

Gyakorló feladatok az 1. zárthelyi dolgozathoz megoldással

1. Követelmények formalizálása temporális logika alkalmazásával

Egy vasúti kereszteződést biztosító fényoszorompó viselkedését az állapotaihoz rendelt következő atomi kijelentésekkel jellemezzük: $\{kikapcsolt, fehér, piros\}$

A kereszteződéshez érkező autós viselkedését az állapotaihoz rendelt következő atomi kijelentésekkel jellemezzük: $\{érkezik, körülnéz, megáll, áthalad\}$

Formalizálja LTL kifejezések segítségével az alábbi követelményeket, amelyek az autós viselkedésére minden esetben vonatkoznak:

- Kikapcsolt állapotú fényoszorompó esetén az autós körülnéz és a következő időpillanatban vagy áthalad, vagy megáll.

$$G(kikapcsolt \Rightarrow (körülnéz \wedge X(\áthalad \vee megáll)))$$

- Az autós előbb-utóbb át fog haladni a vasúti kereszteződésen.

$$GF(\áthalad)$$

- Ha egy autós érkezéskor piros a lámpa, akkor az autós addig nem halad át, amíg fehérre nem vált a fényoszorompó.

$$G((piros \wedge érkezik) \Rightarrow (\neg \áthalad \ U \ fehér))$$

2. Követelmények formalizálása temporális logika alkalmazásával

Egy bonyolult szimulációt futtató szerver állapotait a következő atomi kijelentésekkel jellemezzük: $\{kikapcsolt, várakozó, bemelegítés, szimuláció\}$

A szerverszoba hűtőberendezésének működését az állapotaihoz rendelt következő atomi kijelentésekkel jellemezzük: $\{készlet, normál, maximális\}$

Formalizálja LTL kifejezések segítségével az alábbi követelményeket, amelyek a rendszer működésére minden esetben vonatkoznak:

- Ha egy adott pillanatban a szimuláció a hűtőberendezés készlet állapota mellett zajlik, akkor a következő pillanatban a szerver várakozó állapotra kapcsol.

$$G((szimuláció \wedge készlet) \Rightarrow X \text{várakozó})$$

- Előbb-utóbb elkezdhető a szimuláció.

$$GF(\text{szimuláció})$$

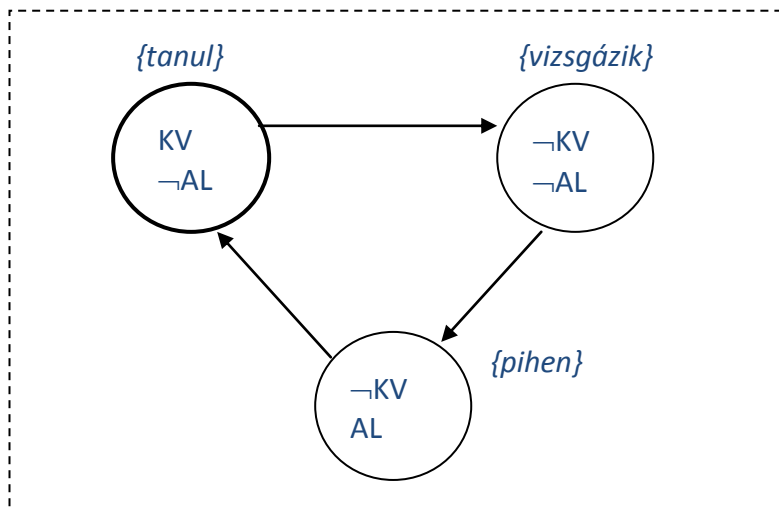
- Csak úgy hajtható végre szimuláció, ha volt bemelegítés a hűtőberendezés normál működése mellett.

$$G((X \text{szimuláció}) \Rightarrow (\text{bemelegítés} \wedge \text{normál}))$$

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: $\{pihen, tanul, vizsgázik\}$



(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: $E(\neg vizsgázik \ U \ pihen)$! Válaszát indokolja meg!

Az ellenőrzés az iteratív állapocímkezési eljárással elvégezhető:

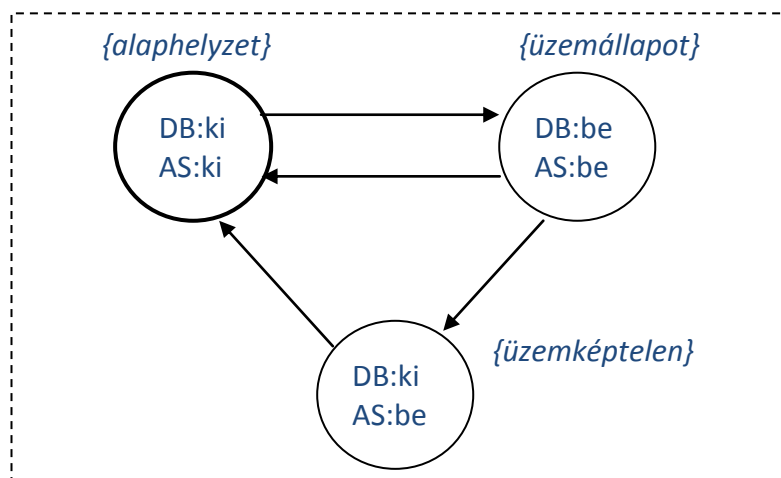
1. lépés: A $pihen$ címkéjű állapotra helyezhető fel az $E(\neg vizsgázik \ U \ pihen)$ címke.
2. lépés: A címkézett megelőző állapotok vizsgálata (kompatibilis-e címkéje a $\neg vizsgázik$ címkével) alapján kiderül, hogy további állapotra nem helyezhető fel az $E(\neg vizsgázik \ U \ pihen)$ címke. A címkézés további bővítése nem lehetséges.

A $tanul$ címkéjű állapot (mint vizsgálandó kezdőállapot) nincs ott a címkézettek között, tehát az $E(\neg vizsgázik \ U \ pihen)$ követelmény nem teljesül.

4. Modellezés és modellellenőrzés

Egy informatikai rendszer két erőforrásból áll, ezek egy adatbázisszerver és egy alkalmazáserver, 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 alkalmazáservert 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}*



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

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

A bizonyítás az iteratív állapotcímkézési eljárással elvégezhető:

1. lépés: Az *alaphelyzet* címkéjű állapotra helyezhető fel az $E(\neg\text{üzemképtelen} \cup \text{alaphelyzet})$ címke.
2. lépés: A címkézett megelőző állapotok vizsgálata (kompatibilis-e címkéje a $\neg\text{üzemképtelen}$ címkével) alapján az *üzemállapot* címkéjű állapotra helyezhető fel az $E(\neg\text{üzemképtelen} \cup \text{alaphelyzet})$ címke.

A címkézés további bővítése nem lehetséges.

Az *üzemállapot* címkéjű állapot (mint vizsgálandó kezdőállapot) ott van a címkézettek között, tehát teljesül a követelmény.