5. fejezet

Megegyezés vonalhibák esetében

Ebben és a következő két fejezetben a *megegyezés elérésének* problémájával foglalkozunk osztott hálózat esetében. Az effajta problémáknál a hálózati folyamatok egy bizonyos típusú kezdeti értékkel indulnak és feltételezhető, hogy végül a kimenetként adott értékük ugyanabba a típusba tartozó érték lesz. A kimenetektől elvárjuk, hogy azonosak legyenek – a folyamatoknak meg kell *egyezniük* –, jóllehet a bemenetek tetszőlegesek lehetnek. Rendszerint adott egy *érvényességi feltétel*, mely leírja a megengedett kimeneti értékeket minden egyes bemeneti mintára.

A megegyezési problémák az üzenetek egyszerű cserélgetésével általában könnyen megoldhatók, amikor a rendszer alkotóelemei hibamentesek. Izgalmasabbá válik a témánk, ha a problémákat hibákat tartalmazó környezetben tanulmányozzuk. Ebben a fejezetben az alapvető megegyezési problémát kommunikációs hibák jelenlétében fogjuk vizsgálni, míg a hatodik fejezet a folyamatok hibáival foglalkozik. A hetedik fejezet szintén az alapprobléma néhány olyan változatát tartalmazza, amelyekben folyamathibák állnak a középpontban.

Megegyezési problémák számos osztott számítógépes alkalmazás során felmerülnek. Például a folyamatok megpróbálhatnak megegyezni abban, hogy egy osztott adatbázisbeli tranzakció eredményét véglegesítsék vagy elvessék; vagy a folyamatok megpróbálhatnak egyetérteni egy repülőgép repülési magasságának megbecslése során, több magasságmérő adataira alapozva döntésüket; vagy megpróbálhatnak egyetérteni abban, hogy különböző folyamatok által végzett vizsgálatok alapján vajon hibásnak mondható-e a rendszer egy alkotóeleme.

Azt a sajátságos megegyezési problémát, amit ebben a fejezetben mutatunk be, *összehangolt támadási problémának* nevezzük; ez a megegyezés elérésének egy alapvető problémája arra az esetre, amikor az üzenetek elveszhetnek. Kezdetnek bemutatunk egy alapvető megoldhatatlansági eredményt determinisztikus rendszerekre vonatkozólag, majd megvizsgáljuk, mire juthatunk a véletlenített megoldás esetében. Megmutatjuk, hogy a probléma megoldható egy véletlenített algoritmussal, egy bizonyos (bár számottevő) hibaszázalék mellett. Továbbá belátjuk, hogy ez a bizonyos hibaszázalék elkerülhetetlen.

5.1.. Az összehangolt támadási feladat – determinisztikus változat

A probléma vázlatos (így kissé pontatlan) megfogalmazásával kezdjük, egy harctéri ütközet előtti párbeszéd forgatókönyvén keresztül.

> Adottak tábornokok, akik különböző irányokból összehangolt támadást terveznek egy közös célpont ellen. Tudják, hogy a támadás csak akkor lehet sikeres, ha mindannyian támadnak; ha csak néhány tábornok támad, seregeik megsemmisülnek. Minden tábornoknak van egy kezdeti véleménye arról, hogy vajon a serege kész-e a támadásra.
>
> A tábornokok különböző helyeken tartózkodnak. A szomszédos tábornokok képesek üzenetet váltani, de csak gyalog közlekedő hírnökökön keresztül. A hírnökök azonban eleshetnek vagy fogságba eshetnek, és ennek következtében üzeneteik is elveszhetnek. A kommunikációnak csak ezt a megbízhatatlan formáját használva a tábornokoknak egyeztetniük kell, hogy támadjanak-e vagy sem. Ezenfelül a tábornokoknak támadniuk kell, amennyiben ez lehetséges.
>
> (Feltételezzük egyrészt, hogy a tábornokok „kommunikációs gráfja" irányítatlan és összefüggő gráf, valamint, hogy az összes tábornok ismeri a gráfot. Feltételezzük továbbá azt is, hogy ismert egy felső korlát arra az időre, amennyi idő alatt egy sikeres hírnök kézbesíti az üzenetet.)

Ha mindegyik hírnök megbízható, minden tábornok küldhet üzenetet az összes többi tábornoknak (lehetséges, hogy több lépésben), megüzenve, hogy akar-e támadni. A „kommunikációs gráf" átmérőjével megegyező számú „menet" után minden tábornok ismerni fogja az összes információt. Ezután alkalmazhatják a közösen megállapított szabályt, hogy ugyanazt a döntést hozzák a támadásról. Kizárólag akkor dönthetnek úgy, hogy támadnak, ha az összes többi tábornok is támadni szándékozik.

Egy olyan modellben, amelyben a hírnökök eltűnhetnek, ez az egyszerű algoritmus nem működik. Kiderül, hogy nem csak az algoritmussal van probléma: bebizonyítjuk, hogy nincs olyan algoritmus, amely mindig hibátlanul megoldja ezt a feladatot.

A leírás mögött álló valóságos számítástudományi probléma az osztott adatbázisok véglegesítési problémája. Ez a feladat folyamatoknak egy csoportját tartalmazza, melyek részt vesznek egy adatbázis-tranzakció feldolgozásában. A feldolgozás után minden folyamat elérkezik egy kezdeti „véleményhez", arról, hogy a tranzakció *véglegessé* váljon-e (azaz, az eredmények tartóssá váljanak-e és szabadon használhatók legyenek-e a többi tranzakció által) vagy *elvessék* (azaz, az eredményeit eldobják). Egy folyamat általában a tranzakció véglegesítését támogatja, ha a tranzakció szerinti összes helyi számítást sikeresen végrehajtotta, máskülönben a tranzakció elvetését javasolja. Feltételezzük, hogy a folyamatok kommunikálnak és végül egyetértenek az eredmények egyikével, a *véglegesítéssel* vagy az *elvetéssel*. Amennyiben lehetséges, a *véglegesítést* kell választani.

Mielőtt bebizonyítanánk, hogy nem lehet eredményre jutni, fogalmazzuk meg a problémát formálisan is, megszüntetve a többértelműségeket. Vegyünk n folyamatot, indexeik legyenek rendre $1, \ldots, n$, amelyek egy tetszőleges irányítatlan gráf szerinti hálózatba rendezettek, és minden folyamat ismeri a teljes gráfot, beleértve a folyamatok indexeit is. Mindegyik folyamat egy kiválasztott állapot összetevőjének $\{0, 1\}$-beli bemeneti értékével kezd. Az 1 jelenti a „támadást" vagy *véglegesítést*, a 0 pedig a „nem támadást" vagy *elvetést*. Ugyanazt a szinkron modellt használjuk, mint amivel eddig dolgoztunk, kivéve, hogy most megengedjük, hogy az üzenetek közül néhány elvesszen egy végrehajtás alatt. (Meghatározást lásd a 2.2. alfejezetben.) A cél, hogy végül mindegyik folyamat kiadjon egy döntést a $\{0, 1\}$ halmazból, a *döntési* állapot összetevőjét 0 vagy 1 értékre állítva. A folyamatok által hozott döntéssel szemben a következő három elvárásunk van.

Megegyezés. Nem lehet két olyan folyamat, mely különböző érték mellett dönt.

Érvényesség.

1. Ha minden folyamat 0-val kezd, akkor csak a 0 a lehetséges döntési érték.

2. Ha minden folyamat 1-gyel kezd és az összes üzenetet kézbesítették, akkor csak az 1 a lehetséges döntési érték.

Befejeződés. Végül minden folyamatnak döntésre kell jutnia.

A megegyezés és a befejezés követelményei természetesen adódnak. Az érvényesség feltételei csak egy lehetőséget írnak le – itt több használható alternatíva is létezik. Az érvényesség feltételei általában azt az elképzelést fejezik ki, hogy a kialakított döntési érték „ésszerű"; ebben az esetben például az érvényességi feltételek második része kizárja azt a triviális protokollt, hogy a döntés mindig 0. A fentebb megadott érvényességi feltétel meglehetősen gyenge: például ha minden folyamat 1-gyel kezd, akkor elvárható, hogy az algoritmus 1-es döntést hozzon, de ha minden folyamat 1-gyel kezd és akár csak egyetlen üzenet is elveszett, akkor megengedett, hogy az algoritmus döntése 0 legyen. Ez a gyenge szabály is megfelelő, mivel ebben a fejezetben egy megoldhatatlansági eredmény bizonyítására összpontosítunk. Látni fogjuk, hogy a problémának még ezt az egyszerű változatát sem lehet megoldani tetszőleges, legalább két csúcspontot tartalmazó gráf esetében.

Bebizonyítjuk a megoldhatatlansági eredményt arra az egyedi esetre, amikor két csúcspont egy éllel van összekötve. Az olvasóra bízzuk annak belátását, hogy az erre az esetre vonatkozó megoldhatatlanság magában foglalja a megoldhatatlanságot tetszőleges, legalább két csúcsot tartalmazó gráf esetében. A bizonyításban a végrehajtások megkülönböztethetetlenségének 2. fejezetben megadott formális meghatározását ($\sim$) fogjuk használni.

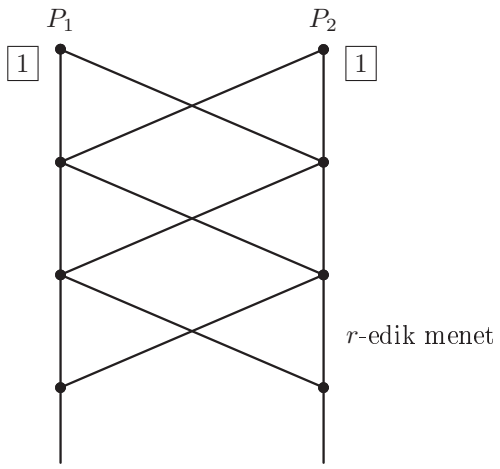
5.1. tétel . *Legyen G egy olyan gráf, amelynek csúcsai P_1 és P_2, egyetlen éllel összekötve. Ekkor nincs olyan algoritmus, amely megoldja az összehangolt támadási problémát G-ben.*

Bizonyítás. Indirekt módon bizonyítunk feltételezzük, hogy létezik egy megoldás; legyen ez az A algoritmus. Az általánosság megsértése nélkül feltételezhetjük, hogy minden folyamat esetében egy adott bemeneti értékhez pontosan kezdeti állapot tartozik, amelyik minden kezdeti értéket tartalmaz; ebből következik, hogy ha rögzített a bemenet és a sikeres üzenetek mintája, akkor a rendszernek pontosan egy végrehajtási sorozata van. Szintén az általánosság megszorítása nélkül feltételezhetjük, hogy a folyamatok minden menetben küldenek üzenetet A-ban, mivel rákényszeríthetjük őket arra, hogy álüzenetek küldjenek.

Legyen α az a végrehajtási sorozat, mely akkor valósul meg, amikor mindkét folyamat 1-es értékkel kezd és az összes üzenetet kézbesítették. A befejezés követelménye szerint végül mindkét folyamat dönt és az érvényességi feltétel második részének megfelelően mindkettő az 1-es értéket választja. Tegyük fel, hogy mindkettő r meneten belül dönt. Most legyen α_1 azonos α-val, kivéve, hogy az első r menet utáni üzenetek elvesznek. α_1-ben szintén mindkét folyamat az 1-et választja r meneten belül. Az α_1 kommunikációs mintája az 5.1. ábrán látható. Az élek olyan üzeneteket jelentenek, amelyeket kézbesítettek; az elküldött, de nem kézbesített üzeneteket egyszerűen nem rajzoljuk meg.

α_1-gyel kezdve képezzük a végrehajtási sorozatok egy sorozatát, melynek minden eleme megkülönböztethetetlen a sorban előtte lévőtől az egyes folyamatok szempontjából: ebből az következik, hogy mindezen végrehajtási sorozatok eredménye ugyanaz a döntési érték lesz.

Legyen α_2 ugyanaz a végrehajtási sorozat, mint α_1, kivéve, hogy az utolsó (r-edik) menetben a P_1 folyamatnak a P_2-höz küldött üzenete nem érkezik meg (lásd az 5.2. ábrát). Bár emiatt P_2 az r-edik menet után eltérő állapotba kerülhet α_2 végrehajtási sorozatban, mint az α_1-ben, ezt az eltérést nem tudatja P_1-gyel; ezért $\alpha_1 \overset{1}{\sim} \alpha_2$. Minthogy



5.1.. ábra. Az üzenetváltások mintája α_1 végrehajtási sorozatban.

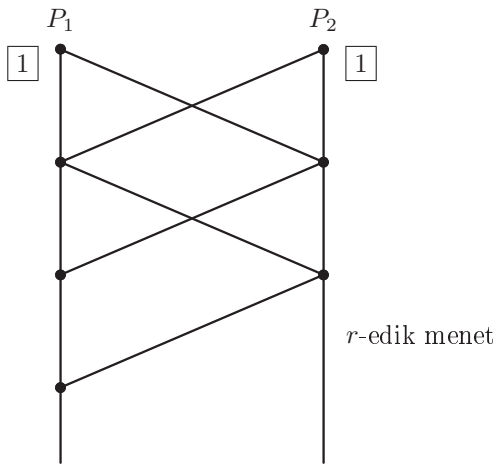
P_1 döntése α_1-ben 1 volt, 1 lesz α_2-ben is. A befejezési és megegyezési tulajdonságokból következik, hogy P_2 folyamat is 1-es döntést hoz α_2-ben. Következő lépésben legyen α_3 ugyanaz mint α_2, kivéve, hogy P_2 utolsó P_1-hez küldött üzenete elvész. Minthogy $\alpha_2 \overset{2}{\sim} \alpha_3$, P_2 folyamat 1-est dönt α_3-ban, és a befejezési és megegyezési tulajdonságok miatt P_1 is így tesz.

Ezt folytatva felváltva eltávolítjuk hol a P_1, hol a P_2 folyamat utolsó üzenetét, végül egy α' végrehajtási sorozathoz jutunk, melyben mindkét folyamat 1-essel kezd és egyetlen üzenet sincs kézbesítve. A fenti okok miatt mindkét folyamat arra kényszerül, hogy ebben az esetben is 1-es döntést hozzon.

Vegyünk azonban egy α'' végrehajtási sorozatot, amelyben P_1 folyamat 1-essel kezd, de P_2 folyamat 0-val és nincs kézbesített üzenet. Ekkor $\alpha'' \overset{1}{\sim} \alpha'$ fennáll, következésképp a P_1 folyamat még mindig 1-es mellett dönt és P_2 is így tesz a befejezési és megegyezési tulajdonságok következtében. Minthogy $\alpha'' \overset{2}{\sim} \alpha'''$, ahol α''' az a végrehajtási sorozat, melyben mindkét folyamat 0-val kezd és nincs kézbesített üzenet, P_2 folyamat 1-est dönt α'''-ban. Ez azonban ellentmondás, mivel az érvényességi feltétel első része szerint α'''-ban mindkét folyamatnak 0-ás döntést kell hoznia. □

Az 5.1. tétel az osztott hálózatokban rejlő képességek egy alapvető korlátját mutatja be. Azt sugallja, hogy kevés az esélyünk, hogy megoldást találjunk az alapvető megegyezési problémára, csakúgy mint az osztott adatbázisok véglegesítési problémájára, megbízhatatlan kommunikáció esetében. Mindezeken túl a probléma néhány változatát valós rendszerekben meg kell oldani. Az 5.1. tételben kimondott korláton két módon kerekedhetünk felül: vagy megerősítjük a modellt, vagy gyengítünk a probléma követelményein.

Az egyik megközelítés, hogy megtartva a folyamatok determinisztikusságát valószínűségi feltevéseket állítunk az üzenetek elvesztésével kapcsolatosan, majd engedélyeznünk kell a megegyezési és/vagy érvényességi szabályok megsérülését néhány esetben. Egy ilyen környezetben működő algoritmus kifejlesztését meghagyjuk gyakorlatnak(lásd 5-3. gyakorlat). Egy másik megközelítésben megengedjük a folyamatoknak a véletlenszerűség használatát, valamint ismét megengedjük a megegyezési és/vagy érvényességi szabályok megsértését néhány esetben. Ezt a megközelítést tárgyaljuk az 5.2. alfejezetben.



5.2.. ábra. Az üzenetváltások mintája α_2 végrehajtási sorozatban.

5.2.. Az összehangolt támadási probléma – véletlenített változat

Ebben a részben az összehangolt támadási problémát olyan körülmények között tekintjük át, ahol a folyamatok véletlenszerűen viselkedhetnek. Az előző részhez hasonlóan, vegyünk n folyamatot egy tetszőleges, irányítatlan, ismert gráf szerkezetű hálózatban. Mindegyik folyamat egy kijelölt rendszerösszetevőben beállított $\{0,1\}$ beli bemenő értékkel kezd; feltesszük, hogy mindegyik folyamat egy adott bemeneti értékhez pontosan egy kezdeti állapottal rendelkezik. Feltesszük továbbá, hogy a protokoll rögzített számú $r \geq 1$ meneten belül befejeződik, még pontosabban azt, hogy az r-edik menet végére mindegyik folyamat kimenete egy $\{0,1\}$ halmazbeli döntés, amely a folyamat *döntési* változóját a 0 vagy 1 értékre állítja. Minden k, $1 \leq k \leq r$ menetben és mindegyik él mentén történik egy üzenetküldés, ezek közül néhány elveszhet.

A cél lényegében ugyanaz, mint az előbb volt, kivéve, hogy most gyengítünk a problémát leíró állításokon, bizonyos valószínűséggel megengedjük a hiba előfordulását. Nevezetesen, ugyanazt az érvényességi feltételt használjuk, mint az előbb, de gyengítünk a megegyezés feltételén (kis ϵ valószínűséggel megengedjük azt az esetet, amelyben nem jön létre megegyezés). Alsó és felső korlátot adunk ϵ elérhető értékeire az r függvényében, ahol r a menetek száma. Ahogy látni fogjuk, ϵ elérhető értékei nem kicsik.

5.2.1.. Formális modellezés

A probléma formális megfogalmazásához tisztázni kell a valószínűségi állításokat – a helyzet bonyolultabb, mint az MFH problémánál volt a 4.5. alfejezetben. A bonyolultságot az okozza, hogy a megvalósuló végrehajtási sorozat nem csak a véletlenszerű választásoktól függ, hanem attól is, mely üzenetek vesznek el. Nem azt fogjuk feltételezni, hogy az üzenetek elvesztését a véletlen határozza meg. Inkább úgy képzeljük, hogy azt egy „ellenfél” irányítja, amely megpróbálja olyan nehézzé tenni a dolgokat az algoritmus számára, amennyire csak lehet; az algoritmust úgy értékeljük ki, hogy feltesszük a lehetséges legrosszabb viselkedést a szóba jöhető ellenfelek osztályán.

Formálisan, meghatározunk egy *kommunikációs mintát*, amely az alábbi halmaz tet-

szőleges részhalmaza:

$$\{(i,j,k) : (i,j) \text{ egy éle a gráfnak, és } 1 \leq k\}.$$

A γ-val jelölt kommunikációs minta definíció szerint *jó* lesz, ha $k \leq r$ esetében minden $(i,j,k) \in \gamma$ (ez csak erre a fejezetre vonatkozik – a „jóság" egy másik fogalmát használjuk a 6. fejezetben). Egy jó kommunikációs minta olyan üzenetek halmazát ábrázolja, amelyeket kézbesítettek a végrehajtási sorozat során: ha (i,j,k) eleme a kommunikációs mintának, az azt jelenti, hogy P_i-nek a k-adik menetben P_j számára küldött üzenetét sikeresen átadták.

Az *ellenfél* fogalma, amelyet itt használunk, egy tetszőleges választás az alábbiakból:

1. bemeneti adatok hozzárendelése az összes folyamathoz;
2. egy jó kommunikációs minta.

Egy adott ellenfél esetében a folyamatok véletlenszerű választásai egy-egy egyedi végrehajtási sorozatot határoznak meg. Így minden egyes ellenfél esetében a folyamatok véletlenszerű választása egy valószínűségi eloszlást alkot a végrehajtási sorozatok halmazán. Ezt a valószínűségi eloszlást használva kifejezhetjük a valószínűségét az olyan eseményeknek, mint például: az összes folyamat egyetért. Az ellenfél szerepét hangsúlyozva, a Pr^B jelölést használjuk egy adott B ellenfél valószínűségi függvényének jelölésére.

Nézzük elölről az összehangolt támadási problémát ebben a véletlenített környezetben. Az állításnak paramétere ϵ, $0 \leq \epsilon \leq 1$.

Megegyezés. Minden B ellenfél esetében,

$$Pr^B[\text{a folyamatok közül néhány 0, néhány 1 mellett dönt}] \leq \epsilon \ .$$

Érvényesség. Az előbbivel azonos.

A befejezésre nem írunk elő feltételt, mivel feltettük, hogy a folyamatok r meneten belül döntést hoznak. Célunk az, hogy találjunk egy algoritmust, a lehető legkisebb ϵ mellett, és bebizonyítsuk, hogy ennél kisebb ϵ érték nem érhető el.

5.2.2.. Egy algoritmus

Az egyszerűség kedvéért ebben és a következő fejezetben figyelmünket az n-csúcspontú teljes gráfokra korlátozzuk. Az általános gráfokra való kiterjesztést gyakorlatnak hagyjuk meg. Erre az egyedi esetre bemutatunk egy egyszerű algoritmust, mely eléri az $\epsilon = \frac{1}{r}$ értéket.

Az algoritmus azon alapul, hogy a folyamatok mit tudnak egymás kezdeti értékéről, valamint mit tudnak arról, hogy a többi folyamat mit tud a kezdeti értékekről, és így tovább. Szükségünk van néhány Meghatározásra, hogy a tudás effajta fogalmát megfoghassuk.

Először tetszőleges γ kommunikációs mintára meghatározunk egy $\leq_\gamma$ reflexív parciális rendezést az (i,k) párokra, ahol i egy folyamat indexe, k egy időtényező, $0 \leq k$. (Emlékezzünk vissza a 2. fejezetre, „k időpont" a végrehajtási sorozat azon pontjára vonatkozik, amikor k menet már lezajlott.) Ez a rendezés a különféle folyamatok, különféle időpontjai közötti információáramlást ábrázolja. A reláció definíciója az alábbi:

1. $(i,k) \leq_\gamma (i,k')$ igaz minden i, $1 \leq i \leq n$, és minden k, k', $0 \leq k \leq k'$ esetében;

2. ha $(i,j,k)\,\epsilon\,\gamma$, akkor $(i,k-1) \leq_\gamma (j,k)$;
3. ha $(i,k) \leq_\gamma (i',k')$ és $(i',k') \leq_\gamma (i'',k'')$, akkor $(i,k) \leq_\gamma (i'',k'')$.

Az első pont az azonos folyamaton belüli információáramlást ábrázolja. A második pont a küldőtől az elfogadóhoz áramló információt írja le. A harmadik pont a tranzitivitást definiálja. Az információáramlás hasonló elgondolása jelenik majd meg a könyv későbbi fejezeteiben, például a 14., 16., 18. és 19. fejezetekben.

Most rekurzívan definiáljuk tetszőleges γ jó kommunikációs mintára az *információ szintjét*, legyen ez a $szint_\gamma(i,k)$, ahol i tetszőleges P_i folyamat és k, $0 \leq k \leq r$ tetszőleges időpont. Három eset lehetséges:

1. $k = 0$ esetében
   $szint_\gamma(i,k)$ legyen 0;
2. $k > 0$ és van olyan $j \neq i$, hogy $(j,0) \not\leq_\gamma (i,k)$ esetében
   $szint_\gamma(i,k)$ legyen 0;
3. $k > 0$ és $(j,0) \leq_\gamma (i,k)$ minden $j \neq i$ esetében
   minden $j \neq i$ értékre jelölje l_j a $\max\{szint_\gamma(j,k') : (j,k') \leq_\gamma (i,k)\}$ értéket. (Ez az a legmagasabb szint, ahol elérjük, hogy P_i ismeri P_j-t.) Vegyük észre, hogy $0 \leq l_j \leq k-1$ igaz minden j értékre. Ezután $szint_\gamma(i,k)$ definíció szerint legyen $1 + \min\{l_j : j \neq i\}$.

Más szavakkal, minden egyes folyamat a 0. szinten kezd; amikor értesül az összes többi folyamatról, előre lép az 1. szintre. Amikor értesül arról, hogy az összes többi folyamat elérte az 1. szintet, tovább lép a 2. szintre, és így tovább. Előfordul néhányszor, hogy a $szint_B(i,k)$ jelölést használjuk egy γ kommunikációs mintával megadott B ellenfélre, de ekkor ezen a $szint_\gamma(i,k)$ információs szintet értjük.

5.2.1. példa. Az információ szintje

Tegyük fel, hogy $n = 2$ és $r = 6$. Legyen γ egy jó kommunikációs minta, mely pontosan a következő hármasokból áll:

$$(1,2,1),(1,2,2),(2,1,2),(1,2,3),(2,1,4),(1,2,5),(2,1,5),(1,2,6)$$

Az 5.3. ábra mutatja a γ kommunikációs mintát. A P_1 és P_2 folyamatok információ szintjét a k, $0 \leq k \leq 6$ időpontokban a címkék jelzik.

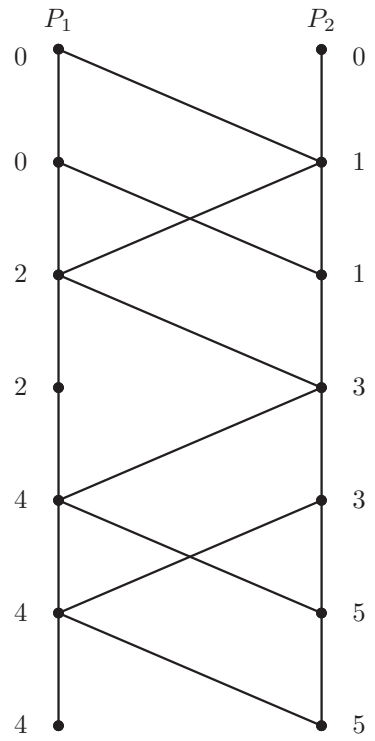
A következő lemma kimondja, hogy a különböző folyamatok információ szintje közötti eltérés nem lehet egynél nagyobb.

5.2. lemma . *Bármely γ jó kommunikációs minta és bármely k, $0 \leq k \leq r$ és bármely i és j esetében $|szint_\gamma(i,k) - szint_\gamma(j,k)| \leq 1$.*

Bizonyítás. A bizonyítást meghagyjuk gyakorlatnak (lásd 5-5. gyakorlat.). □

A következő lemma azt mondja ki, hogy abban az esetben, amikor minden üzenet sikeresen eljut a címzetthez, az információ szintje a menetek számával egyenlő.

5.3. lemma . *Amennyiben γ egy „teljes" kommunikációs minta, mely az összes (i,j,k) hármast tartalmazza $1 \leq k \leq r$ esetében, akkor $szint_\gamma(i,k) = k$ minden i és k értékre.*



5.3.. ábra. A γ jó kommunikációs minta.

Bizonyítás. A bizonyítást meghagyjuk gyakorlatnak (lásd 5-6. gyakorlat). □

A VÉLETLENÍTETTTÁMADÁS nevű algoritmus alapgondolata a következő.

VÉLETLENÍTETTTÁMADÁS algoritmus (vázlatosan)

A *szint* változó tárolja, hogy az egyes folyamatok (jelölésük P_i) a végrehajtási sorozatnak megfelelő kommunikációs minta szerint melyik szintnél tartanak. P_1 folyamat választ egy véletlen értéket, nevezzük ezt *kulcsnak*, ami egy egész szám az $[1, r]$ intervallumból. Ezt az értéket az algoritmus végigcipeli az összes üzeneten. Emellett a folyamatok kezdeti értékét is végigcipeljük az összes üzeneten.

Az r-edik menet után az egyes folyamatok döntése pontosan akkor lesz 1, ha a folyamat szintje legalább akkora, mint a *kulcs* értéke és a folyamat tudomására jutott, hogy az összes többi folyamat az 1 kezdeti értékkel indult.

VéletlenítettTámadás algoritmus (formálisan)

Az üzenet ábécéjét az (S, E, k) alakú hármasok alkotják, ahol S vektor minden egyes eleme egy $[0, r]$ intervallumba tartozó egész értéket rendel a megfelelő indexű folyamathoz, hasonlóan E vektor minden egyes eleme a $\{0, 1, \textit{nem\_ meghatározott}\}$ halmazba tartozó értéket rendeli a megfelelő indexű folyamathoz, k értéke pedig vagy egy egész érték az $[1, r]$ intervallumból, vagy lehet *nem_ meghatározott*.

állapotok$_i$:
menetek $\in \mathbb{N}$, kezdetben 0
döntés $\in \{\textit{ismeretlen}, 0, 1\}$, kezdetben *ismeretlen*
kulcs $\in [1, r] \cup$ *nem_ meghatározott*, kezdetben *nem_ meghatározott*
minden j, $1 \leq j \leq n$ esetében:
    érték$(j) \in \{0, 1, \textit{nem\_ meghatározott}\}$; kezdetben *érték*$(i)$ P_i kezdeti értéke és *érték*$(j) =$ *nem_ meghatározott* minden $j \neq i$ esetében
    szint$(j) \in [-1, r]$; kezdetben *szint*$(i) = 0$ és *szint*$(j) = -1$ minden $j \neq i$ esetében

A *szint*(j) változó feladata, hogy nyomon kövesse P_j folyamat azon legmagasabb szintjét, amiről - az üzenetváltásokon keresztül - P_i folyamat már tudomást szerzett. Mielőtt P_i bármit is hall P_j felől, minden $j \neq i$ esetében *szint*(j) alapértéke -1. A *véletlen*$_i$ függvényben használt *véletlen* egy $[1, r]$ intervallumba eső, egyenletes eloszlású véletlen értéket jelöl.

véletlen$_i$:
if $i = 1$ **and** *menetek* $= 0$ **then** *kulcs* $:=$ *véletlen*

üzenetek$_i$:
küldd el $(S, E, \textit{kulcs})$ hármast minden P_j folyamatnak, ahol S a *szint* vektor és E az *érték* vektor

átmenet$_i$:
menetek $:=$ *menetek* $+ 1$
legyen (S_j, E_j, k_j) a P_j folyamattól érkező üzenet az összes olyan P_j esetében, amelyiktől üzenet érkezett
if van olyan j, amire $k_j \neq$ *nem_ meghatározott* **then** *kulcs* $:= k_j$
for minden $j \neq i$ **do**
    if van olyan i', hogy $E_{i'}(j) \neq$ *nem_ meghatározott* **then**
        érték$(j) := E_{i'}(j)$
    if van olyan i', hogy $S_{i'}(j) >$ *szint*(j) **then** *szint*$(j) := \max_{i'}\{S_{i'}(j)\}$
szint$(i) := 1 + \min\{\textit{szint}(j) : j \neq i\}$
if *menetek* $= r$ **then**
    if *kulcs* $\neq$ *nem_ meghatározott* **and** *szint*$(i) \geq$ *kulcs* **and**
        érték$(j) = 1$ minden j esetében
            then *döntés* $:= 1$
            else *döntés* $:= 0$

A kód harmadik sora beállítja a *kulcs* komponenst; az nem okoz hibát, hogy ezt az értékadást többször is végrehajtja a kód, hisz ugyanaz az érték jár körbe *kulcs* értékként. Az ötödik sor a $j \neq i$ indexű folyamatok *érték* komponensét állítja be, ismét az egymásnak ellentmondó hozzárendelés veszélye nélkül. A hatodik sor a $j \neq i$ indexű folyamatok *szint* komponensét frissíti, ezekben tároljuk az egyes folyamatok P_i által már ismert szint értékei közül az éppen aktuális legnagyobbat. Majd P_i frissíti a saját *szint*

komponensét, beállítva, hogy eggyel nagyobb legyen, mint a többi folyamatról megtudott érték minimuma. Végül, ha az utolsó menetben vagyunk ($menetek = r$) P_i döntést hoz az előzőleg megadott szabály szerint.

5.4. tétel . *A* VÉLETLENÍTETTTÁMADÁS *algoritmus megoldja az összehangolt támadási probléma véletlenített változatát* $\epsilon = \frac{1}{r}$ *értékkel.*

Bizonyítás. A bizonyítás kulcsa az a segédállítás, hogy az algoritmus helyesen számítja ki a szint értékeket. Azaz, a VÉLETLENÍTETTTÁMADÁS bármely végrehajtási sorozata esetében, tetszőleges γ jó kommunikációs mintára és tetszőleges k, $0 \leq k \leq r$ értékekre igaz, hogy k menet után bármely i-re $szint(i)_i$ értéke egyenlő $szint_\gamma(i,k)$ értékkel. Úgyszintén, k menet után, ha $szint(i)_i \geq 1$, akkor $kulcs_i$ érték és $érték(j)_i$ érték – az utóbbi minden j esetében – meghatározott, továbbá az előbbi egyenlő a P_1 által választott *kulcs* értékkel, az utóbbiak pedig egyenlőek az egyes folyamatok kezdeti értékével.

Nyilvánvaló, hogy a VÉLETLENÍTETTTÁMADÁS algoritmus mindig befejeződik. Az érvényességet vizsgálva, ha minden folyamatnak 0 a kezdeti értéke, akkor nyilvánvalóan az egyetlen lehetséges döntés a 0 érték. Most tételezzük fel, hogy az összes folyamat az 1 értékkel kezd és minden üzenet megérkezik. Ekkor az 5.3. lemmából, valamint abból a tényből, hogy az algoritmus a szint értékeket helyesen számítja ki, az következik, hogy az r-edik menet azon pontján, ahol a döntés megszületik, minden i esetében $szint(i)_i = r$. Abból, hogy $szint(i)_i = r \geq 1$ az adott pontban, az következik, hogy $kulcs_i$ érték és $érték(j)_i$ érték is meghatározott, az utóbbi minden j esetében. Mivel a *kulcs* lehetséges értékei kisebb vagy egyenlők mint r, az egyetlen lehetséges döntési érték az 1.

Végül tekintsük a megegyezést. Legyen B tetszőleges ellenfél. Megmutatjuk, hogy

$$Pr^B[\text{néhány folyamat döntése 0, és néhány folyamat döntése 1}] \leq \epsilon.$$

Jelölje l_i tetszőleges i esetében a $szint(i)_i$ pillanatnyi értékét akkor, amikor P_i meghozza a döntését (az r-edik menetben). Az 5.2. lemmából következik, hogy l_i értékek egymástól legfeljebb eggyel térhetnek el. Ha a választott *kulcs* érték szigorúan nagyobb, mint $\max\{l_i\}$, vagy volt olyan folyamat, amely a 0 kezdeti értékkel indult, akkor mindegyik folyamat a 0 döntést hozza meg. Másrészről viszont, ha $kulcs \leq \min\{l_i\}$, és mindegyik folyamat kezdeti értéke 1, az összes folyamat az 1 döntést hozza meg. Ezért a „sikertelen megegyezés" csak akkor fordulhat elő, ha $kulcs = \max\{l_i\}$. Ennek az eseménynek a valószínűsége $\frac{1}{r} = \epsilon$, minthogy $\max\{l_i\}$ értéket a B ellenfél határozza meg, és a *kulcs* értéke egyenletes eloszlású a $[0, r]$ intervallumban. □

5.2.2. példa. A VÉLETLENÍTETTTÁMADÁS algoritmus viselkedése

Tekintsük azt az esetet, amikor $n = 2$ és $r = 6$. Tegyük fel, hogy a B ellenfél az 5.2.1. példában megadott γ jó kommunikációs mintát és a folyamatok bemeneteként az 1 értéket szolgáltatja. Legyen $\epsilon = \frac{1}{6}$. Az 5.4. tétel állítása szerint a megegyezés hiányának valószínűsége legfeljebb ϵ. Valójában ennek valószínűsége pontosan ϵ: ha a P_1 által választott *kulcs* értéke 5, akkor P_1 döntése 0, P_2 döntése 1 lesz; ha *kulcs* ≤ 4, akkor mindkettőjük döntése 1 lesz; és ha *kulcs* $= 6$, akkor mindketten a 0 mellett döntenek.

Másrészről viszont, ha az ellenfél a γ kommunikációs mintával együtt a bemenetek egy bármilyen másik kombinációját nyújtja, akkor a sikertelen megegyezés valószínűsége 0, mivel mindkét folyamat döntése 0 lesz.

Az 5.4. tétel bizonyításához használt gondolatainkra támaszkodva beláthatjuk, hogy a VÉLETLENÍTETTTÁMADÁS algoritmus erősebb érvényességi feltételt is kielégít, mint ahogy azt állítottuk. Nevezetesen megmutathatjuk, az alábbiakat.

Érvényesség.

1. Ha van olyan folyamat, amelyik 0-val kezd, a lehetséges döntési érték csak a 0.
2. Tetszőleges olyan B ellenfél esetében, amelyben az összes bemenet értéke 1,

$$Pr^B[\text{minden folyamat 1 mellett dönt}] \geq l\epsilon,$$

ahol l a folyamatok szintjei közül a legkisebb az r időpontban, B ellenfél mellett.

A második tulajdonság hasznos lehet számos alkalmazásnál, például a hadviselés, vagy az osztott adatbázisokban a véglegesítés, ahol kívánatos, hogy előnyben részesüljön a pozitív kimenetel. Ha például csak egy üzenet vész el, az összehangolt támadás valószínűsége garantáltan magas, legalább $\frac{r-1}{r}$. Annak bizonyítását, hogy a VÉLETLENÍTETTTÁMADÁS algoritmus kielégíti az erősebb érvényességi feltételeket, gyakorlatnak hagyjuk meg.

5.2.3.. Alsó korlát a megegyezés hiányának valószínűségére

Most belátjuk, hogy az 5.4. tételben adott korlátnál jobbat nem érhetünk el. (Emlékezzünk vissza az előző alfejezetre, hogy csak az n-csúcsú teljes gráfokat vizsgáljuk.)

5.5. tétel . *Az összehangolt támadási probléma véletlenített változatának tetszőleges r-menetes algoritmusára igaz, hogy a megegyezés hiányának valószínűsége legalább* $\frac{1}{r+1}$.

A fejezet további részében feltesszük, hogy egy bizonyos A algoritmus egy n-csúcsú teljes gráfban megoldja az összehangolt támadási problémát, úgy hogy a megegyezés hiányának valószínűsége ϵ; bebizonyítjuk, hogy $\epsilon \geq \frac{1}{r+1}$.

A tétel bizonyításához egy további meghatározásra van szükségünk. Vegyünk egy B ellenfelet, γ legyen az ellenfél kommunikációs mintája, P_i pedig egy tetszőleges folyamat, definiáljunk egy másik ellenfelet is, nevezzük *ritkít*(B, i)-nek. A *ritkít*(B, i) a B ellenfelet egyszerűen „megritkítja", azaz eltávolítja az olyan információkat, amelyek nem jutnak el P_i folyamathoz B-ben. $B' = \textit{ritkít}(B, i)$ meghatározása így szól:

1. ha $(j, 0) \leq_\gamma (i, r)$, akkor P_j bemenete B'-ben ugyanaz, mint B-ben, egyébként pedig 0;
2. a (j, j', k) hármas pontosan akkor eleme a B' kommunikációs mintának, ha eleme a B kommunikációs mintának, és $(j', k) \leq_\gamma (i, r)$.

Azaz B' ellenfél mindazon üzeneteket tartalmazza, amelyekről P_i a B-ben értesült, de a többit nem, és B' azon bemeneti értékeket, amelyekről P_i nem értesült, B-ben 0-nak határozza meg. A következő lemma szerint az ellenfél megritkított változata elegendő, hogy meghatározzuk a kimenetek valószínűségi eloszlását.

5.6. lemma . *Ha B és B' mindketten ellenfelek, P_i egy folyamat, és ritkít(B,i) = ritkít(B',i), akkor $Pr^B[P_i$ döntése 1$] = Pr^{B'}[P_i$ döntése 1$]$.*

Bizonyítás. A bizonyítást meghagyjuk gyakorlatnak (lásd 5-11. gyakorlat). □

Az 5.5. tétel bizonyítása a következő lemmán alapul.

5.7. lemma . *Legyen B egy tetszőleges ellenfél, amelyre minden bemeneti érték 1, és legyen P_i egy tetszőleges folyamat, akkor*

$$Pr^B[P_i \text{ döntése } 1] \leq \epsilon(szint_B(i,r)+1).$$

Bizonyítás. $szint_B(i,r)$ szerinti indukcióval.

Alapeset: tegyük fel, hogy $szint_B(i,r) = 0$. Legyen $B' = ritkít(B,i)$. Ekkor $ritkít(B',i) = B' = ritkít(B,i)$, ezért az 5.6. lemma szerint

$$Pr^B[P_i \text{ döntése } 1] = Pr^{B'}[P_i \text{ döntése } 1]. \tag{5.1}$$

Mivel $szint_B(i,r) = 0$, van olyan P_j folyamat, $j \neq i$, melyre $(j,0) \not\leq_\gamma (i,r)$, ahol γ a B kommunikációs mintája. Ekkor B' ellenfél a 0 kezdeti értéket rendeli P_j folyamathoz, és kommunikációs mintája nem tartalmaz P_j-nek címzett üzenetet. Ebből az következik, hogy $ritkít(B',j)$ az a triviális ellenfél, amelyben minden kezdeti érték 0 és a kommunikációs mintája nem tartalmaz üzeneteket. Jelölje B'' ezt a triviális ellenfelet. Innen $ritkít(B'',j) = B'' = ritkít(B',j)$, ezért az 5.6. lemma szerint

$$Pr^{B'}[P_j \text{ döntése } 1] = Pr^{B''}[P_j \text{ döntése } 1].$$

Az érvényességi feltételből következik, hogy

$$Pr^{B''}[P_j \text{ döntése } 1] = 0,$$

ezért

$$Pr^{B'}[P_j \text{ döntése } 1] = 0.$$

Mivel a megegyezés hiányának valószínűsége legfeljebb ϵ, azt kapjuk, hogy

$$|Pr^{B'}[P_i \text{ döntése } 1] - Pr^{B'}[P_j \text{ döntése } 1]| \leq \epsilon.$$

Ekkor

$$Pr^{B'}[P_i \text{ döntése } 1] \leq \epsilon,$$

az 5.1. egyenletből pedig az következik, hogy

$$Pr^{B}[P_i \text{ döntése } 1] \leq \epsilon,$$

amire szükségünk volt.

Indukciós lépés: tegyük fel, hogy $szint_B(i,r) = l > 0$, és tegyük fel, hogy a lemma igaz minden olyan szintre, amely kisebb, mint l. Legyen $B' = ritkít(B,i)$. Ekkor az 5.6. lemmából következik, hogy

$$Pr^B[P_i \text{ döntése } 1] = Pr^{B'}[P_i \text{ döntése } 1]. \tag{5.2}$$

Minthogy $szint_B(i,r) = l$, a *szint* definíciójából következik, hogy van olyan P_j folyamat, amelyre $szint_{B'}(j,r) \leq l-1$. Az indukciós feltevésből adódik, hogy

$$\begin{aligned} Pr^{B'}[P_j \text{ döntése 1}] &\leq \epsilon(szint_{B'}(j,r)+1) \\ &\leq \epsilon l. \end{aligned}$$

De mivel a megegyezés hiányának a valószínűsége legfeljebb ϵ, így fennáll, hogy

$$|Pr^{B'}[P_i \text{ döntése 1}] - Pr^{B'}[P_j \text{ döntése 1}]| \leq \epsilon.$$

Ekkor

$$Pr^{B'}[P_i \text{ döntése 1}] \leq \epsilon(l+1),$$

az 5.2. egyenletből pedig az következik, hogy

$$Pr^{B}[P_i \text{ döntése 1}] \leq \epsilon(l+1),$$

amire szükségünk volt. □

Most már bebizonyíthatjuk a tételt.

Bizonyítás. (5.5. tétel) Legyen B az az ellenfél, amelyre minden bemenet értéke 1, és nem vesznek el üzenetek. Annak valószínűsége, hogy mindegyik folyamat döntése 1 lesz, legfeljebb annyi lehet, mint annak a valószínűsége, hogy a folyamatok egyikének a döntése 1, ami az 5.7. lemma szerint legfeljebb $\epsilon(szint_B(i,r)+1) \leq \epsilon(r+1)$. De az érvényességi feltétel szerint az összes folyamatnak az 1 mellett kell döntenie valamennyi olyan végrehajtási sorozatban, melyet ez az ellenfél állít elő; ennélfogva annak valószínűsége, hogy mindannyiuk döntése 1 lesz, pontosan 1. Ebből következik, hogy $\epsilon(r+1) \geq 1$, amiből viszont adódik, hogy $\epsilon \geq \frac{1}{r+1}$, amit bizonyítani akartunk. □

5.3.. Megjegyzések a fejezethez

Az összehangolt támadási probléma megfogalmazása Graytől származik [142], aki ezt az osztott adatbázisokban előforduló véglelegesítési probléma modellezéséhez használja. A determinisztikus modell megoldhatatlansági eredménye is Graynek köszönhető [142]. Az összehangolt támadási probléma véletlenített változatának eredményeit Varghese és Lynch munkájában találhatjuk meg [281].

5.4.. Gyakorlatok

5-1. Mutassuk meg, hogy az összehangolt támadási probléma (determinisztikus változatának) megoldása bármely nem triviális, összefüggő gráf esetében magában foglalja a probléma megoldását arra az egyszerű, két pontból álló gráfra, mely egy éllel van összekötve. (Ebből következik, hogy a probléma megoldhatatlan tetszőleges, nem triviális gráf esetében.)

5-2. Tekintsük a (determinisztikus) összehangolt támadási probléma következő változatát. Tegyük fel, hogy a hálózat $n > 2$ résztvevőből álló teljes gráf. A befejezési és érvényességi feltételek az 5.1. alfejezetben leírtakkal azonosak. Azonban a megegyezési feltételt gyengítjük: „Ha van olyan a folyamatok között, amelyik döntése 1, akkor legalább kettőnek 1-est kell döntenie." (Azaz szeretnénk kizárni azt az esetet, amikor egy tábornok magányosan támad, de megengedjük azt, hogy két vagy több tábornok együtt támadjon.) Vajon ez a probléma megoldható, vagy nem? Bizonyítsuk.

5-3. Tekintsük az összehangolt támadási problémát vonalhibák esetében arra az egyszerű esetre, amikor két folyamat egy éllel van összekötve. Tegyük fel, hogy a folyamatok determinisztikusak, de az üzenetrendszer véletlenített, abban az értelemben, hogy mindegyik üzenetnek van egy független p valószínűségi értéke $0 < p < 1$, ami annak a valószínűségét adja meg, hogy az üzenet sikeresen megérkezik. (Ahogy általában, most is megengedjük, hogy a folyamatok menetenként csak egy üzenetet küldjenek.) Tervezzünk ezekkel a beállításokkal olyan algoritmust, mely rögzített r számú meneten belül befejeződik, a megegyezés hiányának valószínűsége legfeljebb ϵ, és ehhez hasonlóan az érvényességi feltétel megsértésének valószínűsége is legfeljebb ϵ. A lehető legkisebb ϵ érték elérésére törekedjünk.

5-4. Az előző gyakorlat kikötései szerinti modellben adjunk alsó korlátot az ϵ értékére, bizonyítsuk be, hogy ez az elérhető legalacsonyabb érték.

5-5. Bizonyítsuk be az 5.2. lemmát.

5-6. Bizonyítsuk be az 5.3. lemmát.

5-7. Bizonyítsuk be nagyon körültekintően az 5.4. tétel bizonyításának első segédtételét, mely szerint a VÉLETLENÍTETTTÁMADÁS algoritmus helyesen számolja ki a *szint* értékeket, és helyesen szállítja a kezdeti értékeket és a kulcsot.

5-8. Bizonyítsuk be a VÉLETLENÍTETTTÁMADÁS algoritmussal kapcsolatos erősebb érvényességi feltételt – melyet az 5.2.2. szakasz végén adtunk meg, azaz bizonyítsuk be a következőket:

(a) ha van egy olyan folyamat, amelyik 0-val kezd, csak 0 lehet a végleges döntés;
(b) tetszőleges olyan B ellenfél esetében, amelyben az összes bemenet értéke 1,

$$Pr^B[\text{minden folyamat 1 mellett dönt}] \geq l\epsilon,$$

ahol l a folyamatok szintjei közül a legkisebb az r időpontban, B ellenfél mellett.

5-9. Általánosítsuk az összehangolt támadási probléma véletlenített változatát úgy, hogy megengedjük ϵ valószínűséggel mind az érvényességi, mind a megegyezési szabályok megsértését. Írjuk át a VÉLETLENÍTETTTÁMADÁS algoritmust úgy, hogy a módosított feltételek mellett elérje a lehető legkisebb ϵ értéket. Végezzünk elemzést.

5-10. Általánosítsuk a VÉLETLENÍTETTTÁMADÁS algoritmust, és az elemzését az általános (nem szükségszerűen teljes), irányítatlan gráfokra.

5-11. Bizonyítsuk be az 5.6. lemmát.

5-12. Általánosítsuk az 5.5. tételben kapott alsó korlátot az általános (nem szükségszerűen teljes), irányítatlan gráfokra.

5-13. Mi történne a fejezetben tárgyalt, véletlenített környezettel kapcsolatos eredményekkel, ha az ellenfél kommunikációs mintája nem lenne előre rögzítve, mint ahogy eddig feltettük, hanem az ellenfél közvetlen irányítással határozhatná meg azt. Pontosabban szólva, tegyük fel, hogy az ellenfél képes arra, hogy megvizsgálja a végrehajtási sorozatot bármely k-adik menettől visszafelé a kezdetig, mielőtt döntene, hogy a k-adik menetbeli üzenetek közül melyek legyenek kézbesítve.

(a) Milyen ϵ korlát garantálható a VÉLETLENÍTETTTÁMADÁS algoritmus esetében a megegyezés hiányára, ilyen közvetlen irányításra képes ellenfelek esetében?
(b) Adhatunk-e valamilyen érdekes alsó korlátot az elérhető ϵ értékekre?

Tárgymutató