



Hibadiagnosztika elosztott rendszerekben

Bevezető

A mérés során megvalósítandó feladatok két csoportba oszthatók: determinisztikus és valószínűségi diagnosztikai algoritmusokra.

- A **determinisztikus algoritmusok** adott feltételek teljesülése esetén képesek teljes és konzisztens diagnosztikai képet előállítani. *Teljesnek* akkor nevezünk egy diagnosztikai eredményt, ha a rendszer minden egysége kapott valamilyen (hibátlan/hibás) minősítést, *konzisztensnek* pedig akkor, ha minden egység valós és az algoritmus által feltárt hibaállapota megegyezik. A teljes és konzisztens diagnosztika eléréséhez teljesítendő további feltétel(ek)re példa a mérési segédletben tárgyalt, a hibás egységek számára adott felső korlát, az ún. *t*-korlát.
- A **valószínűségi algoritmusok** nem garantálnak biztosan teljes és konzisztens megoldást, de nagy valószínűséggel helyes eredményt szolgáltatnak. A tévedés lehetőségért cserébe ezek a módszerek csak a teszteredményekből kinyert információt használják fel, azaz nem igényelnek a rendszerre nézve olyan további előírásokat, mint a *t*-korlát a determinisztikus módszereknél.

A valószínűségi algoritmusok lehetséges diagnosztikai tévedései három csoportba sorolhatók:

1. *Az ismeretlen egység* egy olyan processzort jelent, amelynek hibaállapotát az algoritmus nem tudta meghatározni. Ebben az esetben a diagnosztika nem teljes.
2. *Jóindulatú tévedés* esetén egy hibátlan processzort hibásnak minősítünk. Így csökkentjük ugyan a rendelkezésre álló erőforrások számát, de a biztonság javára tévedünk.
3. *Rosszindulatú tévedés* esetén egy hibás processzort hibátlannak minősítünk. Ekkor a rendszerből nem távolítjuk el a hibás komponenst, ami hibás adatokkal és hibaterjesztéssel az egész folyamat hibáját okozhatja. Az erőforrások számát nem csökkentjük.

A mérési feladat

A mérési feladat a következő pontban leírt valószínűségi diagnosztikai algoritmus megvalósítása egy C nyelven megírt diagnosztikai szubrutin formájában. A szubrutint a szimulációs környezetbe ágyazva le kell fordítani, majd az algoritmus helyességét próbafuttatásokkal ellenőrizni. (A futtatások eredményét statisztikailag értékelve a mérési jegyzőkönyvben is fel kell tüntetni!)

A mérés kiértékelése

A valószínűségi algoritmusok „tévedhetnek”. A jó- vagy rosszindulatú tévedések száma az adott valószínűségi algoritmus fontos jellemzője. Ezért a valószínűségi módszerek kiértékelésekor az algoritmus értékeléséhez több pontban elvégzett mérések alapján (adott rendszer méret esetén növekvő hibaszám, vagy adott számú hiba mellett növekvő rendszer méret, stb.) adjon statisztikát (táblázatos és diagram alakban) a jóindulatú és rosszindulatú diagnosztikai tévedések (és ha van, az ismeretlen egységek) számáról!

Szorgalmi feladat: ha tud, adjon ötleteket az adott algoritmus továbbfejlesztéséhez (és ha van idő, valósítsa is meg)! Cél, hogy a diagnosztikai tévedések száma csökkenjen.

A megvalósítandó algoritmus leírása

Scheinerman algoritmusa valószínűségi diagnosztikai algoritmusa eljárásának első lépésében előállítja a tesztelési gráfnak azt a részgráfját, amely csak 0-éleket (hibátlan teszteredményekkel jelölt éleket) tartalmaz. Ennek a részgráfnak megkeresi az összes *erősen összekötött komponensét*. (Azokat a részgráfokat, amelyben csak a 0-val jelölt hibátlan teszteredményeket jelképező irányított éleken lépkedve a részgráf bármely csomópontjából bármelyik másik csomópontja irányított úttal elérhető). Mint a mérési útmutatóban bemutatott példán is láthattuk, a csupa 0 teszteredménnyel címkézett irányított körökben szereplő egységek *azonos hibaállapotúak*, a 0-éleket tartalmazó erősen összekötött komponensek keresése tehát az ilyen halmazok összegyűjtését célozza meg.

Ezek után kiválasztja a megtalált erősen összekötött komponensek közül azt, amelyik a legtöbb egységet tartalmazza, és ezeket az egységeket hibátlan címkével látja el. A rendszer megmaradó részét az így felfedezett hibátlan egységek által elvégzett tesztekkel levonható következtetésekkel minősíti. (Természetesen a módszer lényegéből az is következik, hogy Scheinerman algoritmusa nem minden esetben szolgáltat teljes diagnosztikát.)

Valósítsa meg Scheinerman algoritmusát! Fejlessze tovább az algoritmust úgy, hogy az összefüggő komponensek keresése során az ellentmondásokat (ezek tipikusan a $t_{i,i+1} = 0, \dots, t_{k-1,k} = 0, t_{k,i} = 1$ jellegű hurkok) is fel tudja fedezni, és az ezekből levonható biztos következtetéseket is kezelni tudja!