (state 9) [forks[my_id]?fork]
297: proc 5 (phil:1) philo:22 (state 10) [l_h = 1]
298: proc 1 (phil:1) philo:21 (state 9) [forks[my_id]?fork]
299: proc 5 (phil:1) philo:25 (state 11) [((r_h==0))]
300: proc 1 (phil:1) philo:22 (state 10) [l_h = 1]
301: proc 1 (phil:1) philo:25 (state 11) [((r_h==0))]
spin: trail ends after 301 steps
#processes: 6
301: proc 5 (phil:1) philo:25 (state 12)
301: proc 4 (phil:1) philo:25 (state 12)
301: proc 3 (phil:1) philo:25 (state 12)
301: proc 2 (phil:1) philo:25 (state 12)
301: proc 1 (phil:1) philo:25 (state 12)
301: proc 0 (:init::1) philo:70 (state 21)
6 processes created
Exit-Status 0
```

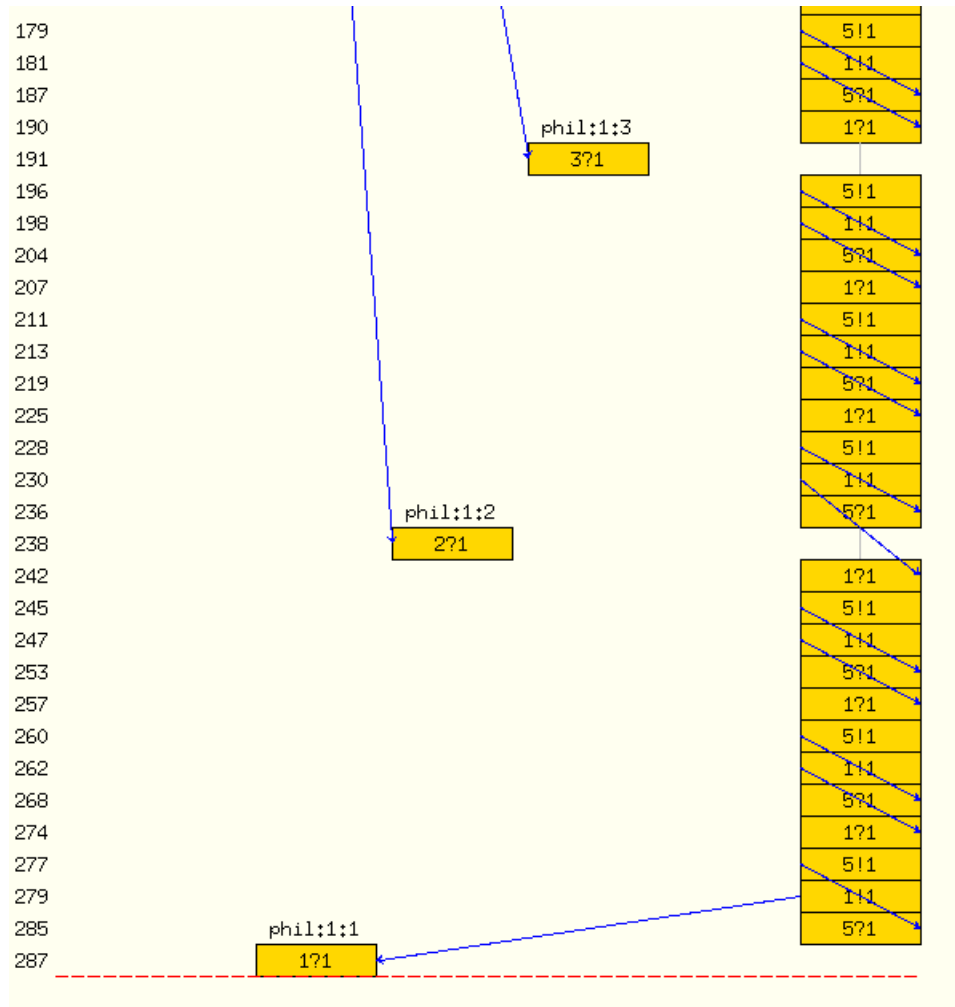
Dinning philosophers problem – 5man

- 추적



Dinning philosophers problem – 5man

- 추적



Q & A

END