

Peregi Tamás

Aknakereső játék logikai modellezése

Mesterséges intelligencia házi feladat

Neptun kód: A0QX87

2010. december 8.

Bevezetés

Az aknakereső egy népszerű egyszemélyes logikai játék, amely szinte minden mai személyi számítógépen megtalálható. A játék első változatai a 80-as évek elején jelentek meg, ma ismert formájában 1990-ben, a Microsoft Windows Entertainment Pack nevű játékgyűjteményének részeként látott napvilágot. A játék (a paszinással egyetemben) a Windows 3.1 kiadása óta része az operációs rendszernek, így felhasználók milliói játsszák rendszeresen. A legtöbb számítógépes játékkal ellentétben az aknakereső során a játékosnak sokkal inkább az eszére, semmint ügyességére kell hagyatkoznia. A játék elé egyszerű ahhoz, hogy könnyedén meg lehessen érteni a szabályokat, ugyanakkor elég komplex ahhoz, hogy ne váljon gyorsan unalmassá. A későbbiekben látni fogjuk, hogy a játék még egy számítógép számára is kellően bonyolult, jelenleg nem ismert hatékony algoritmus, amely képes lenne hibátlanul megoldani minden aknakereső-táblát.

A játék szabályai

A játékot (alapvetően) egy személy játssza egy téglalap alakú, cellákra osztott pályán. A játék elején a számítógép előre megadott számú aknát helyez el véletlenszerűen kiválasztott cellákban, a játékos elől elrejtve. A játékos ezután bármelyik cellára rákattinthat. Ha a kiválasztott cellán akna van, a játékos veszített, ellenkező esetben a gép ráírja a cellára, hogy a vele közvetlenül szomszédos 8 cella közül pontosan hányon van akna. A játékos a kapott információk alapján ismét kattint, és ez így megy addig, amíg egyszer vagy aknára nem kattint, vagy fel nem fedi az összes aknamentes mezőt.

A játéknak van egy kétszemélyes változata is, amely leginkább az MSN Messenger szoftverbe épített kliens által ismert. Ebben a változatban ellentétes a cél: a játékosoknak az aknamentes mezők helyett az aknára kell kattintaniuk. Ha valaki eltalál egyet, újra ő jön; ha nem aknára kattint, a másik játékos használhatja fel a kattintás során szerzett új információkat az aknák keresésére. A két játékmód között alapvető különbségek vannak - míg egyjátékos módban csak arra kell ügyelnünk, hogy nem biztos kattintás esetén minél nagyobb valószínűséggel kattintsunk aknamentes mezőre, két játékos esetén azt is figyelembe kell vennünk, hogy minél kevesebb információt szolgáltassunk a másik játékosnak.

A felmerülő kérdések

Ha matematikai szempontból vizsgáljuk a játékot, az alábbi kérdések merülhetnek fel egy adott játékállapottal kapcsolatban:

- Konzisztens-e ez az állapot, azaz van-e olyan aknaelrendezés, ami nem mond ellent a már ismert adatoknak?
- Mely mezők azok, amelyeken biztosan van akna?
- Mely mezők azok, amelyeken biztosan nincs akna?
- Az előző két kategória egyikébe sem tartozó mezők mekkora valószínűséggel tartalmaznak aknát, ha minden lehetséges aknaelrendezést azonos valószínűségűnek tekintünk?

Szóhasználat

A későbbi tárgyalás egyszerűsítése érdekében vezessük be az alábbi kifejezéseket:

- biztonságos mező: olyan mező, amiről biztosan tudjuk, hogy nincs rajta akna
- megjelölt mező: olyan mező, amelyről biztosan tudjuk, hogy van rajta akna
- mező aknaszáma: egy mező aknaszáma azt fejezi ki, hogy hány vele szomszédos mezőn van akna
- felfedett mező: olyan (biztonságos) mező, amelynek ismerjük az aknaszámát
- üres mező: nem biztonságos és nem megjelölt mező
- telített mező: olyan mező, aminek aknaszámával megegyező számú megjelölt szomszédja van

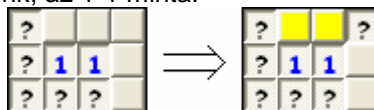
Kézi megoldás mintaillesztéssel

A későbbiekben látni fogjuk, hogy a játék által felvetett problémák még egy számítógép számára is meglehetősen bonyolultak. Van azonban pár stratégia, amelyek alkalmazásával emberi játékosok is meg tudnak oldani különböző állásokat.

Az egyik legegyszerűbb ilyen a "fennmaradó helyek" stratégiája: ha egy mezőnek pontosan annyi üres szomszédja van, mint amekkora az aknaszámának és a vele szomszédos megjelölt mezők számának különbsége, akkor az összes fel nem fedett szomszédján akna van, így azok megjelölt mezőkké válnak. Hasonlóan egyszerű stratégia az, hogy egy telített mező összes üres szomszédja biztonságossá tehető (és felfedhető), mert a telített mezőnek már az összes szomszédos aknáját felderítettük. A harmadik, legegyszerűbb stratégia az, hogy egy 0 aknaszámú mező minden szomszédja biztonságos, így felfedhetjük őket. (Ezt a stratégiát a játék tipikus implementációi automatikusan megvalósítják, ezért lehetséges az, hogy egyetlen kattintással több mezőt is felfedjünk egyszerre.)

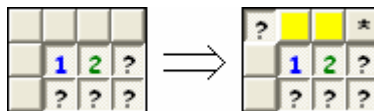
Az előbb vázolt két (illetve három) stratégia igen hasznos, hiszen egy csomó következtetést levonhatunk gondolkodás nélkül. Ráadásul a második stratégia segítségével nagy mennyiségű új mezőt felfedhetünk, amelyekre újra alkalmazható az első stratégia... Ám nagyon hamar belefutunk egy olyan helyzetbe, amikor már nem segítenek ezek a módszerek. Ilyenkor jönnek jól a különböző ismert minták.

Az első minta, amivel megismerkedünk, az 1-1 minta:



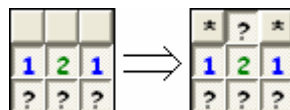
Az bal oldali ábrán kérdőjellel jelölt mezők aknaszáma lényegtelen, a lényeg csak az, hogy biztonságos mezők. (Az is előfordulhat, hogy a tábla szélén találjuk meg a mintát, így a ?-es mezők hiányoznak.) Ha valahol meglátjuk a fenti mintát, levonhatjuk a következtetést, hogy a bal oldali 1-es miatt a sárga háttérű mezők közül pontosan egyen van akna. Bármelyik mezőn is van az akna, az telítetté teszi a jobb oldali 1-est, így a minta alapján levonható a következtetés, hogy a második ábrán (újonnan) kérdőjellel jelölt mező biztonságos. (Sőt, a jobb oldali 1-es összes eddig üres szomszédja is biztonságosnak minősíthető. Azért épp a kérdőjellel jelöltet emeltem ki, mert a mintát legtöbbször felfedett mezők összefüggő tartományainak egyenes határán szoktuk alkalmazni.)

A másik alapminta az 1-2 minta:

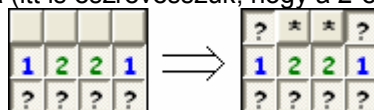


A gondolatmenet hasonló: a két sárgával jelölt mező közül *legfeljebb* az egyikben van akna (az 1-es miatt). Levonható következtetésként, hogy a csillaggal jelölt mezőn biztosan akna van, így azt meg kell jelölnünk. Mivel a 2-es aknaszámú mező szomszédságában kell lennie még egy aknának, kimondhatjuk, hogy a sárga mezőkön *pontosan* egy akna van. Ettől viszont az 1-es mező telítetté válik, így (többek között) a második ábrán kérdőjellel jelölt mezőről megállapíthatjuk, hogy biztonságos.

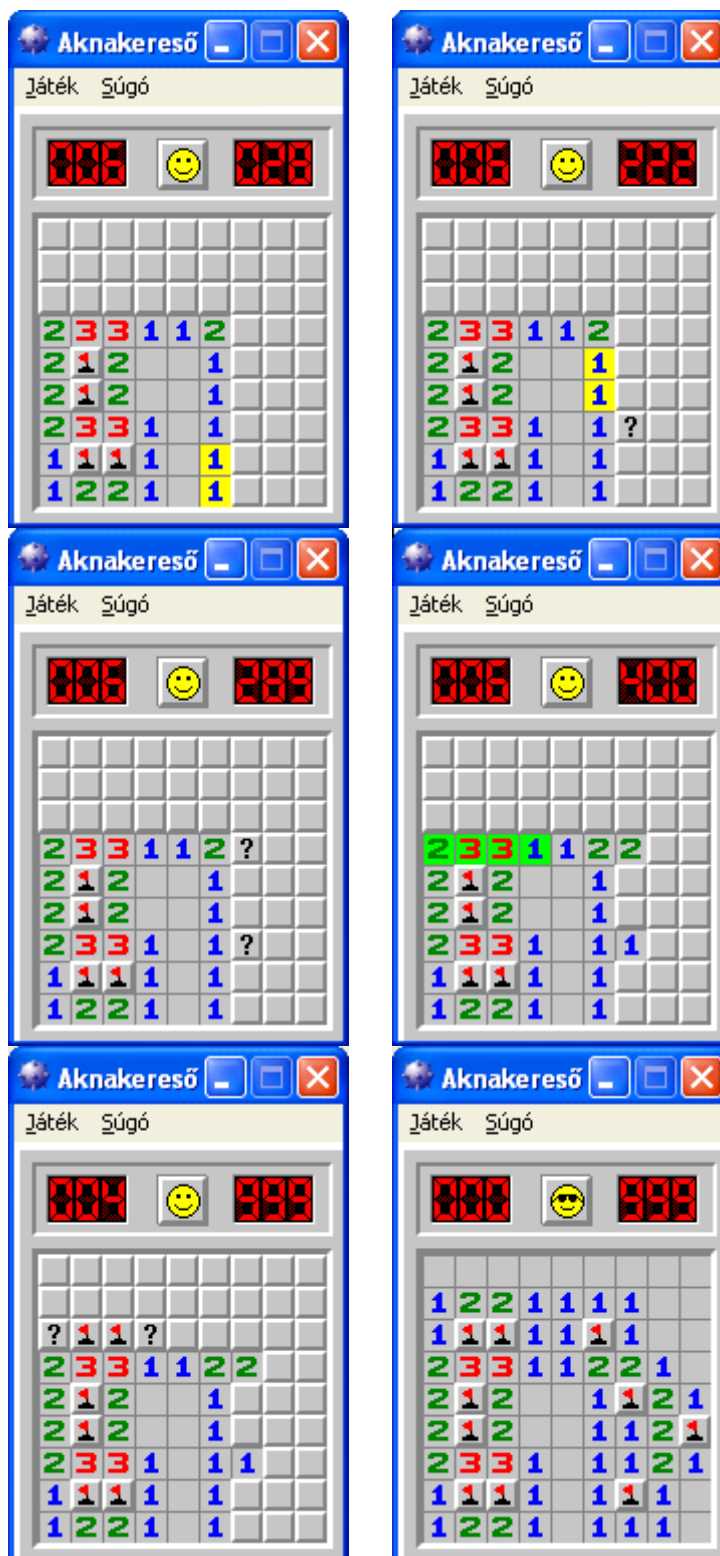
Az 1-2 minta kétszeri alkalmazásával keletkezik az egyik leghíresebb minta, az 1-2-1 minta (a két mintaillesztés után még egy telítésellenőrzést is végzünk a 2-es mezőre):



A másik híres minta az 1-2-2-1 minta (itt is észrevesszük, hogy a 2-es mezők telítettek):



Ezzel a néhány mintával (főleg az utolsó kettővel) már meglehetősen jó eredményeket lehet elérni. Álljon itt egy példa! Néhány kattintás után az alábbiak közül a legelső állapotba jutottam egy akna kereső-játzsma során:



A 1. ábrán sárgával jelölt mezőkön felismertem egy 1-1 mintát, amiből következően a 2. ábrán kérdőjellel jelölt mezőről megállapítottam, hogy biztonságos. Ekkor újabb 1-1 mintát találtam, amiből a 4. ábrán látható következtetést vonhattam le. Első pillantásra itt nem találtam semmit, ezért felfedtem a korábban biztonságosnak minősített mezőket. Ez látható a 4. ábrán. Ekkor vettem észre egy "rejtett" 1-2-2-1 mintát (a 4. ábrán zölddel jelölve): ha a zölddel jelölt mezők aknaszámából levonjuk a már megtalált aknaszomszédok számát, épp egy 1-2-2-1 mintát kapunk, így az 5. ábrán látható következtetéseket vontam le. Ezután felnyitottam a kérdőjeles mezőket, és kizárólag a két alapstratégia (fennmaradó helyek és telítettség vizsgálata) segítségével megoldottam a feladványt.

Ennek a néhány mintának az alkalmazása igen hatékony tud lenni, ám közel sem old meg minden problémát, amivel szembekerülünk. Vannak olyan esetek, amikor egyszerűen nem lehet eldönteni, hol van az akna (ilyenkor nincs mit tenni, tippelnünk kell), és van, amikor ötletesebb, gondolkozósabb módszereket kell alkalmazni. Ezek azonban bonyolultabbak annál, hogy egy ilyen esszében könnyedén leírhatóak legyenek, és sokszor nem is eléggé "megfoghatóak", mert csak a tapasztalt játékosok fejében léteznek.

Az emberi játékosok stratégiáiról térjünk most át arra, mihez kezdhet egy program egy aknakereső-táblával! Ehhez először a számítástudomány egyik fontos fogalmát kell megismernünk.

További stratégiai tanácsok az [1] oldalon találhatóak, ahol azt is megtudjuk, hogyan érdemes tippelni, ha nem egyértelműen megoldható a tábla.

NP-teljesség fogalma

Az NP problémák az eldöntési problémák (olyan problémák, amelyekre egyetlen igen/nem válasz adható) egy igen érdekes és fontos osztályát alkotják. Az ebbe az osztályba tartozó problémák megoldására ugyan jelenleg nem ismert polinomiális idejű algoritmus, ám mindegyikhez létezik "pozitív tanú" algoritmus, azaz olyan algoritmus, amely egy megfelelő "bizonyíték" alapján az "igen" választ ellenőrizni tudja. Példa erre a Hamilton-kör létezésének problémája: jelenleg nem ismert polinomiális idejű algoritmus annak meghatározására, hogy van-e egy gráfban Hamilton-kör, de egy konkrét Hamilton-kört bizonyítéknak használva polinomiális időben ellenőrizhető, hogy az tényleg kielégíti a feltételeket. Igen sok fontos NP-beli probléma van, ezek közé tartozik a már említett Hamilton-kör létezésének problémája, ill. például annak vizsgálata, hogy egy ráfban létezik-e k elemű független ponthalmaz.

Az NP-beli problémák egy kitüntetett csoportját alkotják az ún. NP-teljes problémák. Ezek olyan NP problémák, amelyekre bármelyik másik NP probléma polinomiálisan visszavezethető. Emiatt, ha valakinek valaha sikerül polinomiális idejű algoritmust adni egy NP-teljes probléma megoldására, azzal egy csapásra polinomiális időben megoldhatóvá teszi az összes NP-beli problémát. Az amerikai Clay intézet 2000-ben egy millió dolláros díjat tűzött ki annak, aki megold egy NP-teljes problémát polinomiális időben, vagy bebizonyítja, hogy ez lehetetlen.

Konzisztencia vizsgálata, NP-teljesség bizonyítása

A bizonyítás a [2] cikkben található, ám az nem elérhető az Interneten. A bizonyítás elvét és az áramkört elemek megvalósításának egy részét a [3] és [4] oldalokról szereztem, egy részét (és a NOR-kapu bizonyítását) én dolgoztam ki.

Richard Kaye 2000-ben bebizonyította, hogy egy aknakereső-tábla konzisztenciájának ellenőrzése NP-teljes. A bizonyítás során egy ismert NP-teljes problémát, az ún. SAT problémát (egy logikai formula kielégíthetőségének vizsgálata) vezette vissza az aknakereső-tábla konzisztenciájára. A visszavezetés során a kielégítendő logikai kifejezést először logikai áramkörre alakítjuk, majd az egyes komponenseket (kapuk, vezetékek, vezetékmetszések, kanyarok) megfelelő aknakereső-táblarészletekkel helyettesítjük. Ha az aknakereső-áramkört ki tudjuk tölteni aknákkal úgy, hogy a "kimenetén", azaz egy megjelölt mezőben igaz érték (azaz akna) legyen, és a kitöltés ellentmondásmentes, akkor a logikai formula kielégíthető. Ha viszont az aknakereső-áramkör nem tölthető ki, akkor inkonzisztencia van jelen, tehát nem kielégíthető az eredeti logikai formula.

A bizonyítás alapvető irányvonalai után lássuk az egyes ekvivalenseket!

Az első alkatrész, amire szükségünk van, az egyenes (vízszintes) vezeték:



Látható, hogy a kék pöttyel jelölt "kimeneten" akkor és csak akkor van akna, ha a pirossal jelölt bemeneten akna van.

Részletesebb magyarázattal az alábbi ábra szolgál:



Ha a legbaloldali kérdéses négyzetben akna van, akkor a mellette lévő biztosan nincs, mert különben a pirossal jelölt 1-esnek túl sok akna szomszédja lenne. Hasonlóan, ha nincs ott akna, akkor a szomszédjának lennie kell. Azaz a két mező "akna-állapota" épp ellentétes: ha az egyikben akna van, akkor a másikon nincs, és fordítva. Ezt úgy jelöljük, hogy az egyik mezőbe "A"-t, a másikba "a"-t írunk. Ezt a jelölést a továbbiakban is használni fogjuk: a kis- és nagybetűk mindig ellentétes állapotú mezőket jelölnek, a különböző betűk pedig független mezőket. Az ábrát tovább töltve láthatjuk, hogy a vezeték két végén tényleg ugyanolyan állapotú mezők vannak.

Mivel időnként minden kapcsolási rajzban meg kell törnünk a vezetéket, szükségünk van "kanyar" elemre is, azaz olyan elemre, amihez egy vízszintes és egy függőleges vezeték csatlakoztatható, és a kimenetén ugyanaz az érték jelenik meg, mint a bemenetén. Egy lehetséges megvalósítás:

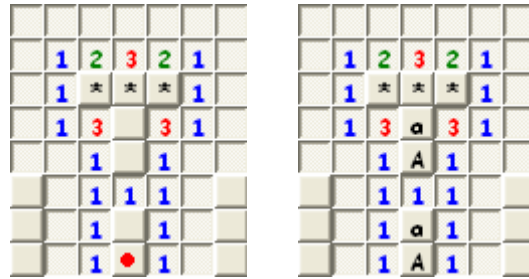


Bejelölve az állapotokat:

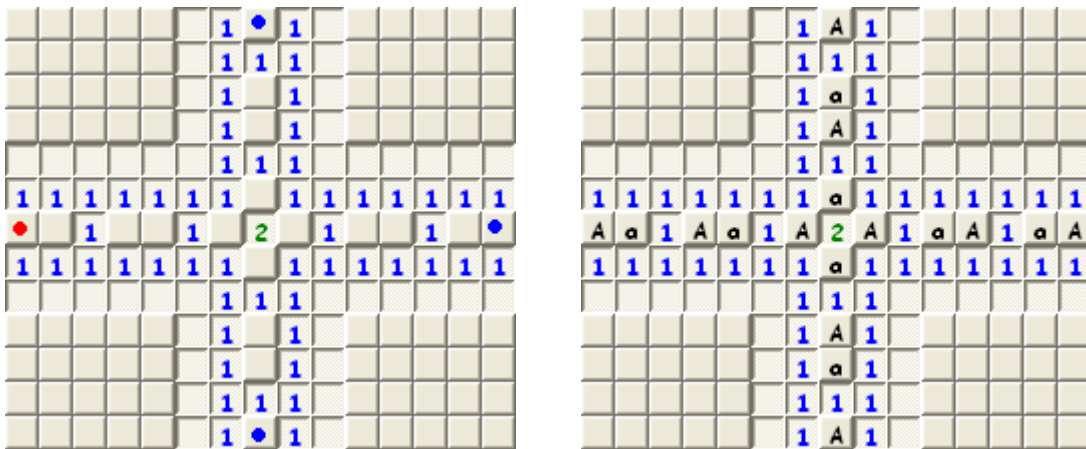


Időnként vannak felesleges vezetékvégek, amiket le kell zárni. Az alábbi összeállítás pirossal jelölt

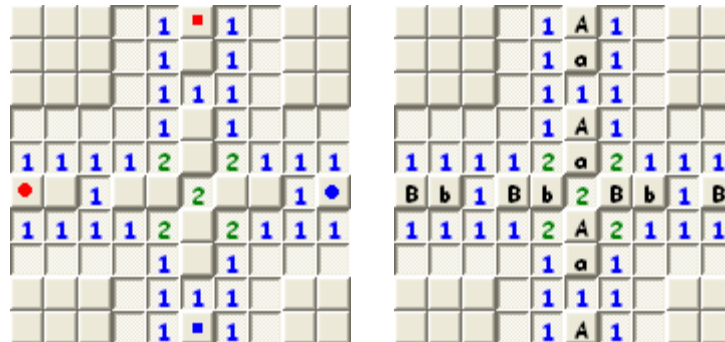
bemenetén mindegy, mi hogy mi van, mindkét esetben érvényes, ellentmondásmentes állapotot találunk:



Szükségünk van arra is, hogy egy jelet két-, vagy akár háromfelé osszunk, erre nyújt lehetőséget a következő "splitter" elem (ha csak kétfelé akarunk ágazni, az előbb bevezetett lezáró elemmel megoldhatjuk a problémát):

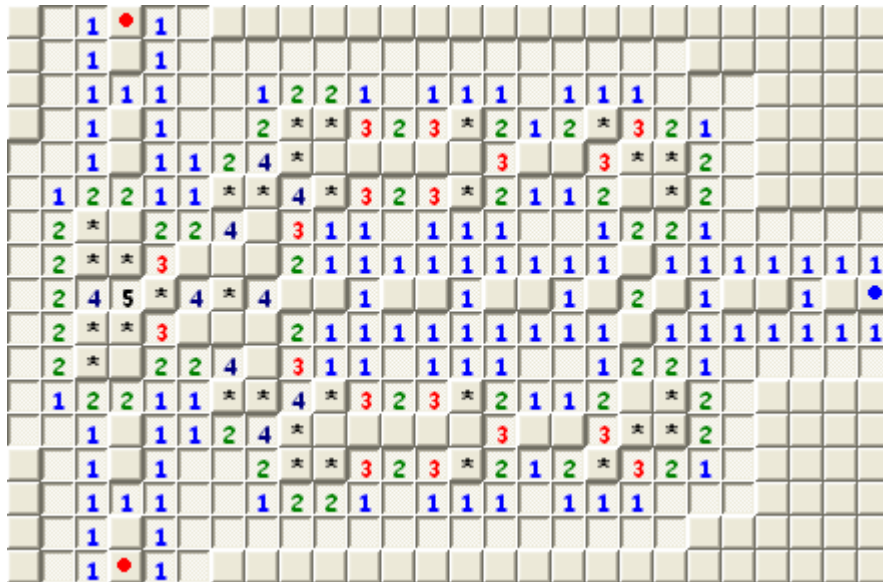


Kapcsolási rajzokon előfordul, hogy két vezeték metszi egymást, ennek megoldására az alábbi táblarészletet használhatjuk fel:

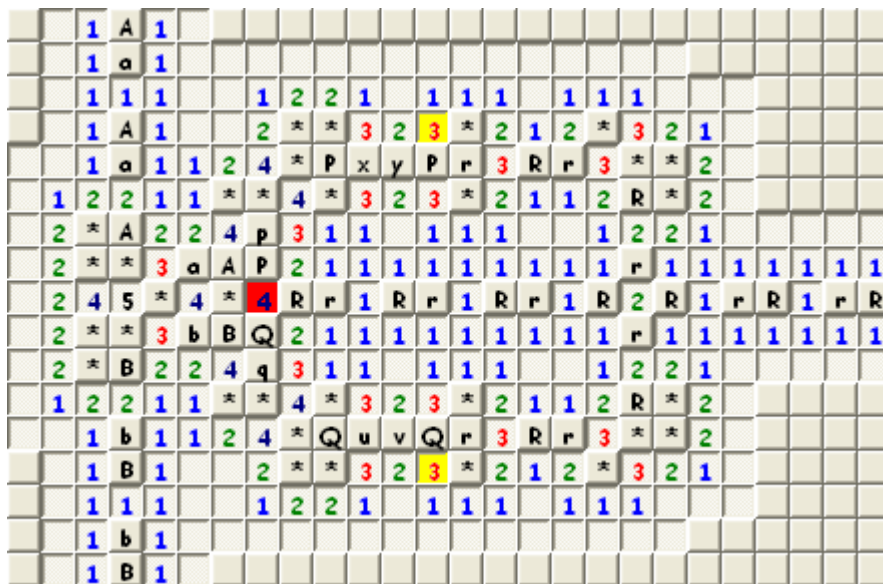


Látható, hogy a piros körrel jelölt bemenet értéke a kék körrel jelölt kimenetre kerül, a piros négyzettel jelölt a kék négyzettel jelöltre, egymástól függetlenül.

Az utolsó szükséges, és egyben legbonyolultabb áramköri elem a NOR (nem-vagy, sem-sem) kapu, ami az alábbi ábrán látható (pirossal jelölve a bemenetek, kékkel a kimenet):



A működés bizonyításához végezzük el az áramkör analízisét (egyelőre pár új ismeretlent bevezetve):



A NOR kapu elvi igazságtáblája:

A	B	A NOR B
hamis	hamis	igaz
hamis	igaz	hamis
igaz	hamis	hamis
igaz	igaz	hamis

Vizsgáljuk meg, milyen következtetéseket vonhatunk le, ha a két bemeneten "hamis" logikai értéket adunk az áramkörre, azaz ha az A-val és B-vel jelölt mezőkön nincs akna, az a-val és b-vel jelöltek pedig van! Ahhoz, hogy a piros háttérrel megjelölt 4-es mellett 4 akna helyezkedjen el, muszáj, hogy a P-vel, Q-val és R-el jelölt mezőkön is legyen akna. Könnyen belátható, hogy ekkor az y-nal és v-vel jelölt mezőkön lesznek még aknák, az u-val és x-szel jelöltek pedig nem lesznek, és a kapott állás ellentmondásmentes. Azt kaptuk tehát, hogy ha a két bemenet egyikén sincs "igaz" jel, akkor a kimeneten "igaz" jelet kapunk, tehát ebben az esetben a kapu jól működik.

Vizsgáljuk meg, milyen következtetést tudunk levonni abból, ha a kimenet "igaz" állapotú, azaz az R jelű mezőkön van, az r jelű mezőkön pedig nincs akna! Ebben az esetben a sárga háttérű 3-asokat megvizsgálva azt kapjuk, hogy a v-vel, Q-val, y-nal és P-vel jelölt mezőkön muszáj aknának lennie. A piros háttérű 4-est vizsgálva láthatjuk, hogy már "megvan" mind a négy aknája, így az A és B jelű mezőkön, azaz a bemeneteken semmiképpen sem lehet akna. Az u és x jelű mezőknek "hamis" (aknátlan) értéket adva egyértelmű, ellentmondásmentes állapotot kaptunk.

Az előző két bekezdés során kiderült, hogy ha a kapu bemenetére két "hamis" értéket kapcsolunk, akkor a kimenete "igaz" lesz, és ha a kimenet "igaz", akkor a bemeneten biztosan két "hamis" érték van. Ennek a jellemzésnek a kétváltozós logikai függvények közül kizárólag a NOR felel meg, így az ábrán látható összeállítás valóban egy NOR kaput valósít meg.

Ismert, hogy NOR kapuk és vezetékek segítségével tetszőleges logikai függvényt megvalósító áramkör felépíthető. Ha a SAT problémában szereplő logikai függvényt megvalósítjuk egy áramkörrel, majd az áramkör egyes komponenseit az aknakereső-ekvivalenseikkel helyettesítjük, az áramkör bemeneteire "záró" áramkört, kimenetére pedig az alábbi ábrán látható "konstans igaz" elemet tesszük, olyan aknakereső-táblát kapunk, amelynek konzisztenciája ekvivalens a logikai formula kielégíthetőségével. Mivel az ekvivalens generálása polinomiális ideig tart, és egy állítólag kielégítő aknaelrendezés ellenőrzése is polinom idejű, a probléma valóban NP-teljes.

	1	1	1	1	1
A	1	a	A	1	
	1	1	1	1	1

Az aknakereső, mint kényszerkielégítési probléma

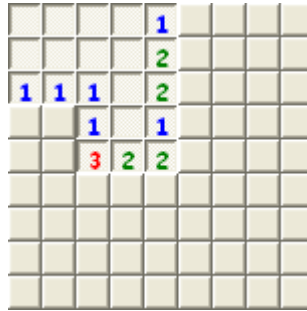
Az aknakereső játék megoldása (azaz annak eldöntése, melyik mezőkön van biztosan akna, és melyikeken nincs biztosan) modellezhető kényszerkielégítési problémaként (CSP, constraint satisfaction problem). Az ilyen problémák leírása során megadunk néhány változót, azok értékkészletét (lehetséges értékeit), valamint a változókra vonatkozó kényszereket. A feladat az, hogy meghatározzuk a változók összes olyan értékét, amelyek kielégítik a kényszereket.

Modellezzük az aknakereső játékot a következőképpen: a tábla i . sorának j . mezőjéhez rendeljük egy x_{ij} változót. A tábla minden mezőjére írunk fel egy kényszert az alábbi módon:

- ha az (i,j) mező biztonságos, vegyük fel az $x_{ij}=0$ kényszert
- ha az (i,j) mező megjelölt, vegyük fel az $x_{ij}=1$ kényszert
- ha az (i,j) mező felfedett, vegyünk fel egy olyan kényszert, ami a vele szomszédos üres és megjelölt mezőkhöz tartozó változók összegére előírja a mező aknaszámának értékét

Ha a kapott CSP-t kielégítő minden megoldásban $x_{ij}=0$, akkor az (i,j) mező biztosan biztonságos. Ha az összes megoldásban $x_{ij}=1$, akkor a mezőn biztosan akna van.

A következőkben bemutatok egy példát, hogy hogyan fogalmazhatunk át egy aknakereső-problémát CSP problémává. Tekintsük a következő aknakereső-táblát:



Felírva a kényszereket (a triviális, $x_{ij}=0$ alakúakat kihagyva):

- (1,5) $x_{16}+x_{26}=1$
- (2,5) $x_{16}+x_{26}+x_{36}=2$
- (3,5) $x_{26}+x_{36}+x_{46}=2$
- (4,5) $x_{36}+x_{46}+x_{56}=1$
- (5,5) $x_{46}+x_{56}+x_{66}+x_{65}+x_{64}=2$
- (5,4) $x_{63}+x_{64}+x_{65}=2$
- (5,3) $x_{42}+x_{52}+x_{62}+x_{63}+x_{64}=3$
- (4,3) $x_{42}+x_{52}=1$
- (3,3) $x_{42}=1$
- (3,2) $x_{41}+x_{42}=1$
- (3,1) $x_{41}+x_{42}=1$

A fenti típusú CSP problémák megoldására található módszereket az [5] cikkben, amely azt is bemutatja, hogyan lehet meghatározni a nem egyértelmű változók értékének valószínűségi eloszlását hatékonyan.

Források:

- [1] <http://www.minesweeper.info/wiki/Strategy>
- [2] Richard Kaye. Minesweeper is NP-complete. The Mathematical Intelligencer, 22(2):9–15, 2000.
- [3] Richard Kaye's Minesweeper page (<http://web.mat.bham.ac.uk/R.W.Kaye/minesw/minesw.htm>)
- [4] Minesweeper is NP-complete (<http://sed.free.fr/complex/mines.html>)
- [5] Raphael Collet: Playing the Minesweeper with Constraints (<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.139.4635&rep=rep1&type=pdf>)