

2. ZH

A **kérdések**, válaszok és **magyarázatok** lehetőleg jól különüljenek el.

1. ZH kidolgozás 2014

Kidolgozott feladatok:

https://www.dropbox.com/sh/34n3so80smluf4e/AADtPeFi3MMbJDrTCjCDASela/FM-2015_ZH1t_gyakorlo_feladatok_megoldassal.pdf?dl=0

Kidolgozott feladattal kapcsolatban kérdések:

Első feladat:

Mi az a G, GF, X?

Állapotokra, illetve útvonalakon található állapotokra vonatkozó operátorok.

Gp: (Global) globálisan igaz p, minden állapotban igaz p, minden állapotban fennáll p.

Fp: (Future) jövőben valamikor egy állapotban igaz lesz p, fenn fog állni p valamikor egy állapotban.

Xp: (neXt) a következő állapotban igaz lesz p, fenn fog állni a következő állapotban p.

GFp: mindenhol igaz az, hogy bármely állapotból elérhető lesz a jövőben egy olyan állapot, ahol igaz lesz p, fenn fog állni p.

Második feladat:

“Csak úgy hajtható végre szimuláció, ha volt bemelegítés a hűtőberendezés normál működése mellett.”

$G(X \text{ szimuláció}) \rightarrow (\text{bemelegítés} \ \&\& \ \text{normál})$

Azt jól érzem, hogy az X a “volt” szócska miatt van ott? Igen, mert normál állapot ÉS bemelegítés után a következő lépés lesz a szimuláció.

(Ergő kicsit átforgatva a mondatot, ha lesz szimuláció, akkor kellett bemelegítés, és normál működés)

Minden állapotra igaz (G)/mindig igaz, hogy a következő lépésben (X) akkor állhat fenn a szimuláció (lehet igaz)/lehet szimuláció, ha normál működés ÉS bemelegítés is fennállt (igaz volt egyszerre).

Elmélet összeszedve:

<https://docs.google.com/document/d/10SuQA5F8wksmkRZZh18kLgT2srPHFeAoX3ebU4hohx8/edit>

Kérdések:

2db kiadott mintazh feladatok:

2. Bináris döntési diagramok (14 pont)

Adottak az alábbi logikai függvények:

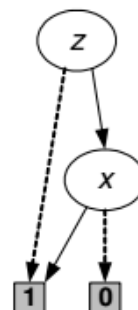
$f :=$ lásd a jobboldali ábrát

$g := (x \vee y) \wedge (y \vee z)$

$m := f \wedge g$

(A szaggatott vonal tartozik a változó 0 behelyettesítéséhez, a folytonos pedig az 1-hez.)

Adott az f logikai függvény ROBDD-je:



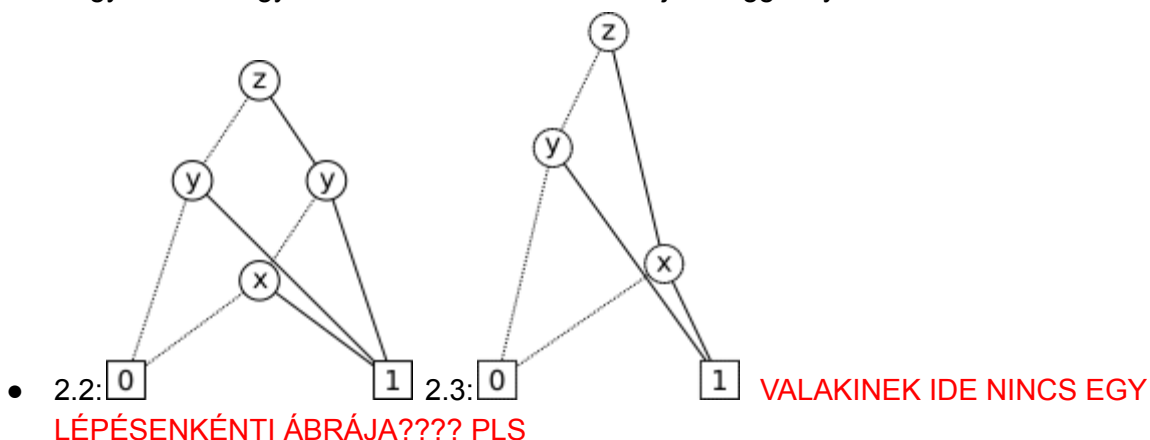
2.1. Írja fel az f logikai függvény képletét algebrai alakban az ROBDD ábrából leolvasva! 3 pont

2.2. Rajzolja fel a lenti üres helyre a g logikai függvényt redukált rendezett bináris döntési diagram (ROBDD) alakban! 3 pont

2.3. Számítsa ki és ábrázolja az m logikai függvényt ROBDD alakban! A számítást közvetlenül az ROBDD-ken értelmezett műveletekkel végezze el! A függvényeket az f függvény ábrájával konzisztens változósorrendben ($z \rightarrow y \rightarrow x$) értékelje ki. 8 pont

2.1: $!z || x \Rightarrow$ EZ NEM: $!z || z \&\& x$??? de, $!z || (z \square x)$ EHHEZ EGY MAGYARÁZÓ ÁBRA VKI?

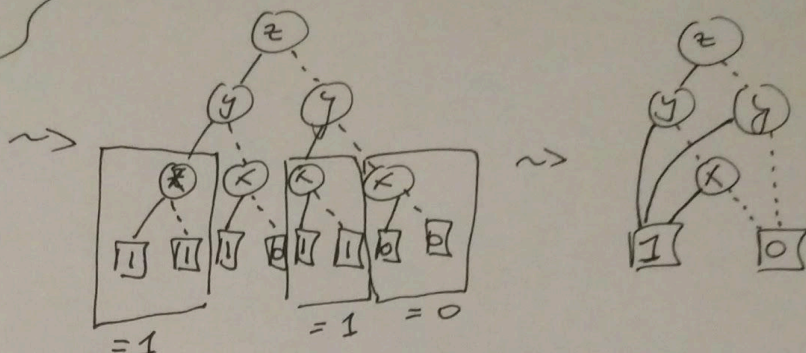
De ezek egyenlőek! Vagy van valami "hivatalos" formája a függvényeknek?



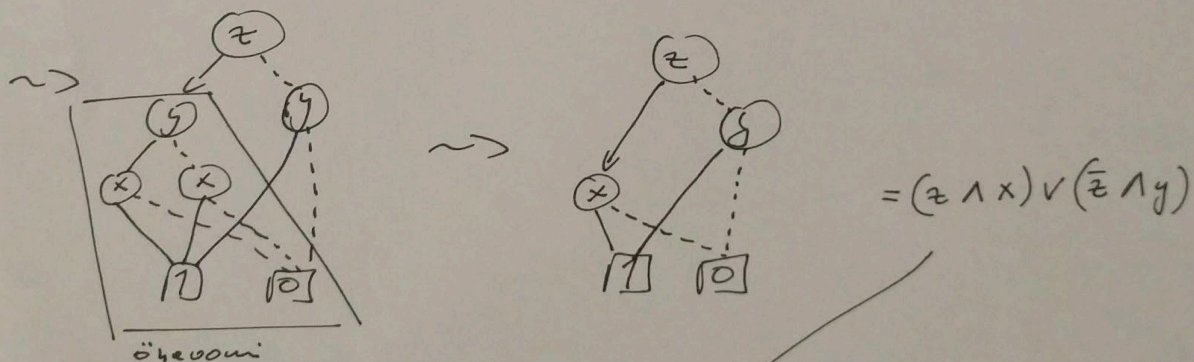
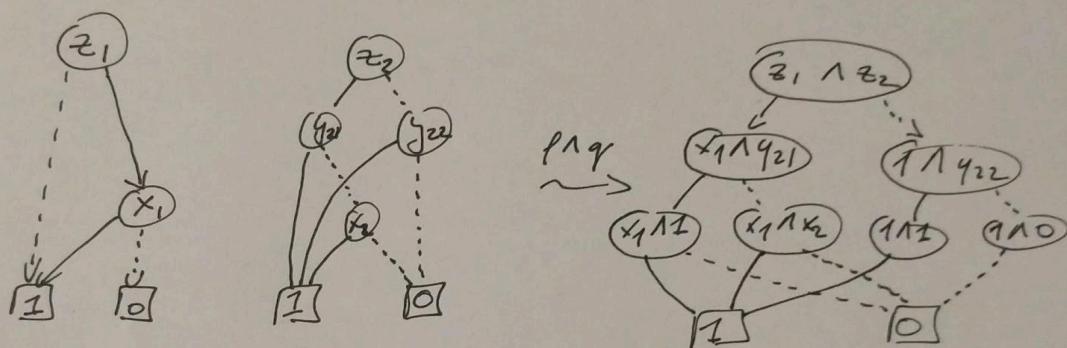
Az itt nem gond, hogy a 2.2 -ben a táblázat x,y,z sorrendben van, de a fa fentről z,y,x ???

2.2/ $(x \vee y) \wedge (y \vee z) = y \vee (x \wedge z)$

x	y	z
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1



2.3



Ell:

x	y	z	f	g	f ∧ g	z ∧ x	z ∧ y
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	1	0	1	1	1	0	1
0	1	1	0	1	0	0	0
1	0	0	0	0	0	0	0
1	0	1	1	1	1	1	0
1	1	0	1	1	1	0	1
1	1	1	1	1	1	1	0



2.2-re alternatív megoldás (annak aki jobban szeret számolni, mint rajzolgatni :D):

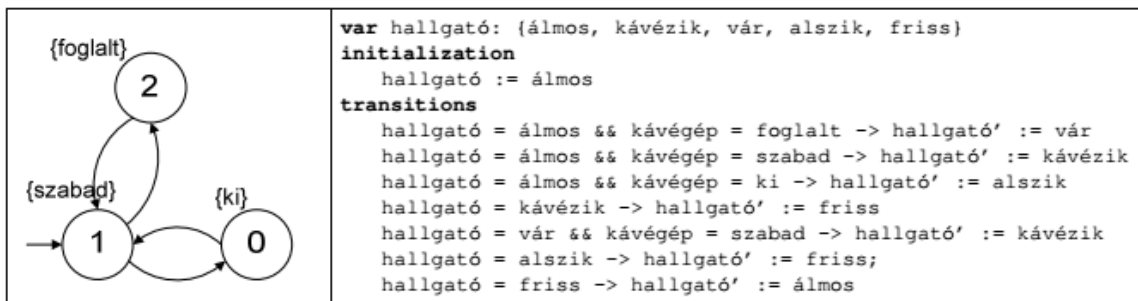
$z \rightarrow y \rightarrow x$

$$(x \vee y) \wedge (y \vee z) = y \vee (x \wedge z)$$

$g = z \rightarrow g_z, g_{\bar{z}}$
 $g_z = y \vee (x \wedge 1) = 1$ $g_{\bar{z}} = y \vee (x \wedge 0) = y$
 $g_{\bar{z}} = y \rightarrow g_{zy}, g_{z\bar{y}}$
 $g_{zy} = 1 \vee (x \wedge 1) = 1$ $g_{z\bar{y}} = 0 \vee (x \wedge 1) = x$
 $g_{\bar{z}} = y \rightarrow g_{\bar{z}y}, g_{\bar{z}\bar{y}}$
 $g_{\bar{z}y} = 1 \vee (x \wedge 0) = 1$ $g_{\bar{z}\bar{y}} = 0 \vee (x \wedge 0) = 0$
 $g_{\bar{z}\bar{y}} = x \rightarrow g_{\bar{z}\bar{y}x}, g_{\bar{z}\bar{y}\bar{x}}$
 $g_{\bar{z}\bar{y}x} = 0 \vee (1 \wedge 1) = 1$ $g_{\bar{z}\bar{y}\bar{x}} = 0 \vee (0 \wedge 1) = 0$

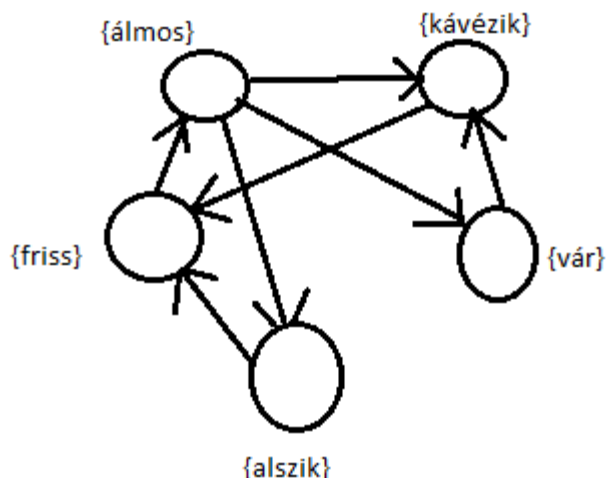
3. Temporális logika és modellellenőrzés (16 pont)

Egy egyetemi hallgató életében az egyik legfontosabb objektum a kávéautomata, aminek működésére nap, mint nap rá van utalva. A kávégép működését a baloldali Kripke struktúra (kapcsos zárójelben a logikai állítások, mint címkék találhatók: *ki*, *szabad*, *foglalt*), míg a hallgatók magatartását a jobboldali tranzíciós rendszer (program) írja le.



3.1. Rajzolja fel először a hallgató alrendszerének, majd az egész rendszernek (tehát a hallgató és a kávégép együttesének) Kripke struktúráját!

4 pont



s15

Az egész rendszer KS-je megvan valakinek ?

ebben a doksiban benne van, két féle is, de nem tudjuk, hogy a melyik jó így nem akarom átmásolni...

<https://docs.google.com/document/d/1-tD7XbaW8Wp8uFicF2DCVxd50xWnlcfM8ZelWhcEM3c/edit?usp=sharing>

Tavalyi emlékeket elővéve: az van, hogy mivel első körben csak a hallgató alrendszere kell, ezért a fenti Paint-es ábra jóságos, csak a hallgatóra vonatkozó állapotok vannak rajta. (Amire még figyelni kell nagyon - szintén tavalyi emlék - hogy ezeket a címkéket ne a gömbökbe írjuk, mert az hiba, úgy más dolgokat jelölünk, van is valamelyik doksiban olyan feladat).

A közös rendszer pedig a két alrendszer direkt szorzata, vagyis 5x3 bubi lesz rajta, nyilak pedig nyilván azok között mennek, ami a megadásokban értelmes.

Az 5x3-al azokat az állapotokat is felvesszük, hogy pl. {alszik,ki}, {alszik,szabad}, {alszik,foglalt} amik nem relevánsak a feladatot nézőpontjából. A kávégép alrendszerére nézve persze be tudjuk húzni az állapotátmeneteket, de a hallgatóra nézve pont tökmindegy, hogy mi történik a kávégéppel amíg alszik, márpedig ez a feladat középpontjában a hallgató van. Szóval?

Azokat utána igazából szerintem ki lehet hagyni tekintve, hogy tényleg nem relevánsak. Passz, tényleg, sehol nincs ez normálisan leírva. Én szerintem úgy csinálnám meg hogy leírnám azt, hogy a két alrendszer alkotta közös rendszerben 5x3 állapot van, azok "ezek", viszont közülük nem lényeges mindegyik, úgyhogy a végleges ábra meg "itt" van. Ebből szerintem látszódik is, hogy érted + a lapon van a jó megoldás, akár mind a 15 állapot kell, akár csak a relevánsak :)

3.2. Formalizálja LTL kifejezések segítségével az alábbi követelményeket!

Vigyázat: úgy tekintse, hogy a követelmények formalizálásakor nem ismerjük magát a rendszert, csak a logikai állítások, azaz címkék halmazát! (válaszonként 2 pont)

6 pont

3.2-1. A hallgató nem ihat kávét addig, amíg a kávégép foglalt.

3.2-2. Ha a kávégép szabad, a hallgató a következő időpillanatban kávéhoz jut.

3.2-3. A hallgatónak nem kell örökké várnia a kávégépre.

3.3. Igaz-e a kávégép viselkedését leíró tranzíciós rendszerben (azaz a Kripke struktúráját vizsgálva), hogy **GF(szabad)**? Válaszát indokolja a tabló módszer segítségével!

6 pont

3.2-1: $G(\neg \text{kávézik} \cup \neg \text{foglalt})$ \\ Szerintem: $G(\text{foglalt} \cup \text{kávé})$, ? Szerintem is $G(\text{foglalt} \cup \text{kávé})$, mert

$G(\neg \text{kávé} \cup \neg \text{foglalt}) = \neg \text{kávé} \neg \text{kávé} \neg \text{kávé} \neg \text{kávé} \dots \neg \text{foglalt}$

$G(\text{foglalt} \cup \text{kávé}) = \text{foglalt} \text{ foglalt} \text{ foglalt} \dots \text{kávé}.$

Bsz-esen vegyük észre, hogy $\neg \text{kávé}$ az nem csak a foglaltból eredhet, hanem abból is, hogy a gép ki van kapcsolva, ezért nem a feladat szövegére (amíg a kávégép foglalt) referál.

$G(\text{foglalt} \cup \text{kávé}) = \text{foglalt} \text{ foglalt} \text{ foglalt} \dots \text{kávé}$ - de nem ez a feladat szövege... arról nincs szó, hogy kávézni is fog a hallgató, csak arról, hogy addig nem kávézhat. Tehát az első megoldás a jó, nincs kávézás amíg foglalt.

Okés ez jogos, de attól még a $\neg \text{kávé}$ nem jelenti azt, hogy foglalt

de a feladat szövege ezt se mondja :) tehát "nem ihat kávét addig, amíg a kávégép foglalt"... persze még ezer ok lehet, amiért nem ihat kávét, de a 3.2-1 szerint most csak a foglalt ok számít

És ehhez egy alternatívaként a $G(\text{foglalt} \rightarrow !\text{káv })$? ez a legeg rtelm bb... tuti
kimagyar zhat  :-)-viszont hib s. Az until nem  gy m k dik.

a 2014-es kidolgoz sban valaki azt  rta, hogy feladat megold skor ez volt a helyes:

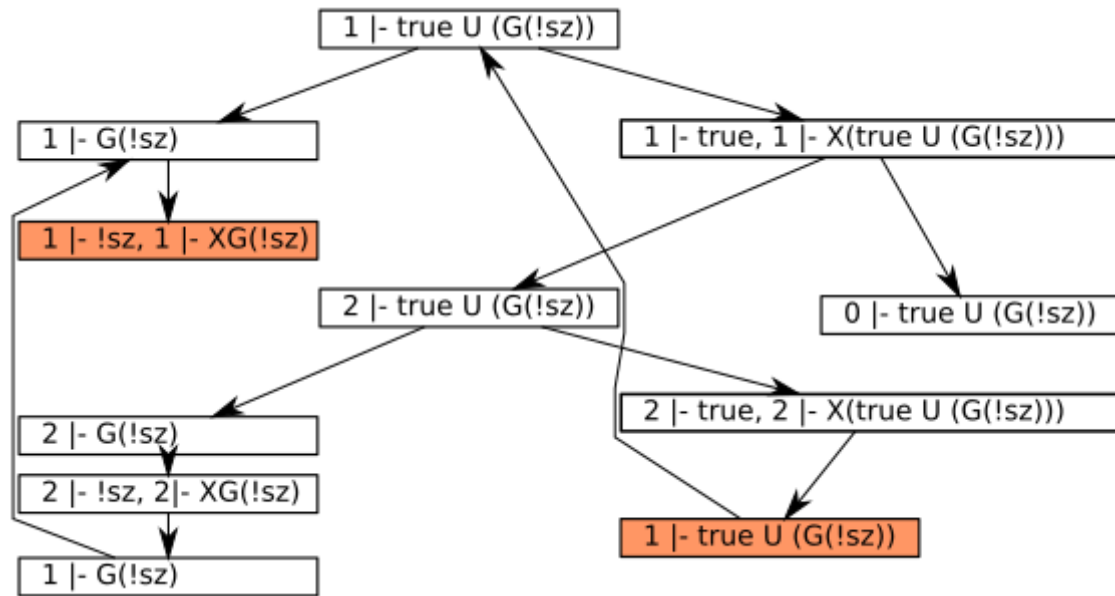
$G(\neg \text{k v zik} \cup \neg \text{foglalt})$

Esetleg **$G(!\text{k v zik} \cup \text{szabad})$** (szerintem szabadot kell kapnia, hisz, ha !foglalt,
azaz mondjuk ki  llapotba ker l, akkor sem tud k v zni a hallgat )

3.2-2: $G(\text{szabad} \Rightarrow X(\text{k v zik}))$

3.2-3: $GF(!\text{v r})$

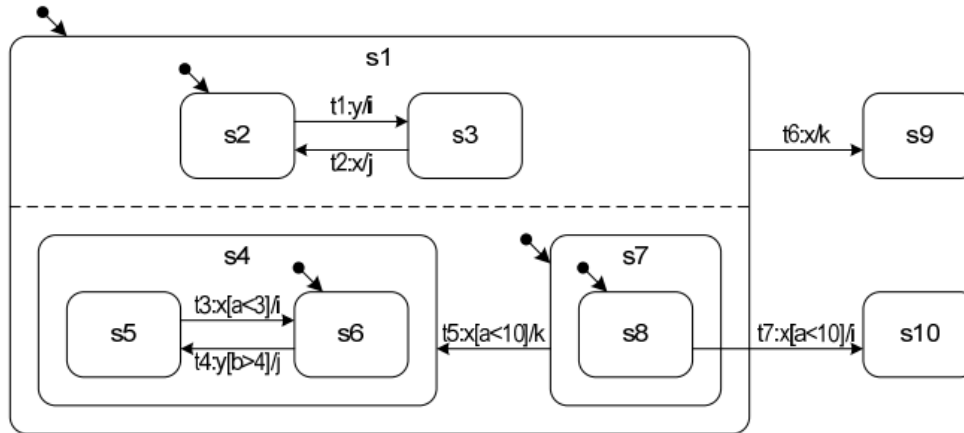
3.3: $\neg GF(\text{szabad}) = \neg(\neg F(\neg F(\text{sz}))) = FG(\neg \text{sz}) = \text{true} \cup (G(\neg \text{sz}))$



Mindenhogy ellentmondásra jutunk, mert $\neg \text{szabad}$ nem teljesül az 1-es állapotban, és olyan ciklus alakul ki, amiben $p \cup q$ -ban p teljesül, de q nem $\rightarrow$ nincs ellenpélda $\rightarrow$ helyes az eredeti állítás.

4. UML állapottérképek (8 pont)

Tekintsük az alábbi állapottérképet, melyben minden s_k állapothoz tartoznak $s_k.entry$ és $s_k.exit$ akciók is, melyek az adott állapot belépési és a kilépési akcióit jelentik! A nyilakra írt kifejezések átmenetnév: trigger [őrfeltétel] / akció alakúak.

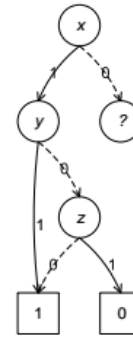


A rendszer a kezdő konfigurációjában van, az a változó értéke 8. Az esemény-sorrendezőtől egy „x” esemény érkezik.

- 4.1. Mi az állapottérkép kezdő állapotkonfigurációja? 1 pont
 - 4.2. Melyek az engedélyezett állapotátmenetek? 1 pont
 - 4.3. Mely engedélyezett állapotátmenetek állnak egymással konfliktusban? 1 pont
 - 4.4. Hogy néz ki a tüzelhető állapotátmenetek halmaza (konfliktusfeloldás után)? 1 pont
 - 4.5. Hogy néz(nek) ki a következő stabil állapotkonfiguráció(k)? (Ha több is lehetséges, mindet adja meg!) 2 pont
 - 4.6. Milyen akciók hajtódnak végre tüzelés közben és milyen sorrendben? (Ha több tüzelés lehetséges, válasszon egy olyat, ami megváltoztatja az állapotkonfigurációt, arra adja meg!) 2 pont
- 4.1:** s1, s2; s7, s8
s1 és azon belül s2, továbbá s1-en belül még s7 és azon belül s8 (= {s1, s2, s7, s8})
- 4.2:** t5, t6, t7 (ahol a trigger esemény „x” és egy aktuális állapotból tüzelne)
- 4.3:** (t5,t6), (t5,t7) (t6,t7) mivel azonos állapot különböző finomságából akar átvinni
- 4.4:** t7, a legalacsonyabb finomsági állapot szintből kimenő átmenet a prioritásos
- 4.5:** s10, mivel $a < 10$
- 4.6:** t7: ((s8.exit), (s7.exit) | (s2.exit)); (s1.exit); i; (s10.entry)

2. Bináris döntési diagramok (14 pont)

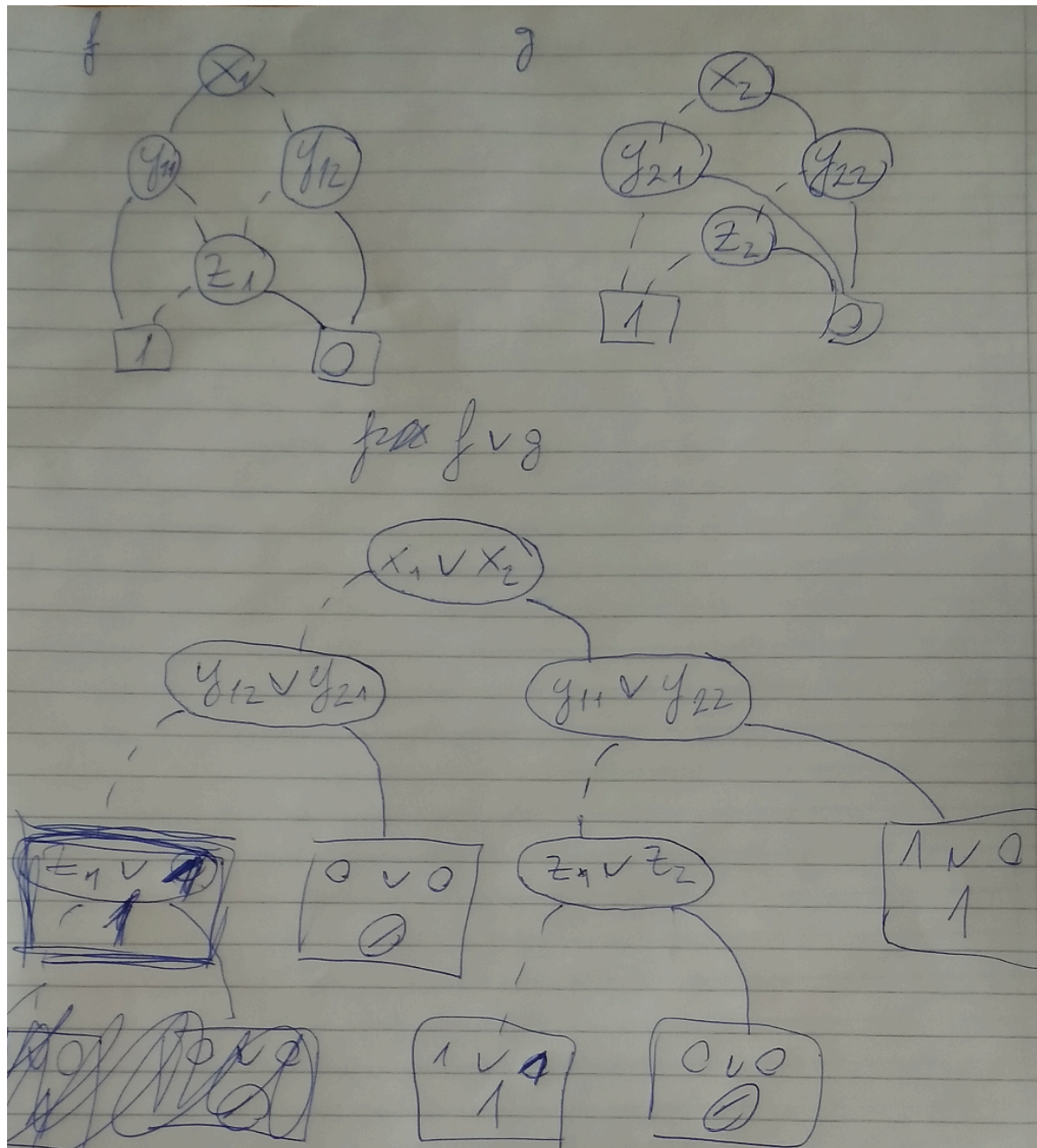
Az f függvény ROBDD alakban a jobb oldali ábrán látható, azonban az egyik csomópont alatti részek törlődtek. A hiányos csomópont (amit ? jelez az ábrán) által reprezentált logikai függvényt azonban ismerjük: $\neg y \wedge \neg z$.



- 2.1. Írja fel a teljes f függvényt algebrai alakban (képlettel) majd rajzolja fel (külön lapon) redukált rendezett bináris döntési diagram (ROBDD) alakban! A változók kiértékelésének sorrendje legyen x, y, z . 5 pont

- 2.2. A g függvény az alábbi alakban adott: $g = (\neg y \wedge \neg z) \vee (\neg x \wedge \neg y)$
Rajzolja fel a g függvény ROBDD-jét, a változók kiértékelésének sorrendje legyen x, y, z . 4 pont

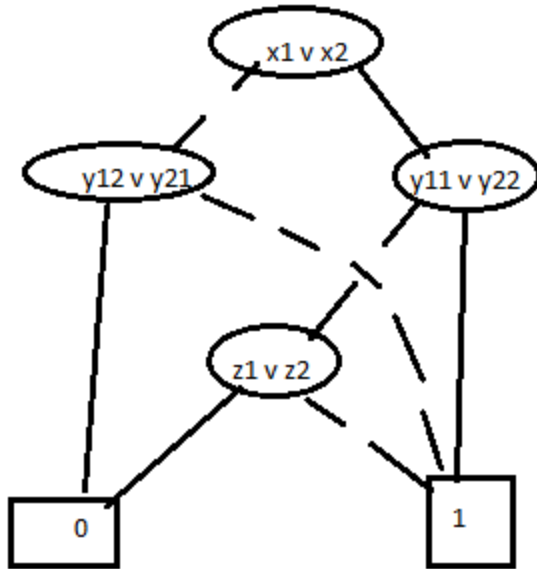
- 2.3. Legyen az m függvény a következő: $m = f \vee g$.
Rajzolja fel az m logikai függvényt ROBDD alakban! Az ROBDD konstruálását közvetlenül az f és g ROBDD alakjának felhasználásával végezze el! A változók kiértékelésének sorrendje legyen ez esetben is x, y, z . 5 pont



ezt még össze kéne vonni, nem?-de

Itt nincs jelentősége, viszont a diák alapján fontos megjegyezni, hogyha az egyikről egy adott szinten hiányzik egy változó, akkor az alatta lévő szerepel a kérdéses szinten és a "saját" szintjén is. FM-2015_EA04a_ROBDD_modellellenorzes.pdf / 42. dia

Tessék: #paintskill

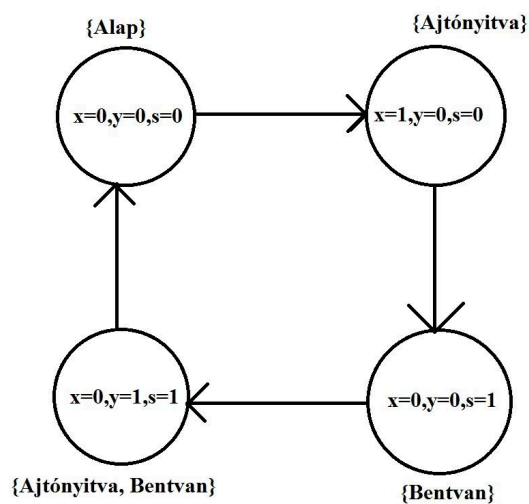


Még annyi hogy itt $x_1 \vee x_2$ helyére x , $y_{12} \vee y_{21}$ helyére y és a többi....

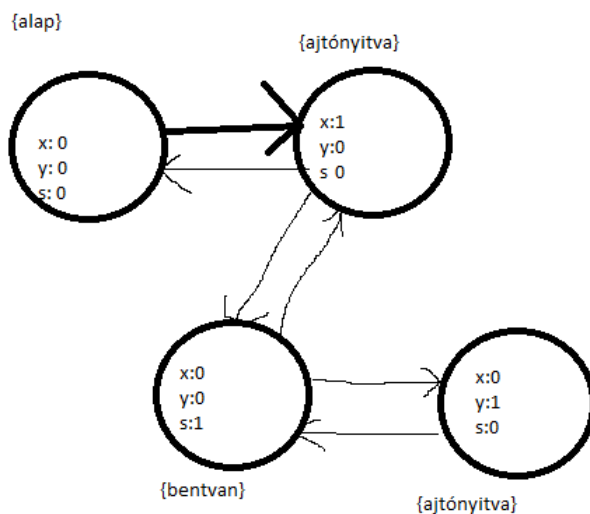
3. Szemantika alapú modellellenőrzés (8 pont)

Egy zsilipkamra két ajtóval rendelkezik, melyek soha nem lehetnek egyszerre nyitva. Az ajtók állapotát x és y bináris állapotváltozók írják le, egy harmadik s változó pedig azt tárolja, hogy valaki halad-e át a zsilipen. *Alapállapotban* mindkét ajtó zárva van, és senki nincs az ajtók között ($x=y=s=0$). Ha valaki át szeretne haladni, először kinyílik az első ajtó (ez legyen x), majd a rendszer észleli a belépő embert, ezért (ezzel egy időben) újra bezáródik az ajtó. Ezután a folyamat a másik ajtóval (ez legyen y) megismétlődik: az ajtó kinyílik, majd az ember távozik és az ajtó bezárul, ezzel a rendszer visszatér az alapállapotba.

- 3.1. Rajzolja fel (külön lapon) a rendszer Kripke struktúra modelljét, jelölje be a rendszerállapotok kódolását, és címkézze meg az állapotokat az *alap*, *ajtónyitva* és *bentvan* címkéket használva! 1 pont



Szerintem meg a fölső. Vélemény? szerintem is a felső jó, +1 tutihogy a fölső.



ez

a jó (valszeg NEM)

:nem térsz vissza alapállapotba a második ajtónyitás után közvetlenül. csak úgy térhet vissza az ábrád szerint alapállapotba, ha gyorsan beszökik a nyitott ajtón a másik irányba valaki
miért lehet visszamenni, kinyitom az ajtót, aztán meggondolom magam?

- 3.2. Adja meg annak az állapothalmaznak a karakterisztikus függvényét, amely azokat az állapotokat tartalmazza, amelyek el vannak látva az *alap* vagy az *ajtónyitva* címkével! 1 pont
- 3.3. Adja meg annak az állapotátmenet halmaznak a karakterisztikus függvényét, amely az *ajtónyitva* címkével ellátott állapotokból kiinduló állapotátmeneteket tartalmazza! 2 pont
- 3.4. A tanult iteratív állapotcímkezési eljárást végrehajtva ellenőrizze a modellen, hogy az alapállapotból, mint kezdőállapotból kiindulva teljesül-e az alábbi CTL kifejezés: $A((ajtónyitva \vee alap) \wedge bentvan)!$ Az iteráció minden lépéséhez adja meg a címkéző kifejezést és (felsorolással) a címkézett állapotok halmazát! 4 pont

3.2: tehát...

$(!x \wedge !y \wedge !z) \wedge (x \wedge !y \wedge !z) \wedge (!x \wedge y \wedge z) \text{ ????$ //szerintem a különböző állapotok között VAGY kapcsolat van, nem ÉS, tehát:
 $(!x \wedge !y \wedge !z) \vee (x \wedge !y \wedge !z) \vee (!x \wedge y \wedge z)$

Szerintem: $(!x \wedge !y \wedge !s) \vee (x \wedge !y \wedge !s) \vee (!x \wedge y \wedge s)$ //fixme

az új ábra szerint: $(!x \wedge !y \wedge !s) \vee (x \wedge !y \wedge !s) \vee (!x \wedge y \wedge s)$

Állapothalmaz karakterisztikus függvénye:

$\{s1, s2\}$ állapothalmaz:

$$C_{\{s1, s2\}} = C_{s1} \vee C_{s2} = (\neg x \wedge \neg y) \vee (\neg x \wedge y)$$

3.3: itt kell az ÉS kapcsolat:

$(s1, s2) \in R$ állapotátmenet:

$$C_{(s1, s2)} = (\neg x \wedge \neg y) \wedge (\neg x' \wedge y')$$

igen, de különböző állapotátmenetek között vagy van, nézd tovább a diát!

új ábra szerint: $(x \wedge !y \wedge !s \wedge !x' \wedge !y' \wedge !s') \vee (x \wedge !y \wedge !s \wedge !x' \wedge y' \wedge s') \vee (!x \wedge y \wedge s \wedge !x' \wedge !y' \wedge !s')$

3.4: ezen az ábrán igaz(kérdés, hogy az ábra jó-e), kezdve a bentvan állapottal visszafelé felcímkezve eljuthatunk az első állapothoz. Kérdés: mennyire kell vajon itt használni a rendes jelölésrendszert az épp címkézett halmazokra? kb megtriplázza a munkát mindennek a jelölése.. egy felsorolás a lépésekről elég?!

Valaki erősítsen meg!

- 1- Ráarakom a megadott címkét az kezdő állapotra
- 2- Ráarakom a megadott címkét a 3. állapotra ($x=0, y=1, z=1$), mert $z=1$ tehát bent van
- 3- Ráarakom a másodikra ($x=0, y=0, z=1$) mert szintén bent van
- 4- Ráarakom az elsőre ($x=1, y=0, z=0$) mert az ajtó nyitva van tehát az E(PUQ) alakú kifejezés P része igaz és utána következőn már rajta van E(PUQ)

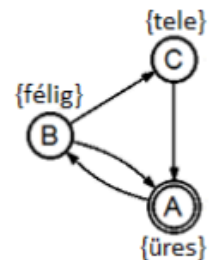
4. LTL követelményformalizálás (8 pont)

Egy doktorandusz hallgató tudományos cikket ír egy nemzetközi konferenciára. A leadási határidőhöz közeledve *hócipőjének* állapotait a jobb oldali ábrán látható Kripke struktúra modellezi.

A *doktorandusz* viselkedését a következő szabályok határozzák meg (ezek $\langle \text{feltétel} \rangle \rightarrow \langle \text{állapotváltozás} \rangle$ alakúak):

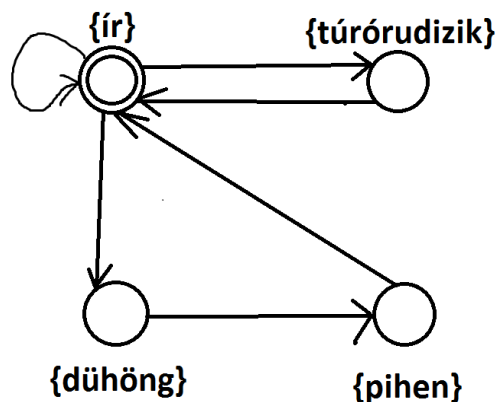
```

var doktorandusz: {ír, pihen, túrórudizik, dühöng}
initialization:
    doktorandusz := ír
transition rules:
    doktorandusz = ír && hócipő = félig -> doktorandusz := ír
    doktorandusz = ír && hócipő = félig -> doktorandusz := túrórudizik
    doktorandusz = túrórudizik && hócipő = üres -> doktorandusz := ír
    doktorandusz = ír && hócipő = tele -> doktorandusz := dühöng
    doktorandusz = dühöng -> doktorandusz := pihen
    doktorandusz = pihen && hócipő = üres -> doktorandusz := ír
    doktorandusz = pihen && hócipő = félig -> doktorandusz := ír
    
```



4.1. Rajzolja fel azt a Kripke struktúrát, ami a *doktorandusz* viselkedését modellezi! Használja az {ír, pihen, túrórudizik, dühöng} állapotcímkéket!

2 pont



4.2. Formalizálja LTL kifejezések segítségével az alábbi követelményeket, amelyek a doktorandusz viselkedésére minden esetben vonatkoznak!

6 pont

4.2-1. Ha tele van a hócipője, akkor dühöng, majd a következő időpillanatban pihen.

4.2-2. A doktorandusz előbb-utóbb folytatja az írást.

4.2-3. Ha a doktorandusz túrórudizik, akkor addig nem pihen, amíg nincs tele a hócipője.

4.2-1. $G(\text{tele} \rightarrow (\text{dühöng} \ \&\& \ X \text{ pihen}))$

4.2-2. $GF(\text{ír})$

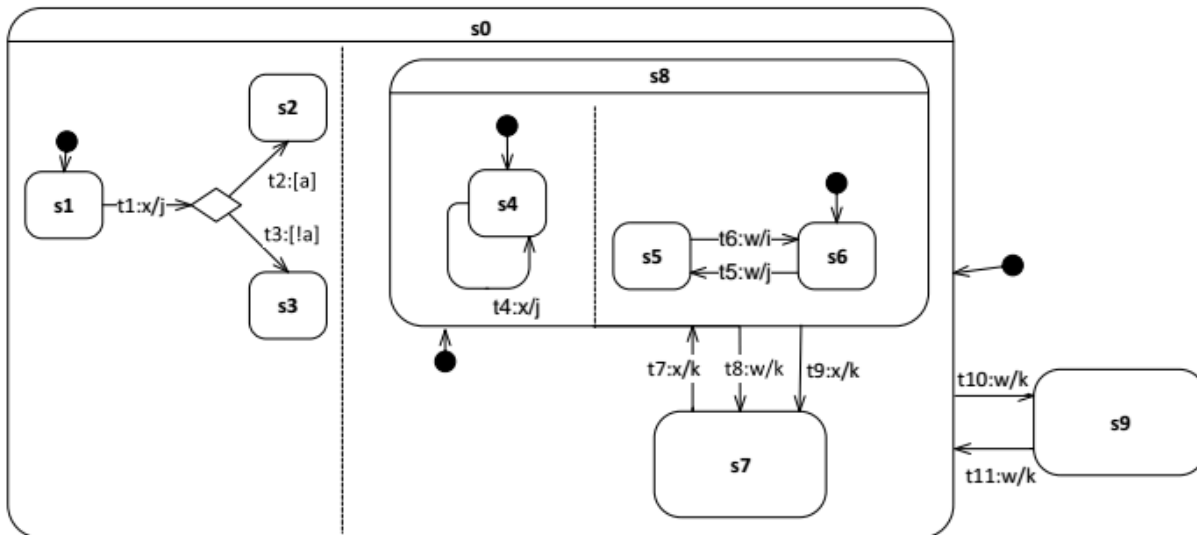
4.2-3. $G(\text{túrórudizik} \rightarrow (!\text{pihen} \ U \ !\text{tele}))$

Ennek itt nem ennek kellene lennie: $(!\text{pihen} \ U \ \text{tele})$? Mert ugye $!\text{pihen} \ !\text{pihen} \ !\text{pihen}$, közben gondolom telik a hócipője, aztán tele lesz, és utána már pihenhetne. (Szép dolog a magyar nyelvben a dupla tagadás) <-Jogos, nekem is ez jött ki

De igen. Érdekes átgondolni, mit is jelent a mondat (vagy lefordítani angolra :D), mert a magyar nyelv sajátosságai miatt ezt könnyű elrontani.

5. UML állapottérképek (8 pont)

Tekintsük az alábbi állapottérképet, melyben minden s_k állapothoz tartoznak $s_k.entry$ és $s_k.exit$ akciók is, melyek az adott állapot belépési és a kilépési akcióit jelentik! A nyilakra írt kifejezések átmenetnév: trigger [őrfeltétel] / akció alakúak.



A rendszer kezdőállapotban van, az a boolean változó értéke $false$. Az esemény-sorrendezőtől egy „x” esemény érkezik.

5.1. Melyek az engedélyezett állapotátmenetek?

1 pont

5.2. Mely engedélyezett állapotátmenetek állnak egymással konfliktusban?

1 pont

5.3. Hogy néz ki a tüzelhető állapotátmenetek halmaza (konfliktusfeloldás után)?

1 pont

5.4. Hogy néz(nek) ki a következő stabil állapotkonfiguráció(k)? (Ha több is lehetséges, mindet adja meg!) Adja meg azt is, hogy milyen akciók hajtódnak végre tüzelés közben és milyen sorrendben!

2 pont

Kezdő állapotok:

s_0 és azon belül s_1 illetve s_0 -n belül még s_8 és azon belül s_4 és s_6

5.1: t_1 , t_3 , t_4 , t_9 (ahol x a trigger esemény és egy olyan állapotból jön ki, amiben éppen vagyunk)

5.2: (t_4, t_9) ; - ugyan abból az állapotból tüzelne, csak különböző finomsági szintről

5.3: t_1, t_3, t_4 (állapot finomság alapján a t_4 alacsonyabb szintű mint a t_9)

5.4: s_0 és azon belül s_2 vagy s_3 „a” értékétől függően, illetve s_0 -n belül még s_4 és s_6

szerintem ez s_0, s_3, s_8, s_4, s_6

A megoldás szerint akkor t1-et nem követi automatikusan t2? tehát a következő állapot egy "vagy" kapcsoló? ez így valid? mért követné automatikus t2? örfeltétel van, ha a=true akkor t2 tüzel és s2-be kerülünk, ha a=false akkor t3 és s3-ba kerülünk. De "a" kezdetben false, így t3 tüzel, így az 5.4-nél is s3 kellene nem? De, ott S3 van S2 nincs.

5.5 ?

minta_3 zh ?

4. Modellezés és modellellenőrzés

Egy informatikai rendszer két erőforrásból áll, ezek egy adatbázisszerver és egy alkalmazásszerver, amelyek kikapcsolt vagy bekapcsolt állapotban lehetnek. Az erőforrásokat hibamentes esetben egyszerre kapcsolják ki és be. Alaphelyzetben mindkét erőforrás ki van kapcsolva. A megfelelő üzemállapot az, amikor mindkét erőforrás be van kapcsolva. Ha az üzemállapotban az adatbázisszervert hiba következtében kikapcsolják, az rendszer szinten üzemképtelen állapotnak tekinthető. Ezután az alkalmazásszervert is kikapcsolják, majd mindkét erőforrás bekapcsolásával indítják újra a rendszert.

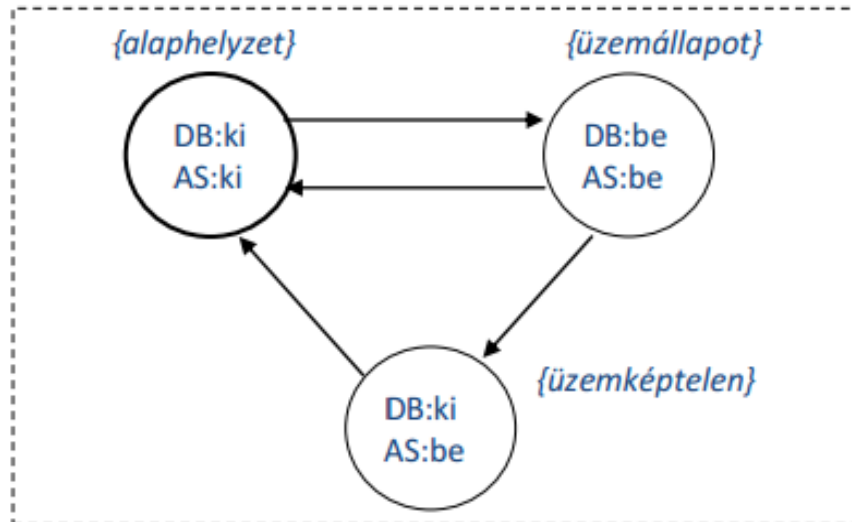
Rajzolja fel a rendszer itt leírt működését modellező Kripke-struktúrát, amelynek állapotait az egyes erőforrások állapotainak (kikapcsolás és bekapcsolás) fentiek szerint lehetséges kombinációi jelölik ki! Az egyes állapotokat címkézzé a következő atomi kijelentésekkel: {alaphelyzet, üzemállapot, üzemképtelen}

KS esetén:

AP = { alaphelyzet, üzemállapot, üzemképtelen } atomi kifejezések (címkek)

S = {s1, s2, s3 } állapotok

Erőforrások: DB, AB (ez nem kell a KS ábrához igazából, csak a CTL-hez később)



(Az állapotokba csak a jobb érthetőség kedvéért kerültek az erőforrások állapotára utaló megjegyzések, ezek nem részei a Kripke-struktúrának.)

Honnan jön az $\{alaphelyzet\} \leftrightarrow \{üzemállapot\}$ nyíl?*

"Az erőforrásokat hibamentes esetben egyszerre kapcsolják ki és be." Alaphelyzetből bekapcsolás után ha nem volt hiba, üzemállapotba kerül a rendszer, üzemállapotból pedig kikapcsolt (alaphelyzetbe) is vissza lehet állítani a rendszert, ezért kell oda-vissza nyíl.

Ellenőrizze a modellen, hogy az üzemállapotot kezdőállapotnak tekintve teljesül-e a következő CTL kifejezés: $E(\neg \text{üzemképtelen} \ U \ \text{alaphelyzet})$! Válaszát indokolja meg!

$E(\neg \text{üzemképtelen} \ U \ \text{alaphelyzet})$: Létezik olyan útvonal (Exists) amelyen egyszer fennáll majd az alaphelyzet, de addig ($U = \text{Until}$) mindenhol az útvonalon üzemképes áll fenn (negált üzemképtelen).

A bizonyítás az iteratív állapotcímkézési eljárással elvégezhető:

1. lépés: Az alaphelyzet címkéjű állapotra helyezhető fel az $E(\neg \text{üzemképtelen} \ U \ \text{alaphelyzet})$ címke.
2. lépés: A címkézett megelőző állapotok vizsgálata (kompatibilis-e címkéje a $\neg \text{üzemképtelen}$ címkével) alapján az üzemállapot címkéjű állapotra helyezhető fel az $E(\neg \text{üzemképtelen} \ U \ \text{alaphelyzet})$ címke. A címkézés további bővítése nem lehetséges. Az üzemállapot címkéjű állapot (mint vizsgálandó kezdőállapot) ott van a címkézettek között, tehát teljesül a követelmény.

Itt nem csak annyi, h megnézem, hogy a üzemállapotból eljuthatok-e a alaphelyzetbe, úgy hogy közben végig nem üzemképtelen, és akkor a random** nyíl miatt sikerül?

3. Modellezés és modellellenőrzés Egy informatikus hallgató „tevékenységeit” aszerint különböztetjük meg, hogy éppen kávézik vagy nem, valamint éppen alszik vagy nem. A hallgató három állapotát különböztetjük meg: tanulás közben kávézik és nem alszik; ezután vizsgázik,

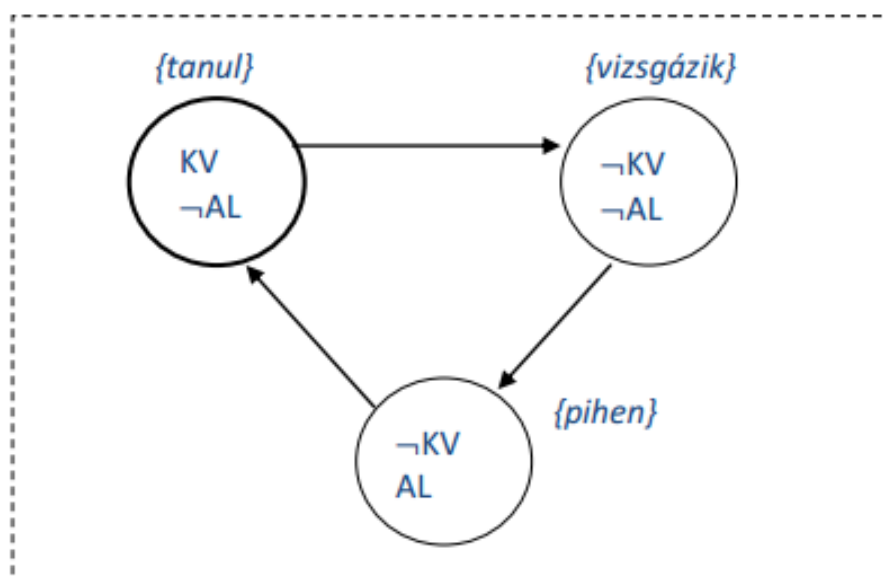
ahol nem kávézik és nem is alszik; a vizsgázás után pihen, ekkor alszik és nem kávézik. A hallgató alapállapota a tanulás, amit a vizsgázásig nem is hagy abba. Tanulás nélkül a hallgató nem vizsgázik; a vizsgázás után közvetlenül pedig nem tanul (csak pihenés után).

Rajzolja fel a hallgató itt leírt viselkedését modellező Kripke-struktúrát, amelynek állapotait a hallgató egyes tevékenységeinek (kávézás és alvás) fentiek szerint lehetséges kombinációi jelölik ki! Az egyes állapotokat címkézzé a következő atomi kijelentésekkel: {pihen, tanul, vizsgázik}

KS esetén:

AP = { pihen, tanul, vizsgázik } atomi kijelentések (címkek)

S = { s1, s2, s3 } állapotok



(Az állapotokba csak a jobb érthetőség kedvéért kerültek a kávézásra és az alvásra utaló megjegyzések, ezek nem részei a Kripke-struktúrának.)

A vastagon kihúzott állapot a kezdőállapot.

Ellenőrizze a modellen, hogy a hallgató alapállapotából (ami a tanulás) kiindulva teljesül-e a következő CTL kifejezés: $E(\neg \text{vizsgázik} \cup \text{pihen})$! Válaszát indokolja meg!

$E(\neg \text{vizsgázik} \cup \text{pihen})$: Létezik olyan útvonal, amelyen egyszer a pihen állapot igaz lesz, de addig mindenhol a nem vizsgázik áll fenn (vizsgázik negált).

A pihen állapotot a vizsgázik előzi meg, amelynél nem áll fenn a negáltVizsgázik, továbbá előtte a tanul nem szerepel a kifejezésben. A vizsgázik esetén nem teljesül a negáltVizsgázik.

Az ellenőrzés az iteratív állapotsímkézési eljárással elvégezhető:

1. lépés: A pihen címkéjű állapotra helyezhető fel az $E(\neg \text{vizsgázik} \cup \text{pihen})$ címke. **MIÉRT??**

Belülről indulunk kifelé <- mert először azt címkézed meg $p \cup q$ esetén amin a q van és onnan

nézed visszafelé hogy addig végig teljesül-e a p mindegyikre és így lesz-e címke a kezdőállapoton

2. lépés: A címkézett megelőző állapotok vizsgálata (kompatibilis-e címkéje a $\neg$ vizsgáló címkével) alapján kiderül, hogy további állapotra nem helyezhető fel az $E(\neg$ vizsgáló U pihen) címke. A címkézés további bővítése nem lehetséges. A tanul címkéjű állapot (mint vizsgálandó kezdőállapot) nincs ott a címkézettek között, tehát az $E(\neg$ vizsgáló U pihen) követelmény nem teljesül.

Itt nem csak annyi, h megnézem, hogy a tanulból eljuthatok-e a pihenben, úgy hogy közben végig nem vizsgáló??

De igen.

Egyéb gyakorló feladatok:

- Indokolja meg, hogy az $A(X \text{ Stop } \vee F \text{ Start})$ kifejezés szintaktikailag helyes kifejezés-e CTL illetve CTL* temporális logikában!

Meg kell előznie a kvantoroknak a temporális operátorokat, E: létezik kvantor, A: univerzális kvantor

- Írja le a korlátos modellellenőrzés alapötletét!

A korlátos modellellenőrzés alapja

- Az állapotteret nem „egyben” kezeljük
- Az **útvonalak hosszát korlátozva végezzük az ellenőrzést**

- Írja le, hogy mire használjuk az ROBDD-ket a modellellenőrzésben?

Logikai függvények reprezentációjára (?).

- Definiálja, hogy mit jelent UML állapottérképek esetén a konfliktus fogalma (mely átmenetek vannak egymással konfliktusban) és hogyan történik a konfliktus feloldása! Iméleti kérdések

Azon átmenetek vannak konfliktusban melyek ugyan azt az állapotot hagyják el, pontosabban az elhagyott állapothalmazaik metszete nem üres.

- Indokolja meg, hogy következő ekvivalencia helyes-e: $F \text{ Stop } \vee F \text{ Start} \equiv F (\text{Stop } \vee \text{Start})$ ahol $\vee$ a logikai VAGY operátort jelöli, a Stop és Start pedig atomi kijelentések.

A jövőben igaz lesz, hogy Stop, vagy a jövőben igaz lesz, hogy Start, akkor a jövőben igaz lesz, hogy vagy Start, vagy Stop.

• Indokolja meg, hogy következő ekvivalencia helyes-e: $AG\ Stop \equiv not\ EF\ (not\ Stop)$ ahol not a logikai negálás operátort jelöli, a Stop pedig egy atomi kijelentés.

Helyes, a $G(Stop)$ -ot fel lehet írni $not\ F(not\ Stop)$ -ként, az A (“minden”) útvonal pedig felírható $not\ E$ (“nem létezik”)-ként.

Informálisan: minden útvonalra igaz az, hogy mindig Stop állapotban vagyunk = nem létezik olyan útvonal, ahol valahol a jövőben nem Stop állapotban vagyunk.

2. ZH

2011-es minta ZH wikiről:

https://wiki.sch.bme.hu/images/1/10/Form_ZH2_2011_minta_PN_feladattipusok.pdf

2014-es minta ZH:

https://inf.mit.bme.hu/sites/default/files/materials/category/kateg%C3%B3ria/oktat%C3%A1s/msc-t%C3%A1rgyak/form%C3%A1lis-m%C3%B3dszerek/14/FM-2014_ZH2_minta.pdf

3-as feladatnál nagyon érdekes a mátrix, és az ábra kapcsolata... :)

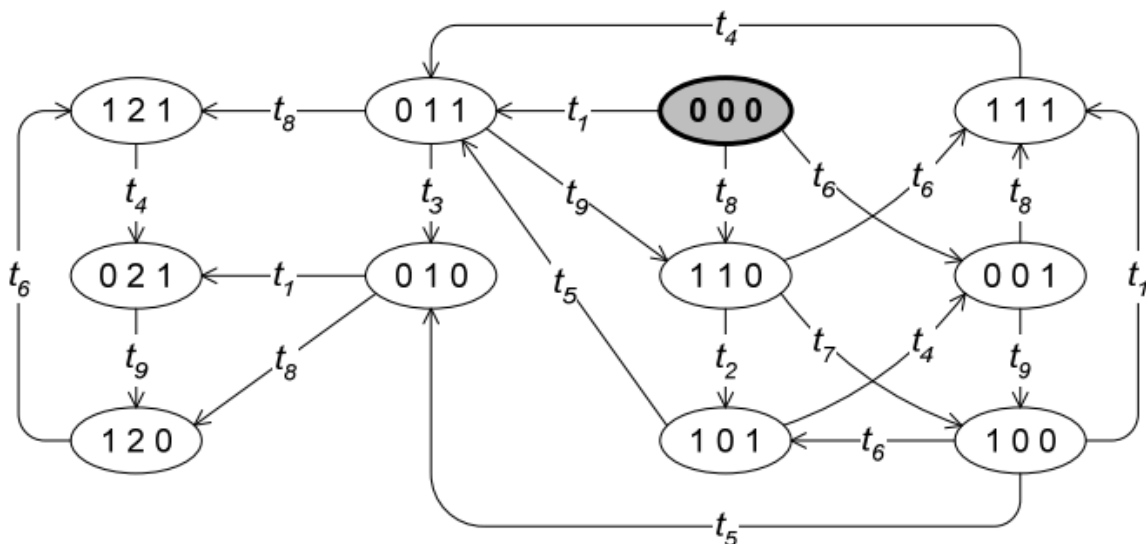
Kérdések, válaszok, definíciók, igaz-hamis. Nem elavult?

<https://wiki.sch.bme.hu/ForxmModZH2Ossz>

Példafeladatok kidolgozása:

1.1. Petri hálók dinamikus tulajdonságai

Az alábbi ábra egy Petri háló állapotterét mutatja be. A hálóban 3 hely és 9 darab tranzíció található, amelyeket a p_1, p_2, p_3 és $t_1, \dots, t_9$ címkékkel jelölünk. Az állapotokat a token eloszlás vektorral címkéztük meg, tehát pl. $0\ 1\ 0$ jelentése: $M = (0, 1, 0)$, azaz $m(p_1) = 0$, $m(p_2) = 1$ és $m(p_3) = 0$. A kezdőállapot az $M_0 = (0, 0, 0)$ állapot (vastag vonalú, szürke háttérű csomópont).



1.1-1. Vizsgálja meg az ábrát, majd az állapottér jellemzői és a tranzícióknak az ábrából leolvasható szemantikája alapján válaszoljon a Petri háló dinamikus tulajdonságaival kapcsolatos kérdésekre!

10 pont

	igaz	hamis	nem dönthető el		igaz	hamis	nem dönthető el
(a) A Petri háló élő	☐	☐	☐	(f) t_9 tranzíció L_4 -élő	☐	☐	☐
(b) A háló megfordítható	☐	☐	☐	(g) t_3 tranzíció L_3 -élő	☐	☐	☐
(c) Az állapottér lefedhető T-invariánsokkal	☐	☐	☐	(h) t_4 és t_9 tranzíció korlátos fair	☐	☐	☐
(d) A hálóban létezik visszatérő állapot	☐	☐	☐	(i) t_2 és t_5 tranzíció korlátos fair	☐	☐	☐
(e) Lehet szabad választású (FC) a Petri háló ¹	☐	☐	☐	(j) A háló globális fair tulajdonságú	☐	☐	☐

- A Petri háló élő: Hamis.** Mert az $121, 021, 120, 011, 010$ állapotokból nem érsz el minden tranzakciót.
- A háló megfordítható: HAMIS.** Mert nem megy el a kiinduló állapotba.
- Az állapottér lefedhető T-invariánsokkal:** 3×9 -es mátrix mint irányított gráf és ehhez keresünk olyan oszlop vektort ami 0-ra hozza ki össze szorozva
- A hálóban létezik visszatérő állapot: IGAZ.** Pl. az 121 a bal felső sarokban. Ide még később vissza lehet térni a $\rightarrow 021 \rightarrow 120 \rightarrow$ állapotokon keresztül.

- e. **Lehet szabad választású (FC) a Petri háló: HAMIS.** A feltétel hogy minden állapotból az összes kimutató élre teljesüljön, hogy: - vagy az egyedüli kimutató állapotból a 011-es állapotba mutató él esetén. Itt nem is az egyedüli kimutató és nem is az egyedüli bemutató, tehát nem lehet FC a háló.
- f. **t_9 tranzakció L_4 -élő: ?**