

Házi feladat (Modellellenőrzés)

A feladat címe: **Processzorok színezése**

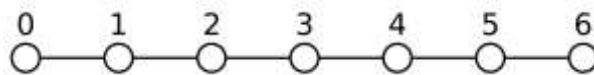
Konzulense: **Marussy Kristóf**

A modellezendő probléma

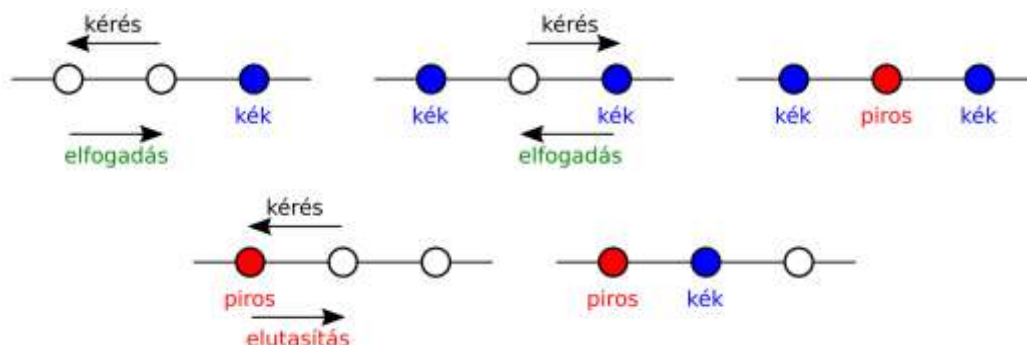
Készítse el az alábbi probléma modelljét az UPPAAL eszköz időzített automata formalizmusát használva!

Többprocesszoros rendszerekben, amikor két processzor szeretne egy objektumot egyszerre elérni, szükséges lehet, hogy meghatározzuk, melyik processzornak van elsőbbsége. Erre egy módszer a *szimmetria megtörése*.

A feladatban egy $N = 7$ processzorból álló rendszerben vizsgálunk egy egyszerű szimmetria-megtörő algoritmust. A processzorok egy 7 csúcsú út mentén vannak egymással összekötve (a 0. és a 6. processzornak csak egy szomszédja van, a többi processzor az út belső csúcsa). Az algoritmus célja az, hogy kék és piros színekkel színezzék ki az út csúcsait úgy, hogy két szomszédos csúcs ne legyen piros. A **pirosra** színezett processzorok férhetnek hozzá az út éleihez rendelt erőforrásokhoz (pl. nagyobb méretű osztott memória), míg a **kék** processzoroknak várakozniuk kell. Az algoritmus során a szomszédos processzorok üzeneteket válthatnak egymás között; az üzenetek elveszését vagy egyéb hibákat nem kell modellezni.



A processzorok az algoritmus során valamilyen sorrendben felébredhetnek, akár egyszerre. A felébredő processzorok először a baloldali szomszédjuknak küldenek egy kérést. Ha a szomszéd válaszában elfogadja a kérést, akkor a processzor jobboldali szomszédjának is kérést küld. A 0. és a 6. processzor értelemszerűen csak egy szomszédjának kell kérést küldjön. Ha a szomszédok rendre elfogadták a kéréseket, akkor a processzor piros színű lesz. Ha valamelyik szomszéd elutasítja a kérést, a processzor kék színű lesz és több kérést már nem küld.



Ha egy processzor még nem kapott színt, a bejövő kérésekre elfogadóan válaszol, kékké válik és szintén nem indít több kérést. A már kékre színezett processzorok minden kérést elfogadnak, míg a piros processzorok minden kérést elutasítanak. Ez azt jelenti, hogy egy processzor akkor színezhető pirosra, ha egyik szomszédjától sem kap kérést, de az ő kérését mindkét szomszédja elfogadja. Ügyeljünk arra, hogy egy processzor akkor is kaphat választ egy szomszédjától az annak küldött kérésére, ha a másik szomszédjától kapott kérés hatására már kékre színeződött. Kezeljük ezt az esetet is!

Az eljárást addig folytatjuk, amíg minden processzor kék vagy piros színűre nem lesz színezve. A színezést akkor tekintjük helyesnek, ha nincs két szomszédos piros processzor.

Megjegyzés: A gyakorlatban ennél hatékonyabb szimmetria megtörő technikák is léteznek, lásd például a 30.4.3. fejezetet és a 30.5. fejezetet a következő könyvben: T. H. Cormen, C. E. Leiserson és R. L. Rivest: Algoritmusok. Műszaki Könyvkiadó, Budapest, 2003.

Az ellenőrzendő követelmények

Az alábbi feladatok közül válasszon ki legalább hármat, és temporális logikai kifejezések és modellellenőrzés segítségével oldja meg ezeket (ha egy követelmény nem teljesül, akkor ellenpélda segítségével magyarázza meg, miért nem teljesül)!

Tipp: A modellben felvehető egy monitor processz, ami hozzáfér a processzorok állapotához és ki tudja értékelni adott kifejezések (függvények) értékét.

1. Igazolja, hogy a szimmetria-megtörő algoritmus előbb-utóbb leáll.
2. Igazolja, hogy soha nem színezünk pirosra két szomszédos processzort.
3. Igazolja, hogy mindig lesz legalább egy piros színű processzor.
4. Határozza meg (továbbra is a modellellenőrző segítségével), hogy legfeljebb hány processzort lehet pirosra színezni az algoritmussal!