

## 2. gyakorlat – Állapot alapú modellezés

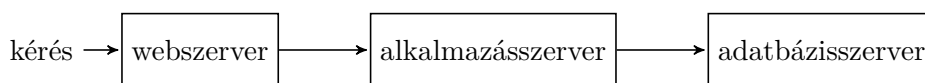
### 1. Közlekedési lámpa

Közlekedési lámpát vezérlő elektronikát tervezünk.

- Készítsük el egy egyszerű piros–sárga–zöld közlekedési lámpa olyan állapotterét, amely kellően finom ahhoz, hogy a lámpák vezérlését ez alapján lehessen végezni! Győződjünk meg arról, hogy az állapottér kizárólagos és teljes!
- A három égőnek külön-külön mi az állapottere? Milyen absztrakciós viszony áll fent a lámpa és az egyes égők állapottere közt? Hogy viszonyul a lámpa állapottere a három állapotváltozó direkt szorzatához?
- Mik az érvényes állapotátmeneti szabályok? Készítsük el az (egyszerű) állapotgráfot!
- Hogyan fejezhető ki ugyanez a működés tömörebben hierarchikus állapotokkal?
- Amikor a lámpa elektromos fogyasztását vizsgáljuk, csak az érdekel, hogy a három égőből hány ég egyszerre. Absztraháljuk az állapotgépet úgy, hogy az állapotokat csak a fogyasztásuk különböztesse meg!
- A piros jelzés végén van egy olyan időszak, amikor a merőleges gyalogosátkelő zöld lámpája már villog. Finomítsuk úgy az (absztrakció előtti) állapotgráfot, hogy ez az állapot elkülöníthető legyen!
- Egy út mentén 10 jelzőlámpa található, egyenként 4 állapottal. Legfeljebb hány állapota lehet a teljes rendszernek? Várhatóan kell-e minden állapotot kezelniük?

### 2. Háromrétegű architektúra

Egy informatikai rendszert szeretnénk modellezni, melyet *háromrétegű architektúra* valósít meg az alábbiak szerint:

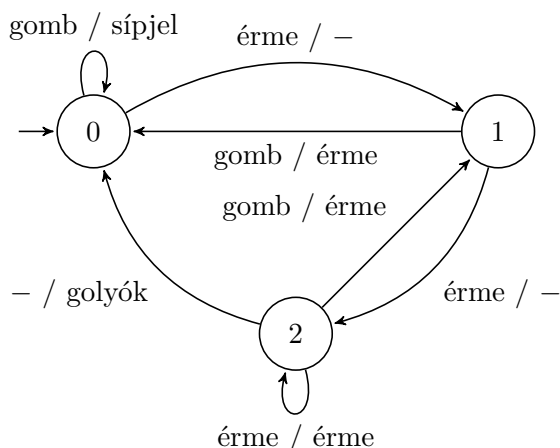


Háromrétegű kiszolgáló infrastruktúránk viselkedésének modellezésére megfelelő állapotterek-e az alábbiak?

- { Webszerver dolgozik, Alkalmazáserver dolgozik, Adatbázisszerver dolgozik }
- { Leállítva, Tétlen üzemel, Aktívan dolgozik }
- $\mathbb{N}$  (mint a pillanatnyilag feldolgozás alatt álló kérések száma)
- { A kérés feldolgozása még nem kezdődött el, A szerverek épp dolgoznak a kéréssel, A kérés kiszolgálása befejeződött }
- { Igaz } \*

### 3. „Mit ír ki?”

Tekintsük az  $M$  állapotgépet!



Megjegyzés: a „–” jel a bemeneti jel pozíciójában spontán állapotátmenetet jelöl, a kimenet helyén pedig kimenet hiányát, egyik esetben sem a digitális áramkörtervezésben használt *don't care* szimbólum.

- Milyen valós rendszer lehet az  $M$  állapotgép mögött, hogyan működik?
- Determinisztikus-e ez a viselkedésmoделl? Hozzávehető-e, ill. elhagyható-e egyetlen állapotátmeneti szabály, hogy ez megváltozzon?
- Absztraháljuk az  $M$  állapotgráfot a  $\{\{0\}, \{1, 2\}\}$  állapotpartíció szerint!
- Hol és milyen jellegű nemdeterminizmus figyelhető meg az így kapott absztrakt modellen?

## 4. Érintőképernyős billentyűzet

Modellezzük állapotgéppel egy mobiltelefon érintőképernyőjére tervezett, az előadáson is bemutatott virtuális billentyűzetet! A billentyűzeten egyszerre vagy a kisbetűk, vagy a nagybetűk, vagy a számok és fontosabb szimbólumok, vagy ritkább szimbólumok láthatóak. Az elsődleges üzemmódváltó gomb a betűk és a számok/szimbólumok beírása között vált, a másodlagos üzemmódváltó pedig ezen kategóriákon belül. Létezik továbbá egy olyan nagybetűs állapot is, amely egy betű leütése után automatikusan kisbetűsre vált. Vegyük figyelembe a bal felső gombot (q/Q/1/=), ill. a két üzemmódváltó gombot mint bemenetet, és a szövegmezőbe begépett karaktereket mint kimenetet!

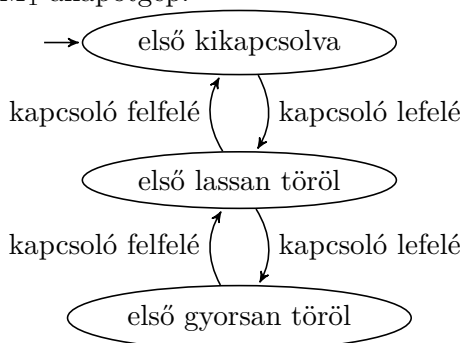


## 5. Ablaktörlő

Egy autóban az első ablaktörlőnek három állapota van (*első kikapcsolva*, *első lassan töröl*, *első gyorsan töröl*), a hátsó ablaktörlőnek kettő (*hátsó kikapcsolva*, *hátsó töröl*). Az első ablaktörlő működését az  $M_1$  állapotgép, a hátsóét az  $M_2$  állapotgép modellezi.

- Készítsük el az  $M_1$  és  $M_2$  állapotgépek aszinkron szorzatát!
- Hány állapota és átmenete van az így kapott modellnek?
- (Kiegészítő feladat) Kifejezhető-e a kapott állapotgépen, ill. a komponensre vetített modellek segítségével olyan kocsi, ahol a hátsó ablaktörlő csak akkor kapcsolható be, ha megy az első is?

$M_1$  állapotgép:



$M_2$  állapotgép:

