

Házi feladat (Modellellenőrzés)

A feladat címe: **Lynch-Shavit algoritmus**

Konzulense: **Tóth Tamás**

A probléma leírása

A Lynch-Shavit algoritmus [1] egy megosztott memória segítségével kölcsönös kizárást megvalósító protokoll, mely lehetővé teszi N folyamat számára egy globális erőforráshoz való konfliktusmentes hozzáférést.

Globális változók és konstansok:

```
// x is an integer between 0 and N
[0..N] x := 0;
bool y := false;
```

Az i -edik folyamat pszeudokódja:

```
// Entry protocol
L:
if (x ≠ 0) {
    goto L;
}
x := i;
pause(delay);
if (x ≠ i) {
    goto L;
}
if (y) {
    goto L;
}
y := true;
if (x ≠ i) {
    goto L;
}

// Critical section

// Exit protocol
y := false;
x := 0;
```

Készítse el a protokoll modelljét az alábbi elvárások figyelembe vételével!

- A modellezésnél egy értékadás (pl. $x := i$) végrehajtása, illetve egy atomi formula (pl. $x \neq i$) kiértékelése tekinthető atomi műveletnek. Az ennél összetettebb műveletek végrehajtása nem atomi.
- A folyamatok ütemezéséről egy ütemező gondoskodik, amelyet szintén modellezni kell. Az ütemező minden időegységben léptet egy folyamatot. Az ütemező működésében minden M -edik időegységben teljesülnie kell, hogy az utolsó M időegységben minden folyamat legalább K -szor, de legfeljebb L -szer lépett (tehát $L \cdot N \geq M$ és $M \geq K \cdot N$). Ezt a kényszert leszámítva az ütemező a léptetendő folyamatot nemdeterminisztikusan választja.
- A modell legyen *delay*-ben, K -ban, L -ben, M -ben és N -ben paraméterezhető.

Az ellenőrzendő követelmények

Temporális logikai kifejezések és modellellenőrzés segítségével igazolja az alábbi követelmények teljesülését egy *legalább három folyamatot* tartalmazó rendszeren! A követelmények nem teljesülése esetén ellenpélda segítségével magyarázza meg, miért nem teljesül az adott követelmény.

1. A rendszer holtpontmentes.
2. A kölcsönös kizárás megvalósul, azaz egyszerre legfeljebb egy folyamat tartózkodhat a kritikus szakaszban.
3. A kritikus szakasz elérhető valamelyik folyamat számára.
4. A rendszer éhezésmentes, azaz minden folyamat előbb-utóbb eljut a kritikus szakaszba.

Tippek

Az ütemező modellezéséhez gondolja meg a *select* konstrukció használatát!

Az ütemező nélküli modell is beadható a hozzá kapcsolódó ellenőrzésekkel, de az nem tekinthető teljes megoldásnak.

Hivatkozások

- [1] N. Lynch and N. Shavit: Timing-based mutual exclusion. Real-Time Systems Symposium, pp. 2-11, 1992.