Kockavilág

A mellékelt hat db JPEG kép egy hat oldalas kézírásos anyagot tartalmaz. Ugyanez a hat oldal megtekinthető egyben, összefűzve egy PDF típusú állományban is, ugyanebben a mappában. A 6. oldalon látható feladat alapján ismertetjük a kitűzött problémát.

A feladat műfaja úgy fogalmazható meg, hogy a kockavilágbeli tételbizonyításra programot kell készíteni két-három megoldási módszerrel: a logikai alapú, az állapottér reprezentációs (MI-s) és a funkcionális programozás alapú megoldási módszerek alapján. Az esszé is szóba jöhet az automatikus tételbizonyítás témaköréből, illetve a mellékelt logikai leíráshoz készített feladatok és bizonyítások formájában. Személyes konzultációra mindenképpen szükség lehet.

A kockavilágot egy meghatározott szabályszerűséggel rendelkező rendszerként tekintjük, amelyben logikai következmények bizonyítása a feladat. Megadott számú, konkrétan megnevezett egyforma kocka van előttünk az asztalon, pl. az a, b, c és d nevű kockák. Ezeket „természetes módon” pakolhatjuk egymásra. Az összes szabályszerű konfiguráció alkotja a kockavilágot, amely csak hátteret képez állításaink bizonyításához.

Kimondunk néhány input állítást, amelyek érvényesnek tekintünk. A példában négy állítás szerepel. (Alatta láthatók azok a konfigurációk, amelyekre ezek az állítások teljesülnek.) Ezután kimondunk egy (vagy több) olyan output állítást, amely(ek) az input állítások teljesülése esetén fennáll(nak). Az a feladat, hogy bizonyítsuk be ezt a logikai következményt. A példában egy output állítás szerepel, amelyről meggondolható, hogy valóban következménye a négy feltétel állításnak.

Erre a tételbizonyítási feladatra kellene két, illetve három megoldást adni a következő módszerekkel: logikai alapú következtetés, állapottér reprezentáció és visszalépéses bejárás (MI), illetve funkcionális programozás (ha sikerül).

1. Logikai alapú megoldás

Felírjuk a kockavilág axiómáit. Ezeket tartalmazza az 1-4. oldal. Ezeket fel lehet használni az output állítás bizonyításában, a feltétel állítások mellett. A tételbizonyítás rezolúcióval történik. Egy igen egyszerű tétel bizonyítása látható az 5. oldalon.

Jó lenne ezt számítógéppel végezni. Ha valaki Prolog-ot szeretne használni, akkor a Sixtus Prolog rendelkezésre áll. Nem hiszem, hogy saját rezolváló program írása reális feladat lenne.

2. Állapottér reprezentáció és gráfkeresés (MI)

A kockákból álló konfigurációk egy n x n-es mátrix kitöltéseivel adhatók meg. Esetünkben egy 4-szer 4-es mátrix alkalmas kitöltésével ábrázoljuk az állapotokat. Egy teljes kitöltésre példa a következő:

c			
a			
d	b		

Ha a kitöltés folyamatát nézzük, akkor az üres mátrixszal együtt négy további mátrix rögzíti az újabb kockák elhelyezésének folyamatát. Rendezzük egy 5-szintű fába ezeket a kitöltéseket: a fa gyökere az üres mátrix (0. magasság), a további szinteken újabb kocka neve jelenik meg a mátrixban.

Ezt a reprezentációs fát generáljuk mélységi bejárással vagy visszalépéses kereséssel. Ha egy állapot sérti az input állításokat, akkor ott vágunk és visszalépünk. Végül annyi levelet generálunk, ahány teljesen kitöltött állapot felel meg az input állításoknak.

Megadható a kitöltésben olyan rendezettségi előírás, amellyel minden állapot pontosan egyszer jön létre.

A tételbizonyítás itt annyit jelent, hogy a létrehozott konfigurációkban ellenőrizzük az output állítás(ok) teljesülését.

3. Funkcionális programozás alapú megoldás

Ennek kidolgozását teljes egészében az érdeklődő hallgatókra bíznánk.