

- 1 Zgodnie ze specyfikacją wymodelowany został czas dyskretny wraz z funkcjami:
  - **urgent(inst1, instr2, T)**, która wymusza wykonanie instrukcji *instr2* najpóźniej *T* sekund po wykonaniu instrukcji *instr1*.
  - **sleep(T)**, która wstrzymuje proces na *T* sekund.
- 2 protokół wzajemnego wykluczania dla wyżej opisanego czasu ma w Promeli następującą postać:

```
//algorithm
active [N] proctype P() {
    byte id = _pid;
    do
        :: urgent(kto == 0, kto = id, T1);
        sleep(T2);
pre_cs: kto == id;
cs:    skip;
        kto = 0;
    od
}
```

implementacja rozwiązania zadania (bez formuł *LTL* znajduje się w pliku *timer.pml*)

- 3 Warunek na  $T1$  i  $T2$  gwarantujący wzajemne wykluczanie to:  $T1 < T2$ .  
Zauważmy, że dla  $T1 \geq T2$  nie spełniony jest warunek bezpieczeństwa. Rozważmy następujący bieg procesów:

- $P_1$  i  $P_2$  przechodzą warunek  $kto == 0$  (mają  $T1$  jednostek czasu na wykonanie operacji przypisania);
- następnie obsługiwany jest tylko proces  $P_1$ , który wykonuje przypisanie  $kto = 1$ , zasypia na  $T2$  sekund, budzi się, sprawdza warunek  $kto == id$  i wchodzi do sekcji (wykonanie zajmuje  $T2$  sekund);
- sterowanie przechodzi na drugi proces  $P_2$ , któremu pozostało  $T1 - T2 \geq 0$  sekund na wykonanie przypisane, czyli warunek *urgent* jest spełniony. Następuje przypisanie, zawieszenie na sleep, sprawdzenie warunku  $kto == id$  i wejście do sekcji;
- warunek bezpieczeństwa został naruszony.

Z powyższego przykładu wnioskujemy, że warunek  $T1 < T2$  jest konieczny, ale i wystarczający, gdyż gwarantuje, wykonanie przypisania  $kto = i$ , na co najmniej jedną jednostkę czasu przed wybudzeniem procesu sprawdzającego warunek  $kto == j$ .

- 4 Dla wartości:

$T1 = 2$ ;  $T2 = 3$ ;  $N = 4$ ;

sprawdzano kolejne własności przy pomocy podanych formuł *LTL*. Ze względu na zachowaną symetrię procesów formuły pisane są dla ustalonej pary procesów: (1,2) (jeśli przeplot istnieje, to zostanie on wykryty także w szczególnym przypadku). W celu wykonania odpowiedniego testu należy wywołać podaną komendę.

- *wzajemne wykluczanie - spełnione:*

```
ltl { ([ ! (P[1]@cs && P[2]@cs) ) }
```

```
$ ./spina timer_LTL_1.pml
```

- *żywołność - spełnione:*

```
ltl {[ (P[1]@pre_cs -> <> (P[1]@cs) )}]
```

```
$ ./spina timer_LTL_2.pml
```

- *jeśli któryś proces rozpocznie protokół wstępny, to po upływie nie więcej niż  $T1 + T2$  jednostek czasu któryś proces znajdzie się w sekcji krytycznej - niespełnione*

```
ltl { !( ( !P[1]@cs && !P[2]@cs && !P[3]@cs && !P[4]@cs ) U  
        ( (tmp_clock == T1 + T2 + 1) && protocol_started == true) ) }
```

```
$ ./spina timer_LTL_3.pml
```

- *ten proces, który jako pierwszy ustawi zmienną kto, wejdzie jako pierwszy do sekcji krytycznej - niespełnione*

```
ltl { (!P[2]@ch_id U P[1]@ch_id) -> (!P[2]@cs U P[1]@cs) }
```

```
$ ./spina timer_LTL_4.pml
```