Állapot alapú modellezés

1. feladat

Fogalmak:

- **Adatbázisszerver**: hosszútávon tároljuk az adatokat
- **Alkalmazásszerver**: az üzleti logikáért felelős alkalmazást futtatja
- **Webszerver**: megjelenítést felelős, generálja a HTML oldalakat

Válaszok:

a) Nem. A három közül *pontosan egynek* kell igaznak lennie minden időpillanatban.
b) Nem. Egyszerre több gép is dolgozhat.
c) Igen. Természetesen elképzelhetők további állapotok is, pl. *hibernált*.
d) Igen, ez egy végtelen állapotteret eredményez. A kizárólagosság mellett fontos vizsgálni, hogy teljes-e. Pl. 3,5 vagy −9 kérés nem lehet a rendszerben, ezért teljes.
e) Igen, ha egyszerre egy kérés lehet a rendszerben. Ezzel azonban a modellünk nem lesz alkalmas terhelések vizsgálatára.
f) Igen, ez teljes és kizárólagos, ezért alkalmas állapotpartíciónak. Ez az állapotmentes modell. Állapotmentes protokollokat (pl. a HTTP-t) érdemes lehet így modellezni.

2. feladat

a) {piros, sárga, zöld, piros-sárga, villogó sárga, kikapcsolt}, röviden: $S = \{p, s, z, ps, \bar{s}, \emptyset\}$. Ez kizárólagos és teljes. A kezdeti állapot: p.

b) $S_p = \{p, \emptyset\}$, $S_s = \{s, \bar{s}, \emptyset\}$, $S_z = \{z, \emptyset\}$. Másik színű filccel érdemes berajzolni az egy és a három állapotváltozós modell közötti kapcsolatokat, pl.

   - p esetén $S_p = \{p\}, S_s = \{\emptyset\}, S_z = \{\emptyset\}$
   - direkt szorzat: $S_p \times S_s \times S_z$, $2 \times 3 \times 2 = 12$ elemű lesz.

   Ebből természetesen nem minden állapot fordul elő: $S \subset S_p \times S_s \times S_z$

	$S_p \times S_s \times S_z$	s	$\bar{s}$	$\emptyset$
p	z	psz	$p\bar{s}z$	pz
	$\emptyset$	$\underline{ps}$	$p\bar{s}$	$\underline{p}$
$\emptyset$	z	sz	$\bar{s}z$	$\underline{z}$
	$\emptyset$	$\underline{s}$	$\underline{\bar{s}}$	$\underline{\emptyset}$

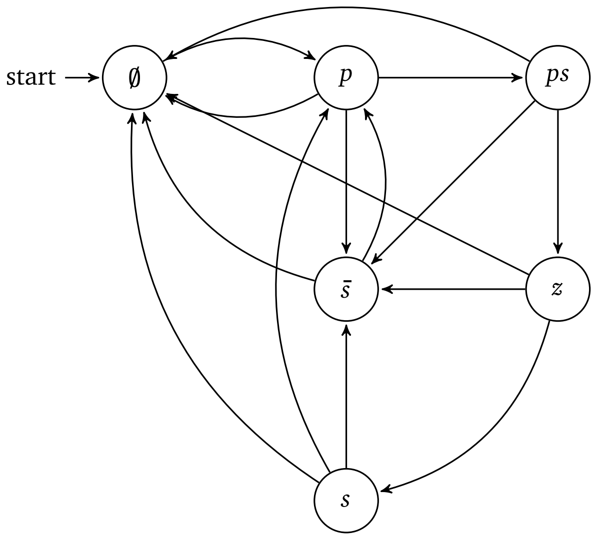
   Az aláhúzott állapotok az eredeti állapottérben is megjelennek.

c) Érvényes állapotátmeneti szabályok állapotgráfja a 6 állapot között. A kezdőállapotot „villámmal” szokás jelölni (ld. diasor). Sok tervezői döntéssel szembesülünk:

- Kötelezően pirosnak kell lennie bekapcsolás után? (igen)
- Csak a szabályos működést modellezzük? (igen)
- Ha elmegy az áram, az hibás működés? (vegyük úgy, hogy csak tervezett áramszünetek vannak)
- Pirosból pirosba mehet? (nem rajzoljuk fel, mert semmi sem történik)
- Piros-sárgából mehet pirosba, pl. baleset esetén?

Az állapotgép bárhonnan kerülhet kikapcsolt ($\emptyset$), ill. villogó ($\bar{s}$) állapotba (utóbbi esetén kivéve kikapcsolt állapotból).

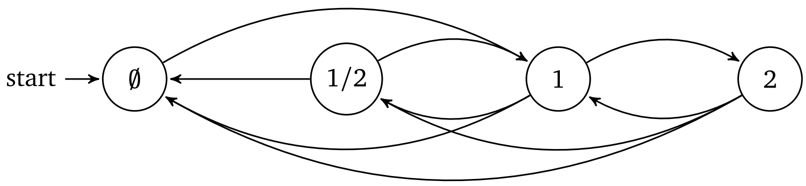
Egy lehetséges megoldás:



d) Bontsuk fel a piros állapotot két állapotra: piros és a gyalogosoknak folyamatos zöld (p_{fz}), piros és a gyalogosoknak villogó zöld (p_{vz}). Az állapotgráfra is vigyük át a változtatást: a két állapot között $p_{fz} \to p_{vz}$ átmenet van, a többi értelemszerűen berajzolandó.

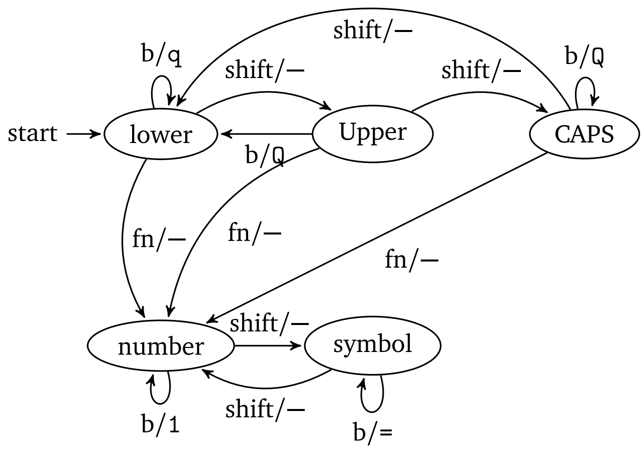
Itt tovább lehet gondolni azt, hogy szükség van-e olyan állapotra, amikor az autósoknak és a gyalogosoknak is piros a lámpa. Ha az előző részben úgy döntöttünk, hogy a lámpának kötelezően pirosnak kell lennie bekapcsolás után, akkor itt is érdemes ezt megvalósítani: ebben az esetben a piros állapotot három állapotra kell felbontani: p_p, p_{fz}, p_{vz}.

e) 4-állapotú állapotpartíció, fogyasztások: $0, 1/2, 1, 2$ (egyszerre mindhárom nem világíthat). Az 1-es állapotra rajzolhatnánk hurokélet (pl. zöld → sárga → piros esetén mindig csak egy lámpa világít), de ezt nem tesszük meg, mivel nem figyelhető meg kívülről. Többszörös éleket (ha nincs rajtuk különböző bemenet/kimenet) nem rajzolunk be.



3. feladat

Fontos, hogy csak egy billentyűt kell feltüntetnünk az állapotgépen. Egy lehetséges megoldás az alábbi:



4. feladat

A teljes állapottér legfeljebb 4^{10} állapotból áll. $4^{10} = 2^{20} = (2^{10})^2 \approx 10^6$. Természetesen nem biztos, hogy valóban ennyiből fog állni: az első feladatban a három állapotváltozó direktszorzata egy 12 állapotú állapotteret feszített ki, de a valódi állapotgép ezeknek csak egy valódi részhalmazán működött.

5. feladat

a) Egy úgy dönthető el, ha végignézünk minden állapotot és a kimenő élekre ellenőrizzük, hogy mindegyikből csak egy megy-e ki egy adott bemenetre. Ennek megfelel az állapotgépünk, azonban van olyan állapotátmenet, amelynél — karakter van a bemeneten.

   A — karakter (formális nyelvekben gyakran ε) jelentése: bemeneten „nem olvasunk semmit” (bármikor megtörténhet az állapotátmenet), ill. kimeneten „nem adunk ki semmit”. Ilyen van a c és az a állapotok között, ezért a modell nemdeterminisztikus, c állapotban spontán z-t adhat ki.

   Fontos, hogy a — jelentése nem ugyanaz, mint a digitális technikában – ott „don’t care”-t jelent, vagyis bemenet esetén olvasunk, de mindegy, hogy mit; kimenet esetén írunk, de mindegy, hogy mit.

   Az állapotgép mindenképp el fog jutni c-be, így (prioritások nélkül) nem tudjuk determinisztikussá tenni állapotok/átmenetek felvételével.

b) Az absztrakciót definiáló fv. szerint mindig csak egyféleképp lehet csinálni. Az a és a b állapotok összeolvadnak, az $1/y$ állapotátmeneti szabályok (élek) is összeolvadnak.

c) A lehetőségek száma a következő. A kezdőállapot lehet e és f is (2). A három ki-, ill. bemenő állapotátmeneti szabály külön-külön háromféle lehet változhatnak (pl. c-ből 0-ra e-be, f-be vagy mindkettőbe menjen), ez $3^3 = 27$. Az $1/y$ hurokél 4 helyre is kerülhet, de legalább egy helyre kerülnie kell. Ezek behúzására összesen $2^4 - 1 = 15$-féle lehetőség van (mindegyik egyformán „jó”). Ez összesen $2 \times 27 \times 15 = 810$ lehetőség.

   Ennek magyarázata, hogy a finomított modell több információt tartalmaz, mint az absztrakt.

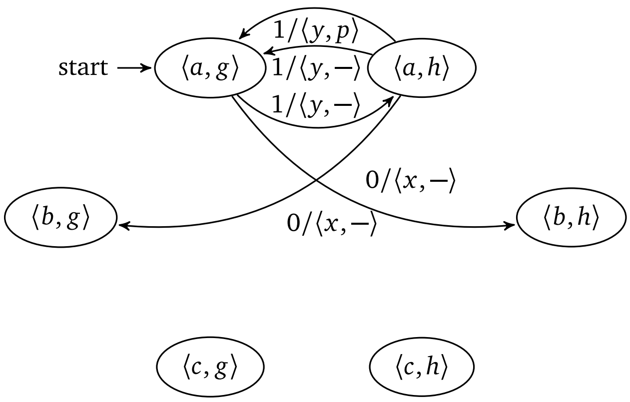
d) A b) részhez hasonlóan ismét csak egy lehetőségünk van.

e) Azon átegmenetek esetén, ahol z van a kimeneten ($c \to a, c \to c, c \to b$), háromféle lehetőségünk van: csak z_1-es átmenetet veszünk fel, csak z_2-es átmenetet veszünk fel vagy mindkét átmenetet felvesszük (párhuzamos élekként). Így összesen $3^3 = 27$ lehetőségünk van.

6. feladat

Két állapotgép partícióinak szorzata az állapothalmazok Descartes-szorzata. A gépek ugyanazt az i inputot olvassák mindketten.

Szinkron szorzat: az állapotgépeket egyszerre léptetjük. Nagyon fontos, hogy minden állapotra megvizsgáljuk, hogy 0, 1, ill. — bemenetre milyen állapotba kerülnek és az átmenet során milyen kimenetet adnak. Az alábbi ábrán az $\langle a, g\rangle$ és az $\langle a, h\rangle$ állapotokat vizsgáltuk meg a lehetséges bemenetekre.



Felmerül a kérdés, hogy az M_1 állapotgép c állapot — bemenetű hurokéle bekerül-e a szinkron szorzatba. Nem, mert az M_2 állapotgépnek nincs olyan állapotátmenete, ami — bemenetre történne meg.

Aszinkron szorzat: minden állapotátmeneti él lemásolódik a szorzatállapotokra. Ezt két állapotgép egyszerűen elkészíthetjük, ha felvesszük az állapotok Descartes-szorzatát és az átmeneteket úgy rajzoljuk be, hogy a M_1 átmenetei függőlegesen, M_2 átmenetei vízszintesen szerepelnek az ábrán.