Mesterséges Intelligencia előadás jegyzet

MI alkalmazási területei

1. ÍRÁSFELISERÉS
   – ORC – optikai karakter felismerés
2. BESZÉDFELISMERÉS
   – beszélt szövegből text file
   – a betűk nem azonos módon ejtődnek ki (fonémák jelalakjai különböznek egymástól)
   – hol a szavak között a szünet, és hol a betűk között? (nehezebb a betűfeldolgozás)
3. ÚTVONALKERESÉS
   – Pl.: Szegedről Párizsba – útvonal javaslat (leggyorsabb, legrövidebb, legkevesebb fogyasztás…)
   – adatbázis – gráf → gráf csomópontjai s városok, gráf bejárása
   – keresőeljárás – dijkstra
   – GPS – általános helyzetérzékelő (műholdas)
4. JÁTÉKOK
   – sakk, reverse
   – a programok felveszik-e a versenyt az emberrel?
5. ROBOTIKA
   – autógyártás
   – űrkutatás (definiálatlan körülmények között is működniük kell)
   – Csernobil
   – háztartási robotok
6. WEB-KERESÉS
   – kulcsszavas keresés
   – tematikus keresés (nem csak egy szót talál meg, hanem a szöveg értelme alapján keres)

MI definíciója

Intelligencia:

– értelmi felfogóképesség, ítélőképesség

Mesterséges:

– emberi tevékenységgel, beavatkozással alkotott, előidézett, történő.

MI:

- az a tudományág, ami azzal foglalkozik, hogy hogyan lehet megtanítani a számítógépet emberi képességekre
- eljárásokat kidolgozni ehhez
- mérés: mennyire voltunk eredményesek

- *Definíció 1:* Izgalmas újszerű kísérlet, hogy a számítógépet gondolkodásra késztessük… tudatos gépek, e fogalom teljes és szószerű értelmében (Haugeland, 1985)
- *Definíció 2*: Az emberi gondolkodással asszociálható olyan aktivitások (automati-zálása), mint pl. a döntéshozatal, a problé-mamegoldás, a tanulás (Bellman, 1978)
- *Definíció 3*: Az olyan funkciót teljesítő gépi rendszerek létrehozásának a művészete, amikhez az intelligencia szükséges, ha azt emberek teszik. (Kurzweil, 1990)
- *Definíció 4*: Annak tanulmányozása, hogy hogyan lehet számítógéppel olyan dolgokat művelni, amiben pillanatnyilag az emberek jobbak. (Rich and Knight, 1991)
- *Definíció 5*: A mentális képességek tanlmányozása számítógépes modellek segítségével. (Charniak and McDermott, 1985)
- *Definíció 6*: Az észlelést, a következtetést és a cselekvést biztosító számítási mecha-nizmusok tanulmányozása (Winston, 1992)
- *Definíció 7*: Egy olyan kutatási terület, amely a számítási folyamatok segítségével megkísérli megmagyarázni és emulálni az intelligens viselkedést. (Schalkoff, 1990)
- *Definíció 8*: A számítástudomány egy ága, amely az intelligens viselkedés automatizá-lásával foglalkozik. (Luger and Stubblefield, 1993)

Intelligencia teszt:

- a programunk emberi-e

Turing teszt:

- 2 terem - egyikben kérdező, másikban számítógép, vagy ember
- akkor állta ki a program a Turing tesztet, ha a kérdező nem tudja eldönteni a kérdéseire kapott válaszok alapján, hogy a másik szobában ember, vagy számítógép volt.

Kínai szoba:

- szobában ül egy személy, akinek az ablakon át kínai írásjeleket mutatnak
- egy szabálykönyve van, amiből kikeresi azt az írásjelet, amit lát
- a szabálykönyv alapján ő is felmutat egy jelet
- a látott jel egy kérdés volt, az általa felmutatott jel pedig a válasz
- kérdés: tud-e az illető kínaiul?

Gyenge MI kérdés:

- lehet-e a gépi rendszerek cselekvését úgy alakítani, mintha intelligensek lennének?
- pl.: Turing teszt

Erős MI kérdés:

- van-e a tudatosan cselekvő rendszereknek valódi tudatuk?
- filozófikus kérdéseket vet fel:
    - mi a tudat?
    - mi a gondolkodás?
    - mennyire vagyunk determinisztikusak?
- pl.: Kínai szoba

MI 4 fő osztálya:

- emberi módra gondolkodó rendszerek
  (emberként gondolkodást próbál meg utánozni)
  kognitív tudományok, hogyan működik az emberi agy?
- emberi módra cselekvő rendszerek
  emberként cselekedni: Turing teszt
- racionálisan – ésszerűen gondolkodó rendszerek
  ≠emberként gondolkodni
  ésszerűen gondolkodni: formális következtetési szabályok, logika
- racionálisan – ésszerűen cselekvő rendszerek
  ésszerűen cselekedni: egy adott feladatot a lehető legjobban megoldani (agent)
  (ez lesz a mi megközelítésünk)

Univerzalitás:

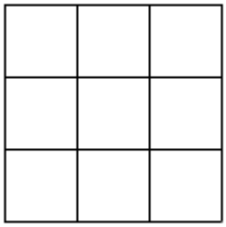
- számítógépekkel a világ megismerése
- 1990-es évek eleje
- amerikai érettségi tárgyakból minden ismeretet → adatbázis
- átlag ismeretanyag a világról

Keresések

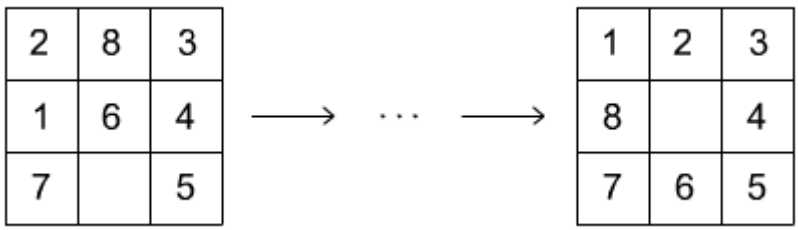
- fontos jellemzők: teljes-e a stratégia, idő- és tárbonyolultság
- állapottér meghatározása → gráffal reprezentáljuk a problémát
- a gráf csúcsai az egyes állások (állapotok)
- kezdőállapotból a végállapotba eljutni a gráfban
- élek az átmenetek az egyes csúcsok (állapotok) között
- a gráf folyamatos megadása → nincs explicit módon megadott gráf
  (csak bizonyos részeit figyeljük a gráfnak, állítjuk elő)

„Nincs ingyen ebéd" tétel: Ha korlátozásokkal oldjuk meg a problémát, és nem teljesülnek a korlátozó feltételek, akkor fizetnünk kell érte.

8 kirakós játék

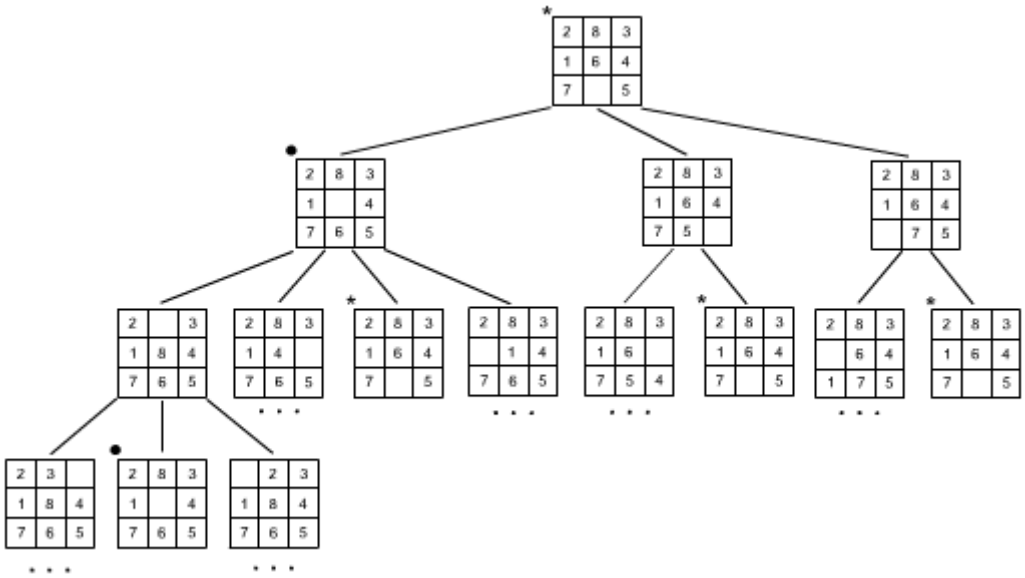


- 9 helyen 8 szám
- 1 üres hely segítségével megfelelő sorrendet kialakítani (tologatni)
- 9! Lehetőség
- n x n-es táblával a probléma exponenciális



kezdőállapot végállapot

- üres helyre egyik szomszédját betolhatjuk – ezt lehet ismételni addig, míg a végállapotig el nem jutunk
- állapottér meghatározása
- gráffal reprezentáljuk a problémát
- a gráf csúcspontjai itt is az egyes állások (9!)
- élek az átmenetek (áttolások)
- ritka gráf, mert a max. szomszédszám: 4
- kezdőállapotból (bemenő adat/állás) → végállapotba (kimenő adat/állás)
- azaz a kezdőpontból a végpontba eljutni a gráfban
- „nincs ingyen ebéd” tétel → ismétléseket figyeljük és ha egy állapot ismétlődik, azt ne vizsgáljuk tovább – ez olcsóbb, de fizetnünk kell érte – ellenőriznünk kell.
  (az ábrán az az azonos állapotokat azonos jelek jelölik)
  (… pedig a további kifejtésre váró állapotok alatt áll)

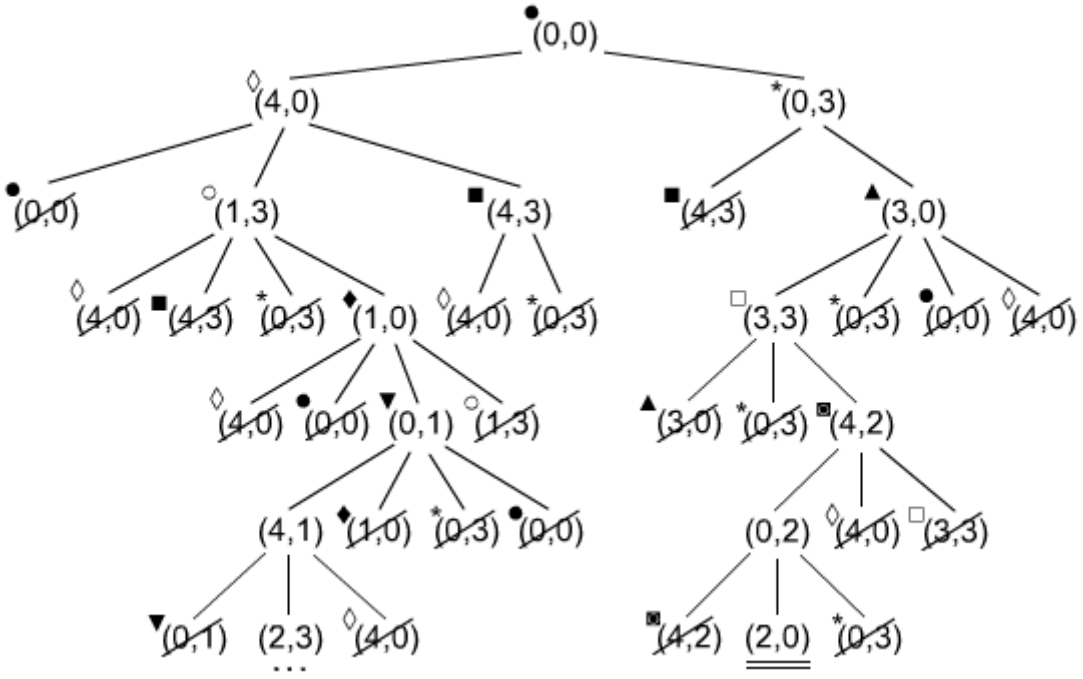


- először a 12, majd a 3, utána a 6, és végül a 9 óránál lévő szomszédot próbálom betolni.
- ha egyik levél sem a végállapot még, akkor haladok tovább
- addig folytatom, amíg meg nem találom a megoldást, a célállapotot

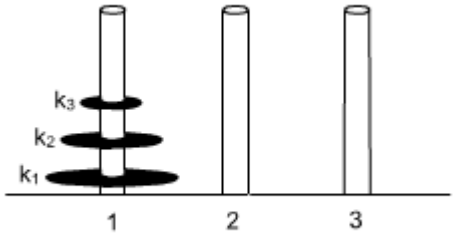
<u>Vízöntős játék</u>

- egy 4 literes, és egy 3 literes korsónk van
- cél: a 4 literesben 2 liter víz legyen
- nincs mérőedényünk

– műveletek, amiket végezhetünk:
  1) kiönteni
  2) teletölteni
– megoldás:
  - a 3 literest megtöltjük
  - áttöltjük a 4 literesbe
  - a 3 literest újra teletöltjük
  - áttöltjük amennyit tudunk a 3 literesből a 4 literesbe (1 liter – a 3 literesben 2 liter marad)
  - a 4 literesből kiöntjük a vizet
  - a 3 literes tartalmát (2 liter) áttöltjük a 4 literesbe
– állapottér:
  rendezett számkettes (x, y),
  ahol x a 4 literes korsó taralma (ezért $0 \leq x \leq 4$), y a 3 literes korsó tartalma ($0 \leq y \leq 3$)
– a feltételek miatt összesen 20 különböző érték lehet, azaz 20 csúcsa van a gráfnak
– (0,0) kezdőállapotból kell eljutni a (2,0) végállapotba
– ismétlődéseket itt is ellenőrizzük
  (az áthúzott állapotokat, már nem vizsgáljuk tovább, mert korábban már előfordultak – hogy hol, azt mutatják az azonos jelölések)

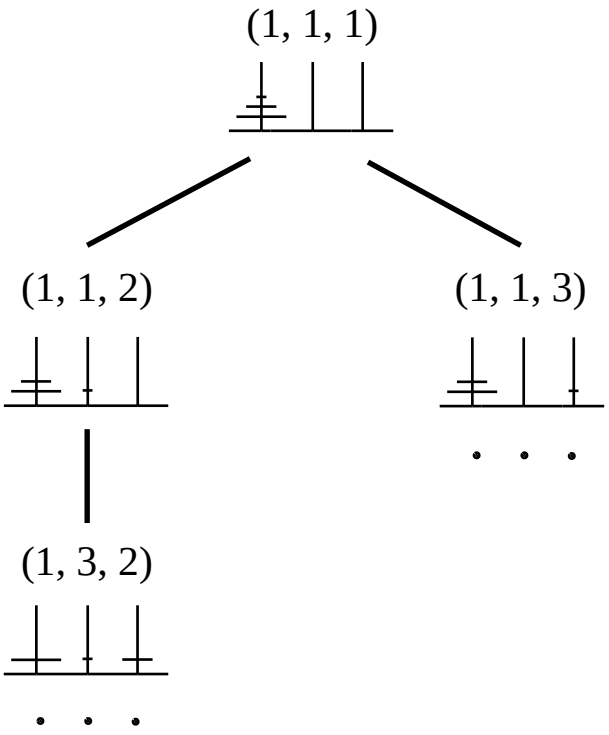


Hanoi tornyai



(, ,)
k_1 k_2 k_3
↑
k_1-es korong (azaz a legnagyobb korong) melyik pálcán van {1, 2, 3}

- k = 3-ra, azaz három korongra
- állapottér reprezentáció: rendezett számhármas
  k_1, k_2, k_3 – a három korong
  1, 2, 3 számok jelölik a pálcákat
- kiindulási állapotból eljutni a végállapotba
  (1,1,1) → (3,3,3)
  (mindhárom korong az első pálcán van → mindhárom korong a harmadik pálcán van)
- 1 lépésben egy korongot lehet csak átrakni
- 1 korongot csak egy nálánál nagyobbra rakhatunk át
- ha egy szám az adott pozíción szerepel, újra előfordulása esetén nem lehet változtatni, csak a legjobboldalibbat, azaz azonos előfordulás esetén a legjobboldalibbat változtatom olyanra, ami tőle jobbra nem szerepel (mert kisebbre nem rakhatom)
  pl. az (1,1,2) állást csak (1,3,2)-re változtathatom (ld. ábra)
- szélességben először keres



ÁLLAPOTTÉR REPREZETÁCIÓ

8 királynő

- 8 x 8-as sakkmező
- 8 királynő
- mind a 8 királynő fenn van a táblán úgy, hogy semelyik kettő nem üti egymást
- *állapottér:* rendezett számnyolcas, ahol a komponensek az oszlopokat jelölik, azaz az adott királynő melyik sorban van
- *kezdőállapot:* (0,0,0,0,0,0,0,0)
- *operátorok:*
  - 0-át cseréljük bármely másik számra {1,2,3,4,5,6,7,8}
  - bármely számot cseréljük bármelyikre (önmaga kivételével)
- 8^8 elemből állna az állapottér, de bele kell venni azokat az eseteket is, amikor egy sorba még nem helyeztünk el királynőt, így 9^8 eset van
- szélességben keresés

  (0, 0, 0, 0, 0, 0, 0, 0) – kezdőállapotból

  (az első királynő elhelyezésének lehetőségei)

  (1, 0, 0, 0, 0, 0, 0, 0)

  (2, 0, 0, 0, 0, 0, 0, 0) első 8 állás

  … ennek hány szomszédja van? (8 x 8 = 64 állás)

  (8, 0, 0, 0, 0, 0, 0, 0)

  (0, 1, 0, 0, 0, 0, 0, 0)

  (0, 2, 0, 0, 0, 0, 0, 0) második 8 állás

  …

  (0, 8, 0, 0, 0, 0, 0, 0)

  … **(**így folytatva a nyolcadik 8 állásig)

- jó állapot = ahol nincs ütés,

  de ez még nem biztos, hogy a jó megoldást adja, mert lehet, hogy még nincs fent minden királynő
- ütő állásokat nem érdemes tovább nézni

– tehát olyan lépést kell keresni, ami megszünteti az ütést
– van lineáris megoldás O(n)

Szám kitalálós

```
  O T
+ O T
-----
T I Z
```

– a betűk helyére írjunk olyan számjegyeket, hogy érvényes összeadást kapjunk!
– *állapottér:* (T, I, Z, O) rendezett számnégyes (a sorrend nem számít)
– *kezdőállapot:* lehetne (0, 0, 0, 0), de ez nem célszerű

T és O nem is lehet nulla

jó kezdőállapot pl.: (T, I, Z, O), ahol

T: 1, …, 9

I: 0, …, 9

Z: 0, …, 9

O: 1, …, 9 (az állapotteret később lehet szűkíteni…)

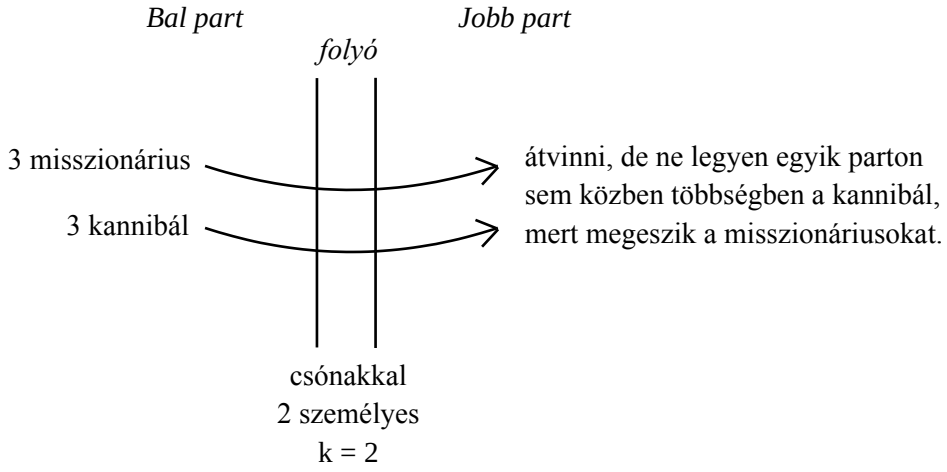
– *operátorok:*
  - betűt cseréljük számra {1,2,3,4,5,6,7,8,9}
  - számot cseréljük egy másik számra (nem önmagára)

(T, I, Z, O) szomszédai:

(1, I, Z, O)	(T, 0, Z, O)	(T, I, 0, O)	(T, I, Z, 1)
(2, I, Z, O)	(T, 1, Z, O)	…	…
(3, I, Z, O)	(T, 2, Z, O)		
…	…		

– jól kell bizonyos eseteket kizárni, nehogy jó megoldás maradjon ki

Kannibál-misszionárius



- n – n – k probléma → hogyan lehet átvinni n db misszionáriust és n db kannibált, egy k személyes csónak segítségével?
- *állapottér:* rendezett számhármas:
  (bal parton hány kannibál áll, bal parton hány misszionárius áll, melyik parton áll a csónak (bal (B) / jobb (J)))

  a probléma folytonos, mert van olyan állapot, amikor megy a csónak
  B → a bal parton kötött ki a csónak, és nincs benne épp senki
  J → a jobb parton kötött ki a csónak, és nincs benne épp senki
  ha k = 3 lenne, akkor már ez az állapottér nem biztos, hogy elegendő, mert akkor még arra is kellene figyelni, hogy útközben ne legyenek többségben a kannibálok, mert megeszik a misszionáriusokat! (k = 2-nél nem áll fent ilyen veszély, mert vagy 2 kannibált visz a csónak, vagy 2 misszionáriust, vagy 1-1-et → egyik esetben sincsenek többségben a kannibálok)
- *kezdőállapot:* (3, 3, B)
- *végállapot:* (0, 0, J)
- *operátorok:*
  - utolsó betűt mindig meg kell változtatni

    B → J

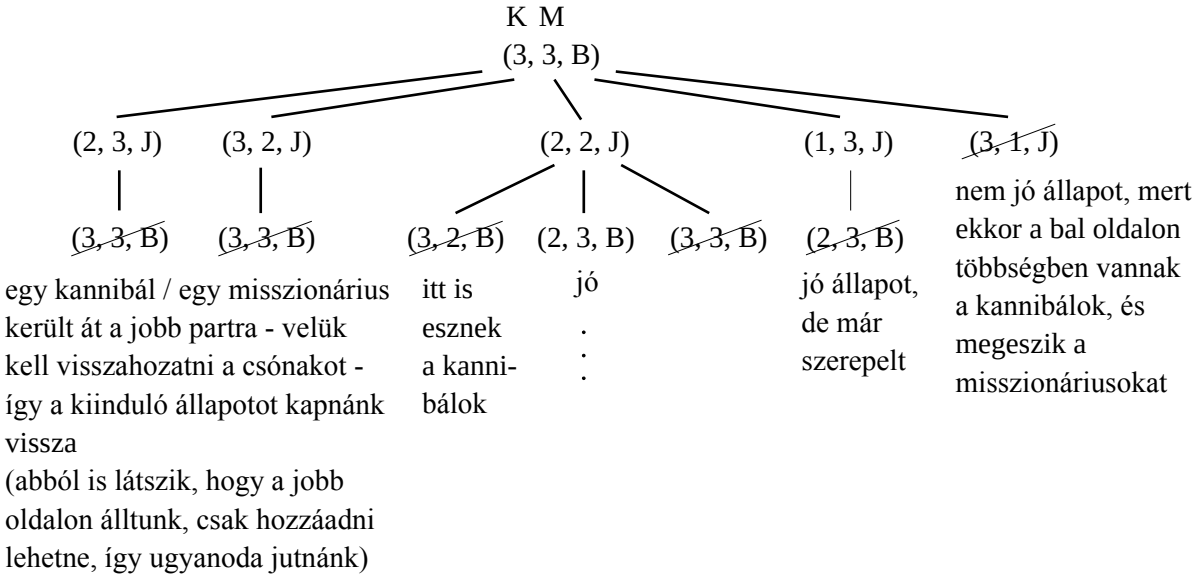
    J → B
  - ha B-ben vagyunk, akkor a következőképp alakíthatjuk a kannibálok, illetve a misszionáriusok számát:

Kannibál	Misszionárius
-1	-1
-2	
	-2

- ha J-ben vagyunk, akkor a következőképp alakíthatjuk a kannibálok, illetve a misszionáriusok számát:

Kannibál	Misszionárius
+1	+1
+2	
	+2

először csak az operátorokat használjuk, később vizsgáljuk a keletkezett állapotokat



Keresési stratégiák

– rendelkezésre álló információ szerint:

- ***informálatlan (vak) keresés:*** nincs információnk a célállapotnak az egyes állapotokhoz képesti elhelyezkedéséről
- ***informált (heurisztikus) keresés:*** valamilyen információval rendelkezünk a célállapotra nézve

– irány szerint:

- ***előre keresés***
- ***hátulról keresés***

(lehet a kettőt kombinálni is: egyszerre h→e és e→h – a megoldás valahol középtájt lesz)

I. SZÉLESSÉGBEN ELŐSZÖR KERES

- legrövidebb utat találja meg
- szintet kell tárolnia
- ezért nagy a memóriaigénye
- a fokszámát felülről tudjuk becsülni az *elágazási faktorral*: hány szomszédja van max.

II. MÉLYSÉGBEN ELŐSZÖR KERES

- pl.: 8 kirakós
- jó, ha messze van a megoldás, és jó irányban haladunk – nincs ingyen ebéd
- spórolós a hellyel
- levág részfákat, ahol nincs jó állapot (későbbi megoldás)

III. ITERATÍVAN MÉLYÜLŐ MÉLYSÉGI KERESÉS

- mélységben csak egy bizonyos határig (lépésszámig) megyünk el *(mélységkorlátozott keresés)* és ha addig nem találtuk meg a célállapotot, akkor korábbi szintekre lépünk vissza → először 1, majd 2, majd 3, … mélységkorláttal végez mélységkorlátozott keresést.
- a megoldás biztos *optimális* lesz, *teljes stratégia*!
- csak 1 utat tárol
- sok a fölös munka (fölösleges utak), de ezek a rövidebb utakon voltak, tehát olcsóbbak
- akkor érdemes használni, ha
  - nagy a fa
  - ha nem tudjuk, milyen mélyen van a célállapot
- szélességben először keres és a mélységben először keres előnyeit ötvözi

GYORSÍTÁSI TECHNIKA – HEURISZTIKUS KERESÉSEK / VAGY GRÁFOK

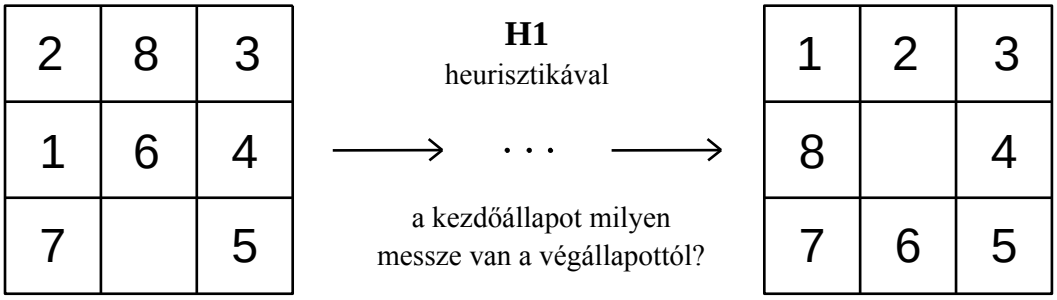
- legyen információnk arról, hogy milyen messze vagyunk a céltól

- a távolságot az operátoraink függvényében adjuk meg
- a megoldásban a célfüggvény értéke 0 lesz, azaz a célfüggvényt minimalizálnunk kell!
- segít abban, hogy merre van a jó irány
- ne lépjünk vissza, hanem közelítsünk a jó megoldáshoz valamilyen függvénnyel
- ezt a jó függvényt keressük
- ha van arra vonatkozó információnk, hogy egy-egy állapot ilyen a megoldáshoz képest, és ezt az információt használjuk is, akkor ***heurisztikáról*** beszélünk
- ezt az információt valamilyen értékelő (jóságra jellemző) függvény segítségével definiáljuk és n állapotban ***h(n)***-nel jelöljük
- pl.: **<u>8 kirakós</u>**

  H1 (egyik lehetséges heurisztika**)**:
  - adjuk össze egy állapotnál, hogy hány darab elem nincs még a helyén, amit oda kellene tolni

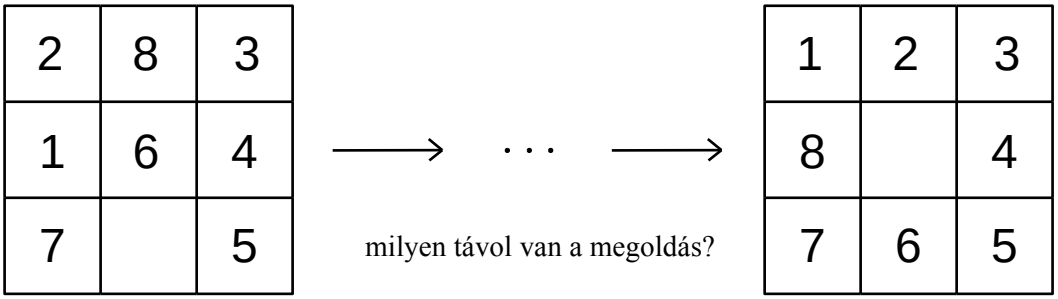
  H2 (másik heurisztika):
  - egy adott állapotnál megvizsgáljuk minden elemre, hogy hány lépésre van a helyétől, majd ezeket összeadjuk – így kapjuk az adott állapot heurisztikáját
  - tehát definiáljuk a távolságot – a célállapotnak megfelelő helyéhez viszonyítva
  - ha messze van, több lépés kell, hogy a helyére toljuk (majd ezeket adjuk össze minden elemre egy adott állapoton belül – ld. példa)

  ha pl. 3 az adott állapot heurisztikája, még egyáltalán nem biztos, hogy ténylegesen 3 áttolással eljutunk a célállapotba, mert ha egy elemet a helyére is rakunk, lehet, hogy közben más elemeket elrontunk



a kezdőállapot heurisztikája **H1** alapján **4**, mert 4 elem nincs a helyén (2, 8, 1, 6)
(látható viszont, hogy 4 lépés biztosan nem lesz elég)

H2
heurisztikával



az 1-es egyre van (azaz ebben az állásban egy lépésre van a végső helyétől)
a 2-es szintén egyre van
a 3-as nullára, azaz már a helyén van
a 4-es is nullára
az 5-ös is nullára
a 6-os egyre
a 7-es is nullára, azaz már ő is a helyén van
a 8-as pedig kettő lépésre van a végső helyétől
ha ezeket az értékeket összeadjuk, akkor 1 + 1 + 1 + 2 = **5**

2	8	3
1		4
7	6	5

ha a 6-ot letoljuk, akkor csak **4** lenne

ha a 7-et, vagy az 5-öt tolnánk be, akkor eggyel nőlne, **6** lenne

2		3
1	8	4
7	6	5

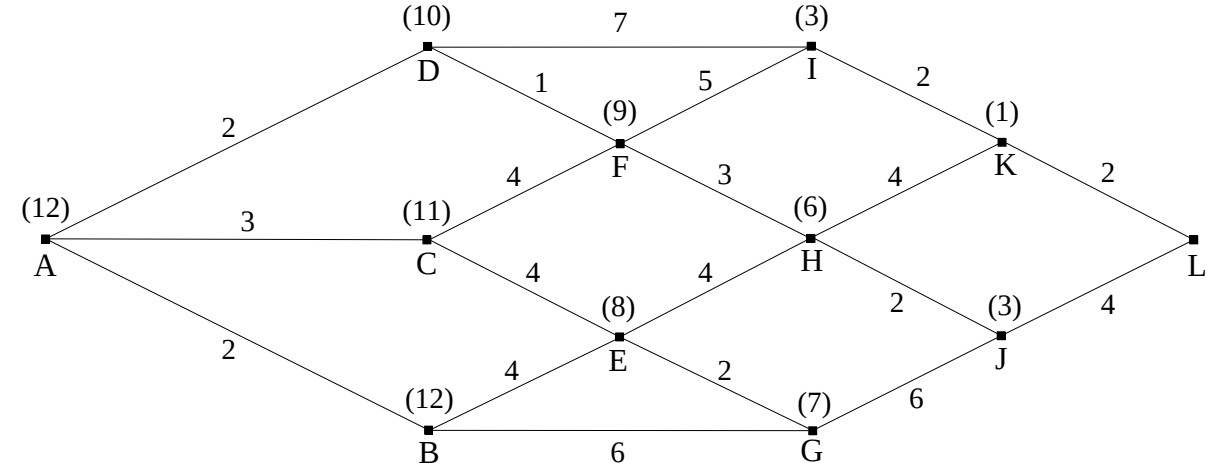
következő lépésként a 8-at lenne a legjobb letolni, mert akkor **3** lenne

míg ha az 1-et, vagy a 4-et tolnánk be, az növekedéssel járna

- milyen jó függvényt tudnánk definiálni a
  - 8 királynő problémára,
  - a vízöntős példára,
  - a kannibál-misszionárius problémára,
  - illetve útvonaltervezésnél?

- **<u>útvonaltervezés</u>**

(a rajz nem arányos)



- adott egy gráf - a gráf csúcsai a városok, a gráf élei pedig a városokat összekötő utak (súlyozottak, az út hosszának megfelelően)
- hogyan lehet eljutni A-ból L-be a legkisebb költséggel (vagyis a legrövidebb úton)
- az optimális megoldást keressük tehát
- a csúcsok fölött zárójelben az adott csúcs heurisztika értéke található, azaz egy becslés, hogy milyen messze van a cél az adott ponthoz képest

f(n) – (*start*-ból az *n*-en keresztül)
n-en áthaladó megoldások optimuma

g(n) – a *start*-ból *n*-be vezető út optimuma

h(n) – *n*-től a *cél*-ig vezető út

optimális út: ***f(n)*** = *g(n) + h(n)*

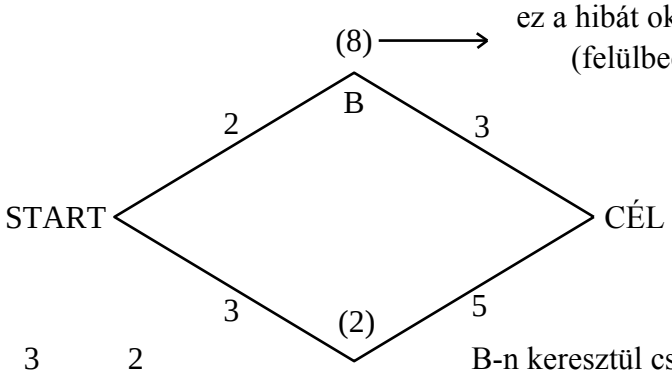
h'(n) – *h(n)* egy becslése

g'(n) – *g(n)* egy becslése

f'(n) = *g'(n) + h'(n)*

f'(n) – az *f(n)* egy heurisztikával kapott becslése

A* eljárás



$f'(A) = \overset{3}{g'(A)} + \overset{2}{h'(A)} = 5$

$f'(B) = \overset{2}{g'(B)} + \overset{8}{h'(B)} = 10$

B-n keresztül csak 5 lett volna a teljes költség, és mi az A pontot választottuk, mert arra probálunk tovább menni, ahol a becslésünk kisebb (A-nál a becslésünk csak 5 volt, míg B-nél 10)

vagyis nem volt jó a heurisztikus függvényünk!
ezért kell egy kikötés a heurisztikus függvényre!

$h' \leq h$ ténylegesen optimális legyen megengedhető

(a példában ez nem teljesült, mert $h' = 8$ és $h = 5$ volt, így nem is adott optimális megoldást)

kannibál - misszionárius példa

H1: a kiindulási oldalon levő személyek száma

jó-e ez a heurisztika???

pl.: (1, 1, 2)

a példán látható, hogy nem jó, mert a H1 heurisztika szerint 1 + 1 = 2 lépés kellene, holott 1 lépéssel meg lehet oldani.

H2: a kiindulási oldalon lévő személyek száma, osztva a csónak férőhelyével

jó heurisztika!!!

$$\frac{K_j + M_j}{CS}$$

állapotok halmaza:

NYITOTT (NY) – ahova már eljutottunk

ZÁRT (Z) – a nyitottak közül azt tesszük bele, amelyikről már azt feltételezzük, hogy megtaláltuk hozzá az optimális utat

A*:

$h' \leq h$ (h az optimális érték)

1) *START* → NY *h'(START)* értékkel
2) ha NY üres, hiba van! (nincs megoldás)
3) NY-beli elemekből válasszuk ki azt az *n* csúcspontot, amelyre *f'(n)* minimális!
   ha *n* = *CÉL*, akkor álljunk le, és adjuk vissza a pointerek segítségével a *START*-ból az *n*-be vezető utat!
4) rakjuk be az *n*-et a Z-be!
   vegyük *n* összes fiát, és csináljuk mindegyikkel a következőt:
   (*n'* az egyik fiú, *n* az apa)
   - *f'(n') = g'(n) + c(n,n') + h'(n')*
   - *n*-re mutató pointer
     a) *n'* ∉ NY és *n'* ∉ Z , ekkor *n'* → NY
     b) *n'* ∈ NY az eddigi *f'(n')*-knél jobb-e az új?
     ha igen, akkor az értékeket és a mutatókat kicserélem
     c) *n'* ∈ Z jobb az eddigieknél?
     ha igen, akkor *n'* → NY (kivesszük a ZÁRT-ból és áttesszük a NYITOTT-ba)
     kicserélem az értékeket és a pointereket
5) vissza a 2)-es ponthoz!

1. példa

az **útvonaltervezés** gráfja alapján

	NY	**Z**
	~~A(12)~~	A(12)
A gyerekei →	B optimuma 14, mert 2 + 12 *g'(n) + c(n,n') + h'(n')* 0+2+12 B(14) C(14) ~~D(12)~~ (D kerül át, mert ő a legkisebb)	D (12)
D gyerekei →	~~F(12)~~ ~~I(12)~~ ~~11~~ (F kerül át, mert az NY-beli elemek közül ő a legkisebb)	F(12)
F gyerekei →	~~I(11)~~ ~~H(12)~~ Eddig mindig az *a)* eset állt fent, most a *b)*, mert az I már benne van az NY-ben, de jobb utat találtunk! Átírjuk az I(12)-t I(11)-re!	I(11)

(folyt.)	NY	Z
I gyerekei →	~~K(11)~~	K(11)
K gyerekei →	~~L(12)~~	H(12)
H gyerekei →	~~J(11)~~ illetve szerepelne még a K(11) is, de az már benne van a Z-ben pont ilyen értékkel, azaz nem jobb az út, amit találtunk hozzá	J(11)
J gyerekei →	L(12), de ez már szerepelt az NY-ben (K gyerekeként), pont ilyen értékkel, azaz nem jobb az út, amit találtunk hozzá	L(12) (vége, mert a cél került bele a Z-be)

2. példa

kannibál-misszionárius (20, 20, 10), ahol (kannibál, misszionárius, csónak)

$$\frac{K_J + M_J}{CS}$$

korábban láttuk, hogy ez jó heurisztika

(ezen képlet alapján számítjuk az értékeket)

(így a kezdőállapot értéke (20 + 20)/10 = 4)

vegyük a gráf éleinek költségét mindenhol 1-nek!

	NY	Z
	~~(20, 20, J)(4)~~	(20, 20, J)(4)
(20, 20, J) gyerekei→	~~(15, 15, B)(4)~~ 4, mert 3 a heur. + 1 az út költsége a startból ebbe az állapotba (16, 16, B)(4,2) (17, 17, B)(4,4) (18, 18, B)(4,6) (19, 19, B)(4,8) a fenti állapotoknál ugyanannyi kannibált vittünk át, mint misszionáriust (összesen max. 10-et) ~~(10, 20, B)(4)~~ ~~(11, 20, B)(4,1)~~ (12, 20, B)(4,2)…(19, 20, B)(4,9) ezek azok az állapotok, amikor csak kannibált vittünk át (több eset nincs, mert azt nem engedhetjük meg, hogy csak misszionáriust vigyünk át, vagy ne egyezzen a m. és a k. száma, mert azon esetekben többségben lesznek az egyik parton a kannibálok)	(15, 15, B)(4)
(15, 15, B) gyerekei→	(16, 16, J)(5,2) 5,2 mert 3,2 a heur. + 2 az út költsége a startból ebbe az állapotba (17, 17, J)(5,4) (18, 18, J)(5,6) (19, 19, J)(5,8)	(10, 20, B)(4)

	(15, 20, J)(5,5) (20, 20, J) ← de itt már voltunk	
(10, 20, B) gyerekei→	(11, 20 J)(5,1) (12, 20, J)(5,2) … … (15, 20, J) ← már volt (16, 20, J)(5,6) … (19, 20, J)(5,9) (20, 20, J) ← már volt	(11, 20, B)(4,1)
(11, 20, B) gyerekei→	…	

- ez azért nem jó, mert az összes (4,…)-et végig kellene nézni… szinte szélességben keresés!!! → figyelembe kellene venni, hogy melyik oldalon van a csónak!
  →heurisztikát módosítani!
- a heurisztikát egy állapothoz rendeljük hozzá, így azt lehet mondani, hogy

  (H1): …, ahol (_ , _ , J)

  (H2): …, ahol (_ , _ , B)
- ha a csónak a jobb oldalon van, akkor az eddigiek alapján számított heurisztikából vonjunk le egyet, azaz ha

  (a, b, J) akkor (H1) : $\frac{a+b}{cs} - 1$

  (a, b, B) akkor (H2) : $\frac{a+b}{cs}$

  így elérjük azt, hogy nem széltében először keres
  (bar még ez sem teljesen mélységi, de a réginél jobb)

- ha a **heur. függvény ≡ 0** lenne , akkor minden *h'(...)* = 0 lenne (DIJKSTRA)
  ekkor az **útvonaltervezés** gráfján:

	NY	Z
A gyerekei →	~~B(2)~~ ~~C(3)~~ ~~D(2)~~	B(2)
B gyerekei →	E(6) G(8)	D(2)
D gyerekei →	I(9) F(3)	C(3)
C gyerekei →	F(7) … de D-n át már volt egy jobb utunk, ezért ezt elvetjük!	…

de a DIJKSTRA-nál nincs vissza út
itt viszont a ZÁRT-ból vissza lehet hozni a NYITOTT-ba

- ha $\boldsymbol{h' = h}$, akkor az optimumot keresnénk
  az optimális úton lenne a legkisebb szám → nem lenne mellékvágány

Tétel: Ha van a startból a célba vezető út, akkor az A* algoritmus optimális megoldást talál.
($h' \leq h$ megengedhető heurisztika az A* előfeltétele)

Megjegyzés: Ha létezik a startból a célba vezető út, akkor létezik optimális út.
(az élek pozitívak, a kezdő és a célcsúcs között van út, összefüggő a gráf → van optimális megoldás)

Bizonyítás:

Lemma: Az előző feltételek mellett az A* algoritmus befejezése előtt mindig lesz olyan n csúcspont az NY listán, amelyre

1) n optimális úton van,
2) az A* algoritmus már talált egy optimális utat n-hez,
3) $f'(n) \leq f(\text{START})$ teljesül.
   $f'(n)$ – az n-en keresztül haladó utak közül az eddig talált legjobb
   $f(\text{START})$ – a START-ból kiinduló legrövidebb út → optimális út

Lemma bizonyítása:

Legyen n_0 (=START), n_1, ..., n_k (=CÉL) egy optimális út.

- Ha START-ot még nem terjesztettük ki, akkor a START $\in$ NY. A megengedhetőség miatt teljesül rá mindhárom tulajdonság.
- Ha kiterjesztettük, akkor n_1 az NYITOTT-ba került ($n_1 \in$ NY).
- Ha a vizsgált időpontban n_1-et még nem terjesztettük ki, akkor rá:
  $f'(n_1) = g'(n_1) + h'(n_1) = g(n_1) + h'(n_1) \leq f(\text{START})$,
  mivel n_1 az optimális úton van, és a heurisztika megengedett.
- Ha n_1-et kiterjesztettük, akkor n_2 került az NYITOTT-ba ($n_2 \in$ NY), és őrá megismételhetjük az előző gondolatmenetet…

Mivel még nem találtuk meg CÉL-t, ezért az n_1, n_2 ,..., n_{k-1} útrészletnek mindig kell lennie olyan elemének, amely az NYITOTT-ban van, és amelyre teljesül a három állítás.

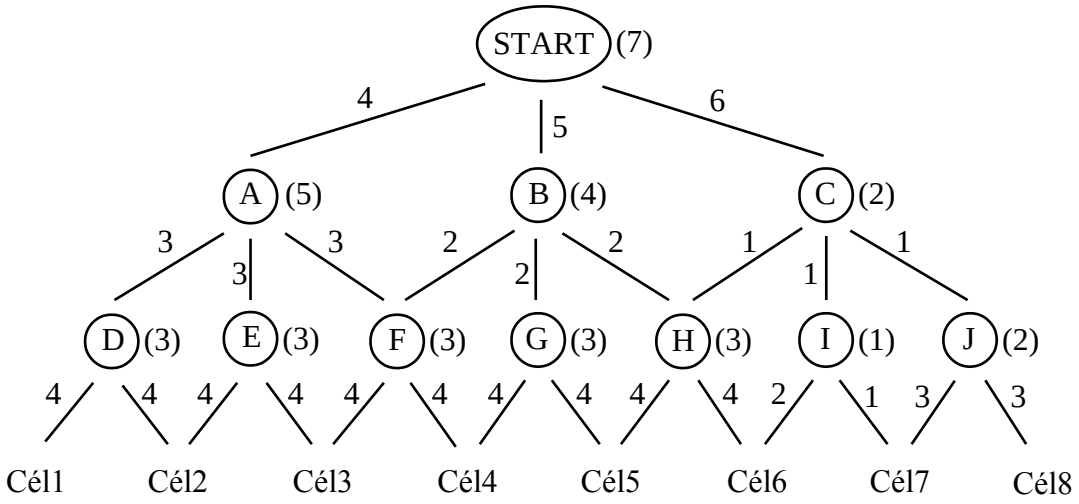
Tétel bizonyítása: (két további állítást látunk be)

- Az A* algoritmus véges lépésben megáll.
  Ha ez nem így lenne, akkor az élek alulról korlátozottsága és a véges elágazási faktor miatt ellentmondásba kerülünk a Lemmával.

- Az A* algoritmus úgy ér véget, hogy megtalálja az optimális megoldást.
  Ha nem az optimumot találná meg, akkor f'(CÉL) > f(START)
  A lemmát alkalmazva közvetlenül a befejezés előtt van olyan olyan n, amelyre $f'(n) \leq f$(START), viszont akkor ezt kellett volna választanunk a CÉL helyett.
  (azaz megtalálnánk a célt, de nem az optimális úton → a lemma 3) pontja miatt nem lehetséges)

Következmény: az A* algoritmus által ki-terjesztett n' csomópontra $f'(n')$ nem nagyobb, mint az optimális megoldás költsége.

Milyen garancia egy jobb heurisztika?
ha nincs heru. → DIJKSTRA → sok csomópont
jó heur. → kevesebb csomópont → gyorsabb



H1: ≡ 0 heurisztika (csupa 0 heur.)

	NY	Z
	~~START(0)~~	START(0)
START gyerekei →	~~A(4)~~ ~~B(5)~~ ~~C(6)~~	A(4)
A gyerekei →	~~D(7)~~ E(7) F(7)	B(5)
B gyerekei →	G(7) H(7)	C(6)
C gyerekei →	I(7) J(7)	D(7)
D gyerekei →	Cél1(11) Cél2(11)	…

Akkor ér véget, ha a valamelyik cél állapot átkerül a ZÁRT-ba, de addig az összes (7)-es állapotot ki kellene fejteni! ☹
→ ha nincs heurisztikánk, akkor a teljes gráfot ki kell fejteni! ☹

H2: ábrán látható értékek

	NY	Z
	~~START(7)~~	START(7)
START gyerekei →	A(9) B(9) ~~C(8)~~	C(8)
C gyerekei →	H(10) ~~I(8)~~ J(9)	I(8)
J gyerekei →	Cél6(9) ~~Cél7(8)~~	Cél7(8)

Jobban informáltság (nagyobb tudás)

Definíció: h_1 heurisztika ***jobban informált*** h_2-nél, ha minden n csúcsra $h_1(n) > h_2(n)$.

(n nem célcsúcs, mert a célcsúcs heurisztikus értéke 0!)

Tétel: Ha h_1 jobban informált, mint h_2, akkor h_2 kiterjeszti a h_1 által kiterjesztett csúcsokat.

Biz.: (teljes indukcióval, a START-ból induló utak hossza szerint)

0 hosszúságra nyílván áll.

Tfh. minden k hosszú útra teljesül

belátjuk, hogy akkor k+1 hosszúakra is

Ha nem így lenne, akkor az utak végén lenne egy olyan n csúcs, amit h_1 kiterjeszt, de h_2 nem.

- Mivel h_2 kiterjesztette n szülőjét, ezért a befejezéskor $n \in$ NY, és mivel n-et nem választottuk ki,

$$f'_2(n) \geq f(\text{START})$$
$$g'_2(n) + h'_2(n) \geq f(\text{START}) \qquad /- g'_2(n)$$
$$h'_2(n) \geq f(\text{START}) - g'_2(n)$$

- Mivel h_1 kiterjeszti n-et,

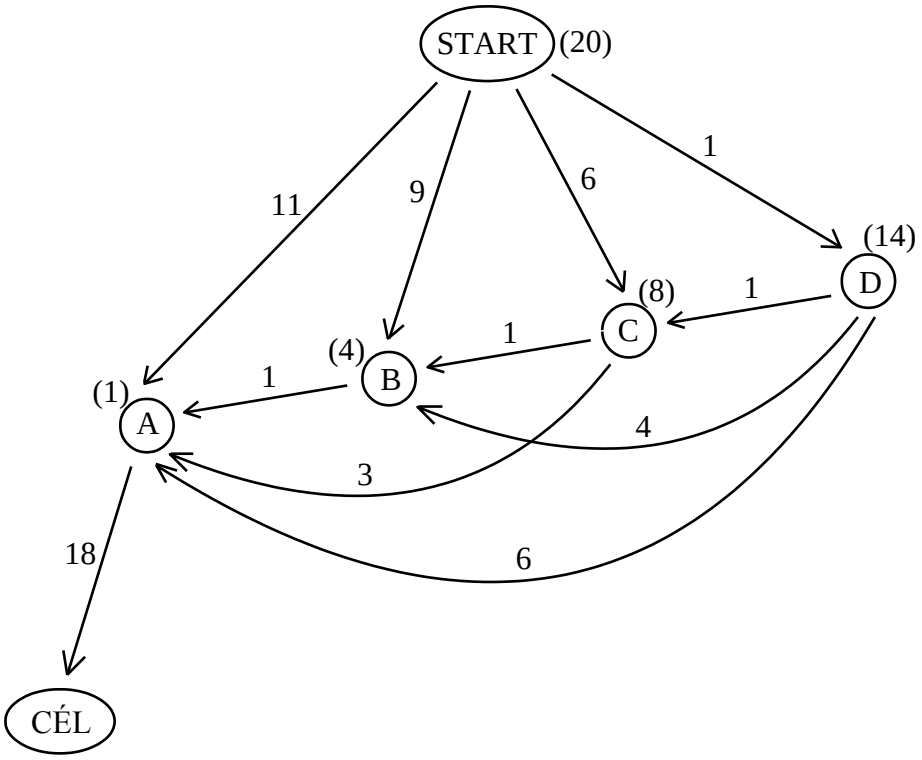
$$f'_1(n) \leq f(\text{START})$$
$$g'_1(n) + h'_1(n) \leq f(\text{START}) \qquad /- g'_1(n)$$
$$h'_1(n) \leq f(\text{START}) - g'_1(n)$$

Mivel bármely, az h_1 által kiterjesztett k hosszúságú úton levő csúcsot az h_2 is kiterjeszt, $g'_2(n) \leq g'_1(n)$, azaz $h'_1(n) \leq f(\text{START}) - g'_1(n) \leq f(\text{START}) - g'_2(n) \leq h'_2(n)$,

ami ellentmond a jobban informáltságra vonatkozó feltételünknek.

Példa (8 kirakós):

A manhattan (h_1) nem jobban informált, mint a helyén nem levők száma (h_2), mert vannak olyan csúcsok, ahol $h_1(n) = h_2(n)$, vagyis nem áll fenn az egyenlőtlenség ($h_1(n) > h_2(n)$)



	NY	Z
	~~START(20)~~	START(20)
START gyerekei →	~~A(12)~~ ~~B(13)~~ ~~C(14)~~ ~~D(15)~~	~~A(12)~~
A gyerekei →	Cél(~~29~~) (~~28~~) (~~27~~) (26)	~~B(13)~~
B gyerekei →	~~A (11)~~ A(12) már szerepelt a ZÁRT-ban, de jobb utat találtunk hozzá → vissza a NYITOTT-ba (*c)* pont miatt)	~~A(11)~~
A gyerekei →	~~Cél(28)~~ Már szerepelt a NYITOTT-ban (29) értékkel → átírom (28)-ra.	~~C(14)~~
C gyerekei →	~~B(11)~~ ~~A(10)~~ Mind a B, mind az A esetében úgy járunk el, mint az előbb…	~~A(10)~~
A gyerekei →	~~Cél(27)~~	~~B(11)~~
B gyerekei →	~~A(9)~~	~~A(9)~~
A gyerekei →	~~Cél(26)~~	D(15)
D gyerekei →	C(10) B(9) A(8) (C-t, B-t és A-t is visszahoz-tuk a ZÁRT-ból a NYITOTT-ba, mert jobb utat találtunk…)	…

Ugyanolyan monoton növekvőt kapunk, mint az elején, csak 4-gyel kisebb értékkel! → sok lépés kellene! → 2^n (azaz exponenciális) → újabb csomópontnál duplázódik!

Definíció: n – apa, n' – gyerek, ekkor a ***heurisztika monoton***, ha teljesül, hogy

$$h'(n) \leq h'(n) + c(n\ ,\ n')$$

Pl.: az előző gráfban A gyereke B-nek

$4 \leq 1 + 1$ (nem teljesül) → nem monoton

Tétel: Ha h monoton, akkor bármely (n) csúcs kiterjesztésekor az A* algoritmus már megtalálta az optimális utat n-ig. (Azaz egy csúcs se kerül be a ZÁRT halmazba többször! Semmit nem viszünk vissza a ZÁRT-ból a NYITOTT-ba!)

Biz.: Tfh. n kiterjesztésekor ez nem teljesül.

Jelölje az n-ig vezető optimális utat a $P = (n_0\,(=\text{START}), n_1, ..., n_k\,(=n))$ csúcssorozat.

Legyen n_l a P út utolsó olyan csúcsa, amely ZÁRT-ban van (ekkor már $n_{l+1} \in$ NY).

A monotonitási kényszer miatt P összes csúcsára:

$$g(n_i) + h'(n_i) \leq g(n_i) + h'(n_{i+1}) + c(n_i\ ,\ n_{i+1}).$$

Mivel n_i és n_{i+1} eleme az optimális útnak,

$$g(n_{i+1}) = g(n_i) + c(n_i\ ,\ n_{i+1}), \text{ ezért}$$

$$g(n_i) + h'(n_i) \leq g(n_{i+1}) + h'(n_{i+1}).$$

Mivel ez P bármely két szomszédos csúcsára teljesül, ezt használva l+1-től k-1-ig:

$$g(n_{l+1}) + h'(n_{l+1}) \leq g(n_k) + h'(n_k), \text{ azaz}$$

$$g(n_{l+1}) + h'(n_{l+1}) \leq g(n) + h'(n).$$

Másrészt, mivel az A* algoritmus az n-et választja az NYITOTT-ból,

$$g'(n) + h'(n) \leq f'(n_{l+1})\ (= g(n_{l+1}) + h'(n_{l+1})) \leq g(n) + h'(n).$$

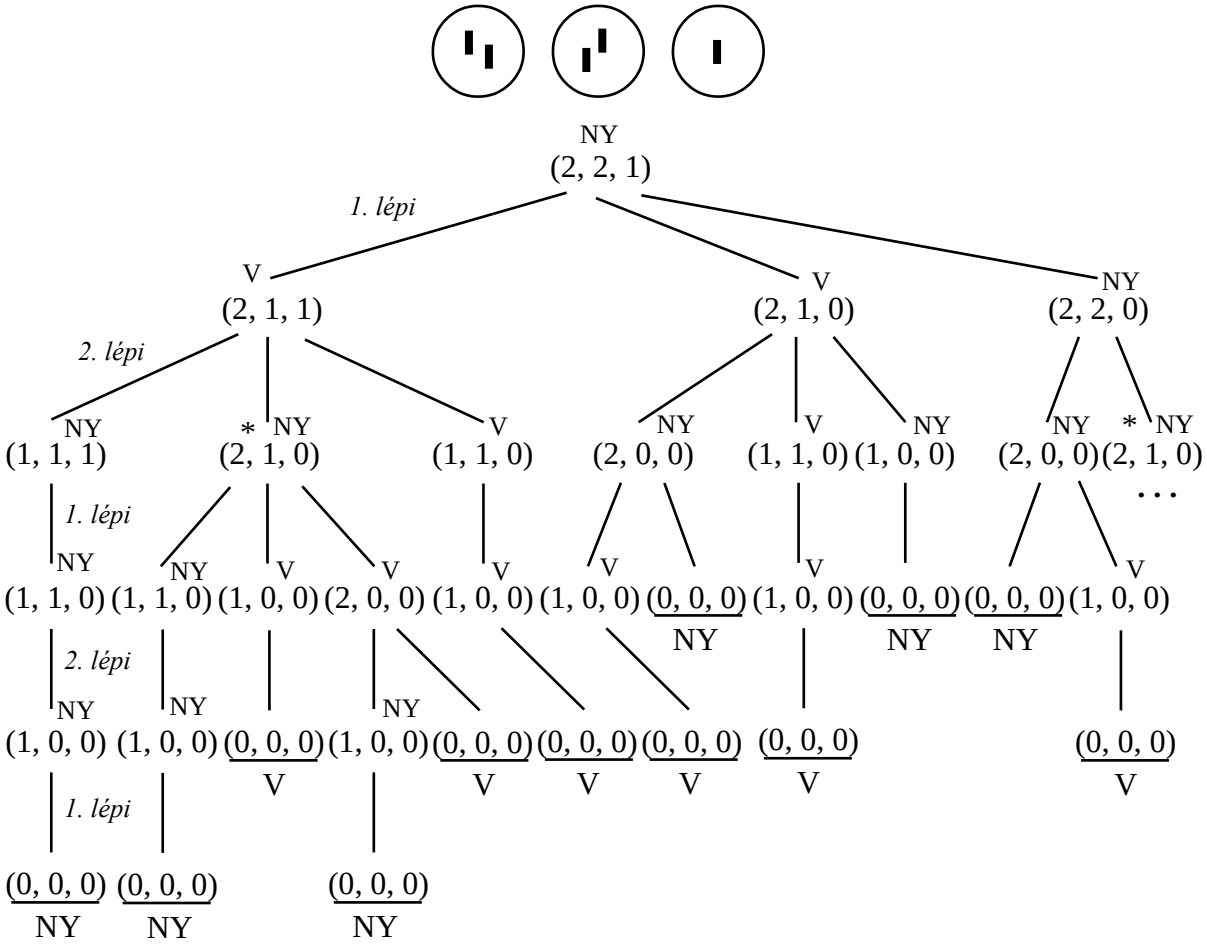
Ebből viszont következik, hogy $g'(n) \leq g(n)$, azaz az n-ig vezető optimális utat már megtaláltuk.

Kétszemélyes játékok

nim játék

- n = 3-ra
- 3 kupac gyufánk van (a kupacokban lévő gyufák száma tetszőleges)
- egy lépésben egyik kupacból vehetünk el gyufát (legalább egyet, maximum a kupacban levő összes gyufát)

- az nyer, aki az utolsó gyufát elveszi
- a kupacok sorrendje nem meghatározott
- az első játékos kezd



- mindkét játékos nyerni akar
- a játékfa minden levelén van egy címke – abban az adott végállapotban ki nyer?
- (azaz a kezdő játékos szempontjából nyerő-e vagy vesztő az adott végállapot)
  - ha a kezdő játékos szempontjából nyerő az adott állás → NY címke jelöli
  - ha a kezdő szempontjából vesztő → V címke jelöli
- a címkéket felvisszük a levelektől egészen a gyökérig (a gyökér címkéje adja majd meg, hogy nyerhet-e a kezdő játékos, azaz van-e nyerő stratégiája)
- itt a kezdő nyer
- (3, 2, 1)-et a *2*. játékos nyeri
- ha +1 a NY, akkor az első játékos *maximalizál*!

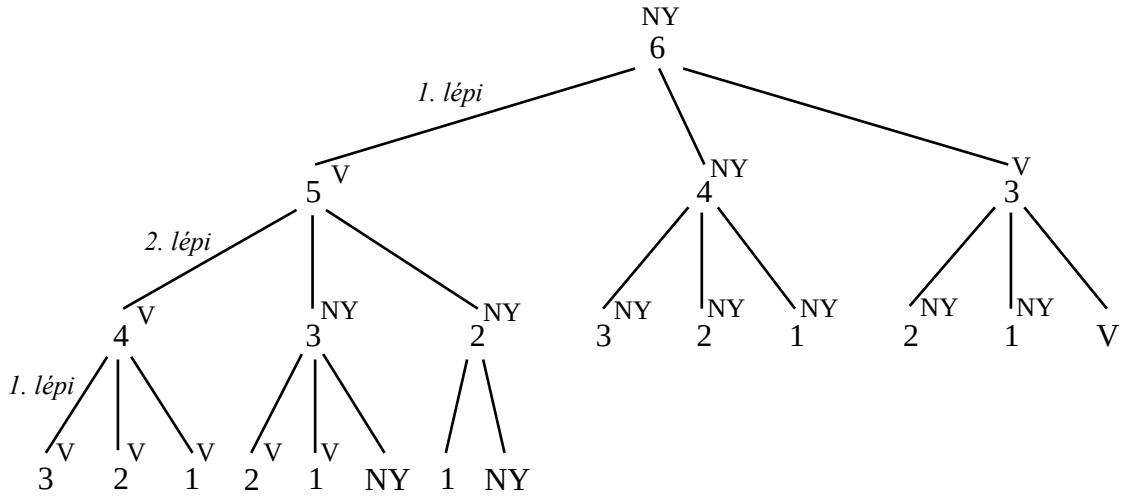
- ha –1 a V, akkor a kettes játékos *minimalizál*!

2 kupacban azonos számú gyufa

- a második játékos nyeri
- mindig ugyanannyit kell a 2-esnek elvennie, amennyit az előző lépésben az 1-es elvett → egyensúly
- különben az 1-es nyer, mert elvesz a többből annyit, hogy az egyensúlyi állapot nála alakuljon ki, amit később is fenn kell tartania

egy kupacban *n* gyufa

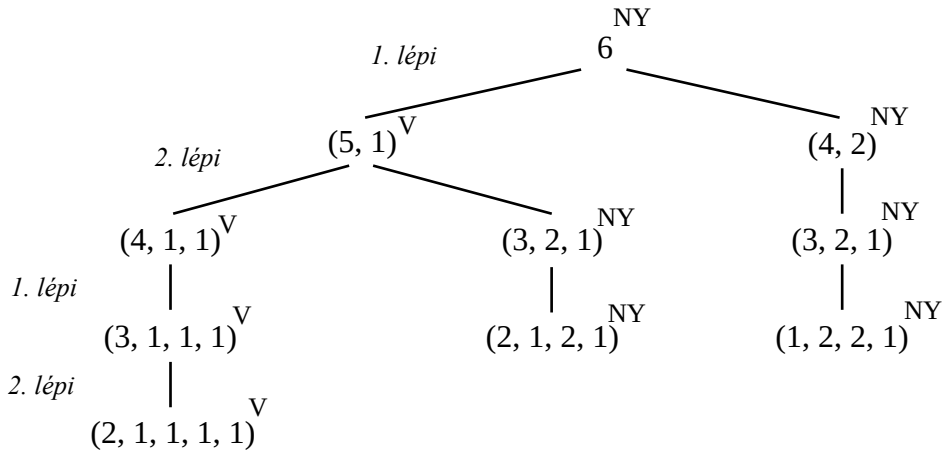
- 1, 2, vagy 3 gyufát vehetnek el egy lépésben.
- *n* = 6-ra → 1-es játékos nyeri!



- a kupac osztható-e 4-gyel?
- ha nem, akkor az egyes játékos elveszi a maradékot, majd
  4 – (a kettes játékos által elvett gyufa) – ennyit kell minden lépésben elvennie → ez a nyerő stratégiája az egyesnek.

grundi játék

- egy kupac
- lépés: egy kupacot → 2 darab, nem azonos elemszámú kupaccá bontani
- aki nem tud már lépni, az veszített



1) Írjuk fel a teljes játékfát!
2) Címkézzük meg!
3) Min-max eljárás

(de van olyan játék, ahol túl nagy a fa – pl. sakk)

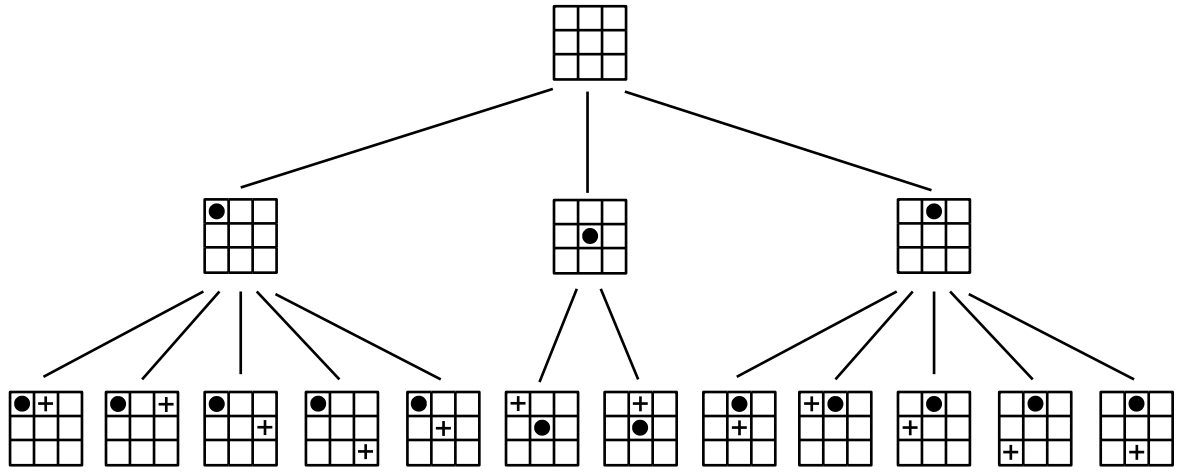
Kétszemélyes, teljes információjú játékok

- 2 játékos játszik
- egymás után lépnek
- pontosan meghatározott szabályok
- mindkét játékos birtokában van a játékkal kapcsolatos összes információnak (ismerik a korábbi állást)
- minden állásban csak véges sok lépés van
- nincs végtelen játék
- a játszma végén vagy valamelyik játékos nyer, vagy döntetlen alakul ki

Áll.: Ha nincs döntetlen, akkor az egyik játékosnak mindig van nyerő stratégiája.
A címkézés egyértelmű.

tic tac toe

- a játékosok felváltva lépnek
- az a játékos nyer, aki 3 jelet bír egymás mellett
- döntetlen a végeredmény, ha mindketten jól játszanak



• • •

- szimmetriát használjuk ki (tükrözzük az állapotokat)
- a leveleket megcímkézzük → szabály: max (*1. játékos*) – min (*2. játékos*)
- alulról felfelé haladunk
- minden belső pontot is megcímkézünk
- ami a gyökér címkéje, az a játék végeredménye (ahogy eddig is)
- a címkék lehetnek:
    - **1** – *1. játékos* nyer
    - **0** – döntetlen
    - **-1** – *1. játékos* veszít

<u>amőba</u>

- korlátlan négyzetrácson
- 15 x 15 vagy 19 x 19-es korlát lehet
- óriási játékfa lenne
- a teljes játékfának felépítjük egy részfáját
- a leveleken állásaink vannak → minden álláshoz valamilyen számértéket kellene rendelni (az adott állás jóságának megfelelően – állásértékelő függvény!
- milyen állásértékelés kellene az amőbánál?
    - nagy, pozitív szám, ha 4 jel van egymás mellett
    - kisebb súllyal, de még mindig jó, ha 2 darab szabad 3-as van … stb.

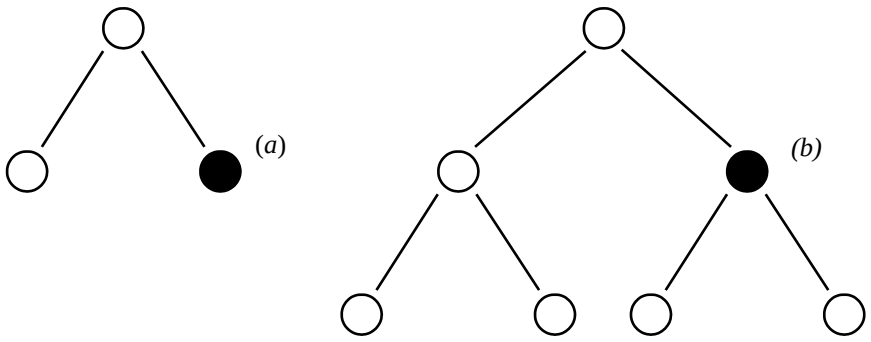
- a kisebb előnyöket is érdemes számolni
  (pl.: olyan 2-est létrehozni, amiből már könnyű 3-ast csinálni)
- 3. szintű lépésfát már nem nagyon tudunk tárolni
- a kiértékelés úgy történik, mint eddig
  - levélig elmegyünk
  - függvény értékeket nézzük
  - min – max szerint választunk

reversie – othello

- nagy, pozitív érték, ha a tábla sarkában vagyunk (+)
- már az is jó, ha eljutottunk a tábla szélére
- 6. oszlopba lépés
- 7. (utolsó előtti) oszlop – rossz → negatív érték (-)
- mennyire vegyük figyelembe a színeket?
- érdemes figyelni, hogy kinek hány korongja van már lent

sakk

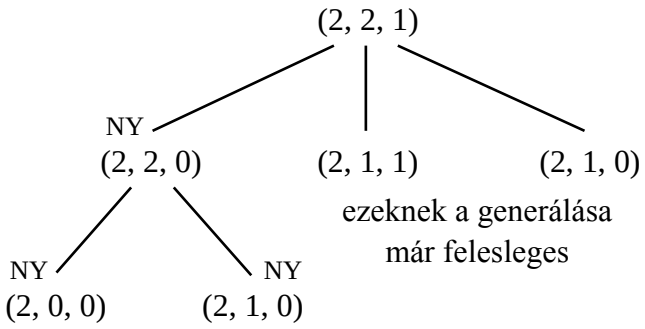
- milyen állásokat, tényezőket vennénk figyelembe?
- minden bábunak lehet különböző értéke
- pl. egy-egy bábu hány mezőre léphet
- állásértékelő játékok hátránya → determinisztikussá válik a játék.
- ha tovább megyünk a fában (mélyebbre), akkor változnak a függvény értékek, azaz



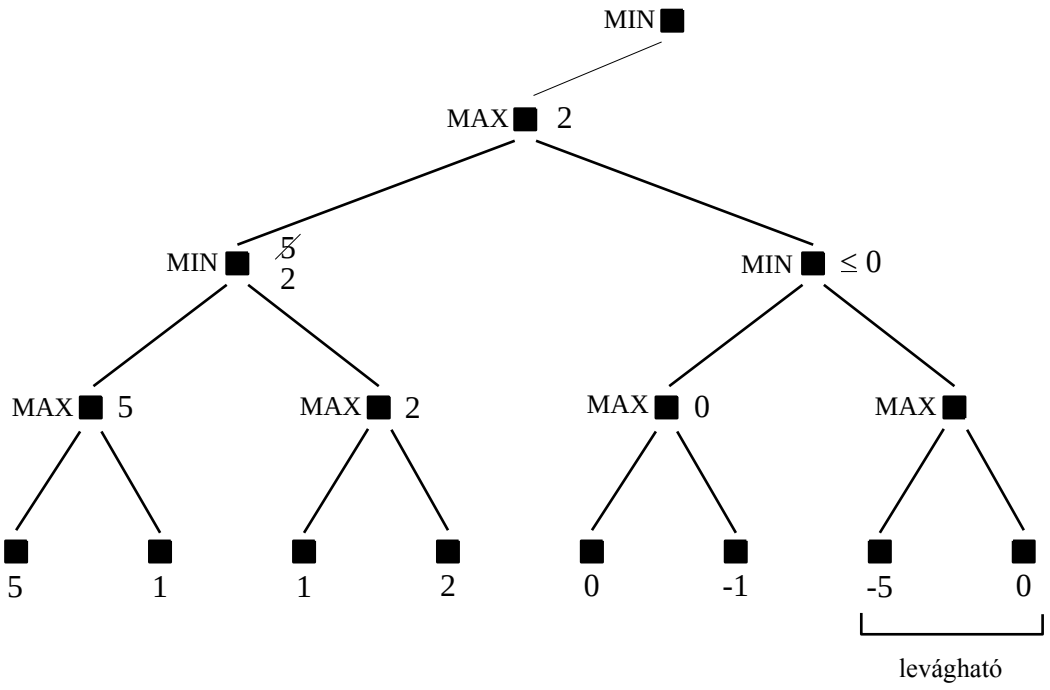
$a \neq b$

α – β vágás

nim játék



eljutott egy olyan címkéhez, amit ő fog nyerni → azt fogja lépni



A minimax eljárás:

- a játékfa adott mélységű generálása,
- a leveleknek megfelelő állások pontértékeinek meghatározása,
- a maximális illetve minimális értékek visszaadása a fában, egészen a gyökérig,
- a választás a gyökér értékével megegyező utódokba vezető lépések között.

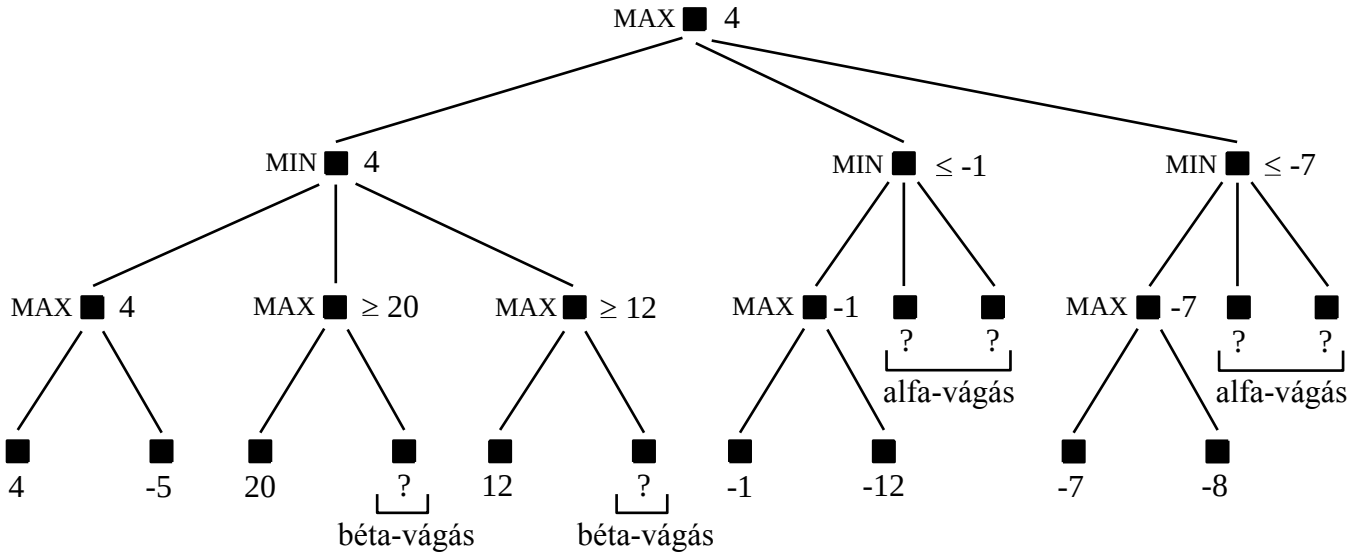
Alfa: az a minimum, amit egy MAX biztosan elér.

Béta: az a maximum érték, amit egy MIN elérhet.

Alfa és béta vágások. Kiindulási pont: ha ismerjük egy max (min) szinten levő csúcs egyik utódjának értékét, akkor ez alsó (felső) korlátja az adott csúcs értékének. Nevezzük ezt az alsó (felső) korlátot alfa (béta) értéknek (MAX - α, MIN - β). Az alfa értékek csak növekedhetnek, a béta értékek csak csökkenhetnek a kiértékelés során.

Alfa (le)vágás: egy min csúcs alatti kiértékelést nem kell folytatni, ha a hozzá tartozó béta érték kisebb, vagy egyenlő, mint ezen csúcs valamely max őséhez tartozó alfa érték. (fenti példa)

Négyzetgyökösíteni szokta a kiszámítandó csomópontok számát!

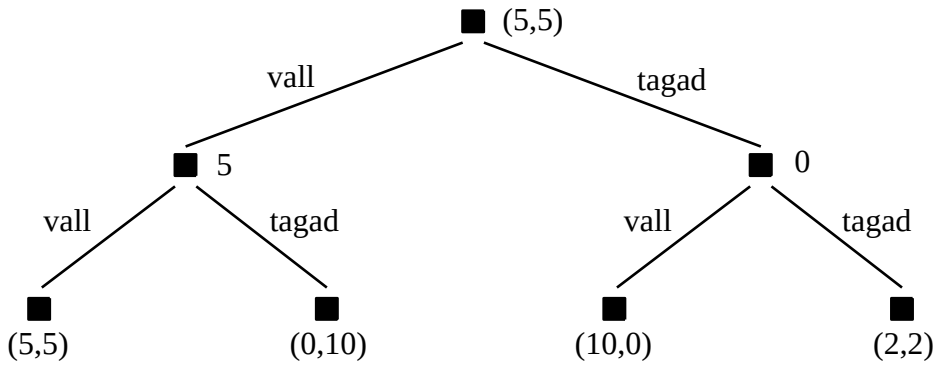


rabok problémája

- 2 betörő
- külön-külön zárkában vannak
- a következő táblázat alapján kapják büntetésüket, annak megfelelően, hogy vallanak, vagy tagadnak.

	Vall	Tagad
Vall	(5, 5)	(0, 10)
Tagad	(10, 0)	(2, 2)

- itt a játékfában fel kell térképezni az összes lehetséges játékpárt
- egyszerre lépnek
- itt eredményes a játékfa



szökevény-üldöző probléma

•

kétszemélyes zérusösszegű játékok

	1	2	3	4
1	-1	3	2	9
2	5	4	4	6
3	8	-2	6	-2

- \- (negatív) szám a 2. játékos nyeresége
- \+ (pozitív) szám az 1. játékos nyeresége
- például, ha az 1. játékos 2-őt lép, a 2. játékos 4-et, akkor az 1. játékos nyer 6-ot.
- ha 1. játékos lép 1-et legrosszabb esetben veszít 1-et
  legjobb esetben nyer 9-et
  2-őt legrosszabb esetben nyer 4-et (mindkét esetben
  legjobb esetben nyer 6-ot nyer)
  3-at legrosszabb esetben veszít 2-őt
  legjobb esetben nyer 8-at

$$\max_i \left(\min_j a_{ij} \right)$$

azaz soronként megnéztük, melyik szám a legkisebb és ennek a maximumát vettük.

- ha 2. játékos lép 1-et legrosszabb esetben veszít 8-at
  legjobb esetben nyer 1-et
  2-őt legrosszabb esetben veszít 4-et
  legjobb esetben nyer 2-őt
  3-at legrosszabb esetben veszít 6-ot (mindkét esetben
  legjobb esetben veszít 2-őt veszít)
  4-et legrosszabb esetben veszít 9-et
  legjobb esetben nyer 2-őt

az oszlopok maximumának a minimumát fogja választani a 2. játékos,
azaz minimalizálja a maximális veszteségét
(azt fogja választani, ahol a lehető legkisebb a maximális vesztesége)

$$\min_j \left(\max_i a_{ij} \right)$$

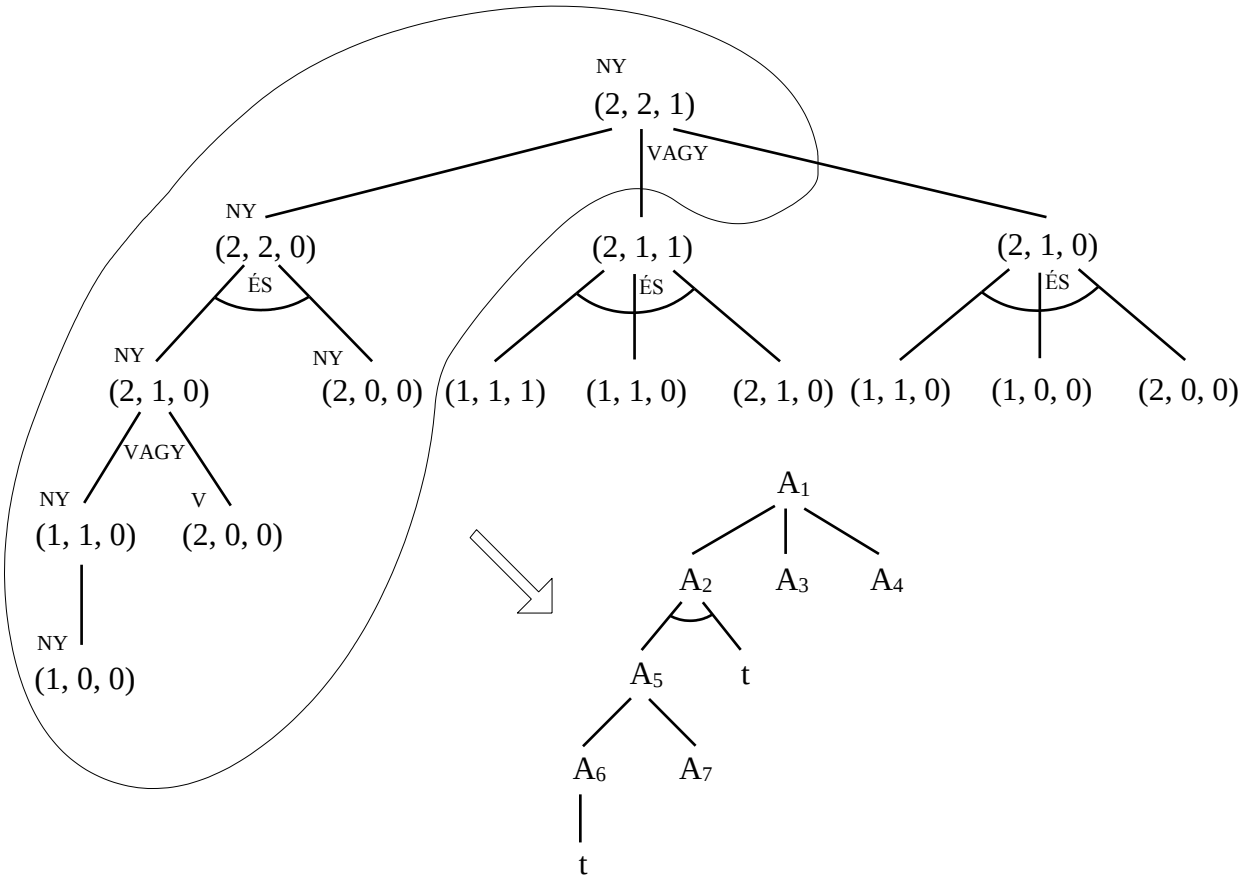
$$\max_i \left(\min_j a_{ij} \right) \leq \min_j \left(\max_i a_{ij} \right)$$

a fenti reláció mindig teljesül
ahol az egyenlőség áll fenn az a ***nyeregpont***
(itt a 2-es az)

Címkéző eljárás

1-es játékos VAGY csomópont

2-es játékos ÉS csomópont



- Problémák felbontása (dekompozíció).
- ***ÉS/VAGY gráfok*** definíciója: egy csúcsnak ÉS és VAGY utódai is lehetnek (probléma-redukció).
- Megoldásgráf: gyökér eleme, ha egy csúcs eleme, akkor legalább egy utódja az, ÉS utód esetében az összes többi ÉS-utód is.
- Levelek primitív (megoldható) problémák (célállapotok) vagy megoldhatatlan problémák.
- Címkézés: kereső eljárásnál egy csúcs megoldott, ha:

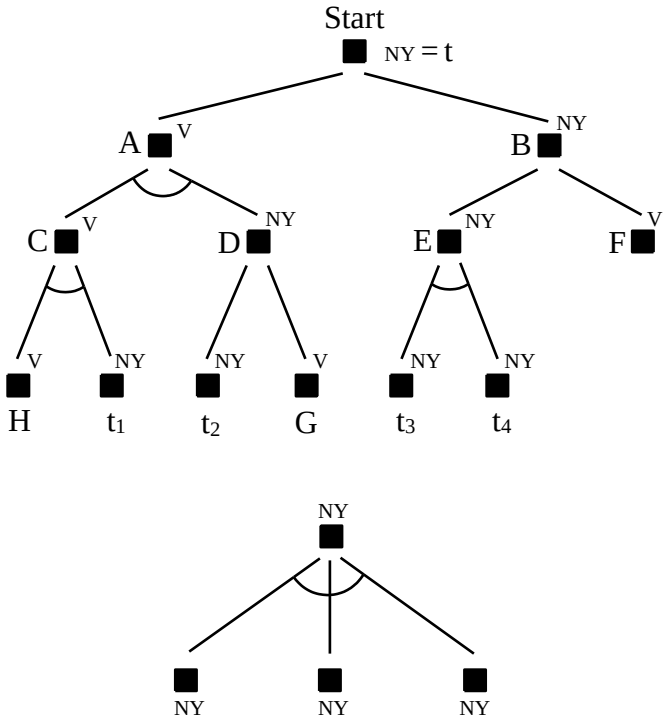
  -primitív probléma (cél, megoldható),

  -van egy VAGY utódja, amelyik már megoldott,

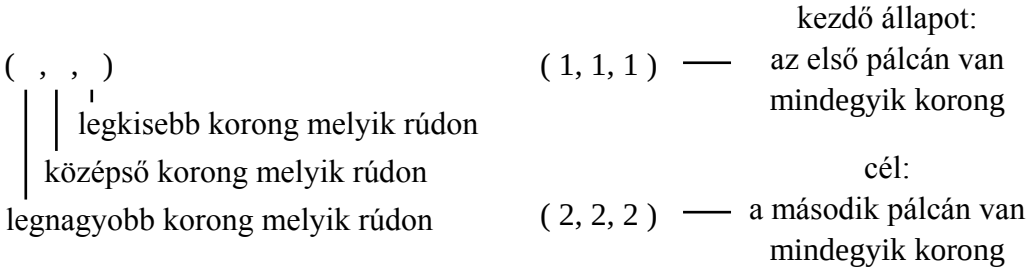
  -az összes ÉS utódja megoldott.

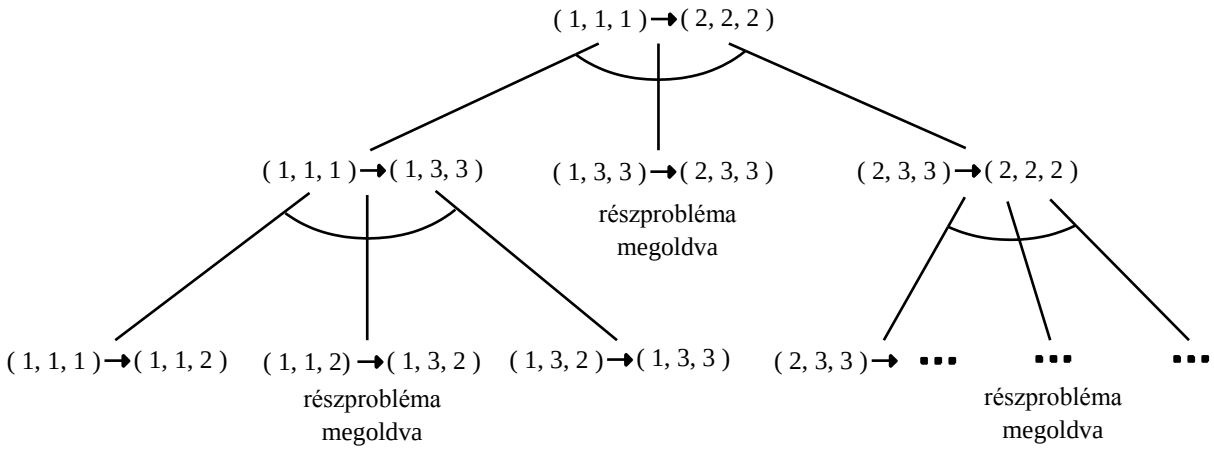
- Megoldhatatlan: nincs utódja és nem primitív probléma; az összes VAGY utódja megoldhatatlan, illetve van legalább egy megoldhatatlan ÉS utódja.
- VÉGSŐ KÉRDÉS: GYÖKÉR CÍMKÉJE

- ***Nyerőfa***
- $t_1, \ldots, t_4$ – célállapotok
- itt nincs meghatározva, hogy melyik szinten lehet ÉS (a játékfánál csak az egyik játékos szintjén voltak az ÉS-ek, a másikon a VAGY-ok)



- hagyományos, állapotteres feladatokat lehet ÉS/VAGY gráffal felírni
- milyen problémáknál találkozunk ÉS/VAGY gráffal?
- például rekurziós fáknál, ilyen a **hanoi tornyai probléma**





- rekurzív megoldás
- a levelek elemi problémák lesznek egy idő után

Alakfelismerés

- adott objektumoknak egy halmaza és osztályoknak egy halmaza
- minden objektumot soroljunk be valamelyik osztályba
- minden objektumnak vannak tulajdonságai → minden objektumhoz tulajdonság vektor fogunk definiálni
- a tulajdonság vektor általában véletlentől függő értékekből épül fel
- ha ez a vektor n dimenziós → n dimenziós teret feszít ki
- k osztályunk van → k részre partícionálni a teret (felosztani) (tulajdonságtér felosztása (döntési =) diszkriminancia függvénnyel)
- ahová mutat a vektor → abba az osztályba tartozik
- hogy csináljunk ilyen felosztást?
- a felosztásnak milyen a jósága?
- példák: számfelismerő, pénzfeldobás, orvos diagnosztikai, nyomtatott irányítószámok felismerése

feltételes valószínűség

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

Példa: (pénz feldobás)

A, B: események

A: ≥ 4 (4-et, vagy annál nagyobbat dobunk)

B: páros

$$P(B|A) = ? = \frac{P(B \cap A)}{P(A)} = \frac{\frac{2}{6}}{\frac{3}{6}} = \frac{1}{3} \cdot \frac{2}{1} = \frac{2}{3}$$

Bayes-formula

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

teljes valószínűség tétele

$B_1, B_2, \ldots, B_n \rightarrow$ teljes eseményrendszert alkot (egyesítésük a teljes eseményt adja)

$$P(A) = P(A|B_1) \cdot P(B_1) + P(A|B_2) \cdot P(B_2) + \ldots + P(A|B_n) \cdot P(B_n)$$

Példa: (dobás)

A: 4-et, vagy annál többet dobtunk

B: páros

$\overline{B}$: páratlan

$\overline{A}$: 1-et, 2-őt, vagy 3-at dobtunk

$$P(A) = P(A|B) \cdot P(B) + P(A|\overline{B}) \cdot P(\overline{B}) = \frac{2}{3} \cdot \frac{1}{2} + \frac{1}{3} \cdot \frac{1}{2} = \frac{1}{2}$$

$$P(B) = P(B|A) \cdot P(A) + P\left(B|\overline{A}\right) \cdot P\left(\overline{A}\right) = \ldots$$

Bayes-tétel (alakfelismerés)

A Bayes-formulában a nevezőt a teljes valószínűség tételével adjuk meg:

|

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B) = P(B|C_1) \cdot P(C_1) + _ + P(B|C_n) \cdot P(C_n)}$$

Példa: (hamis-eredeti érme feldobása)

- 1 szabályos pénz
- 1 hamis pénz
- feldobtuk a pénzt – fejet, vagy írást kaptunk
- hamis, vagy szabályos (eredeti) pénzt dobtunk fel?
-
-

Példa: (orvosi diagnosztika)

- tegyük fel, hogy valamilyen betegség a népesség 0.005-nél fordul elő
- van egy tesztünk, amelyik a betegséget 99% valószínűséggel jelzi
- sajnos a tesz 0.05 valószínűséggel hamis pozitív eredményt ad (azaz nincs betegség, de jelzi)
- osztályok → beteg, nem beteg
- tulajdonság vektor egy elemű → tünet (elemei: beteg – 0, egészséges - 1)
- tudjuk:

  A: beteg $\overline{A}$: egészséges
  B: pozitív teszt $\overline{B}$: negatív teszt

  $P(A) = 0.005$ (mivel teljes eseményrendszert
  $P(\overline{A}) = 0.995$ alkotnak)
  $P(B \mid A) = 0.99$ beteg és potitív eredményt is kap
  $P(\overline{B} \mid A) = 0.01$ beteg, mégis negatív a tesztje
  $P(B \mid \overline{A}) = 0.05$ egeszséges, mégis pozitív
  $P(\overline{B} \mid \overline{A}) = 0.95$ egeszséges, és negatív is a teszt

- kiszámítandó:

P (A | B) = ? ha pozitív, mi a valószínűsége, hogy beteg

P ($\overline{A}$ | B) = ? ha pozitív, mi a valószínűsége, hogy egészséges

P (B) = ? teljes valószínűség tétele alapján a következő:

$$P(B) = P(B|A) \cdot P(A) + P\left(B|\overline{A}\right) \cdot P\left(\overline{A}\right) = 0.99 \cdot 0.005 + 0.05 \cdot 0.995 = 0.0547$$

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)} = \frac{0.99 \cdot 0.005}{0.0547} \approx \underline{0,09}$$

$$P\left(\overline{A}|B\right) = \underline{0.91}$$

Példa: (számfelismerés)

- 0 és 1 számok osztályozása
- magasságuk azonos
- szélességük 1 és 8 között lehet
- (diszkrét valószínűségi változók)

0:	4 széles:	0.2	1:	2 széles:	0.1
	5 széles:	0.3		3 széles:	0.2
	6 széles:	0.2		4 széles:	0.3
	7 széles:	0.2		5 széles:	0.3
	8 széles:	0.1		6 széles:	0.1
	(diszkrét eloszlás)			(diszkrét eloszlás)	

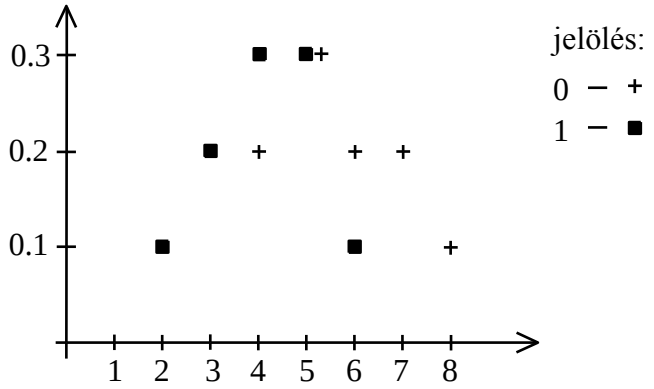
- kérdés: jön egy szám – lemérjük a szélességét – milyen szám jöhetett?

$$\frac{4}{7} = P(1) \qquad \frac{3}{7} = P(0)$$

P (1) – 1-es írásának valószínűsége

P (0) – 0 írásának valószínűsége

- 4, 5, és 6 széles számok érkezése esetén kérdéses (mert ha 2 vagy 3 érkezik, biztos, hogy 1-es volt, ha pedig 7, vagy 8 érkezik, biztos, hogy 0-as volt)



$$P(\xi=4|0)=0.2 \qquad P(\xi=2|1)=0.1$$
$$P(\xi=5|0)=0.3 \qquad P(\xi=3|1)=0.2$$
$$P(\xi=6|0)=0.2 \qquad P(\xi=4|1)=0.3$$
$$P(\xi=7|0)=0.2 \qquad P(\xi=5|1)=0.3$$
$$P(\xi=8|0)=0.1 \qquad P(\xi=6|1)=0.1$$

$$P(\xi=2)=P(\xi=2|0)\cdot P(0)+P(\xi=2|1)\cdot P(1)=0\cdot\frac{3}{7}+\frac{1}{10}\cdot\frac{4}{7}=\frac{4}{70}$$

$$P(\xi=3)=P(\xi=3|0)\cdot P(0)+P(\xi=3|1)\cdot P(1)=0\cdot\frac{3}{7}+\frac{2}{10}\cdot\frac{4}{7}=\frac{8}{70}$$

$$P(\xi=4)=P(\xi=4|0)\cdot P(0)+P(\xi=4|1)\cdot P(1)=\frac{2}{10}\cdot\frac{3}{7}+\frac{3}{10}\cdot\frac{4}{7}=\frac{18}{70}$$

$$P(\xi=5)=P(\xi=5|0)\cdot P(0)+P(\xi=5|1)\cdot P(1)=\frac{3}{10}\cdot\frac{3}{7}+\frac{3}{10}\cdot\frac{4}{7}=\frac{21}{70}$$

$$P(\xi=6)=P(\xi=6|0)\cdot P(0)+P(\xi=6|1)\cdot P(1)=\frac{2}{10}\cdot\frac{3}{7}+\frac{1}{10}\cdot\frac{4}{7}=\frac{10}{70}$$

$$P(\xi=7)=P(\xi=7|0)\cdot P(0)+P(\xi=7|1)\cdot P(1)=\frac{2}{10}\cdot\frac{3}{7}+0\cdot\frac{4}{7}=\frac{6}{70}$$

$$P(\xi=8)=P(\xi=8|0)\cdot P(0)+P(\xi=8|1)\cdot P(1)=\frac{1}{10}\cdot\frac{3}{7}+0\cdot\frac{4}{7}=\frac{3}{70}$$

$$P(0|\xi=2)=\frac{P(\xi=2|0)\cdot P(0)}{P(\xi=2)}=\frac{0\cdot\frac{3}{7}}{\frac{4}{70}}=0$$

$$P(1|\xi=2)=\frac{P(\xi=2|1)\cdot P(1)}{P(\xi=2)}=\frac{\frac{1}{10}\cdot\frac{4}{7}}{\frac{4}{70}}=\frac{4}{70}\cdot\frac{70}{4}=1$$

$$P(0|\xi=3)=\frac{P(\xi=3|0)\cdot P(0)}{P(\xi=3)}=\frac{0\cdot\frac{3}{7}}{\frac{8}{70}}=0$$

$$P(1|\xi=3)=\frac{P(\xi=3|1)\cdot P(1)}{P(\xi=3)}=\frac{\frac{2}{10}\cdot\frac{4}{7}}{\frac{8}{70}}=\frac{8}{70}\cdot\frac{70}{8}=1$$

$$P(0|\xi=4)=\frac{P(\xi=4|0)\cdot P(0)}{P(\xi=4)}=\frac{\frac{2}{10}\cdot\frac{3}{7}}{\frac{18}{70}}=\frac{6}{70}\cdot\frac{70}{18}=\frac{6}{18}=\frac{1}{3}$$

$$P(1|\xi=4)=\frac{P(\xi=4|1)\cdot P(1)}{P(\xi=4)}=\frac{\frac{3}{10}\cdot\frac{4}{7}}{\frac{18}{70}}=\frac{12}{70}\cdot\frac{70}{18}=\frac{12}{18}=\frac{2}{3}$$

$$P(0|\xi=5)=\frac{P(\xi=5|0)\cdot P(0)}{P(\xi=5)}=\frac{\frac{3}{10}\cdot\frac{3}{7}}{\frac{21}{70}}=\frac{9}{70}\cdot\frac{70}{21}=\frac{9}{21}=\frac{3}{7}$$

$$P(1|\xi=5)=\frac{P(\xi=5|1)\cdot P(1)}{P(\xi=5)}=\frac{\frac{3}{10}\cdot\frac{4}{7}}{\frac{21}{70}}=\frac{12}{70}\cdot\frac{70}{21}=\frac{12}{21}=\frac{4}{7}$$

$$P(0|\xi=6)=\frac{P(\xi=6|0)\cdot P(0)}{P(\xi=6)}=\frac{\frac{2}{10}\cdot\frac{3}{7}}{\frac{10}{70}}=\frac{6}{70}\cdot\frac{70}{10}=\frac{6}{10}=\frac{3}{5}$$

$$P(1|\xi=6)=\frac{P(\xi=6|1)\cdot P(1)}{P(\xi=6)}=\frac{\frac{1}{10}\cdot\frac{4}{7}}{\frac{10}{70}}=\frac{4}{70}\cdot\frac{70}{10}=\frac{4}{10}=\frac{2}{5}$$

$$P(0|\xi=7)=\frac{P(\xi=7|0)\cdot P(0)}{P(\xi=7)}=\frac{\frac{2}{10}\cdot\frac{3}{7}}{\frac{6}{70}}=\frac{6}{70}\cdot\frac{70}{6}=1$$

$$P(1|\xi=7) = \frac{P(\xi=7|1)\cdot P(1)}{P(\xi=7)} = \frac{0\cdot\frac{4}{7}}{\frac{6}{70}} = 0$$

$$P(0|\xi=8) = \frac{P(\xi=8|0)\cdot P(0)}{P(\xi=8)} = \frac{\frac{1}{10}\cdot\frac{3}{7}}{\frac{3}{70}} = \frac{3}{70}\cdot\frac{70}{3} = 1$$

$$P(1|\xi=8) = \frac{P(\xi=8|1)\cdot P(1)}{P(\xi=8)} = \frac{0\cdot\frac{4}{7}}{\frac{3}{70}} = 0$$

a hiba valószínűségét minimalizáljuk!!!

mi a valószínűsége annak, hogy hibát követünk el?

(azaz mennyi a valószínűsége annak, hogy nem találjuk el a megfelelő számot?)

$$\begin{aligned}P(Hiba) &= P(Hiba|\xi=2)\cdot P(\xi=2) + P(Hiba|\xi=3)\cdot P(\xi=3)\\ &+ P(Hiba|\xi=4)\cdot P(\xi=4) + P(Hiba|\xi=5)\cdot P(\xi=5)\\ &+ P(Hiba|\xi=6)\cdot P(\xi=6) + P(Hiba|\xi=7)\cdot P(\xi=7)\\ &+ P(Hiba|\xi=8)\cdot P(\xi=8)\\ &= 0\cdot\frac{1}{10} + 0\cdot\frac{2}{10} + \frac{1}{3}\cdot\frac{18}{70} + \frac{3}{7}\cdot\frac{21}{70} + \frac{2}{5}\cdot\frac{10}{70} + 0\cdot\frac{2}{10} + 0\cdot\frac{1}{10}\\ &= 0+0+\frac{6}{70}+\frac{9}{70}+\frac{4}{70}+0+0 = \frac{19}{70}\end{aligned}$$

a.priori $P(A)$ – osztályok előfordulási valószínűsége

a.posteriori $P(A|B)$ – feltételes valószínűség

- d dimenziós tulajdonság vektor
- c osztály : $\omega_1, \omega_2, \ldots, \omega_c$
- ismerjük:
  - $P(\omega_1), P(\omega_2), \ldots, P(\omega_c)$ – *a.priori* valószínűség (osztályba esés valószínűsége)
    $P(\omega_j)$ – j-edik osztályba esés valószínűsége
  - $P(x|\omega_1), P(x|\omega_2), \ldots, P(x|\omega_c)$ – osztályokhoz tartozó eloszlás
    $P(x|\omega_j)$ – j-edik osztályhoz tartozó sűrűség függvény
- Bayes-formula adja az a.posteriori valószínűségeket (a legnagyobbat választom)

(ez a választás a lehető legkisebb hibát eredményezi)

$$P(\omega_j|x) = \frac{P(x|\omega_j) \cdot P(\omega_j)}{P(x) = \sum_{j=1}^{c} P(x|\omega_j) \cdot P(\omega_j)}$$

veszteség függvény *(risk = kockázat)*

veszteség mátrix (négyzetes mátrix)

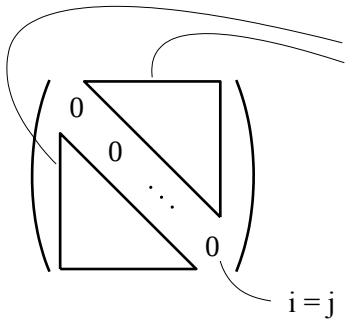
$$\lambda_{ij} = \lambda(\alpha_i, \omega_j)$$

α_i – döntésünk

ω_j – ténylegesen melyik osztályból származott

$\lambda_{i\,j}$ – az ára annak, hogy azt mondjuk, hogy az i osztályból érkezett, holott a j-ből.

- a döntéseket súlyozhatjuk!
  (például sokkal súlyosabb az, ha egy beteg ember kap negatív leletet, mintha egy egészséges ember kapna pozitívat)
- ha azt az osztályt választottuk, amiből való (i = j), akkor 0 a veszteségünk
- ha i ≠ j, akkor ≠0



rossz döntés - az itt álló értékek 0-tól különbözőek
- állhat mindehol 1-es - ekkor minden osztálytévesztés ugyanakkora veszteséggel jár
- állhatnak különböző értékek, ha a veszteségeket súlyozzuk

i = j jó döntés (0-ák)

(átlagos) veszteség

$$R(\alpha_i|x) = \sum_{j=1}^{c} \lambda_{ij} \cdot P(\omega_j|x)$$

- a veszteség várható értékét *kockázatnak* nevezzük. (Risk) (Bayes kockázat)
- a veszteség várható értékét kell minimalizálni!

- 2 osztály esetén: $\alpha_1 \rightarrow \omega_1$ osztály választása esetén

  $\alpha_2 \rightarrow \omega_2$ osztály választása esetén

- *veszteség:*

$$\underline{i=1} \quad R(\alpha_1|x) = \underbrace{\lambda_{11} \cdot P(\omega_1|x)}_{i=1,\ j=1} + \underbrace{\lambda_{12} \cdot P(\omega_2|x)}_{i=1,\ j=2}$$

$$\underline{i=2} \quad R(\alpha_2|x) = \underbrace{\lambda_{21} \cdot P(\omega_1|x)}_{i=2,\ j=1} + \underbrace{\lambda_{22} \cdot P(\omega_2|x)}_{i=2,\ j=2}$$

a kisebb kockázatút választom!

Vegyük úgy, hogy ω_1-et választottuk, azaz

$$R(\alpha_1|x) < R(\alpha_2|x)$$
$$\lambda_{11} \cdot P(\omega_1|x) + \lambda_{12} \cdot P(\omega_2|x) < \lambda_{21} \cdot P(\omega_1|x) + \lambda_{22} \cdot P(\omega_2|x)$$
$$\lambda_{12} \cdot P(\omega_2|x) - \lambda_{22} \cdot P(\omega_2|x) < \lambda_{21} \cdot P(\omega_1|x) - \lambda_{11} \cdot P(\omega_1|x)$$
$$(\lambda_{12} - \lambda_{22}) \cdot P(\omega_2|x) < (\lambda_{21} - \lambda_{11}) \cdot P(\omega_1|x)$$
$$P(\omega_2|x) < \underbrace{\frac{\lambda_{21} - \lambda_{11}}{\lambda_{12} - \lambda_{22}}}_{\frac{1-0}{1-0} = 1} \cdot P(\omega_1|x)$$

ha a veszteséget nem súlyozzuk, akkor a veszteség mátrixban

ahol i = j $\Rightarrow$ 0

különben $\Rightarrow$ 1

(ezért használható a fenti egyszerűsítés)

Bayes-döntés – a.posteriori valószínűség

a nagyobbat választom (a lenti esetben ω_1-et)

$$P(\omega_2|x) < 1 \cdot P(\omega_1|x)$$
$$P(\omega_2|x) < P(\omega_1|x)$$

(a.posteriorik helyére Bayes)

$$\frac{P(x|\omega_2) \cdot P(\omega_2)}{P(x)} < \frac{\lambda_{21} - \lambda_{11}}{\lambda_{12} - \lambda_{22}} \cdot \frac{P(x|\omega_1) \cdot P(\omega_1)}{P(x)}$$

$$P(x|\omega_2) \cdot P(\omega_2) < \frac{\lambda_{21} - \lambda_{11}}{\lambda_{21} - \lambda_{11}} \cdot P(x|\omega_1) \cdot P(\omega_1)$$

$$\frac{P(x|\omega_2)}{P(x|\omega_1)} < \underbrace{\frac{\lambda_{21} - \lambda_{11}}{\lambda_{21} - \lambda_{11}} \cdot \frac{P(\omega_1)}{P(\omega_2)}}_{\text{konstans}}$$

ha a < reláció áll fönt, akkor ω_1-et választom, különben ω_2-őt.

minimális átlagos veszteséget biztosítja, tehát

ilyen térfelosztást kellene találni! → *diszkriminancia függvény*

- minden osztályhoz definiálunk egy függvényt
- azt az osztályt fogjuk választani, amihez tartozó függvényérték a *legnagyobb*
- amelyik térrészbe esik, abba az osztályba fogjuk sorolni

diszkriminancia függvény (határfelületek meghatározása)

$g_i(x)$

$g_i(x) > g_j(x) \quad i \neq j \quad \Rightarrow \quad \alpha_i$ lenne a választásunk, ω_i osztályból származónak tartanánk

diszkriminancia függvény lehet:

1) a.posteriori

   Bayes- féle döntési függvény

$$g_i(x) = P(\omega_i|x) = \frac{P(x|\omega_i) \cdot P(\omega_i)}{\underbrace{P(x)}_{\text{konstans} \Rightarrow \text{elhagyhatjuk}}}$$

2) veszteség függvény

   veszteség mátrix – azt választjuk, ahol a veszteségfüggvény minimális.

$$g_i(x) = \frac{P(x|\omega_i) \cdot P(\omega_i)}{\underbrace{P(x)}_{\text{konstans} \Rightarrow \text{elhagyhatjuk}}}$$

$$g'_i(x) = P(x|\omega_i) \cdot P(\omega_i)$$

$$g''_i(x) = \ln P(x|\omega_i) + \ln P(\omega_i) \qquad (*)$$

ezen függvények metszéspontjaival lehet leírni a határoló felületeket (a térfelosztást)

2 osztály esetén:

$g_1(x)$, $g_2(x)$ a.posterior valószínűség

$g_1(x) > g_2(x)$ az 1. osztályba soroljuk az x objektumot

$$g_1(x) > g_2(x)$$
$$g_1(x) - g_2(x) > 0$$
$$g''_1(x) - g''_2(x) > 0$$
$$\ln P(x|\omega_1) + \ln P(\omega_1) > \ln P(x|\omega_2) + \ln P(\omega_2)$$
$$\ln P(x|\omega_1) - \ln P(\omega_2) > \ln P(x|\omega_2) - \ln P(\omega_1)$$
$$\ln\left(\frac{P(x|\omega_1)}{P(x|\omega_2)}\right) > \ln\left(\frac{P(\omega_2)}{P(\omega_1)}\right)$$
$$\ln\left(\frac{P(x|\omega_1)}{P(x|\omega_2)}\right) - \ln\left(\frac{P(\omega_2)}{P(\omega_1)}\right) > 0$$
$$\ln\left(\frac{P(x|\omega_1)}{P(x|\omega_2)}\right) + \ln\left(\frac{P(\omega_1)}{P(\omega_2)}\right) > 0$$

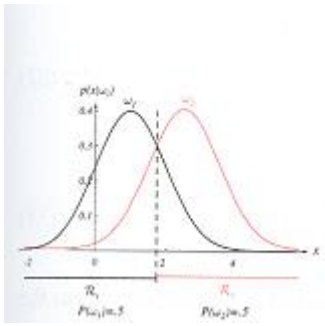
tfh. egyváltozós normális eloszlásról van szó.

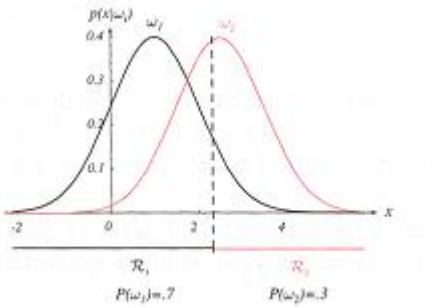
$$P(x) = \frac{1}{\sqrt{2\pi}\cdot\sigma}\cdot e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

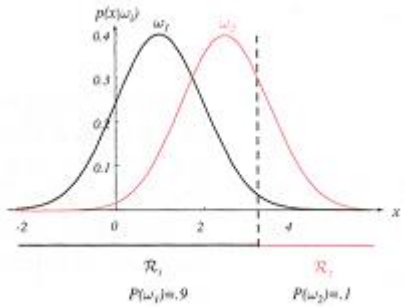
$$f(x) \quad E(f(x)) = \int_{-\infty}^{+\infty} f(x)p(x)\,dx$$

$$\mu = E(x) = \int x\,p(x)\,dx$$

$$\sigma^2 = E((x-\mu)^2)$$







Példa:. (többváltozós normális eloszlás)

- Két osztály, mindkettőnek 4-4 pontja ismert:

  (2,6), (3,4), (3,8), (4,6)

  (1,-2), (3,-4), (3,0), (5,-2)

- Ekkor μ_1, Σ_1, μ_2, Σ_2, továbbá a mátrixinverzek kiszámíthatók, azonos a.priori valószínűségek mellett a döntési felület:

$$x = 3{,}514 - 1{,}125y + 0{,}1825y^2$$

ROC görbék (receiver operating charasteristic)

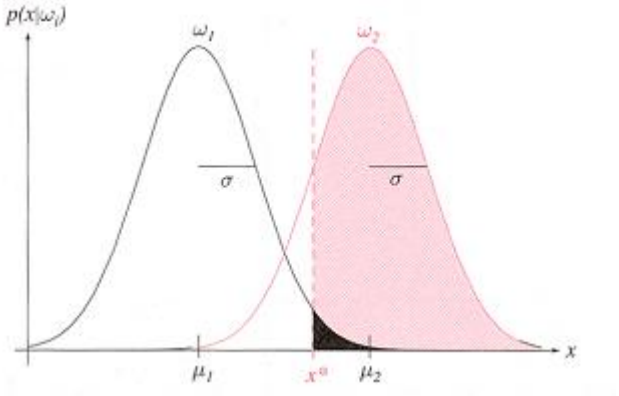
alkalmazás: jelérzékelés

zajos körülmények között (Gauss eloszlás) mérünk jeleket.

ha van jel, μ_2 a várható érték, ha nincs, μ_1 (vagyis $P(x|\omega_i) = N(\mu_i, \sigma^2)$).

megkülönböztethetőség:

$$d' = \frac{|\mu_2 - \mu_1|}{\sigma}$$



Példa: radar

ω_1 – nincs jel

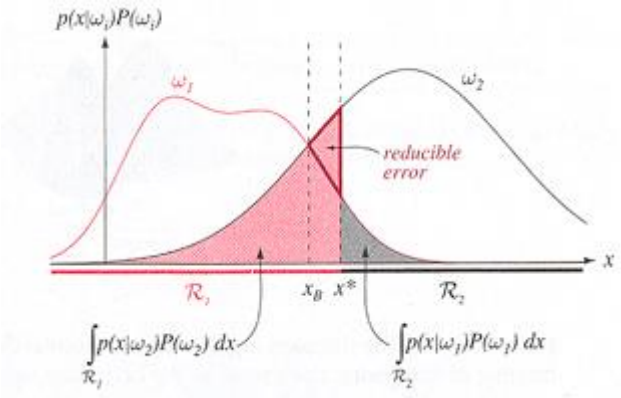
ω_2 – van jel

2 osztály, egydimenziós eset $\Rightarrow$ 2 részre partícionálni az egyenest: R_1-re, és R_2-re

hibavalószínűségek:

$$P(\text{hiba}) = P(x \in R_2|\omega_1) \cdot P(\omega_1) + P(x \in R_1|\omega_2) \cdot P(\omega_2) =$$
$$= \int_{R_2} P(x|\omega_1) \cdot P(\omega_1)\, dx + \int_{R_1} P(x|\omega_2) \cdot P(\omega_2)\, dx$$

4 definíciót vezettek be radarok esetén:

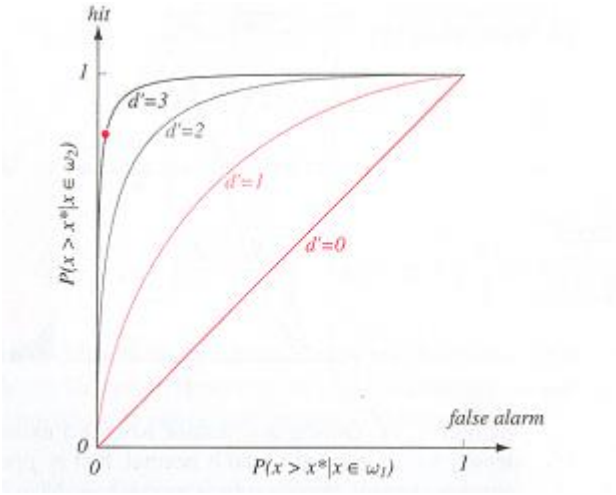


- *találat*: $P(x > x^* \mid \omega_2)$
  - x^*-nál nagyobb értéket mértünk
  - ω_2 –ből jött a jel
- *hamis riasztás*: $P(x > x^* \mid \omega_1)$
  - másodfajú hiba
  - téves pozitív lelet, azaz hamisan azt mondjuk, hogy van jel, holott nincs
- *tévesztés*: $P(x < x^* \mid \omega_2)$
  - elsőfajú hiba
  - pozitív tünet fel nem ismerése

- *helyes elvetés*: $P\ (x < x^* \mid \omega_1)$

különböző *d* értékekhez tartozó *ROC görbék*

- sok kísérlet esetén a valószínűségek x* függvényében becsülhetők: ROC görbék
- minél nagyobb a különbség a két várható érték között, annál magasabban van a görbe



(Dombi József előadása)

Bizonytalanságok melletti következtetés problémája

valszám. megközelítés mellett - emberi hiedelmek → hatékony eljárások

Példa: (tenisz)

$f : x \to v$

$x \in (a_1 \ldots a_n)$ $v = \{Yes, No\}$

a_1 = időjárás = (S, O, R) (S – napos, O – felhős, R – esős)

a_2 = hőmérséklet = (H, M, C) (H – magas, M – közepes, C – hűvös)

a_3 = páratartalom = (H, N) (H – magas, N – normális)

a_4 = szél = (T, F) (T – szél van, F – nincs szél)

a_5 = teniszjáték (Y, N) - érdemes-e belefogni a játékba… (Y – igen, N – nem)

a_1	a_2	a_3	a_4		a_5
S	H	H	F		N
S	H	H	T		N
O	H	H	F		Y
R	M	H	F		Y
R	C	N	F		Y
R	C	N	T		Y
O	C	N	T		N
S	M	H	F		Y
S	C	N	F		N
R	M	N	F		Y
S	M	N	T		Y
O	M	H	T		Y
O	H	N	F		Y
R	M	H	T		N

$$P(I) = \frac{9}{14} \qquad P(N) = \frac{5}{14}$$
$$P(S|I) = \frac{2}{9} \qquad P(S|N) = \frac{3}{5}$$
$$P(C|I) = \frac{3}{9} \qquad P(C|N) = \frac{1}{5}$$
$$P(H|I) = \frac{3}{9} \qquad P(H|N) = \frac{4}{5}$$
$$P(T|I) = \frac{3}{9} \qquad P(T|N) = \frac{3}{5}$$

- Kérdés: (*S, C, H, S*) körülmények között érdemes-e kimenni teniszezni?
- Az attribútumok (tulajdonságok) függetlenek.
- *naiv Bayes* szerint ki kell minden osztály valószínűségét számolni..
- a nagyobb valószínűségű osztályba kell sorolni.

$$P(I) \cdot P(S|I) \cdot P(C|I) \cdot P(H|I) \cdot P(T|I) = \frac{9}{14} \cdot \frac{2}{9} \cdot \frac{3}{9} \cdot \frac{3}{9} \cdot \frac{3}{9} = \frac{1}{189} = \underline{0.005}$$
$$P(N) \cdot P(S|N) \cdot P(C|N) \cdot P(H|N) \cdot P(T|N) = \frac{5}{14} \cdot \frac{3}{5} \cdot \frac{1}{5} \cdot \frac{4}{5} \cdot \frac{3}{5} = \frac{18}{875} = \underline{0.02}$$

- tehát nem érdemes kimenni teniszezni, mert az N osztályba esés valószínűsége a nagyobb

Valószínűségi következtető rendszerek

A tudásom leképezése egy hálóba.

Valószínűségi háló (belief network)

- a gráf csomópontjai (csúcsai) valószínűségi változók
- a csomópontokat irányított, élekkel kötjük össze
- feltételes valószínűségeket rendelünk a csúcsokhoz – valószínűségi tábla („szülők hatása”)
- körmentes a gráf

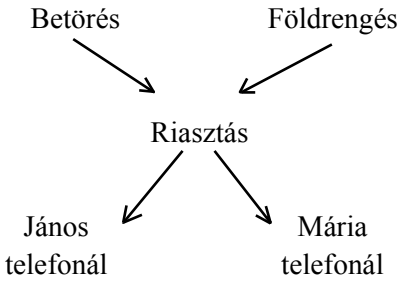
Példa: (Pearl példája)

- riasztó beszerelése – jelzi a földrengést is

- két szomszéd: Mária és János, akik telefonálnak, ha szól a riasztó
- de a riasztó nem tökéletes, János nem mindig tudja a riasztót a telefontól megkülönböztetni, és Mária fülhallgatóval hallgat zenét
- mindezeket valószínűségi táblával adjuk meg
- János és Mária nem érzékelik a földrengést, csak a riasztást; a betörőt sem, és egymást sem, csak a riasztást! – tehát bizonyos kapcsolatok elhanyagolhatók, azaz csak a tényleges függőségben levők számítanak!

Szomszédsági hálózat

topológia



A riasztás valószínűségének megadása

CPT-tábla – feltételes valószínűségi táblázat (ez az, amit minden csomóponthoz megadunk)

Betörés	*Földrengés*	*P* (*Riasztás* | *Betörés*, *Földrengés*)	
		igaz	hamis
igaz	igaz	0.95	0.05
igaz	hamis	0.95	0.05
hamis	igaz	0.29	0.71
hamis	hamis	0.001	0.999

P (*Betörés*) = *P*(*B*) = 0.01

P (*Földrengés*) = *P*(*F*) = 0.02

P (*Riasztás*) = *P*(*R*) – riasztáshoz tábla (a fenti táblázat első három oszlopa)

(hatalmas tábla lehet bizonyos esetekben)

János

	P(J) – János telefonál	János nem telefonál
igaz	0.9	0.1
hamis	0.05	0.95

Mária

	P(M) – Mária telefonál	Mária nem telefonál
igaz	0.7	0.3
hamis	0.01	0.99

$$P(X_1 = x_1 \wedge \ldots \wedge X_n = x_n) = \prod_{i=1}^{n} P(x_i | SZULOK\, X_i)$$

$$P(J \wedge M \wedge R \wedge \neg B \wedge \neg F) = P(J|R) \cdot P(M|R) \cdot P(R|\neg B \wedge \neg F) \cdot P(\neg B) \cdot P(\neg F)$$
$$= 0.9 \cdot 0.7 \cdot 0.001 \cdot 0.999 \cdot 0.998 = 0.000628$$

Nagyon kicsi a valószínűsége tehát annak, hogy János is és Mária is telefonált, volt is riasztás, de nem volt sem földrengés, sem betörés.

(rekurzívan végzem a következőket:)

$$P(x_1 \ldots x_n) = P(x_n | x_{n-1} \ldots x_1) \cdot P(x_{n-1} \ldots x_1) = \ldots$$
$$= P(x_n | x_{n-1} \ldots x_1) \cdot P(x_{n-1} | x_{n-2} \ldots x_1) \cdot \ldots \cdot P(x_2 | x_1) \cdot P(x_1)$$

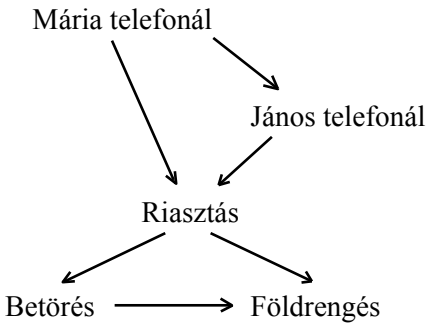
$$P(X_i | X_{i-1} \ldots X_1) = P(X_i | SZULOK(X_i))$$

<u>Hálók építése</u>

- befolyásoló tényezők, vagyis a változók meghatározása
- sorrend kijelölése
- amíg van érintetlen változó
    a) vegyünk egy ilyet, adjuk a csúcsokhoz
    b) határozzuk meg a szüleit
    c) adjuk meg a feltételes valószínűségek tábláját

Ha megváltoztatjuk a sorrendet, már más hatásmechanizmusok munkálnak.

a fenti riasztásos példa így már sokkal bonyolultabb:



- *determinisztikus* csomópontok nincs zaj

Példa: ha Észak-Amerika vagy USA, vagy Kanada, vagy Mexikó

- *nemdeterminisztikus* csomópontok zajos

Példa: megfázás, influenza, malária | láz

(ahol csak egy igaz van, ott nincs zaj, ahol több igaz van, ott már van zaj)

Megfázás	*Influenza*	*Malária*	*P* (*Láz*)	*P* (¬*Láz*)
hamis	hamis	hamis	0	1
hamis	hamis	**igaz**	0.9	0.1
hamis	**igaz**	hamis	0.8	0.2
hamis	igaz	igaz	0.98	$0.2 \cdot 0.1 = 0.02$
igaz	hamis	hamis	0.4	0.6
igaz	hamis	igaz	0.94	$0.6 \cdot 0.1 = 0.06$
igaz	igaz	hamis	0.88	$0.6 \cdot 0.2 = 0.12$
igaz	igaz	igaz	0.998	$0.6 \cdot 0.2 \cdot 0.1 = 0.002$

A megjelölt esetekben nincs zaj.

Következtetés valószínűségi hálókban

Milyen jellegű lekérdezések lehetnek?

- *diagnosztikai következtetés*:

hatásról → okra következtetünk

pl.: riasztásos példa esetében $P(B \mid J)$ (azaz, ha adott, hogy János telefonált,

akkor kiszámítható, hogy betörés volt)

- *okozati következtetés*:

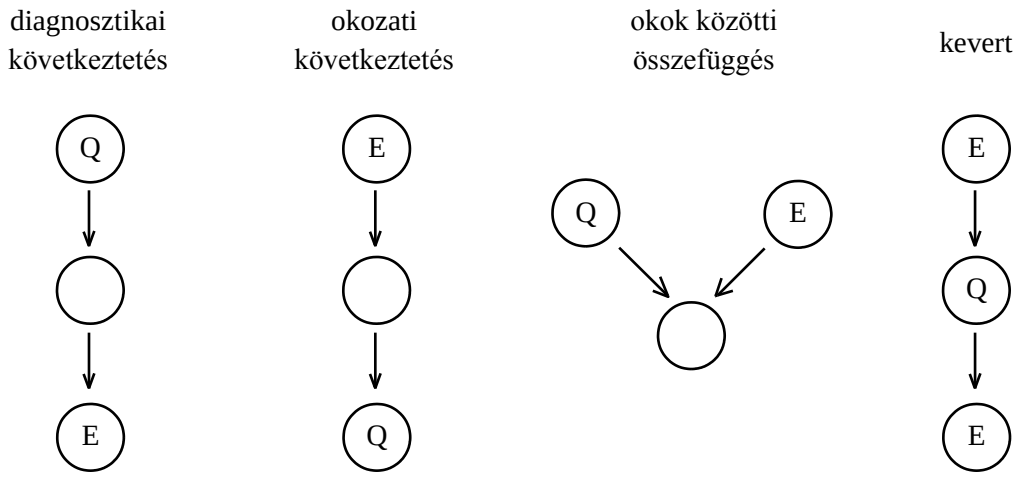
  okról → hatásra következtetünk

  pl.: riasztásos példa esetében $P(J \mid B)$

- *okok közötti összefüggés*

- *kevert*

E - tényváltozó
Q - lekérdezéses változó



<u>Következtetés többszörösen összekötött valószínűségi hálóban</u>

<u>A hálók kezelésére alkalmas algoritmusok:</u>

1) *összevonós eljárás*

   átalakítják a hálót egy gráffá a nem megfelelő csomópontokat összevonva

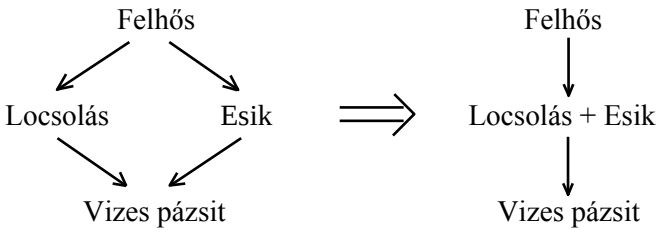
2) *feltételezéses eljárás*

   átalakításnál a változók értékét rögzítik, majd kiértékelik a gráfot minden lehetséges értékkombinációra. (szétbontjuk → csinálunk 2 hálót)

3) *szimulációs eljárás (sztohasztikus)*

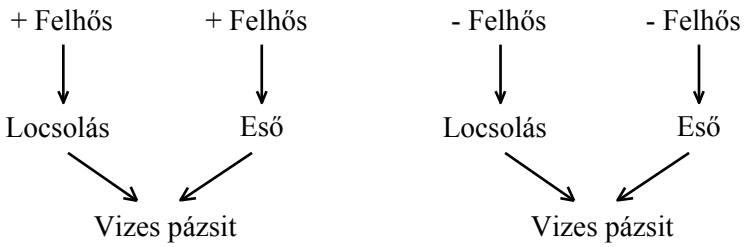
   tárgytartomány nagyon nagy számú, konkrét modelljét generálják le

Példa az *összevonós eljárásra* (1)



Példa a *feltételezéses eljárásra* (2)

\+ - teljesül a feltétel
\- - nem teljesül a feltétel



(Csirik János előadása)

$$p(x) = \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(\underline{x}-\underline{\mu})^t \Sigma^{-1}(\underline{x}-\underline{\mu})}$$

$$\sigma_{ij}: \Sigma \quad E\left(\underline{x}-\underline{\mu}\right)\left(\underline{x}-\underline{\mu}\right)^t$$

- minden osztályhoz van egy ilyen sűrűség függvény → *diszkriminancia függvény* → *x*-be behelyettesítünk, és a legnagyobb diszkriminanciájút választjuk.
- ezek a diszkr. partícionálják a teret
- $g_i(x)$ - *x*-edik osztályhoz tartozó *diszkriminancia függvény*

$$g_i(x) = \ln P(x|\omega_i) + \ln P(\omega_i)$$

$$\ln P(x|\omega_i) = -\frac{1}{2}\left(\underline{x}-\underline{\mu_i}\right)^t \Sigma_i^{-1}\left(\underline{x}-\underline{\mu_i}\right) - \underbrace{\frac{d}{2}\ln(2\pi)}_{\text{konstans}} - \frac{1}{2}\ln|\Sigma_i| = (*)$$

1) $\Sigma_i = \sigma^2 I$ ahol Σ_i az i-edik osztályhoz tartozó kovariancia mátrix

σ^2 a szórás négyzet

I egységmátrix

a komplemensek függetlenek egymástól és ugyanaz a szórásuk is, ekkor

$$\underbrace{-\frac{1}{2}\ln|\Sigma_i|}_{\text{is konstans}}$$

a főátlóban vesszük a reciprokokat, azaz

$$\Sigma_i^{-1} = \frac{1}{\sigma^2} I$$

$$(*) = -\frac{1}{2\sigma^2}\left(\underline{x}-\underline{\mu_i}\right)^t\left(\underline{x}-\underline{\mu_i}\right) + \underbrace{\underline{C}}_{\text{konstansok}} =$$

$$= -\frac{1}{2\sigma^2}\left(\underline{x}^t\underline{x} - 2\underline{\mu}_i^t\underline{x} + \underline{\mu}_i^t\underline{\mu}_i\right) + C = (**)$$

$$\underline{x}^t \cdot \underline{\mu}_i = \underline{\mu}_i^t \cdot \underline{x}$$

- $\underline{x}^t \cdot \underline{x}$ ez megint nem függ az i-től, tehát nyertünk egy újabb konstansot

$$(**) = -\frac{1}{2\sigma^2}\left(-2\underline{\mu}_i^t\underline{x} + \underline{\mu}_i^t\underline{\mu}_i\right) + C$$

$$= \frac{1}{\sigma^2}\underline{\mu}_i^t\underline{x} - \frac{1}{2\sigma^2}\underline{\mu}_i^t\underline{\mu}_i + C$$

- a két osztály között milyen elválasztó felület lesz?
- ha adott 2 osztály, *i* és *j*, akkor azt választjuk, amelyik *diszkriminanciája* (függvény értéke) nagyobb
- pl.: ($i = 1, j = 2$)

  $g_1 > g_2 \quad \rightarrow \quad$ 1. osztályt választom

$$\frac{1}{\sigma^2}\left(\underline{\mu}_i^t - \underline{\mu}_j^t\right)\underline{x} - \frac{1}{2\sigma^2}\left(\underline{\mu}_i^t\underline{\mu}_i - \underline{\mu}_j^t\underline{\mu}_j\right) =$$

$$= \frac{1}{\sigma^2}\underbrace{\left(\underline{\mu}_i^t - \underline{\mu}_j^t\right)}_{\text{kiemelve}}\underbrace{\left(x - \frac{1}{2}\left(\underline{\mu}_i + \underline{\mu}_j\right)\right)}_{\substack{\text{ha ez 0,} \\ \text{akkor a szorzat is 0}}}$$

- ez egy sík, mert elsőfokú a függvény
- átmegy a két várható vektor középpontján (hipersík)
- ezek a vektorok egyike sem 0, de merőlegesek egymásra
- a számolás során el lett hagyva az osztályvalószínűség, tehát akkor érvényes a képlet, ha az osztályvalószínűségek azonosak

  (+ $ln\ P\ (\omega_i)$ – vel nem számoltunk)

2) $\Sigma_i = \Sigma$ azaz, ha a konvergencia mátrixok megegyeznek

   (ez is lineáris lesz)

3) Σ - k általánosak (kvadratikus felület)

Példa:

- van egy hamis, és egy igazi érménk.
- dobjuk fel d-szer a kiválasztott érmét!
- d alkalom után akarunk valamit mondani a dobásokról
- többváltozós eloszlásként jellemezhető

$\underline{x} = (x_1, \ldots, x_d)$

0 0

1 1

- 2 osztály van: valódi pénz osztálya – ω_1

  hamis pénz osztálya – ω_2

  $P(x_i = 0 \mid \omega_1) = p$ ($x_i = 0$ – fej)

  $P(x_i = 0 \mid \omega_2) = q$ ($x_i = 1$ – írás)

  $\underline{x} = (x_1, x_2, \ldots, x_d)$ (vektor)

- $P(x \mid \omega_1) = ?$

$$P(x|\omega_1) = \prod_{i=1}^{d} p^{1-x_i}(1-p)^{x_i}$$

ha $d = 1$ $P(x = 0 \mid \omega_1) = p$

$P(x = 1 \mid \omega_1) = 1 - p$ } $p^{1-x}(1-p)^{x}$

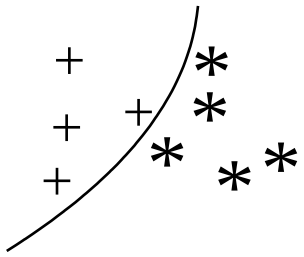
- $P(x \mid \omega_2)$ hasonlóan megadható

Gradiens eljárás

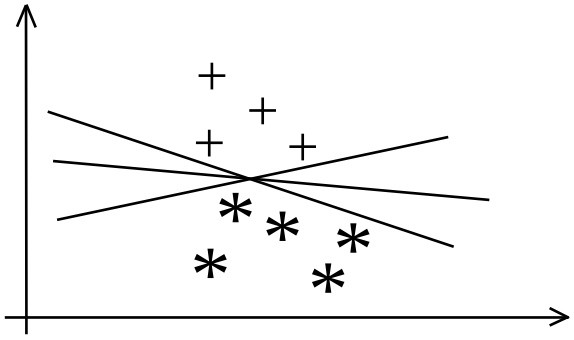
Feladat:

- adott 2 halmaz, amiben pontok vannak
- adjunk meg egy olyan egyenest, ami a 2 halmaz pontjait elválasztja!
- egyszerűbb a kiszámolás, ha egyből a határfelületekre nézem → ahol a két diszkriminancia függvény megegyezik egymással.
- hogyan tudjuk meghatározni azt a hipersíkot, amitől az egyik irányba vannak az egyik osztály pontjai, a másikba a másiké?

+ - 1. osztály pontjai
* - 2. osztály pontjai

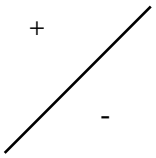


- címkézett pontok → tudjuk , hogy melyik osztályba tartozik
- keressük az egyenest
- végtelen sok megoldás lehetséges



- keressük azt az algoritmust, ami ezt az egyenest megtalálja (pontosabban egy ilyen egyenest talál!)
- 50 évvel ezelőtt született meg az algoritmus (neuronhálók alkalmazása során)
- ha síkban vagyunk, akkor egyszerű :
  attól függően kap *előjelet* az adott pont, hogy melyik oldalára esik az egyenesnek

pl.: +



-

- térben: d – dimenzió lineáris függvény

$$\sum_{i=1}^{d} \omega_i x_i + \omega_0$$

- trükk: *homogén koordináták* használata → d dimenziós térből áttérünk $d + 1$ dimenziósra

$$\sum_{i=1}^{d} \omega_i x_i + \omega_0 = \underline{a}^t \underline{y}$$

- az új dimenzió kezdetben 1-es ($x \to d$ dimenzió $y \to d+1$ dimenzió)

$$\underline{x} \to \underline{y} \begin{pmatrix} 1 \\ x_1 \\ x_2 \\ \vdots \\ x_d \end{pmatrix} \qquad \underline{\omega} \to \underline{a} \begin{pmatrix} \omega_0 \\ \omega_1 \\ \omega_2 \\ \vdots \\ \omega_d \end{pmatrix}$$

ω_1 : $\underline{a}^t \cdot \underline{y} > 0$ ω_2 : $\underline{a}^t \cdot \underline{y} \leq 0$

1. osztálybeli pontok 2. osztálybeli pontok

- a 2. osztályhoz tartozó pontokat szorozzuk meg (–1) - el , így ugyanarra az oldalra fog kerülni az összes 2. osztálybeli pont, mint ahol az 1. osztálybeliek vannak, tehát az összes pont a sík azonos oldalára fog esni

$$\underline{y} \to -\underline{y}$$

- hogyan fognak kinézni ezek a 2 dimenziós egyenesek, illetve hogyan fognak kinézni 3 dimenziós térben?

  3 dimenzióban a homogén koordináta-rendszerben az egyenesek az origón haladnak át

- tehát keresünk egy egyenest, amire igazak a következők:
  - az összes pont a pozitív oldalon van
  - átmegy az origón a homogén koordináta-rendszerben

gradiens eljárásokat fogunk használni

- van egy célfüggvényünk, ennek keressük a minimumát
- erre eljárás a gradiens eljárás (iterációs)
- közelítő eljárás → elindul valahonnan, adott pontban veszi a gradiensét, és abba az irányba, amit a gradiens mutat, lép egyet → ez lesz az új közelítő érték…
- gradiens eljárás keresi az egyenesünket

perceptron gradiens:

$$J_p(\underline{a}) = -\sum_{y \in Y} \underline{a}^t \cdot \underline{y}$$

Y – rosszul osztályozott pontok halmaza

- találjunk egy olyan egyenest, hogy az összes pont a (+) oldalon van!
- ha veszünk egy rosszul osztályozott pontot, akkor negatív érték lesz
- negatív tagok mínuszaiból (+) értéket kapunk → minimuma 0 lesz
- akkor kapunk szélsőértéket, ha nincs rosszul osztályozott pontunk
- mennyi ennek a függvénynek a *gradiense*? Jel.: ∇
- vennünk kell a parciális deriváltakat (a szerint)

$$J_p(\underline{a}) = -\sum_{y \in Y} \underline{a}^t \cdot \underline{y}$$
$$\nabla J_p(\underline{a}) = -\sum_{y \in Y} \underline{y}$$

általános gradiens eljárás:

$$\underline{a}(k+1) = a(k) - \eta(k) \cdot \nabla Jp(\underline{a})$$

$\underline{a}(k+1)$ - a keresett vektor következő közelítése

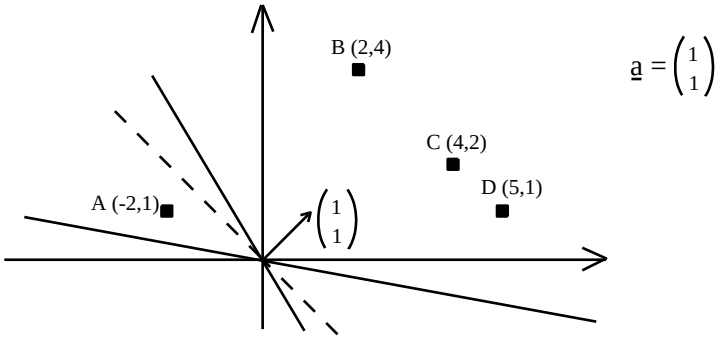
$\eta(k)$ - (skálázó) konstans – a gradiens irányába lépünk valamennyit

addig folytatjuk, míg megkapjuk a jó egyenest!

az általános felírás ebben a konkrét esetben:

$$= a(k) + \eta(k) \cdot \sum_{y \in Y} y$$

Példa:



$\underline{a} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$

- $\eta(k) = 1$ (a konstansok k-tó) függetlenül mindig 1-ek

$$\underline{a}^t \cdot y > 0 \qquad (\underline{a}^t \text{ – vektor , } y \text{ - pontok})$$

- ebbe a képletbe kell behelyettesíteni – ahol nem áll fönt a reláció (azaz nem nagyobb, mint 0), azok a pontok rossz pontok

tehát:

A (-2,1) $\qquad \eta(k) = 1$

B (2,4)

C (4,2) $\qquad a_0 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$

D (5,1)

1) $a_0^t \cdot y > 0$

A pontra $(1, 1) \cdot \begin{pmatrix} -2 \\ 1 \end{pmatrix} = -2 + 1 = -1 \not> 0$ **A rossz pont**

B pontra $(1, 1) \cdot \begin{pmatrix} 2 \\ 4 \end{pmatrix} = 2 + 4 = 6 > 0$ **B pont megfelel**

C pontra $(1, 1) \cdot \begin{pmatrix} 4 \\ 2 \end{pmatrix} = 4 + 2 = 6 > 0$ **C pont megfelel**

D pontra $(1, 1) \cdot \begin{pmatrix} 5 \\ 1 \end{pmatrix} = 5 + 1 = 6 > 0$ **D pont megfelel**

a gradiens: $a(1) = \begin{pmatrix} 1 \\ 1 \end{pmatrix} + 1 \cdot \underbrace{\begin{pmatrix} -2 \\ 1 \end{pmatrix}}_{\text{A rossz pont}} = \begin{pmatrix} -1 \\ 2 \end{pmatrix}$

$$a_1 = \begin{pmatrix} -1 \\ 2 \end{pmatrix} \qquad a_1^t = (-1, 2)$$

2) $a_1^t \cdot y > 0$

A pontra $(-1, 2) \cdot \begin{pmatrix} -2 \\ 1 \end{pmatrix} = 2 + 2 = 4 > 0$ **A pont megfelel**

B pontra $(-1, 2) \cdot \begin{pmatrix} 2 \\ 4 \end{pmatrix} = -2 + 8 = 6 > 0$ **B pont megfelel**

C pontra $(-1, 2) \cdot \begin{pmatrix} 4 \\ 2 \end{pmatrix} = -4 + 4 = 0 \not> 0$ **C rossz pont**

D pontra $(-1, 2) \cdot \begin{pmatrix} 5 \\ 1 \end{pmatrix} = -5 + 2 = -3 \not> 0$ **D rossz pont**

$$a \text{ gradiens}: \underbrace{a(2)}_{a(k+1)} = \underbrace{\begin{pmatrix} -1 \\ 2 \end{pmatrix}}_{a(k)} + 1 \cdot \underbrace{\begin{pmatrix} 9 \\ 3 \end{pmatrix}}_{\Sigma \text{ rossz pontok } (C,\, D)} = \begin{pmatrix} 8 \\ 5 \end{pmatrix}$$

$$a_2 = \begin{pmatrix} 8 \\ 5 \end{pmatrix} \qquad a_2^T = (8,\, 5)$$

rossz pont: A

$$a(3) = \begin{pmatrix} 6 \\ 6 \end{pmatrix} \qquad a_3 = \begin{pmatrix} 6 \\ 6 \end{pmatrix} \qquad a_3^T = (6,\, 6)$$

rossz pont: A

$$a(4) = \begin{pmatrix} 4 \\ 7 \end{pmatrix} \qquad a_4 = \begin{pmatrix} 4 \\ 7 \end{pmatrix} \qquad a_4^T = (4,\, 7)$$

rossz pont: A

$$a(5) = \begin{pmatrix} 2 \\ 8 \end{pmatrix} \qquad a_5 = \begin{pmatrix} 2 \\ 8 \end{pmatrix} \qquad a_5^T = (2,\, 8)$$

nincs rossz pont!

- tehát, ha a (2,8) pontba mutató vektor normálisát (rá merőleges egyenesét) vesszük, akkor kapunk egy megoldás egyenest!

Példa: (az előző példa, csak itt $\eta(k) = 0.1$ azaz sokkal finomabb a lépésköz, amivel toljuk majd a megfelelő irányba az egyenesünket – közelítve folyamatosan a megoldáshoz)

$$a(1) = \begin{pmatrix} 1 \\ 1 \end{pmatrix} + 0.1 \cdot \begin{pmatrix} -2 \\ 1 \end{pmatrix} = \begin{pmatrix} 0.8 \\ 1.1 \end{pmatrix}$$

$$a(2) = \begin{pmatrix} 0.6 \\ 1.2 \end{pmatrix}$$

$$a(3) = \begin{pmatrix} 0.4 \\ 1.3 \end{pmatrix}$$

- apró lépésenként mozgatja lefelé a rossz pont(ok) irányába az egyenest
- ha szeparálhatók a pontok, akkor jó ez az eljárás
- ha, nem szeparálhatók, akkor nincs minimum ☹

egy lépéses eljárás (egyszerűsített eljárás)

- adottak a pontok (*A, B, C, D*)
- ismételjük a pontokat ciklikusan
- menjünk végig rajtuk
- ha a vizsgált pont jó, akkor menjünk tovább
- ha rossz, akkor azt az egy pontot módosítsuk, majd lépjünk tovább
- akkor áll meg az eljárás, ha már mind a négy pont jó

A	*B*	*C*	*D*	*A*	*B*	*C*	*D*	*A*	...
rossz	*jó*	*rossz*	*jó*	*rossz*	***jó***	***jó***	***jó***	***jó***	
$a(1)=\begin{pmatrix}-1\\2\end{pmatrix}$		$a(2)=\begin{pmatrix}3\\4\end{pmatrix}$		$a(3)=\begin{pmatrix}1\\5\end{pmatrix}$					

- tehát *a* (3)-ra nézve jó lesz mind a négy pont
- 3 lépésben megkaptuk tehát a jó megoldást ezzel a módszerrel

$$\underline{a}(k+1)=\underline{a}(k)+\underbrace{\eta(k)}_{1}\cdot\sum_{y\in Y}y=\underline{a}(k)+\underline{y}^k$$

$\underline{y}^k$ adott pillanatban egy rossz pont koordinátái

- konvergens, ha szeparálható ponthalmazunk van
- thf. szeparálható ponthalmazunk van

$$\left\|\underline{a}_{k+1}-\alpha\cdot\underline{\hat{a}}\right\|^2=\left\|\underline{a}(k)-\alpha\cdot\underline{\hat{a}}+\underline{y}^k\right\|^2=(*)$$

$\underline{\hat{a}}$ – a feladat egy m.a. – a

(a kettő különbségének normáját becsülöm
az optimálistól mennyivel tér el az adott – ezt becsüljük)

$$(*)=\left\|\underline{a}(k)-\alpha\cdot\underline{\hat{a}}\right\|^2+2\left(\underline{a}(k)-\alpha\cdot\underline{\hat{a}}\right)^t\underline{y}^k+\left|y^k\right|^2$$

y rosszul osztályozott pont

$y \cdot a$ negatív lesz – ha ezt elhagyjuk, akkor

$$\leq \left|\underline{a}(k) - \alpha \cdot \underline{\hat{a}}\right| - 2 \cdot \alpha \cdot \underline{\hat{a}} \cdot \underline{y}^k + \left|\underline{y}^k\right|^2$$

igaz az állítás, ha az α-t elég nagyra választom, ekkor

$$2 \cdot \alpha \cdot \underline{\hat{a}} \cdot \underline{y}^k < 2 \cdot \left|\underline{y}^k\right|^2$$

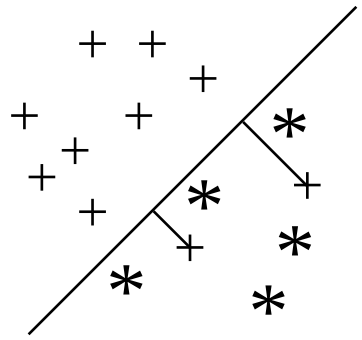
teljesül, ezért

$$\leq \left|\underline{a}(k) - \alpha \cdot \underline{\hat{a}}\right| - \left|\underline{y}^k\right|^2$$

<u>nem elválasztható eset</u>

- nem szeparálható a pontok halmaza
- <u>módszerek</u>:

I. *legkisebb négyzetek módszere:*



$d_1^2 + d_2^2$ - a minimális legyen

azaz húzzunk egy olyan egyenest, amelynél a rossz oldalra eső pontok négyzetösszege (egyenestől való távolsága) minimális

(ez lehetne egy jó módszer, de mi most nem ezt választjuk)

$$\underbrace{\begin{pmatrix} y_{10} & y_{11} & \dots & y_{1d} \\ y_{20} & y_{21} & \dots & y_{2d} \\ \vdots & \vdots & & \vdots \\ y_{n0} & y_{n1} & \dots & y_{nd} \end{pmatrix}}_{Y} \cdot \underbrace{\begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_d \end{pmatrix}}_{\underline{a}} = \underbrace{\begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}}_{\underline{b}} \qquad b_i > 0$$

b komponensek a pontok egyeneshez való helyzetét jelentik

$\| Ya - b \|^2$ *gradiens fgv.*

(Dombi József előadásai)

Irodalom: Fekete István – Nagy Sára: Bevezetés a mesterséges intelligenciába

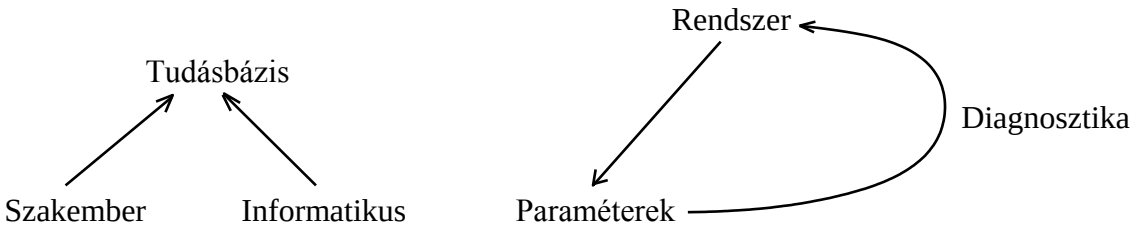
147. old. – 193. old. → előadás anyag

195. oldaltól → gyakorlat anyaga

- logika = intelligencia
- olyan programokat készíteni, ami nem algoritmusokra, hanem a logikára épül.
  - Prolog (elterjedtebb)
  - LISP

  ezek gyakorlati alkalmazásából *szakértői rendszereket* hoznak létre → javítások, szolgáltatások segítése

  pl.: egy programot kap a szerelő, hogy mit kell tennie, ha pl. nincs kep, vagy nincs hang…
- *diagnózis*
- hogyan alkossunk diagnózist, hogyan tudunk következtetni?



- A szakértői rendszert úgy képzelték el, hogy szinte mindenki meg tudná otthon javítani az elromlott berendezést (amihez eddig szerelő kellett). Vagy pl. otthon elvégezhet az ember magán bizonyos vizsgálatokat - vér, vizelet,… - ezt beadják a *szakértői rendszernek* → az diagnosztizál (nincs felelőssége az orvosnak))
- mindehhez hatalmas *tudásbázis* kellene → sok idő és munka, míg feltöltik
- MI-nél a programok egyfajta keresőeljárások – maguk találják meg a megoldást
- nem direkt program – a következtetési rendszert generálja le
- a számítógép architektúráját megváltoztatni, hogy pl. a Prolog nyelvre is optimális legyen → szoftver rendszer is kell
- ’90-es évek végén leálltak a próbálkozással, hogy elterjesszék a szakértői rendszereket
- nagy szükség lett volna rájuk, de nem lehetett jól megoldani
- alig maradtak fenn ilyen rendszerek (OPS5)

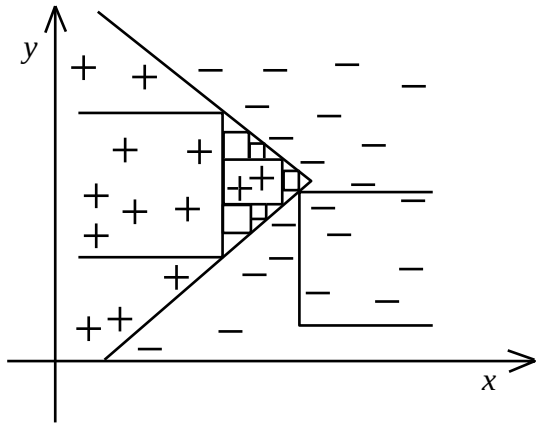
(a logikának a mindenhatóságába vetett hiten alapult, de nem mindig a logikus lépés a jó!)

Példa:

- érzékszervek alapján fogjuk fel a világot:
- szembe jutó fény intenzitása – x_1; hang frekvenciája – x_2; … ; x_n
- (x_1, x_2, …, x_n) – vektor → ezt kell feldolgozni
- digitalizálva lehet csak ezt megtenni → ekkor tud a logika dolgozni vele

  $x \in A_1$ (az $\in$ reláció már digitális)

  pl.: ($x \in A_1 \, y \in B_2$)



ha csak 2 értékünk van -- két dim.
2 esemény között dönteni

\+ −

pl.: paradicsom vásárlása esetén
a ***színt*** és a ***súlyt*** nézem

- logika szerint az *x* és az *y* tengelyt is intervallumokra bontom
- ha teljes pontossággal akarom ilyen téglalapokkal lefedni → végtelen sok kellene
- sok szabály szükséges ahhoz, hogy legalább nagyjából pontos legyen
- ezek a tartományok n dimenzióban nagyon bonyolultak lehetnek
- az agyunk képes ilyen bonyolult hipersíkokat vizsgálni!

Tételbizonyítás az ítéletkalkulusban

(Készítsünk automatikus eljárást!)

zérusrendű logika = ítéletkalkulus

- ítéletváltozók
- relációk

elsőrendű logika: (új elemei:)

- függvények
- kvantorok (∃,∀)

moduláris logika:

- szükségszerű,
- lehetséges szó használható

következtetési szabályok: (példa)

premissza: van egy dollárom

Ha van ennyi pénzem, vehetnék rajta colat.

Ha van ennyi pénzem, vehetnék rajta hamburgert.

a rendszer azt mondaná, hogy akkor van colam, és van hamburgerem is (mert meg tudom venni a colat, és meg tudom venni a hamburgert is) – persze ez nem így van, de honnan tudhatná ezt a logika

→ időben változó logika = *temporális logika* (ez már tudja a fenti problémát kezelni)

(mi az ítéletkalkulussal és az elsőrendű logikával foglalkozunk a továbbiakban)

Feladat:

Lássuk be, hogy az A_1, A_2 és A_3 állításokból logikailag következik az A_4 állítás!

A_1: Ha süt a nap, akkor Péter strandra megy.

A_2: Ha Péter strandra megy, akkor úszik.

A_3: Péternek nincs lehetősége otthon úszni

A_4: Ha süt a nap, akkor Péter nem marad otthon.

A_3: a szakértői rendszer problematikáját is tükrözi.

Tény következményeit a szakértői rendszernek ki kell vizsgálnia.

Például:

tény: Mária a bevásárlóközpontban beverte a fejét és az vérzett.

1. következtetés: Mária aznap a boltba magával vitte a fejét.

Nem tudja a gép a számunkra kézenfekvő dolgokat (hogy a fejünk mindig velünk van ☺)

Ezekkel a triviális dolgokkal is fel kell tölteni a rendszert.

Feladat (folyt.):

Bizonyítandó: „Ha A_1 és A_2 és A_3 , akkor A_4.”

(erre egy programot, eljárást kellene készíteni)

Szintaxis:

1) *elválasztó jelek*: ; ( ) [ ] { }
2) *logikai műveletek*: $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow(\equiv)$
3) *ítéletváltozók*: p, q, r
4) *ítéletkonstansok*: igaz (T), hamis (F)

Formulák: (iteratívan hat. meg)

atomi formulák: (atom)

- ítéletváltozók
- ítéletkonstansok

formulák: (jólformált formula)

- atomi formula
- ha A és B formulák, akkor a

  $(\neg A)$, $(A \wedge B)$, $(A \vee B)$, $(A \rightarrow B)$, $(A \leftrightarrow B)$

  kifejezések is formulák

zárójelek elhagyása → *precedencia sorrend* alapján: $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow(\equiv)$

pl.: $((\neg p) \wedge q) \equiv r$

$\neg p \wedge q \equiv r$

Feladat (folyt.):

p: süt a nap

q: Péter strandra megy

r: Péter úszik

s: Péter otthon marad

Ekkor az $A_1 – A_4$ állításoknak rendre az alábbi $F_1 – F_4$ formulákat feleltetjük meg:

F_1: $p \rightarrow q$

F_2: $p \rightarrow r$

F_3: $\neg(s \wedge r)$

F_4: $p \rightarrow \neg s$

Szemantika:

(a formulákhoz a szemantika szabályai szerint adhatunk jelentést)

1. lépés: formula *interpretálása*
2. lépés: formula *kiértékelése*

interpretálás: a formula minden egyes ítéletváltozójához ítéletkonstansokat rendelünk minden lehetséges módon (a formula egy interpretációja egy lehetséges hozzárendelést jelent).

pl.: I_1: (p, q, r, s) ← (T, T, F, F)

I_2: (p, q, r, s) ← (F, T, T, F)

kiértékelés: az interpretált formula kiértékelését a műveleti jelek szemantikája (*igazságtábla*) alapján végezzük. (a kiértékelés interpretáció függő – a műveleti jelek szemantikája alapján)

A kielégíthetőségi tulajdonság

Kielégíthető egy formula, ha valamely interpretációban igaz.

Érvényes egy formula, ha minden interpretációban igaz (*tautológia*).

Kielégíthetetlen egy formula, ha minden interpretációban hamis a logikai értéke.

Tétel: Egy formula akkor és csak akkor érvényes, ha a negáltja kielégíthetetlen.

Tétel: Minden érvényes formula kielégíthető.

Formulák ekvivalenciája

Két formula ekvivalens, ha minden interpretációban ugyanaz a logikai értékük.

1. $A \equiv B = (A \rightarrow B) \wedge (B \rightarrow A)$
2. $A \rightarrow B = \neg A \vee B$
3.a. $A \vee B = B \vee A$ (kommutativitás)
3.b. $A \wedge B = B \wedge A$
4.a. $(A \vee B) \vee C = A \vee (B \vee C)$ (asszociativitás –
4.b. $(A \wedge B) \wedge C = A \wedge (B \wedge C)$ a művelet kiterjesztése szempontjából fontos)
5.a. $A \vee (B \wedge C) = (A \vee B) \wedge (A \vee C)$ (disztributivitás)
5.b. $A \wedge (B \vee C) = (A \wedge B) \vee (A \wedge C)$
6.a. $A \vee F = A$
6.b. $A \wedge T = A$
7.a. $A \vee T = T$
7.b. $A \wedge F = F$
8.a. $A \vee \neg A = T$
8.b. $A \wedge \neg A = F$
9. $\neg(\neg A) = A$
10.a. $\neg(A \vee B) = \neg A \wedge \neg B$ (De-Morgan)
10.b. $\neg(A \wedge B) = \neg A \vee \neg B$

(nem asszociatív a számtani közép, ha több argumentum van)

(a szorzás nem kommutatív a patikusoknál ☺ $2\times5 \neq 5\times2$)

Tétel: Az A és a B formulák akkor és csak akkor ekvivalensek, ha az $A \equiv B$ formula érvényes.

A logikai következmény

W formula *logikailag következik* az $A_1, \ldots, A_n$ formulákból (másként: ezeknek logikai következménye), ha minden olyan interpretációban, amelyben $A_1, \ldots, A_n$ mindegyike igaz, igaz a W formula is.

Tétel: A W formula akkor és csak akkor következik logikailag az $A_1, \ldots, A_n$ formulákból, ha az $(A_1 \wedge \ldots \wedge An) \rightarrow W$ formula érvényes.

Bizonyítás: (könyv alapján)

Ha W logikailag következik $A_1, \ldots, A_n$-ből, akkor definíció szerint minden olyan interpretációban, amelyben $A_1 \wedge \ldots \wedge An$ igaz, W is igaz.

Ekkor az $(A_1 \wedge \ldots \wedge An) \rightarrow W$ implikáció minden interpretációban igaz, hiszen éppen azt az egyetlen hamis igazságértékű esetet kizártuk, amikor az előtag igaz és az utótag hamis.

Megfordítva, ha $(A_1 \wedge \ldots \wedge An) \rightarrow W$ minden interpretációban igaz, akkor az implikáció igazságtáblája szerint nem valósulhat meg az az eset, hogy $A_1 \wedge \ldots \wedge An$ igaz és W hamis, azaz ha $A_1 \wedge \ldots \wedge An$ igaz, akkor W is igaz értékű – és ez éppen a logikai következmény definíciójával egyezik meg.

Tétel: A W formula akkor és csak akkor logikai következménye az $A_1 \wedge \ldots \wedge An$ formulának, ha az $A_1 \wedge \ldots \wedge An \wedge \neg W$ formula kielégíthetetlen.

Bizonyítás:

$$\neg [(A_1 \wedge \ldots \wedge An) \rightarrow W] = \neg[\neg (A_1 \wedge \ldots \wedge An) \vee W]$$
$$= (A_1 \wedge \ldots \wedge An) \wedge \neg W$$
$$= A_1 \wedge \ldots \wedge An \wedge \neg W$$

A bizonyítandó $(A_1 \wedge \ldots \wedge An) \rightarrow W$ formulát *tételnek* nevezzük.

Az $A_1 \wedge \ldots \wedge An$ kifejezések a tétel *axiomái* vagy *feltételei*, *premisszái*.

A W formula pedig a *következmény* vagy *konklúzió*.

A tételbizonyítás néhány módszere

A tételbizonyítás egy $(A_1 \wedge \ldots \wedge An) \rightarrow W$ típusú formula *érvényességének* belátását kívánja.

$A_1 \wedge \ldots \wedge An$ – *feltételeknek*, *premisszáknak*, vagy *axiomáknak* nevezzük.

W – *következménynek, konklúziónak, célállításnak* hívjuk.

A bizonyítandó $(A_1 \wedge \ldots \wedge An) \rightarrow W$ formulát *tételnek* nevezzük.

1) Tételbizonyítás igazságtáblával

Bizonyítsuk be, hogy az F_1 és F_2 formulákból logikailag következik az F_3 formula, ahol

F_1: $p \rightarrow q$

F_2: $\neg q$

F_3: $\neg p$

Tételbizonyítás formái:

a) minden interpretációban, amelyben $F_1 \wedge F_2$ igaz, igaz F_3 is (teljesül-e)

azaz $(p \rightarrow q) \wedge \neg q$ igaz, akkor igaz $\neg p$ is?

b) $F_1 \wedge F_2 \rightarrow F_3$ érvényes (teljesül-e)

azaz $((p \rightarrow q) \wedge \neg q) \rightarrow \neg p$ formula érvényes-e

c) $F_1 \wedge F_2 \wedge \neg F_3$ formula kielégíthetetlen, azaz minden interpretációban hamis (teljesül?)

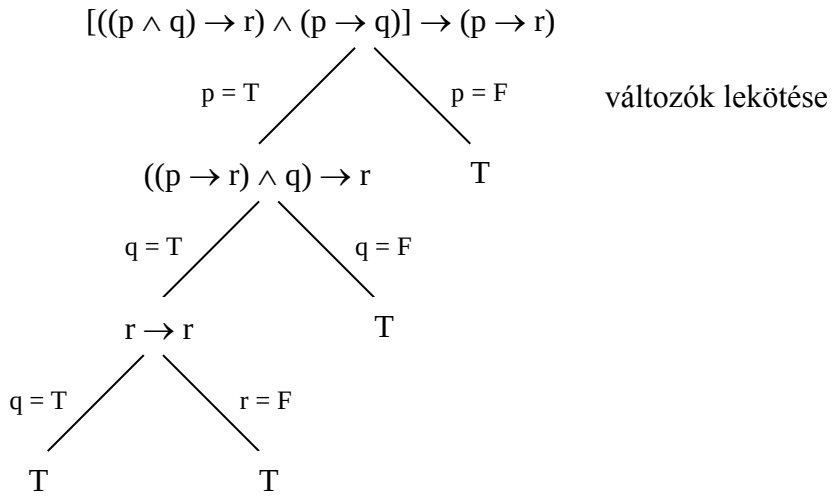
$(p \rightarrow q) \wedge \neg q \wedge p$ formula kielégíthetetlen?

2) Quine algoritmusa

Minden lépésben a formula egy változóját interpretáljuk a logikai igaz és a logikai hamis érték hozzárendelésével. Így két újabb formulához jutunk. → *bináris fa*.

Addig folytatjuk, amíg az így kialakult bináris fa *minden levelén* egy-egy igazságérték található.

Bizonyítsuk be, hogy $[((p \wedge q) \rightarrow r) \wedge (p \rightarrow q)] \rightarrow (p \rightarrow r)$ formula érvényes.



A fa minden levelén a T jelenik meg (minden interpretációra igaz) → a kiindulási formula *érvényes* formula

3) Wang algoritmusa

Implikációt itt nem szabad kiküszöbölni, sőt ha nincs, akkor be kell csempészni.

$$A_1 \wedge \ldots \wedge A_m \rightarrow B_1 \vee \ldots \vee B_n$$

Ilyen alakú formulákon dolgozunk.

($p \wedge q$, $p \vee q$ ezek könnyen kiértékelhetők)

- Ha ugyanaz a formula előfordul az implikáció jel mindkét oldalán (azaz valamelyik A_i és B_j megegyezik), akkor a teljes formula *érvényes.*
- Ha a formulában csupa ítéletváltozó van, de egyik sem szerepel a másik oldalon, akkor a formula *nem érvényes*.

Hogy hozzunk ilyen alakra?

Átalakítási szabályok:

1. De Morgan szabályok segítségével redukáljuk a negáció hatáskörét.
2. Az ekvivalenciát kiküszöböljük.
3. Ha valamelyik oldalon van implikáció, azt kiküszöböljük.
4. Bármelyik oldalról a $\neg X$ átvihető a másik oldalra X-ként.
5. Ha $p \vee q$ a rossz oldalon szerepel, akkor 2 formulára bontjuk

   pl. $A_1 \wedge \mathbf{A_2} \wedge \ldots \wedge A_m \rightarrow B_1 \vee B_2 \vee \ldots \vee B_n$

   (ha az A_2 $\mathbf{(p \vee q)}$ alakú, akkor a következőkepp bontjuk két részre)

   $A_1 \wedge \mathbf{p} \wedge \ldots \wedge A_m \rightarrow B_1 \vee B_1 \vee \ldots \vee B_n$

   $A_1 \wedge \mathbf{q} \wedge \ldots \wedge A_m \rightarrow B_1 \vee B_1 \vee \ldots \vee B_n$
6. Ha $p \wedge q$ a rossz oldalon szerepel, akkor 2 formulára bontjuk

   pl. $A_1 \wedge A_2 \wedge \ldots \wedge A_m \rightarrow B_1 \vee \mathbf{B_2} \vee \ldots \vee B_n$

   (ha az B_2 $\mathbf{(p \wedge q)}$ alakú, akkor a következőkepp bontjuk két részre)

   $A_1 \wedge A_2 \wedge \ldots \wedge A_m \rightarrow B_1 \vee \mathbf{p} \vee \ldots \vee B_n$

   $A_1 \wedge A_2 \wedge \ldots \wedge A_m \rightarrow B_1 \vee \mathbf{q} \vee \ldots \vee B_n$

Példa:

$$[(p \rightarrow q) \wedge (p \rightarrow r) \wedge \neg r] \rightarrow \neg p$$

(3. szabállyal – implikáció kiküszöbölése)

$$[(\neg p \vee q) \wedge (\neg p \vee r) \wedge \neg r] \rightarrow \neg p$$

(4. szabály értelmében – vigyük át a $\neg r$-et a bal oldalról a jobb oldalra r-ként)

$$[(\neg p \vee q) \wedge (\neg p \vee r)] \rightarrow (\neg p \vee r)$$

(5. szabály a konjunkció első tagjára)

$$[\neg p \wedge (\neg q \vee r)] \rightarrow (\neg p \vee r)$$

$$[q \wedge (\neg q \vee r)] \rightarrow (\neg p \vee r)$$

(ismét az 5. szabályt alkalmazása után az egyik formula) (*)

$$(q \wedge \neg q) \rightarrow (\neg p \vee r)$$

(vigyük át ¬q-t a jobb oldalra q-ként)

$$q \rightarrow (\neg p \vee r \vee q)$$

ez a formula érvényes

(* lépés után keletkezett másik formula)

$$(q \wedge r) \rightarrow (\neg p \vee r)$$

$$q \rightarrow \neg p$$

ez a formula is érvényes

4) Formális levezetés

A tételbizonyításnak kétféle megközelítése van. Az egyik a formula érvényességét bizonyítja (igazságtábla módszere), ahol a formula eredeti alakját értékeljük ki.

A másik megközelítés ezzel szemben nem a bizonyítandó formula szemantikáját vizsgálja, hanem megpróbálja azt axiómák egy rögzített halmazából levezetni azt.

Axiómák: (Russel)

1. $p \vee p \rightarrow p$
2. $p \rightarrow p \vee q$
3. $p \vee q \rightarrow q \vee p$
4. $p \vee (q \vee r) \rightarrow q \vee (p \vee r)$
5. $(q \rightarrow r) \rightarrow ((p \vee q) \rightarrow (p \vee r))$

Levezetési szabályok:

1. Szabály: *Helyettesítési szabály*
   Egy tétel bármely változójának minden előfordulását ugyanazzal a kifejezéssel helyettesítve újabb tételt kapunk.
2. Szabály: Ha $A \rightarrow B$ tétel (vagy axióma), és B-t akarjuk tételként belátni, akkor ehhez elegendő A-t bizonyítani.

3. Szabály: a) *Előre láncolás*

Ha $A \rightarrow B$ tétel, és $A \rightarrow C$ a bizonyítandó, akkor elegendő azt belátni, hogy $B \rightarrow C$ tétel.

b) *Visszafelé láncolás*

Ha $B \rightarrow C$ tétel, és $A \rightarrow C$ a bizonyítandó, akkor elegendő azt belátni, hogy $A \rightarrow B$ tétel.

Példa:

Bizonyítsuk be, hogy $(p \rightarrow \neg p) \rightarrow \neg p$

(1. axióma, 1-es szabály → helyettesítsük p-t $\neg p$-vel)

$\neg p \vee p \rightarrow \neg p$

(implikáció definícióját tartalmazó logikai törvény miatt)

$(p \rightarrow \neg p) \rightarrow \neg p$

(ez pedig a bizonyítandó tétel)

De ez a módszer nagyon hosszú lehet – széles a fa.

REZOLÚCIÓ

A konjunktív normálforma (KNF)

A rezolúció alkalmazásához a formulát először KNF alakra kell hozni.

A *konjunktív normálforma* klózok (speciális részformulák) konjunkciója.

Egy *klóz* literálok diszjunkciója, illetve egy literál is klózt alkot.

Literálnak egy ítéletváltozót, vagy annak negáltját nevezzük.

Példa: $(p \vee q) \wedge (r \vee \neg p) \wedge s$

három klózt tartalmaz ez a KNF, és négy literált (p, q, r, s)

Tétel: Minden ítéletkalkulusbeli formula KNF alakra hozható.

Transzformációs szabályok:

1. Ekvivalencia kiküszöbölése.
2. Implikáció kiküszöbölése.
3. Negáció hatáskörének csökkentése – De Morgan szabályok segítségével
4. Disztributivitás → klózok konjunkciójának létrehozása

   $A \vee (B \wedge C) = (A \vee B) \wedge (A \vee C)$

   (ez utóbbi épp két klóz konjunkciója)

Példa:

$$\neg\{[(p \to q) \wedge (q \to r) \wedge \neg(s \wedge r)] \to (p \to \neg s)\}$$

$$\neg\{\neg\,[(p \to q) \wedge (q \to r) \wedge \neg(s \wedge r)] \vee (p \to \neg s)\} \quad (2)$$

$$\neg\{\neg\,[(\neg p \vee q) \wedge (\neg q \vee r) \wedge (\neg s \vee \neg r)] \vee (\neg p \vee \neg s)\} \qquad (2,4)$$

$$\{[(\neg p \vee q) \wedge (\neg q \vee r) \wedge (\neg s \vee \neg r)] \wedge (p \wedge s)\} \qquad (3,4)$$

(a felesleges zárójelek elhagyása után)

$$(\neg p \vee q) \wedge (\neg q \vee r) \wedge (\neg s \vee \neg r) \wedge p \wedge s$$

A rezolúciót a formula *klóz alakjára* alkalmazzuk. Ezt úgy kapjuk meg, hogy a konjunktív normálformából elhagyjuk a $\wedge$ műveleti jeleket, és a formulát *klózok halmazának* tekintjük. Azt kell belátnunk, hogy ez a klóz halmaz kielégíthetetlen!

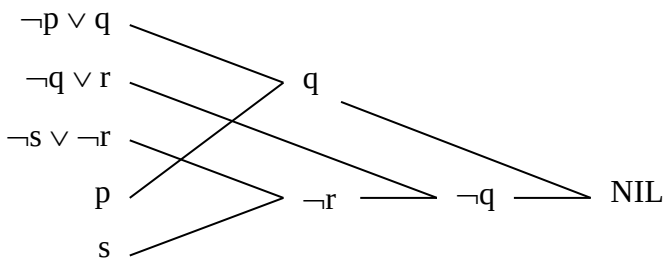
C_1: $\neg p \vee q$

C_2: $\neg q \vee r$

C_3: $\neg s \vee \neg r$

C_4: p

C_5: s



Olyan klózpárokat keresünk, hogy azokban komplemens literálpár szerepeljen, egészen addig, amíg ellentmondásra nem jutunk (NIL – üres klóz)

Rezolúció egy lépése:

Legyen $p \vee A$, $\neg p \vee B$ klózok

Ezek a klózok *rezolválhatók,* mert *egyetlen komplemens literálpárt* tartalmaznak (p és ¬p)

→ $A \vee B$ a rezolvense, ami szintén klóz.

Tétel: A $(p \vee A) \wedge (\neg p \vee B)$ és a $(p \vee A) \wedge (\neg p \vee B) \wedge (A \vee B)$ formulák ekvivalensek.

Biz.:

1) Lemma: Ha $C \to D$ érvényes, akkor $C \equiv C \wedge D$ is érvényes

Biz.: ($C \equiv C \wedge D$ formula mindhárom esetben igaz, tehát érvényes.)

C	D	$C \to D$	$C \wedge D$	$C \equiv C \wedge D$
T	T	T	T	T
F	T	T	F	T
F	F	T	F	T

2) Lemma: $(p \vee A) \wedge (\neg p \vee B) \to (A \vee B)$ érvényes.

Biz.: (indirekt)

Tegyük fel, hogy hamis → ez csak akkor lehet, ha az előtag igaz, az utótag pedig hamis.

Az előtag pedig csak akkor lehet igaz, ha A és B legalább egyike igaz, ekkor azonban $A \vee B$ is igaz. Ez ellent mond a feltételünknek.

(fő tétel bizonyítása)

Alkalmazzuk az 1. Lemmát a 2. Lemmából adódó $C = (p \vee A) \wedge (\neg p \vee B)$ és $D = A \vee B$ helyettesítéssel. Ekkor

$$(p \vee A) \wedge (\neg p \vee B) \equiv (p \vee A) \wedge (\neg p \vee B) \wedge (A \vee B)$$

formula érvényes.

Megjegyzések a rezolúcióhoz:

- KNF
- listába a klózokat
- könnyen programozható
- klózból klózt csinálunk, amíg az üres klózhoz el nem jutunk
- de az eljárás nemdeterminisztikus – túl hosszú
- *teljes*, mert mindig bebizonyítható
- minden számítógépes eljárás ezt használja

Rezolúció az elsőrendű predikátumkalkulusban

Példa: Lássuk be, hogy $F_1 \wedge F_2 \to F_3$

F_1: Van olyan páciens, aki minden doktorban megbízik.

F_2: A kuruzslókban egyetlen páciens sem bízik meg.

F_3: Egyetlen doktor sem kuruzsló.

Szintaxis:

1) *elválasztó jelek*: ; ( ) [ ] { }
2) *logikai műveletek*: $\neg$, $\wedge$, $\vee$, $\to$, $\leftrightarrow(\equiv)$

3) *kvantorok*: $\exists$, $\forall$

4) *elemváltozók*: x, y, z

5) *elemkonstansok:* a, b, c

6) *függvényszimbólumok:* f, g, h (több argumentuma is lehet)

7) *ítéletváltozók*: p, q, r

8) *ítéletkonstansok*: igaz (T), hamis (F)

9) *predikátumszimbólumok*: P, Q, R

Formulák:

term:

- minden elemkonstans term
- minden elemváltozó term
- ha f n-argumentumú függvényszimbólum és $t_1, \ldots, t_n$ termek, akkor $f(t_1, \ldots, t_n)$ is term

atomi formulák:

- minden ítéletkonstans atomi formula
- minden ítéletváltozó atomi formula
- ha P n-argumentumú predikátumszimbólum és $t_1, \ldots, t_n$ termek, akkor $P(t_1, \ldots, t_n)$ is atomi formula

formulák: (jólformált formula)

- minden atomi formula egyben formula is
- ha A és B formulák, akkor a

  $(\neg A)$, $(A \wedge B)$, $(A \vee B)$, $(A \rightarrow B)$, $(A \leftrightarrow B)$

  kifejezések is formulák
- ha az A egy formula és x egy változó, akkor a

  $(\forall x)A$, $(\exists x)A$

  kifejezések is formulák

zárójelek elhagyása → *precedencia sorrend* alapján: $\forall$, $\exists$, $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow(\equiv)$

- ($\forall$x)-ben szereplő x – *univerzálisan kvantált*
- ($\exists$x)-ben szereplő x – *egzisztenciálisan kvantált*
- azok a változók, amelyek nincsenek kvantorral lekötve *szabad változók*
- különben *kötött változók*
- azt a formulát, amelyben minden változó kötött, *mondatnak* nevezzük.

Példa: (folyt.)

P(x): x egy páciens

D(x): y egy doktor

K(y): y egy kuruzsló

M(x,y): x megbízik y-ban

Ekkor a formalizált állítások a következők:

F1: $\exists x\,[P(x) \wedge \forall y\,(D(y) \rightarrow M(x,y))]$

(van olyan páciens, aki minden doktorban megbízik)

F2: $\forall x\,[P(x) \rightarrow \forall y\,(K(y) \rightarrow \neg M(x,y))]$

(a kuruzslókban egyetlen páciens sem bízik meg)

F3: $\forall y\,(D(y) \rightarrow \neg K(y))$

(egyetlen doktor sem kuruzsló)

Szemantika:

1. lépés: formula *interpretálása*
2. lépés: formula *kiértékelése*

– Az *interpretáció* során először megválasztjuk az értelmezés *alaphalmazát, individumtartomá-nyát.* (jelöljük ezt D-vel) D megválasztása után a formula egyes szimbólumaihoz a következő hozzárendeléseket végezzük el:

1) minden konstansszimbólumnak feleltessünk meg D egy elemét

2) minden n-argumentumú függvényszimbólumhoz rendeljünk hozzá egy $D^n \rightarrow D$ leképezést

3) minden n-argumentumú predikátumszimbólumnak feleltessünk meg egy $D^n \rightarrow \{T,F\}$ leképezést

– *kiértékelés*:

1) ha A és B logikai formula igazságértéke ismert, akkor a

$(\neg A)$, $(A \wedge B)$, $(A \vee B)$, $(A \rightarrow B)$, $(A \leftrightarrow B)$

formulák igazságértéke is ismert (igazságtábla)

2) $(\forall x)A$ – ki tudom értékelni

3) $(\exists x)A$ – ki tudom értékelni

Példa: (interpretációra)

$(\forall x)\,[P(f(x,x),b) \rightarrow P(x,b)]$

D tartomány I_1: természetes számok halmaza

I_2: egész számok halmaza

Mindkét esetben:

- legyen b = 1
- f(x,y) = xy (azaz f(x,y) rendelje x-hez és y-hoz az xy szorzatot)
- P(x,y) = x = y (azaz P legyen az egyenlőség relációja)

$(\forall x)\ [(x^2 = 1) \rightarrow (x = 1)]$

Ami I_1-re igaz

I_2-re hamis

A kielégíthetőségi tulajdonság

Kielégíthető egy formula, ha létezik olyan interpretáció, amelyben igaz a formula.

Érvényes egy formula, ha minden interpretációban igaz (*tautológia*).

Például: $(\forall x)\ P(x) \vee (\exists y)\ \neg P(y)$

Kielégíthetetlen egy formula, ha minden interpretációban hamis a logikai értéke.

Például: $(\forall x)\ P(x) \wedge (\exists y)\ \neg P(y)$

Formulák ekvivalenciája

(Az átalakítások 10 szabálya továbbra is érvényes)

1. $A \equiv B = (A \rightarrow B) \wedge (B \rightarrow A)$
2. $A \rightarrow B = \neg A \vee B$

3.a. $A \vee B = B \vee A$ (kommutativitás)

3.b. $A \wedge B = B \wedge A$

4.a. $(A \vee B) \vee C = A \vee (B \vee C)$ (asszociativitás –

4.b. $(A \wedge B) \wedge C = A \wedge (B \wedge C)$ a művelet kiterjesztése szempontjából fontos)

5.a. $A \vee (B \wedge C) = (A \vee B) \wedge (A \vee C)$ (disztributivitás)

5.b. $A \wedge (B \vee C) = (A \wedge B) \vee (A \wedge C)$

6.a. $A \vee F = A$

6.b. $A \wedge T = A$

7.a. $A \vee T = T$

7.b. $A \wedge F = F$

8.a. $A \vee \neg A = T$

8.b. $A \wedge \neg A = F$

9. $\neg(\neg A) = A$

10.a. $\neg(A \vee B) = \neg A \wedge \neg B$ (De-Morgan)

10.b. $\neg(A \wedge B) = \neg A \vee \neg B$

Q – olyan kvantor, amely jelölheti a $\forall$ és a $\exists$ jelek bármelyikét

11.a. $(Qx)A(x) \vee B = (Qx)(A(x) \vee B)$

11.b. $(Qx)A(x) \wedge B = (Qx)(A(x) \wedge B)$

12.a. $\neg((\forall x)A(x)) = (\exists x)(\neg A(x))$

12.b. $\neg((\exists x)A(x)) = (\forall x)(\neg A(x))$

13.a. $(\forall x)A(x) \wedge (\forall x)B(x) = (\forall x)(A(x) \wedge B(x))$

13.b. $(\exists x)A(x) \wedge (\exists x)B(x) = (\exists x)(A(x) \wedge B(x))$

14.a. $(Q_1x)A(x) \vee (Q_2x)B(x) = (Q_1x)(Q_2y)(A(x) \vee B(y))$

14.b. $(Q_1x)A(x) \wedge (Q_2x)B(x) = (Q_1x)(Q_2y)(A(x) \wedge B(y))$

(a kiemelés ára, hogy át kell nevezni a változót)

A logikai következmény

Mint nulladrendűnél.

Klóz alakra hozás

1. Az ekvivalencia és az implikáció kiküszöbölése
2. Negáció hatáskörének csökkentése
   - $\neg(\neg A) = A$
   - $\neg(A \vee B) = \neg A \wedge \neg B$
   - $\neg(A \wedge B) = \neg A \vee \neg B$
   - $\neg((\forall x)A(x)) = (\exists x)(\neg A(x))$
   - $\neg((\exists x)A(x)) = (\forall x)(\neg A(x))$
3. Változók standardizálása

   Ha szükséges, nevezzük át a változókat úgy, hogy az egyes kvantorok által kötött változók különbözzenek.

   Például:

   $(\forall x)\,[P(x) \rightarrow (\exists x)Q(x)]$ formula standardizálás utáni alakja a következő:

   $(\forall x)\,[P(x) \rightarrow (\exists x)Q(y)]$

4. Az egzisztenciális kvantorok kiküszöbölése

   Példa: $(\exists x)P(x)$ ez akkor kielégíthető, ha a P(a) formula kielégíthető.

   $(\exists x)P(x)$ formulát helyettesíthetjük a rezolúcióban P(a)-val

   a – *skolem konstans*

   Példa: $(\forall x)(\exists y)P(x,y)$ formula. Itt minden x-hez létezik legalább egy x-től függő y úgy, hogy P(x,y) igaz. Legyen g(x) egy olyan függvény, amely minden x-hez hozzárendel egy ilyen y-t.

   g(x) – *skolem függvény*

   Balról jobbra küszöböljük ki az egzisztenciális kvantorokat.

   Példa: (kvantorok jobbra tolására)

   $(\forall x)[(\exists y_1)P(y_1) \wedge (\exists y_2)Q(x,y_2)]$

   $(\exists y_1)P(y_1) \wedge (\forall x)(\exists y_2)Q(x,y_2)]$

   $P(a) \wedge (\forall x)Q(x,g(x))$

5. Prenex formára hozás (skolem standardizált forma)

   (a formulában az előző átalakítás után már csak univerzális kvantorok lehetnek)

   az összes kvantort kiemeljük a formulából

   Példa: $(\forall x_1) \ldots (\forall x_n)\, A(x_1, \ldots, x_n)$

6. Klózok kialakítása

   a disztributivitás segítségével klózok konjunkciójává alakítjuk a formulát

7. Az univerzális kvantorok elhagyása

   ha minden kvantorunk le van kötve, akkor elhagyhatjuk a $\forall$ kvantorokat a kifejezés elől

8. A konjunkció elhagyása

   elhagyjuk a konjunkciókat → klózok halmaza (kényelmesebb velük dolgozni)

9. A változók átnevezése

   nevezzük át a változókat úgy, hogy egyetlen változó se szerepeljen két klózban.

Példa: (klóz formára hozásra)

$(\forall x)(\forall y)\{(\exists z)[P(x,z) \wedge P(y,z)] \rightarrow (\exists z)[Q(x,z) \wedge Q(y,z)]\}$

(1) $(\forall x)(\forall y)\{\neg(\exists z)[P(x,z) \wedge P(y,z)] \vee (\exists z)[Q(x,z) \wedge Q(y,z)]\}$

(2) $(\forall x)(\forall y)\{(\forall z)[\neg P(x,z) \vee \neg P(y,z)] \vee (\exists z)[Q(x,z) \wedge Q(y,z)]\}$

(3) $(\forall x)(\forall y)\{(\forall z)[\neg P(x,z) \vee \neg P(y,z)] \vee (\exists u)[Q(x,u) \wedge Q(y,u)]\}$

(4) $(\forall x)(\forall y)\{(\forall z)[\neg P(x,z) \vee \neg P(y,z)] \vee [Q(x,g(x,y)) \wedge Q(y,g(x,y))]\}$

(5) $(\forall x)(\forall y)(\forall z)\{[\neg P(x,z) \vee \neg P(y,z)] \vee [Q(x,g(x,y)) \wedge Q(y,g(x,y))]\}$

(6) $(\forall x)(\forall y)(\forall z)\ \{[\neg P(x,z) \vee \neg P(y,z)] \vee Q(x,g(x,y))]$

$\wedge\ [\neg P(x,z) \vee \neg P(y,z)] \vee Q(y,g(x,y))]\}$

(7,8,9) C_1: $\neg P(x_1,z_1) \vee \neg P(y_1,z_1)] \vee Q(x_1,g(x_1,y_1))$

C_2: $\neg P(x_2,z_2) \vee \neg P(y_2,z_2)] \vee Q(y_2,g(x_2,y_2))$

Elhagyhatók a klózokból a diszjunkciók, és így a klózok *literálok halmazává* válnak:

C_1: $\{\neg P(x_1,z_1), \neg P(y_1,z_1)], Q(x_1,g(x_1,y_1))\}$

C_2: $\{\neg P(x_2,z_2), \neg P(y_2,z_2)], Q(y_2,g(x_2,y_2))\}$

Az egyesítési algoritmus

A rezolúció egy olyan következtetési módszer, amellyel a komplemens literálpárt tartalmazó

$C_1 = p \vee C_1$' és a

$C_2 = \neg p \vee C_2$' klózokból a

$C_3 = C_1$' $\vee$ C_2' klózt állítjuk elő.

Példa: (komplemens literálpár létének megállapítása az elsőrendű predikátumkalkulusban)

$C_1 = P(x) \vee Q(x)$

$C_2 = \neg P(f(x)) \vee R(x)$

(az eredeti formulák)

F_1: $(\forall x)\ [\neg P(x) \rightarrow Q(x)]$

F_2: $(\forall x)\ [P(f(x)) \rightarrow R(x)]$

Látható, hogy C1 és C2 klózzal elvégezhető a rezolúció, ha a komplemens literálpárban az argumentumokat *azonossá tesszük*! („közös nevező")

Ha a C1 klózt x helyett f(x)-re írjuk fel, akkor

C_1' = $P(f(x)) \vee Q(f(x))$

$C_2 = \neg P(f(x)) \vee R(x)$ ezek alapján

$C_3 = Q(f(x)) \vee R(x)$

Helyettesítésnek egy olyan véges $\{v_1/t_1, \ldots, v_n/t_n\}$ halmazt nevezünk, amelyben minden v_i változó, minden t_i term és a v_i változók mind különbözőek.

($v_{1, \ldots,} v_n$ változók minden előfordulását egyidejűleg rendre $t_{1, \ldots,} t_n$-nel helyettesítjük.)

Példa: (így helyettesítünk:)

P(x, f(x), y) {x/a, y/g(b)}

P(a, f(a), y) (x minden előfordulását a-val helyettesítjük)

P(a, f(a), g(b)) (y minden előfordulását g(b)-vel helyettesítjük)

αβ helyettesítés kompozíciója:

α: $\{x_1/t_1, x_2/t_2, \ldots, x_n/t_n\}$

β: $\{y_1/u_1, y_2/u_2, \ldots, y_m/u_m\}$

αβ: $\{x_1/t_1\beta, x_2/t_2\beta, \ldots, x_n/t_n\beta, y_1/u_1, y_2/u_2, \ldots, y_m/u_m\}$

Amiket már egyszer helyettesítettem – nem kell újra helyettesíteni! Azaz

1) törölni kell azon elemeket, amelyekre $x_i/t_i\beta$ teljesül, és
2) azokat az y_j/u_j elemeket is, amelyekre $y_j \in \{x_1, \ldots, x_n\}$ teljesül.

Példa:

α: $\{x/f(y), y/z\}$

β: $\{x/a, y/b, z/y\}$

αβ: $\{x/f(b), y/y, x/a, y/b, z/y\}$

ebből definíció szerint törölhető a y/y, x/a, y/b elemek, így a végleges forma:

αβ: $\{x/f(b), z/y\}$

Az α helyettesítést a $\{A_1, \ldots, A_N\}$ formulák *egyesítőjének* nevezzük, ha $A_1\alpha = \ldots = A_N\alpha$.

Az $\{A_1, \ldots, A_N\}$ halmazt *egyesíthetőnek* nevezzük, ha létezik hozzá α egyesítő.

Formulák egy halmazának több egyesítője is lehet.

Legáltalánosabb egyesítőnek nevezünk egy δ egyesítő helyettesítést, ha bármely α egyesítőhöz létezik α' helyettesítés, amelyre $\alpha = \delta\,\alpha'$.

A legáltalánosabb egyesítő sem egyértelműen meghatározott.

Az $\{A_1, \ldots, A_N\}$ formulahalmaz *különbségi halmazát* úgy kapjuk, hogy először meghatározzuk balról-jobbra azt az első olyan pozíciót, amelyen nem egyezik meg az összes A_i formula, majd vesszük az összes különböző részformulát, amelyek ezen a pozíción kezdődnek.

Példa: (D – különbségi halmaz, δ – helyettesítés)

H = {P(x, u, f(g(x))), P(a, y, f(y))}

D = {x, a} δ = {x/a}

H = {P(a, u, f(g(a))), P(a, y, f(y))}

D = {u, y} δ = {x/a, u/y}

H = {P(a, y, f(g(a))), P(a, y, f(y))}

D = {g(a), y} δ = {x/a, u/g(a), y/g(a)}

H = {P(a, g(a), f(g(a)))}

Mivel H egyértelművé vált, így a legutoljára előállított δ egy legáltalánosabb egyesítő.

A rezolúció

A C1 és C2 klózokat akkor mondjuk rezolválhatónak, ha található bennük egy olyan P predikátum, valamint ehhez egy r és s előfordulási szám, hogy a klózokat

$$C_1: P(t_{11}, \ldots, t_{1n}) \vee \ldots \vee P(t_{r1}, \ldots, t_{rn}) \vee C_1'$$

$$C_2: \neg P(u_{11}, \ldots, u_{1n}) \vee \ldots \vee \neg P(u_{s1}, \ldots, u_{sn}) \vee C_2'$$

alakra írva a

$$H = \{P(t_{11}, \ldots, t_{1n}), \ldots, P(t_{r1}, \ldots, t_{rn}), P(u_{11}, \ldots, u_{1n}), \ldots, P(u_{s1}, \ldots, u_{sn})\}$$

halmaz egyesíthető. Ha δ jelöli ennek legáltalánosabb egyesítőjét, akkor a rezolvens klózt úgy kapjuk, hogy mindkét klózból elhagyjuk az egyesítésben résztvevő literálokat, a megmaradókat pedig a δ helyettesítés alkalmazása után diszjunkcióval összekapcsoljuk, azaz

$$R(C_1,C_2) = C_1'\delta \vee C_2'\delta$$

Példa:

A_1 és A_2 állításokból logikailag az A_3 állítás (ez kellene belátni)

A_1: Némelyik páciens minden doktorban bízik.

A_2: A kuruzslókban egyetlen páciens sem bízik.

A_3: A doktorok nem kuruzslók.

Az állításoknak megfelelő formulák a következők:

F_1: $(\exists x)[P(x) \wedge (\forall y)(D(y)) \rightarrow M(x,y))]$

F_2: $(\forall x)[P(x) \rightarrow (\forall y)(K(y)) \rightarrow \neg M(x,y))]$

F_3: $(\forall x)[D(x) \rightarrow \neg K(x)]$

$F_1 \wedge F_2 \wedge \neg F_3$ — kielégíthetetlen (ezt kell bebizonyítani)

$\neg F_3$: $\neg(\forall x)[D(x) \rightarrow \neg K(x)]$ azaz

F_3': $\neg(\forall x)[D(x) \rightarrow \neg K(x)]$

Alakítsuk az F_1, F_2, és F_3' formulákat klóz formára.

(1) $P(a)$ — F_1-ből

(2) $\neg D(y) \vee M(a,y)$ — F_1-ből

(3) $\neg P(x) \vee \neg K(y) \vee \neg M(x,y)$ — F_2-ből

(4) $D(b)$ — F_3'-ből

(5) $K(b)$ — F_3'-ből

(a változókat nem jelöltük át)

A rezolúció egy lehetséges végrehajtása:

(6) $\neg K(y) \vee \neg M(a,y)$ — (1) és (3) rezolvense

(7)	M(a,b)	(2) és (4) rezolvense
(8)	¬M(a,b)	(5) és (6) rezolvense
NIL		(7) és (8) rezolvense

A rezolúció hatékonyságának növelése

S – alaphalmaz

S klózait rezolváljuk minden lehetséges módon.

A rezolvenseket mindig az S-hez csatoljuk (bővítjük S-et) → újabb klózhalmaz keletkezik

Ez az első szintű klózhalmaz – $R^1(S)$

Majd az $R^1(S)$ – S halmazbeli klózokat rezolváljuk az $R^1(S)$-beli klózokkal minden lehetséges módon. Az így nyert rezolvenseket az $R^1(S)$-hez csatoljuk → $R^2(S)$

Iteratív módon generáljuk tovább őket.

$R^0(S) = S$

Példa:

(1) $P(x)$

(2) $\neg P(x) \vee P(f(x))$

(3) $S(x)$

(4) $\neg S(x) \vee T(x)$

(5) $\neg T(x) \vee U(x)$

(6) $\neg U(x) \vee Q(x)$

(7) $\neg Q(x) \vee W(x)$

(8) $\neg W(x) \vee R(x)$

(9) $\neg R(f(g(x)))$

Képezzük az $R^1(S)$ – S klózhalmazt.

(1 - 2)	(10)	$P(f(x))$	
(3 - 4)	(11)	$T(x)$	(*)
(4 - 5)	(12)	$\neg S(x) \vee U(x)$	
(5 - 6)	(13)	$\neg T(x) \vee Q(x)$	(*)
(6 - 7)	(14)	$\neg U(x) \vee W(x)$	
(7 - 8)	(15)	$\neg Q(x) \vee R(x)$	(*)
(8 - 9)	(16)	$\neg W(f(g(x)))$	

Majd képezzük az $R^2(S)$ - $R^1(S)$ halmaz klózait az $R^1(S)$ – S és az $R^1(S)$ halmazokból vett párok rezolválásával.

(10 – 2)	(17)	$P(f(f(x)))$
(11 - 5)	(18)	$U(x)$

…

(26)

Majd harmadik lépésben a (17) – (26) klózokhoz keresünk lehetséges rezolválható párokat a teljes $R^2(S)$ halmazból. Ez lesz az $R^3(S)$ - $R^2(S)$ halmaz.

(27)

…

(50) □ (üres klóz)

De míg idáig eljutottunk, egy csomó, fölösleges klózt képeztünk.

Összesen 6 lépésből le tudtunk volna vezetni!

Az 1. részből 3 a másodikból 2 a harmadikból 1 szerepelt a bizonyításban. a (*)-os lépések ezek.

Vezérlési stratégia kell!

1) melyik két klózt rezolváljuk
2) ki kell választani, hogy melyik legyen a két komponens literálpár
3) ha kiválasztottunk egy klózpárt – mi legyen az amivel a helyettesítést végezzük

Ha csak a (*)-os elemeket tüntetjük föl → *cáfolati gráf*

De ha korlátozunk, lehet, hogy a megoldást is korlátozzuk ezzel és esetleg nem jutunk el hozzá.

1. Rezolúciós stratégiák (a hatékonyság növelésére)

I. *milyen halmazból* válogatunk
II. szabályozza a rezolvens képzés *sorrendjét*

I. (milyen halmazból válogatunk)

1) Szélességi keresés
   - teljes technológia
   - a legrövidebb levezetést fogja megtalálni
   - de hosszadalmas lehet széles fánál
   - szintek bejárása
   - rezolvens képzése, míg üres klózhoz nem jutunk
   - az előző példa is ez

2) Támogató halmaz stratégia
   - *támogató halmaz*: azok a klózok alkotják, amik a bizonyítandó állítás negáltjait tartalmazzák
   - bővítem az alaphalmazt
   - olyan rezolvenst képez, aminek legalább az egyik szülője ebből a támogató halmazból való
   - ezzel a módszerrel kevesebb klózra van szükség
   - *teljes* – biztosan megtalálom a NIL-re címkézett csúcsot
   - *támogató halmaz általánosítása*:
     - mintha visszafelé való következtetés lenne
     - a célból indulunk ki – a cél a fontos
3) Lineáris input stratégia
   - előrefelé való bizonyítás
   - a kiinduló állítások a fontosak
   - olyat választunk, hogy a kiindulási halmazból legyen az egyik
   - hasonlít a szélességi kereséshez, de korlátozza, hogy mely klózokkal dolgozhatunk.
   - de ez *nem teljes*, mert lehet, hogy két kiinduló elem egy újabb eleméből tudnánk előre lépni.

   Példa:

   (PROLOG, LISP nyelvek így bizonyítanak – lineáris jellegű módszerrel)
   - $(x_1 \wedge x_2 \wedge \ldots \wedge x_n) \rightarrow y$ *Horn klóz*
   - csak ilyen logikai formulánk lehet
   - ezzel korlátoztuk
   - a mindennapi érveléseinkben is ilyeneket használunk
   - prologban csak *Horn formula* van
   - így már nem fordulhat elő olyan, hogy valamit ne tudnánk bebizonyítani
4) Ősre korlátozott stratégia
   - előre felé haladunk
   - csak olyan rezolvensek képezhetők, amelyeknek az egyik szülője vagy a kiinduló klózhalmazból való, vagy a másik szülő egyik őse
   - lineáris input stratégia gyengített változata, mert ha nem tudunk alkalmas szülő klózt választani az alaphalmazból, akkor vehetjük a másik szülő valamely ősét.

- így már *teljes*!
- minél gyorsabban mélyre jussunk (mintha mélységi keresés lenne)

5) Egységklóz stratégia
   - arra koncentrál, hogy az egyszerű kifejezésekkel próbáljon minél gyorsabban előre haladni
   - *egységklóz:* egyetlen literálból áll (Pl.: S(x), P(y))
   - a rövidekből gyorsan eljutni NIL-ig
   - olyan rezolúciót engedünk csak meg, aminek az egyik szülője egységklóz
   - *hatékony*!
   - de *nem teljes*!
   - mélységi korlátot vezetünk be, ezen korlát elérése után más stratégiával folytatjuk
   - majd egy idő után, amint lehet, visszatér az egységklóz stratégiára
   - az a feltétel, hogy a kiegészítő stratégia teljes legyen!

II. (a képzési sorrendre is adhatunk stratégiát)

1) A klózokat az őket alkotó literálok száma szerint vesszük sorba
   a legkevesebbet tartalmazó (kis literálszámú) az első
2) A legrövidebb klóz generálásának elve
   - költséges eljárás
   - ha több komplemens literálpárunk van, akkor
     - abc sorrendbe rendezhetjük,
     - vagy megnézhetjük, hogy a belső része mennyire komplikált → mutató
     - ezek is lehetnek a stratégia alapjai.

<u>2. Bináris rezolúció (a hatékonyság növelésére)</u>

(m × n-et választunk ki)

C_1 : m számú literál

C_2 : n számú literál

- Ekkor a rezolvens képzésre $(2^m - 1)(2^n - 1)$ lehetőségünk van.
- Nagyon nagy számot eredményez → egyszerűsítésre van szükség!
- A *bináris rezolúció* esetén a rezolválandó literálnak csak egy-egy előfordulását választhatjuk ki mindkét szülő klózból a rezolvensképzéshez.

- De ezzel a megszorítással csak mn eset valósul meg és lehet, hogy így nem jutunk el az üres klózig!
- *Nem teljes* eljárás tehát!
- *Faktorizációra* van szükség! → ezzel már *teljes* lesz!
- Egy *klózhalmaz faktorizációja* a benne szereplő összes klóz faktorizációja.
- Egy *klóz faktorizációja* a klóz összes faktorának előállításával azonos.
- Ha δ egy olyan legáltalánosabb egyesítő helyettesítés, amely egy C klóz két vagy több literálját azonossá tesz, akkor a Cδ klózt a C klóz egy *faktorának* nevezzük.
- Egy klóznak egy literál szerinti összes faktorához úgy jutunk, hogy a literál n számú előfordulásából minden lehetséges módon kiválasztunk s = 2, …, n számú előfordulást és az ezeket egyesítő legáltalánosabb helyettesítést alkalmazzuk a klózra.

  Példa:

  $P(x,a) \vee P(y,z) \vee P(z,x)$

  Ebből az összes elsőrendű faktor:

  $P(y,a) \vee P(a,y)$ (az 1. és 2. literál egyesítése)

  $P(a,a) \vee P(y,a)$ (az 1. és 3. literál egyesítése)

  $P(x,a) \vee P(x,x)$ (a 2. és 3. literál egyesítése)

  $P(a,a)$ (az 1., 2. és 3. literál egyesítése)

3. Egyszerűsítési stratégia (a hatékonyság növelésére)

I. Tautológiák elhagyása

Példa:

$P(x) \vee B(y) \vee \neg B(y)$

Mivel $B(y) \vee \neg B(y)$ mindig igaz ← ezek nem segítenek, így a tautológiákat rögtön kihúzzuk.

Példa:

$\forall x \forall y(P(x) \vee \neg P(y))$ → nem tautológia, mert nem ugyanaz a változó!

II. Befoglaló klózok elhagyása

Def.: Egy D klóz magában foglalja a C klózt, ha létezik olyan α helyettesítés, hogy

$C\alpha \subset D$.

A befoglaló klóz törölhető – nem hordoz új információt.

III. Idegen literált tartalmazó klózok elhagyása

- A felhasználhatatlan elemeket elhagyjuk.
- *Idegen literál*: ha nem lehet rezolválni egyetlen másik klóz megfelelő, ellenkező logikai előjelű lietáljával sem.

IV. Procedurális hozzárendelés

- Literálok igazságértékének *közvetlen* meghatározása.
- Közvetlenül kiértékelhető literálok például az *alappéldányok*.
- Az alappéldány olyan klóz, vagy speciális literál, amely nem tartalmaz változót.
- A konstansokat tartalmazó alappéldányokat azonnal kiértékelhetjük.
- A közvetlen kiértékelés eszköze a *procedurális hozzárendelés*. Valahányszor a rezolvensképzés szülő klózaiban előfordul egy ilyen alappéldány, előbb meghívásra kerül egy eljárás, amely meghatározza az alappéldány logikai értékét. Ha ez az érték F, akkor a literált el lehet hagyni a klózból. Ha a kiértékelés értéke T, akkor magát a klózt lehet elhagyni a klózhalmazból.