Alkalmazott mesterséges intelligencia (AMI)

http://www.mit.bme.hu/oktatas/targyak/vimibb01

7. ea. (2023. ősz)

Többágenses (mulitágens) problémák

<u>Előadó:</u> Pataki Béla

a fóliák
Dobrowiecki Tadeusz
anyagainak felhasználásával
készültek

https://www.esrcheck.com/2023/06/05/artificial-intelligence-ai-experts-sign-statement-on-ai-risk/

Pataki Béla

BME I.E. 414, 463-26-79

<u>pataki@mit.bme.hu,</u>

<u>http://www.mit.bme.hu/general/staff/pataki</u>

Eddigiekben **egyetlen ágensben** gondolkodtunk (magányos hős… ☺).

Megoldást keresett az MI ágens/program *önmagában*, ennek érdekében különböző ***modell***eket használt: döntési fát épített adatokból, logikai modellt, neuronhálót használt vagy valószínűségi (Bayes) hálót.

Az élet viszont olyan, hogy leggyakrabban nem egyedül vagyunk, hanem hol minket segítő, hol minket akadályozó más ágensek is vannak körülöttünk.

Ezért érdemes röviden betekinteni a **többágenses**, ún. **multiágens** problémákba.

A való életben sokszor más ágensek is vannak körülöttünk – **kooperatívak (szövetségesek)**, és **kompetitívek (versengők vagy akár ellenségesek)**

Példák:

- Orvosi konzílium, ezek lehetnek az orvosnak tanácsot adó MI ágensek is – az egyik a csonttörések szakértője, a másik a bőrbetegségeké, a harmadik az emésztőrendszeré, a negyedik az idegrendszeré stb.
- A közlekedésben részt vevő önvezető autók – különböző célokkal, ismeretekkel és erőforrásokkal rendelkeznek (alapvetően kompetitív helyzet)
- Robotfoci- a saját csapattársak kooperatívak, az ellenfél játékosai kompetitívek
- Katasztrófáknál robotok menthetnek csapatban
- Fejlesztés team munkában
- stb.

Néhány – a multiágens feladatokhoz tartozó – gondolatot, megoldást példaként megnézünk, mindegyik témát hosszasan lehetne tárgyalni.

- **Feladatmegosztás**
- **Közös döntések** (szavazás)
- **Erőforráselosztás** (aukciók)
- **Kompetitív helyzetek**, ellenséges környezet (modellezés: játékok)

Taszkmegosztás (feladatmegosztás) 1. Master/Slave protokollal

1. Master a megrendelőtől kapott feladatot ***dekomponál***ja, és a Slave ágenseknek kiosztja
2. Slave ágensek a részfeladatukat megoldják, és a megoldásukat a Master ágenssel közlik
3. Master ágens a teljes feladat megoldását összerakja és továbbítja a felhasználó felé.

Jellegzetességek:
Master ágens tudása más és lényegesen több, mint Slave ágenseké: tartalmazza

- a feladat dekomponálási elveit,
- a megoldás szintetizálásának (összerakásnak) képességét
- a Slave ágensek képességi modelljét (hogy tudja, hogy kire mit lehet/érdemes kiosztani).

Taszkmegosztás 2. Vállalkozási Hálók protokollal (*Contract Nets*)

1. A menedzser ágens átveszi a feladatot és kisebb „porciókra”, „taszkokra” bontja
2. A menedzser ágens vállalkozókat keres taszkjához, vagy taszkjának egyes részeihez.
3. Szétküldi (broadcast) a taszkok (problémák) leírását (esetleges megkötésekkel, pl. határidő, minőség), és kedvező ajánlatokra vár. (Az aukciókra hajaz, lásd később.)
4. A vállalkozó ágensek összemérik a meghirdetett taszkok leírását a saját képességeikkel (tudás modellel), és vagy nem reagálnak, vagy beküldik a jelentkezésüket (milyen feltételekkel vállalkoznak a feladatra, milyen minőségű megoldást képesek szállítani, stb.).
5. A menedzser ágens választja ki a legjobbnak tűnő ajánlatokat, és a feladatokat véglegesen kiadja elvégzésre.
6. Vállalkozó ágensek a rájuk bizott feladatokat megvalósítják és a megoldásokat a Menedzsernek beküldik.
7. A befutó megoldásokból a Menedzser ágens összerakja a teljes feladat megoldását, és elküldi a felhasználónak.

Vállalkozási Hálók protokoll (*Contract Nets*)

Jellegzetességek:

Előre nem ismert, hogy egy feladatot melyik ágens végzi majd el.
A **Menedzser ágens** tudása lényegesen több, mint a Vállalkozó ágenseké, abban, hogy az ő feladata

- kapcsolattartás a felhasználóval
- a feladat dekompozíciója
- a feladat megoldásának szintézise

Nem kell tudnia a feladatot megoldani, nem kell mindig ismerni a Vállalkozók képességeit, viszont tudnia kell mérlegelni a beküldött jelentkezéseket (mások modellje, feladat modellje, stb. alapján).

Minden **vállalkozó ágens**nek rendelkeznie kell viszont:
- a **problématerületre** vonatkozó problémamegoldó tudással,
- tudással **önmagáról**, hogy a vállalkozás sikerét mérlegelni tudja.

A vállalkozó ágensnek szabad autonóm módon **elutasítani** is ...

Kooperáció (együttműködés) esetén: **hogyan hozzanak meg egy döntést együttesen?**

Együttműködés versengés közepette – szavazó ágensek

- Versengés: különböző döntési alternatívákat kínálnak az egyes ágensek, mivel más a tudásuk, mások a preferenciáik
- Együttműködés: végül egy közös döntésre akarnak jutni

Vegyük észre, hogy gyakorlatilag az emberek közt is ez megy végbe a szavazásoknál! ***Az MI-nél sokszor visszaköszön a mindennapi élet!***

Szociális választás elmélete (Social Choice Theory):
(Valószínűleg) versengő preferenciák korrekt, kielégítő aggregálása (összegzése, összehangolása) egy közösségi döntésbe (szociális kimenetel). Szavazás:

- erőforrások felosztása,
- koalíciók formálása,
- …

Racionális ágensek

Preferenciák tranzitivitása: humán racionalitás alapvető aspektusa, a helyes világfelfogás kifejezése

Tranzitív preferencia (*i*-edik ágens): ha $x \succ_i y, \;\; y \succ_i z \;\Rightarrow\; x \succ_i z$

legalább olyan jó $x \succcurlyeq_i y$

indifferens (mindegy) $x \; I_i \; y$

Ha: alma $\succ_{János}$ körte, körte $\succ_{János}$ szilva $\Rightarrow$

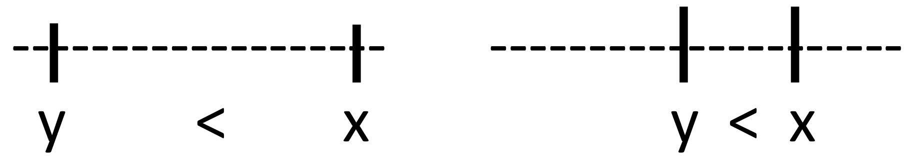
$\Rightarrow$ „elvárt”, hogy alma $\succ_{János}$ szilva

Ha: alma $\succ_{évfolyam}$ körte, körte $\succ_{évfolyam}$ szilva $\Rightarrow$

$\Rightarrow$ „elvárt-e”, hogy alma $\succ_{évfolyam}$ szilva ?

Ordinális vagy kardinális preferencia

Mennyire $x \succ_{\text{János}} y$?



ordinális : csak rangsorolás

kardinális: valamilyen értéket rendelünk preferencia mértékéhez
finomabb, precízebb?

A szavazási mechanizmusok többsége ordinális.

A legelterjedtebb szavazási protokollok

Többségi szavazás (Plurality rule) (**TB**): a győztes az, akié a legtöbb szavazat (a semleges szavazat nem számít)

Minősített többségi szavazás (Majority rule) (**MT**): a győztes az, akié a szavazatok több, mint a fele (csak a mellette szavazat számít, aki semleges, az ellene van)

Példa

Többségi szavazás (**Plurality** rule) (**TB**):

a győztes az, akié a legtöbb szavazat (a semleges szavazat nem számít)

Tegyük fel, hogy 4 jelöltre lehet szavazni: Aladár, Béla, Cecília, Dóra. Minden szavazó felállított egy saját sorrendet köztük, pl. az egyik szavazónál

Dóra > Aladár > Cecília > Béla

és ez a szavazó a leginkább preferáltra szavazott, tehát ez esetben Dórára.

	1 cs	2 cs	3 cs	4 cs
	20	24	26	30
1.	z	y	x	w
2.	x	z	y	z
3.	y	x	z	x
4.	w	w	w	y

TB:

w győz 30 szavazattal (kisebbségi jelölt!)

w a páronkénti felmérésben mindenkivel szemben alulmarad, mégis győz.

Kétfordulós: nincs minősített többség, a két legjobb jelölt: **w** (30), **x** (26)
A 2-ik fordulóban: **x** (70), **w** (30) → **x** a győztes.

Akik a **z** mellett vannak, „joggal?" panaszkodhatnak, hogy miért éppen **x**, ha a többségnél z $\succ$ x ! Mégis **x** miért nyert?

MT szabály 2 jelölt esetén OK, de szokás több jelöltre is alkalmazni.

Tipikus kiterjesztés, pl.: **Kétfordulós** (run-off, RO): az a győztes, aki a minősített többséget kapja (>50%), ha az nincs, akkor a legjobb kettő egymással szemben második fordulóban, sima többséggel.

Többségi szavazás baja közismert, 1700-as évek, Francia Akadémia

Jean-Charles de **Borda**
súlyozott rendezés
(Borda-szabály)

Marie Jean Antoine Nicolas Caritat,
Marquis de **Condorcet**
(páronkénti győztes, Condorcet-kritérium)

de Ramon Llull (1232 - (1299) - c. 1315), www.uni-augsburg.de/llull/

Súlyozott rendezett szavazás - Borda szavazás (BC)

- k alternatíva, egész szám minősítés, mindenki rangsorol, a legrosszabb alternatíva 0, a legjobb (k-1) (semleges = azonos súly)
- alternatívánként összegzés
- eredmény: az alternatívák teljes, tranzitív szociális rendezése

Példa: tegyük fel, hogy heten szavaznak, és 4 alternatíva van: w, x, y, z

Preferencia	pont	1	2	3	4	5	6	7	Borda összpont
1.	3	w	w	x	x	y	y	w	w = 11
2.	2	x	x	y	y	z	z	x	x = 12
3.	1	y	y	z	z	w	w	y	**y = 13**
4.	0	z	z	w	w	x	x	z	z = 6

Borda szavazás (BC) - „győztesből vesztes – paradoxon”

Tegyük fel, hogy z-t ki kell zárni, érvénytelen! Láttuk, hogy z messze az utolsó, nem befolyásolhatná a szavazást!

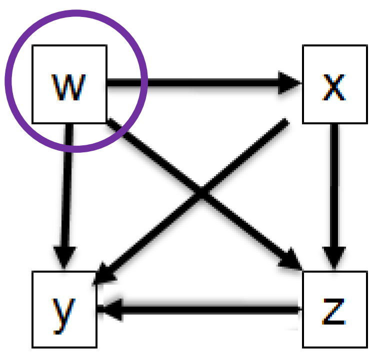
Az y, x és w versenyében z egy „irreleváns” alternatíva.
Borda szavazásban sérül a „**irreleváns alternatívától való függetlenség**” **kritériuma**. Az eredmény nem logikus, manipulálható.

Preferencia	pont	1	2	3	4	5	6	7	Borda összpont (BC)
1.	3	w	w	x	x	y	y	w	w = 11
2.	2	x	x	y	y	z	z	x	x = 12
3.	1	y	y	z	z	w	w	y	**y = 13**
4.	0	z	z	w	w	x	x	z	z = 6

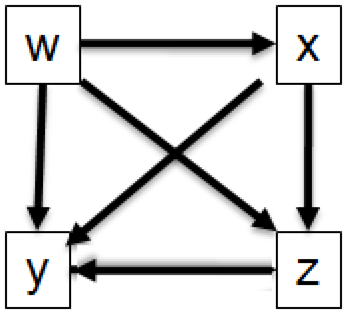
$y \succ_{BC} x \succ_{BC} w$

Preferencia	pont	1	2	3	4	5	6	7	Borda összpont (BC)
1.	2	w	w	x	x	y	y	w	**w = 8**
2.	1	x	x	y	y	w	w	x	x = 7
3.	0	y	y	w	w	x	w	y	y = 6

Condorcet győztes: ha **w** páronként mindenkinél jobb (többségi gráf), akkor **w** a Condorcet győztes.
Az un. **Condorcet kritérium szerint:** ha van Condorcet győztes, akkor az algoritmusnak őt kell megválasztania.
Az alábbi – Borda szavazási – példában ez sérül:



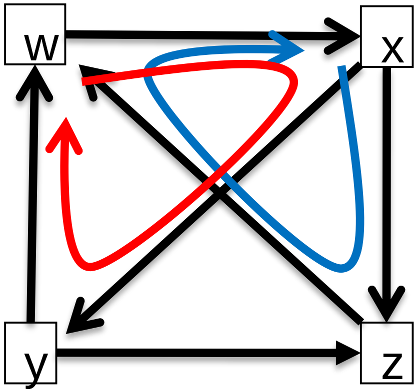
	Pont	**1**	**2**	**3**	**4**	**5**	**BC**
1.	3	w	w	z	x	x	**w = 9**
2.	2	x	x	y	z	y	**x = 10**
3.	1	y	z	w	w	w	y = 5
4.	0	z	y	x	y	z	z = 6



	Pont	1	2	3	4	5	BC
1.	3	w	w	z	x	x	**w = 9**
2.	2	x	x	y	z	y	**x = 10**
3.	1	y	z	w	w	w	y = 5
4.	0	z	y	x	y	z	z = 6

Ciklusok problémája – azaz a Condorcet győztes nem mindig létezik

	1	2	3	4	5	6	7	BC
1.	w	w	x	x	y	y	w	w = 11
2.	x	x	y	y	z	z	x	x = 12
3.	y	y	z	z	w	w	y	**y = 13**
4.	z	z	w	w	x	x	z	z = 6



A „józan paraszti ész” számára kicsit megrázó az alábbi szavazási paradoxon:

Arrow-tétel (Kenneth J. Arrow, 1963) - Nincs olyan szavazási protokoll, ami az alábbi – minimálisnak mondható – alapvető követelményeket egyszerre teljesítené, ha a lehetséges alternatívák halmaza legalább három elemű.

- **Nem diktatorikus**: a csoport eredménypreferenciája nem tükrözheti valamelyik egyén preferenciáját, a többi szavazó preferenciájától függetlenül.
- **Pareto-elvű**: ha minden szavazó egy alternatívát jobbnak tart másnál, akkor a csoportpreferenciának ezt tükröznie kell.
- Ha a csoportpreferencia X-et előbbre helyezi Y-nál, akkor ez nem változhat, ha az összes szavazó X-et előbbre helyezi a többi alternatíva változatlanul hagyása mellett.
- A **lényegtelen alternatívák**tól való függetlenség teljesül.

Azt a választási protokollt kell használnunk, amelynek problémái legkevésbé zavarók az adott közösség számára. (MI ágenseknél?)

Együttműködés versengés közepette – **koordinálás, feladatmegosztás, erőforrás-elosztás árverésekkel (aukció)**

Árverés – klasszikus, humán

humán licitáló

objektum

pénz

Koordinálás MI ágensek közt

ágenslicit

feladat

„költség” (idő, tár stb.)

Miért éppen árverés?

- **ismeretlen** értékű dolgok értékesítése
- **automatizálható**
- **csökkenti a tárgyalás komplexitását**, kedvező a számítógépes implementáció
- „**tisztességes**” megoldás benyomását kelti

Együttműködés versengés közepette – koordinálás, feladatmegosztás, erőforrás-elosztás árverésekkel (aukció)

Előnyök

- árverés lefutása **rövid**
- árverés **kommunikáció-hatékony**: információt licitekbe tömörítünk
- árverés **számítás-hatékony**: liciteket parallel módon lehet processzálni
- árverés **alacsony költségű** szervezetet eredményezhet
- árverést lehet használni akkor is, ha a terep (környezet), vagy a róla alkotott ágenstudás **változó**

Tipikus – aukcióval megoldható - koordinációs feladatok

Szerepek on-line/elosztott kiosztása

- Feladatok hozzárendelése mentő/ tűzoltó/ rendőrségi ágensekhez … misszió-kritikus/ Search and Rescue katasztrófa esetén …
- Különböző megfigyelési célok hozzárendelése külön szenzorokhoz vezeték nélküli (wireless) szenzorhálózatokban.
- Megfigyelő és manipuláló szerepek kiosztása manipulálási feladatkörben.

On-line elosztott ütemezés és vezérlés

- Feladatok ill. folyamatok hozzárendelése min. látencia idő, max. átbocsátó képesség, stb. érdekében.

Erőforrások elosztása

- Például valamilyen áru: búza, kőolaj, vizsgaidőpont ☺ stb.

(így számos NP-teljes optimalitási probléma közelítő, ám gyors megoldása lehetséges)

Egyedi aukció (árverés)

Árverésvezető **egyetlenegy** feladatot vagy árut kínál fel (tipikusan van egy minimális – kikiáltási – ár)

Angol árverés (emelkedő, be/ki részvétel, leütés)
Holland árverés (csökkenő, végleges kilépéssel)

Elsőlicites versenytárgyalás (boritékolt árverés)

Minden licitáló a feladat költségét nyújtja be boritékolt licitjeként.
A legalacsonyabb költséget vállaló licitáló nyer, megkapja a feladatot és beleegyezik, hogy **a licitált költségen** meg is valósítja. (Vagy ha árura licitálunk, akkor az árut a legmagasabb licitált áron vegye meg.)

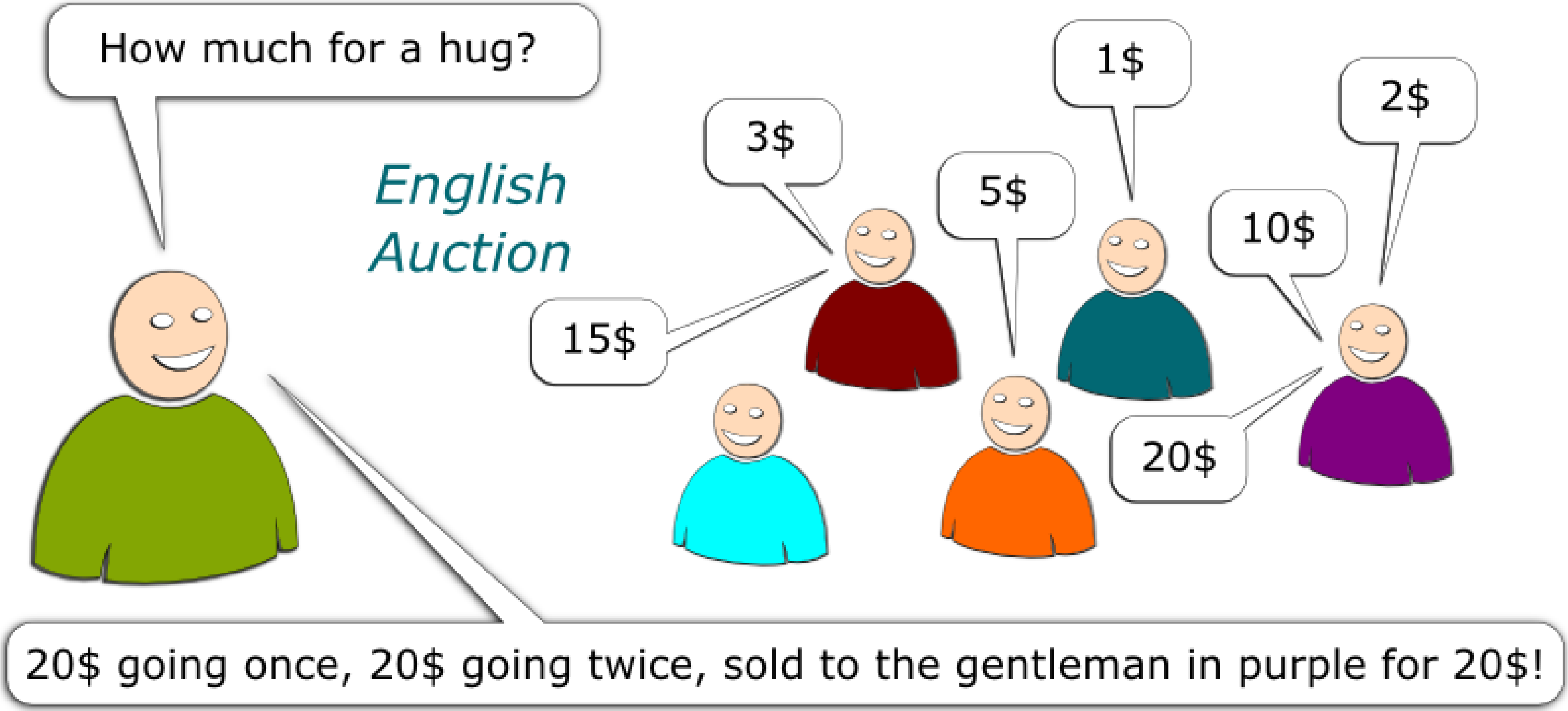
Másodlicites (Vickrey) versenytárgyalás (boritékolt árverés)

Protokoll u.a., csak a győztestől megkövetelik, hogy a **feladatot a második legkisebb licit költségén** valósítsa meg. (Vagy ha árura licitálunk, akkor az árut a második legmagasabb áron vegye meg.)

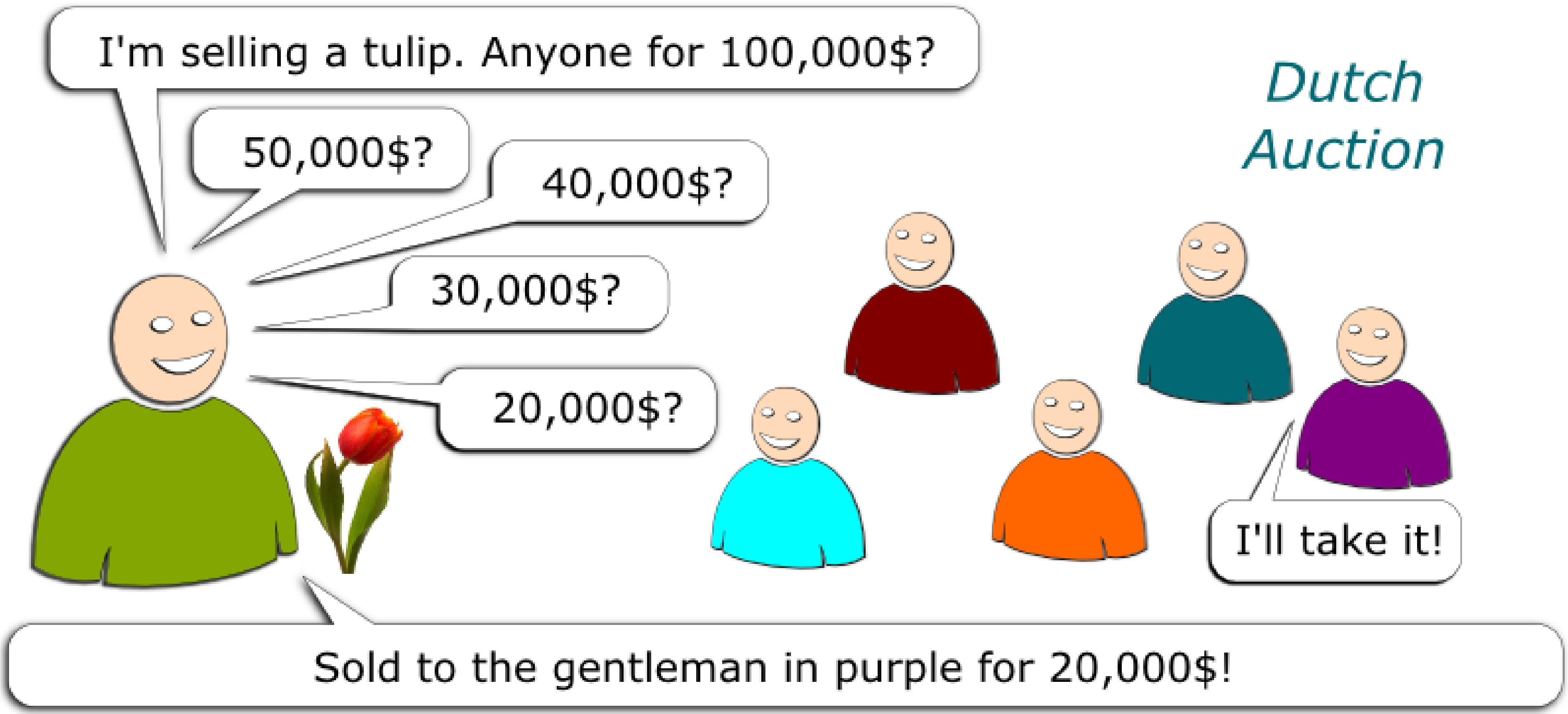
Melyik mechanizmust válasszunk?

Mindegy, amíg az ágensek **hitelesen** licitálnak.

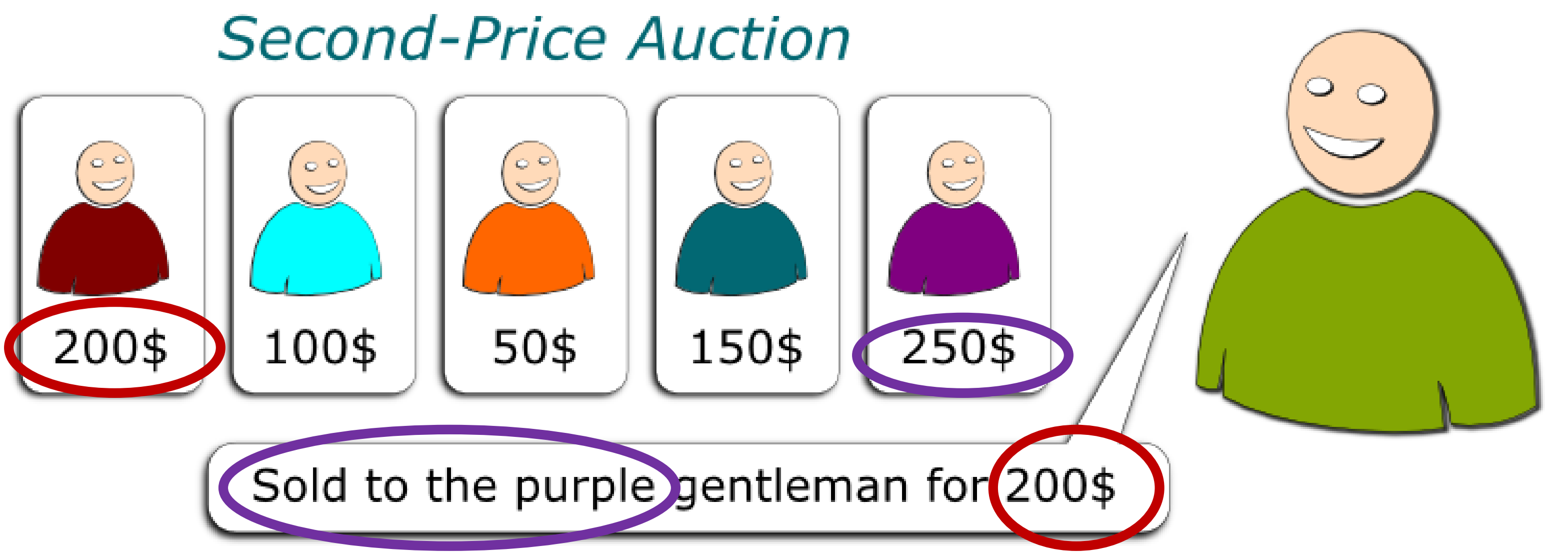
Angol árverés



Holland árverés



Másodlicites (Vickrey) árverés



Vickrey-aukció

A résztvevők zárt borítékban leadják az ajánlataikat, az nyer, aki a legnagyobb licitet adta, de a *második legnagyobb összegért* viheti el az árut/erőforrást, vagy a második legkisebb árért kell elvégezze a vállalt feladatot.

Vickrey-aukció

A résztvevők zárt borítékban leadják az ajánlataikat, az nyer, aki a legnagyobb licitet adta, de a *második legnagyobb összegért* viheti el az árut/erőforrást. **Az optimális stratégia az igazmondás!**

Érjen a *k*-adik licitálónak az áru **vk**-t, és legyen a licitje bk. A többi licitáló közül a legmagasabb licit legyen B.

1. Ha **vk**<bk, tehát a k-adik túllicitált:
   - B≤vk esetén vk=bk-val is nyert volna, és ugyanúgy B-ért viheti el
   - bk<B esetén nem nyer, bk=vk esetén is ez lenne a helyzet
   - Ha vk<B<bk, akkor megnyeri a licitet, de az árura nézve vk-B<0 negatív a nyeresége (veszteség)

2. Ha bk<**vk**, tehát értékén alullicitált:
   - B≤**vk** esetén vk=bk-val is nyert volna, és ugyanúgy B-ért viheti el, viszont növeli a nyerési esélyét
   - **vk**<B esetén nem nyer, bk=vk esetén is ez lenne a helyzet
   - Ha bk<B<**vk**, akkor vk esetén megnyerte volna a licitet, és az árura nézve vk-B>0 pozitív lenne a nyeresége

A licitáló stratégiája határozza meg, hogy a többiek milyen licitjeinél, milyen árszintnél mennyit emel a saját licitjén, vagy kihagy, esetleg kiszáll.

Viszont nem csak „nekem" van stratégiám, a többeiké hat az én stratégiám eredményességére!

A stratégia domináns, ha a többiek stratégiájától függetlenül (bármilyen stratégiát is követnek a többiek) az adott licitálónak maximális hasznot biztosít.

A stratégiák úgynevezett **Nash-egyensúly**ban vannak, ha egyetlen licitáló se tudja úgy megváltoztatni a stratégiáját, hogy többet nyerjen, feltéve, hogy az összes többinek változatlan marad a stratégiája.

Egy súlyos bűntény kapcsán két gyanúsítottat (X-et és Y-t) letartóztat a rendőrség. Mivel nincs elegendő bizonyíték a vádemeléshez, ezért elkülönítik őket egymástól, és mindkettőjüknek ugyanazt az ajánlatot teszik. Amennyiben az első fogoly vall és a társa hallgat, akkor az előbbi büntetés nélkül elmehet, míg a másik, aki nem vallott, 10 év börtönt kap. Ha az első tagadja meg a vallomást és a második vall, akkor a másodikat fogják elengedni, és az első kap 10 évet. Ha egyikük sem vall, akkor egy kisebb bűntényért 1-1 évet kapnak mindketten. Ha mindketten vallanak, mindegyikük 5-5 évet kap. (Tehát mindkettőjüknek kétféle stratégiája lehet: vall vagy tagad.)

	Y tagad	**Y vall**
X tagad	Mindketten 1-1 évet kapnak	X 10 évet kap, Y szabad
X vall	X szabad, Y 10 évet kap	Mindketten 5-5 évet kapnak

X vált stratégiát, Y nem (X rosszul jár)

Y vált stratégiát, X nem (Y rosszul jár)

A világoszöld cella mutatja a Nash-egyensúlyt!
Akár X, akár Y változtat a stratégiáján (vallomás helyett tagad), miközben a másik nem változtat, akkor rosszul jár.

Laborfeladat: Független, szimultán, online aukció, adott értékű licitnövekményekkel

Például önvezető autók licitálhatnak szabad parkolóhelyekre.

Több autó (licitáló) és több parkolóhely (áru, szolgáltatás) van.

Végül egy autó csak egy parkolóhelyet kaphat meg (egy fenékkel egyszerre csak egy lovon lehet ülni ☺). Ez azt jelenti, hogy párhuzamosan folynak az aukciók az egyes parkolóhelyekre, de egy-egy autó mindig csak egyre licitálhat aktívan , a többi licitben passzív. Ha annál már meghaladja az ár azt, amit hajlandó érte adni, akkor kiszáll, és valamelyik másik – még szabad helyre – kezd licitálni.

Példaprobléma-terület kompetitív (ellenséges!) ágensekre – Ellenség elleni játékok. A kétszemélyes játékok *alkalmasak a módszerek vizsgálatára* (de az élet számos területén jelentkezik: pl. üzleti élet...)

Nehézség:

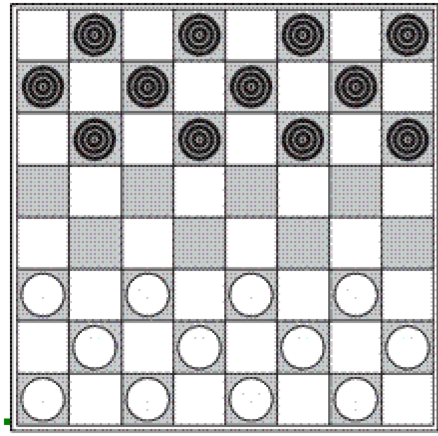
- nem csak mi lépünk, az ellenség is lép
- nem tudjuk, mit fog lépni
- mindenre fel kell készülni megfelelő válaszlépéssel ⇨ stratégia

	determinisztikus	véletlen
teljes információ	sakk, dáma, go, ...	ostábla, Monopoly ...
nem teljes információ	csatahajók, ...	bridzs, póker, atomháború, ...

- zérus-összegű a játék, vagy sem? (pl. fogolydilemma – nem zérus-összegű)
- keresés eredménye = stratégia [$\Rightarrow$ optimális vagy legalább jó válasz az ellenség minden lépésére]

Dámajáték megoldva, **hibátlan játék döntetlenre vezet!**

Dámajáték keresési tere kb. 5×10^{20} pozició.



Történelem

1950 - Arthur Samuel öntanuló programja

1963 – Első győztes parti tehetséges emberi játékos ellen

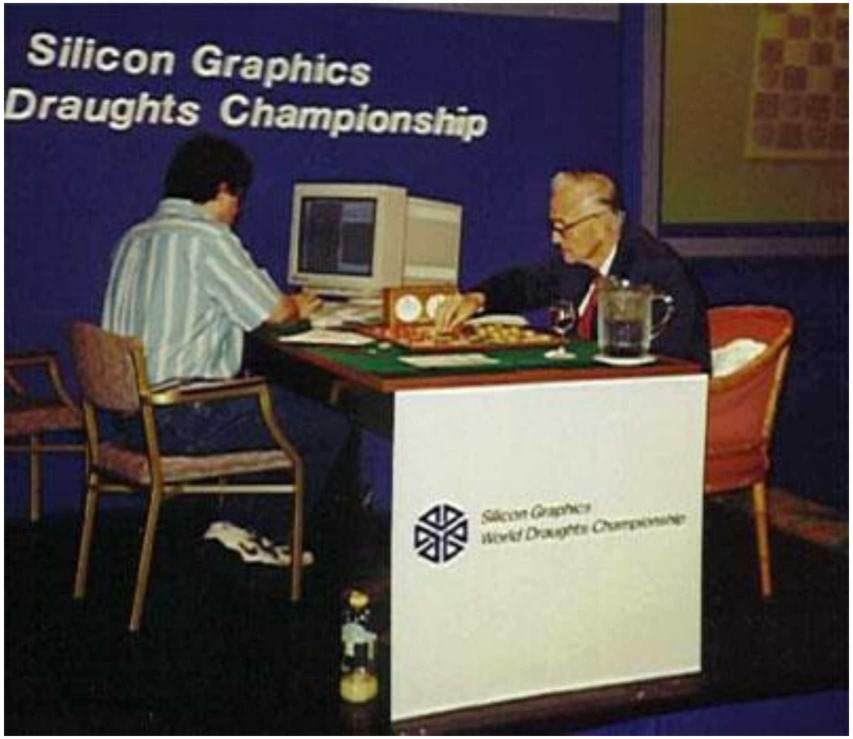
1989 - Chinook projekt, cél: legyőzni az emberi világbajnokot

1990 - Chinook jogosult Világbajnokságban részt venni

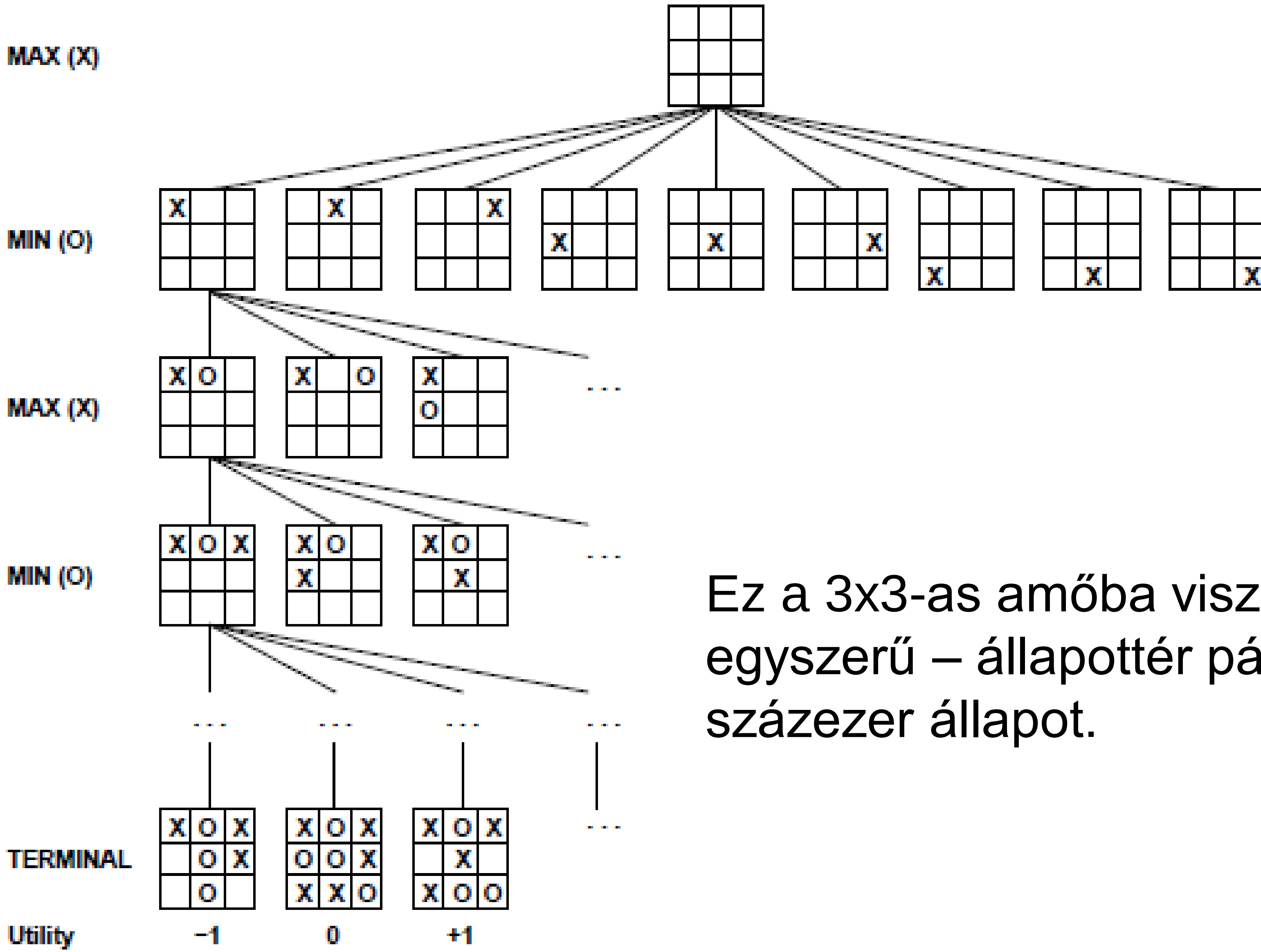
1992 - Marion Tinsley világbajnok a világbajnoki címért folytatott mérkőzésen nehezen, de felülkerekedik

1994 – visszavágó, Tinsley egészségi állapota miatt visszalép, rövidesen meghal

1996 - Chinook minden emberi játékosnál erősebb



példa: 3x3-as amőba – 2 játékos MAX és MIN



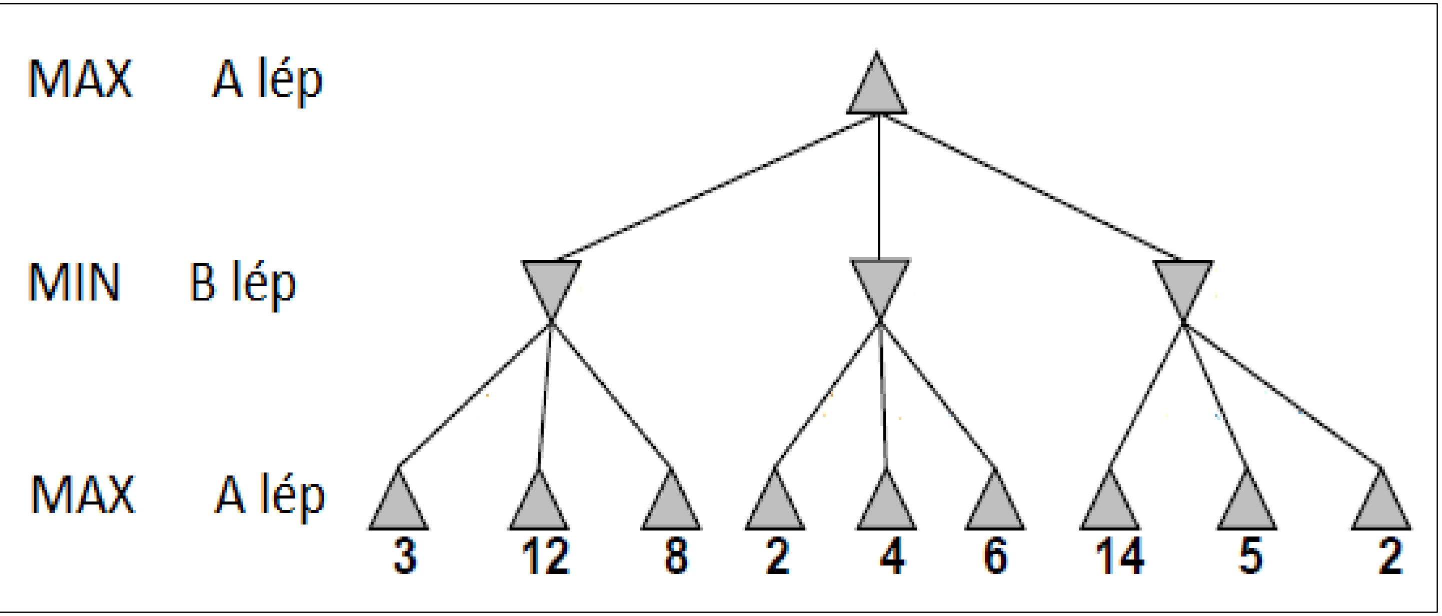
Ez a 3x3-as amőba viszonylag egyszerű – állapottér pár százezer állapot.

Minimax gondolata

Van két játékosunk, az egyiket nevezzük MAX-nak, a másikat MIN-nek.

Értékeljük a játék során az adott helyzeteket, az értékelésünk MAX nyereségét mutatja. Ezért MAX a nagy értékeket szeretné elérni, MIN a minél kisebbeket (hiszen ekkor MAX nem nyer, vagy nem annyit).

Példa: Eladom a házamat, az eladási ár jellemzi e helyzetet, én vagyok MAX a vevő MIN.
Én (MAX) arra törekszem, hogy olyan állapotban végződjenek a tárgyalások, ahol a jellemző összeg nagy, a vevő viszont olyan végállást szeretne, ahol az összeg minél kisebb.

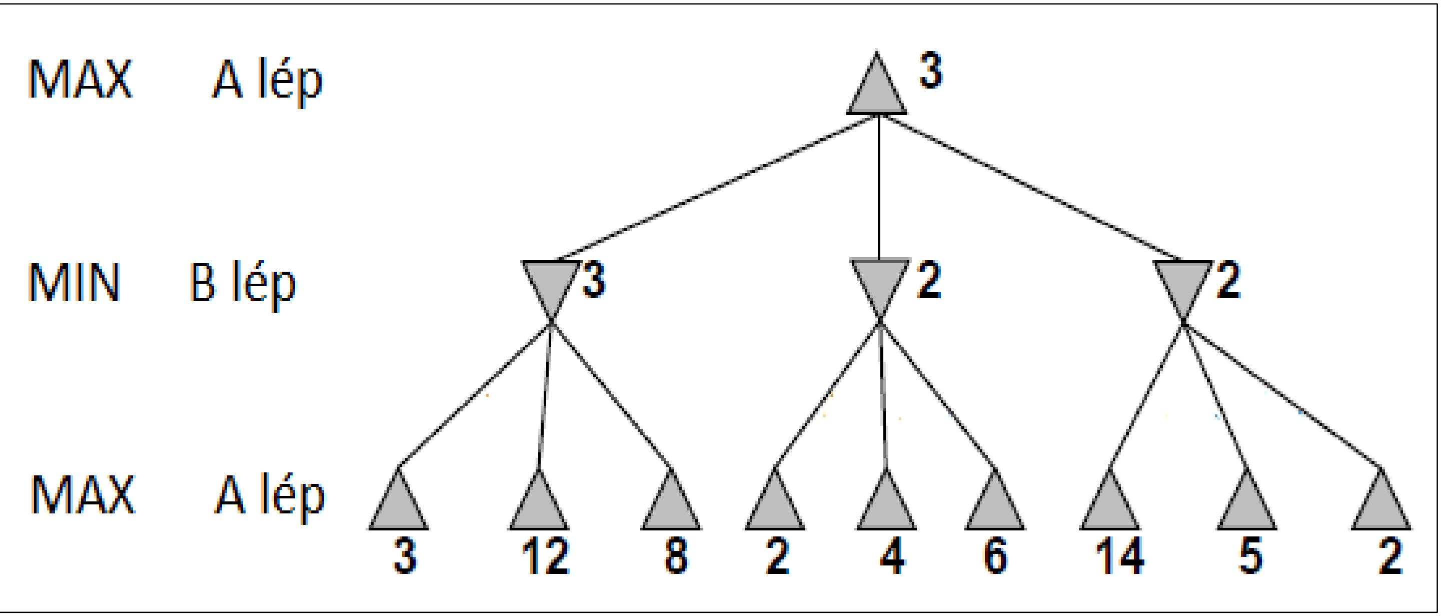


Minimax gondolata

A (vég)állapotok értékét kiértékeljük (MAX játékos szempontjából)

Ebben a példában: valamilyen egyszerű kiértékelő függvény.

Majd a levelektől felfele alkalmazzuk (mint egy fordított szélességi kiértékelés, lentről-felfele, szintenként) a játék elveit (MIN a minimumra, MAX a maximumra törekszik)



Minimax gondolata

Tulajdonságai

Teljes? Igen, ha a fa véges.

Optimális? Igen, egy optimálisan játszó ellenféllel szemben. *Különben?*

Időkomplexitás? $O(b^d)$

Tár? $O(bd)$ (mélységi jelleg miatt)

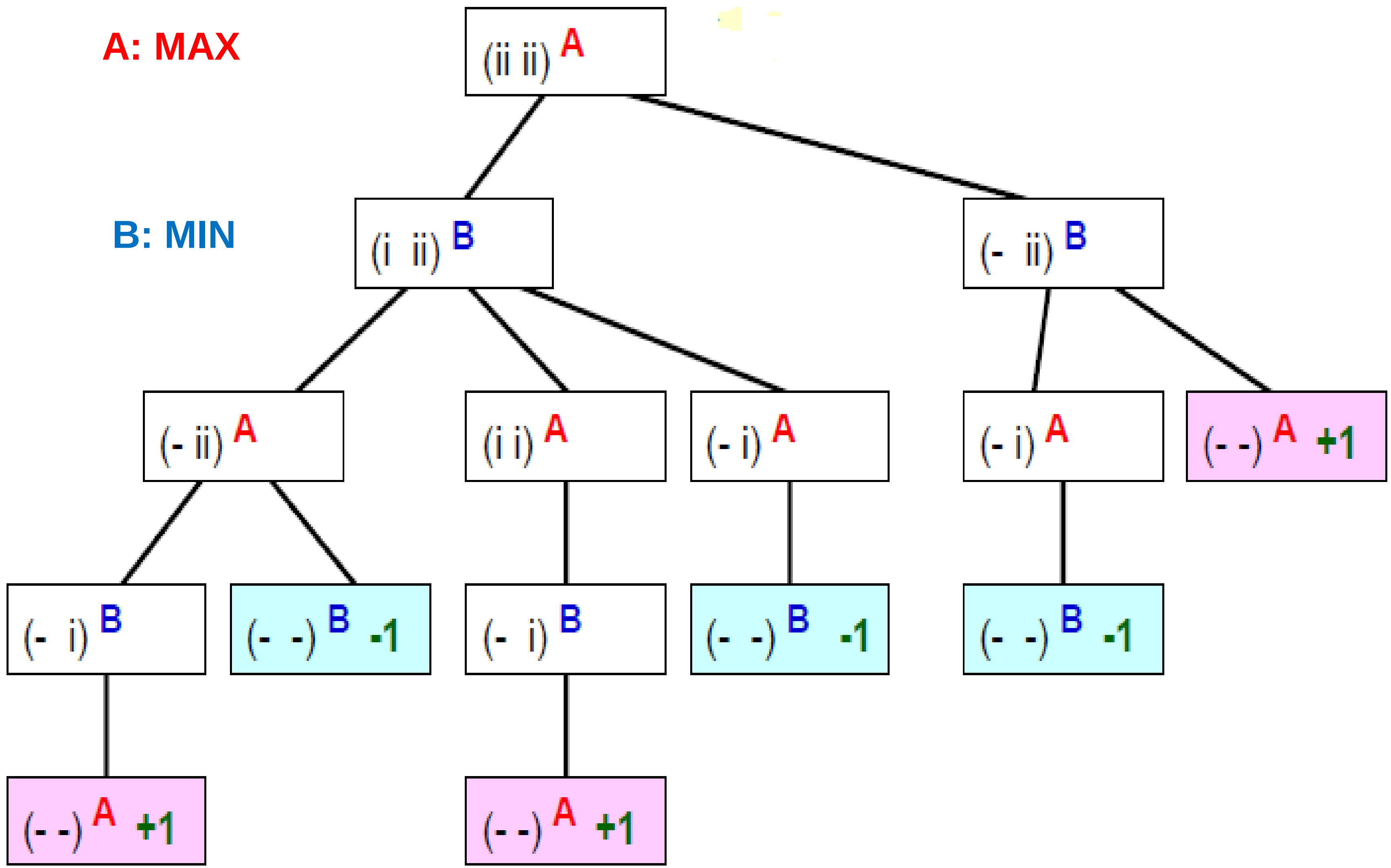
(sakk: $b \approx 35$, $d \approx 80$? $O(b^d)$:$O(10^{123})$)

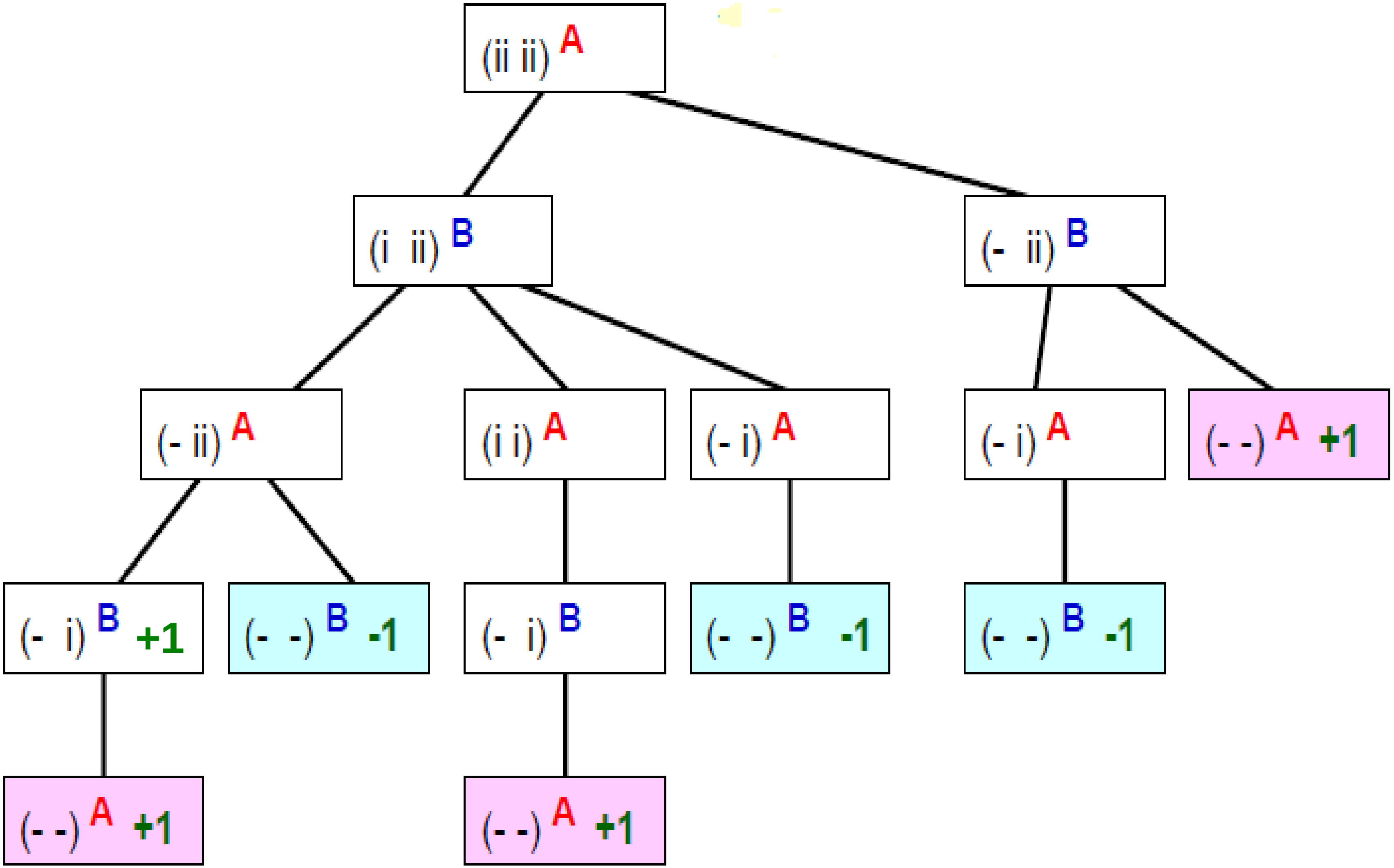
Nim

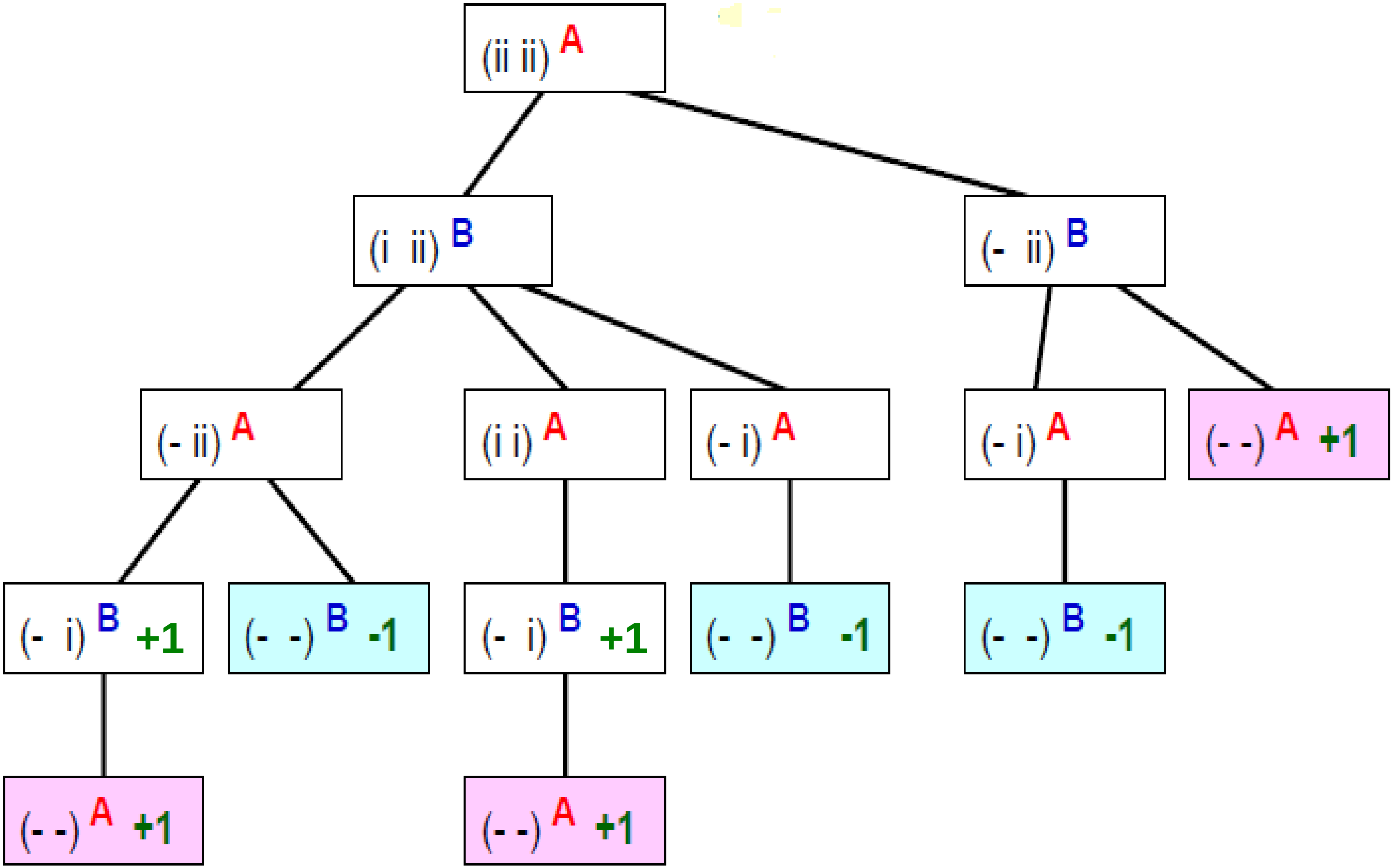
1. Bizonyos számú gyufakupac.
2. Egy lépésben egy játékosnak nem nulla, de tetszőleges számú gyufát szabad elvenni, de csakis egyetlenegy kupacról.
3. Aki az utolsót veszi el, veszít.

Egyszerű példa, II-Nim: két kupac, mindkettőben 2-2 gyufa

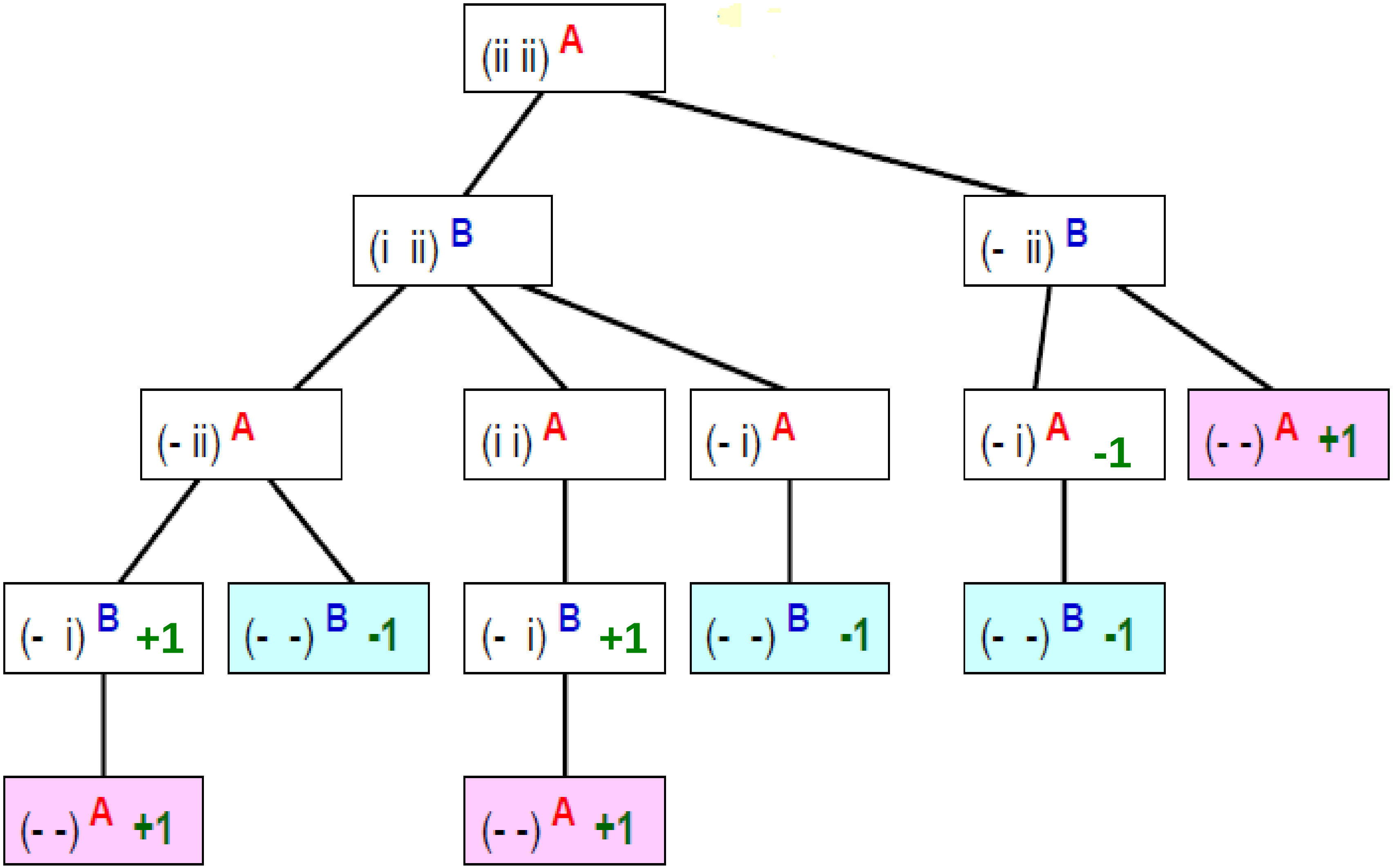
S	=	a finite set of states (note: state includes information sufficient to deduce who is due to move)	(_ , _)-A (_ , i)-A (_ , ii)-A (i , i)-A (i , ii)-A (ii , ii)-A (_ , _)-B (_ , i)-B (_ , ii)-B (i , i)-B (i , ii)-B (ii , ii)-B	
I	=	the initial state	(ii , ii)-A	
Succs	=	a function which takes a state as input and returns a set of possible next states available to whoever is due to move	Succs(_,i)-A = { (_,_)-B } Succs(_,ii)-A = { (_,_)-B , (_,i)-B } Succs(i,i)-A = { (_,i)-B } Succs(i,ii)-A = { (_,i)-B (_,ii)-B (i,i)-B} Succs(ii,ii)-A = { (_,ii)-B , (i,ii)-B }	Succs(_,i)-B = { (_,_)-A } Succs(_,ii)-B = { (_,_)-A , (_,i)-A } Succs(i,i)-B = { (_,i)-A } Succs(i,ii)-B = { (_,i)-A , (_,ii)-A (i,i)-A } Succs(ii,ii)-B = { (_,ii)-A , (i,ii)-A }
T	=	a subset of S. It is the terminal states	(_ , _)-A	(_ , _)-B
V	=	Maps from terminal states to real numbers. It is the amount that A wins from B.	V(_ , _)-A = +1	V(_ , _)-B = -1

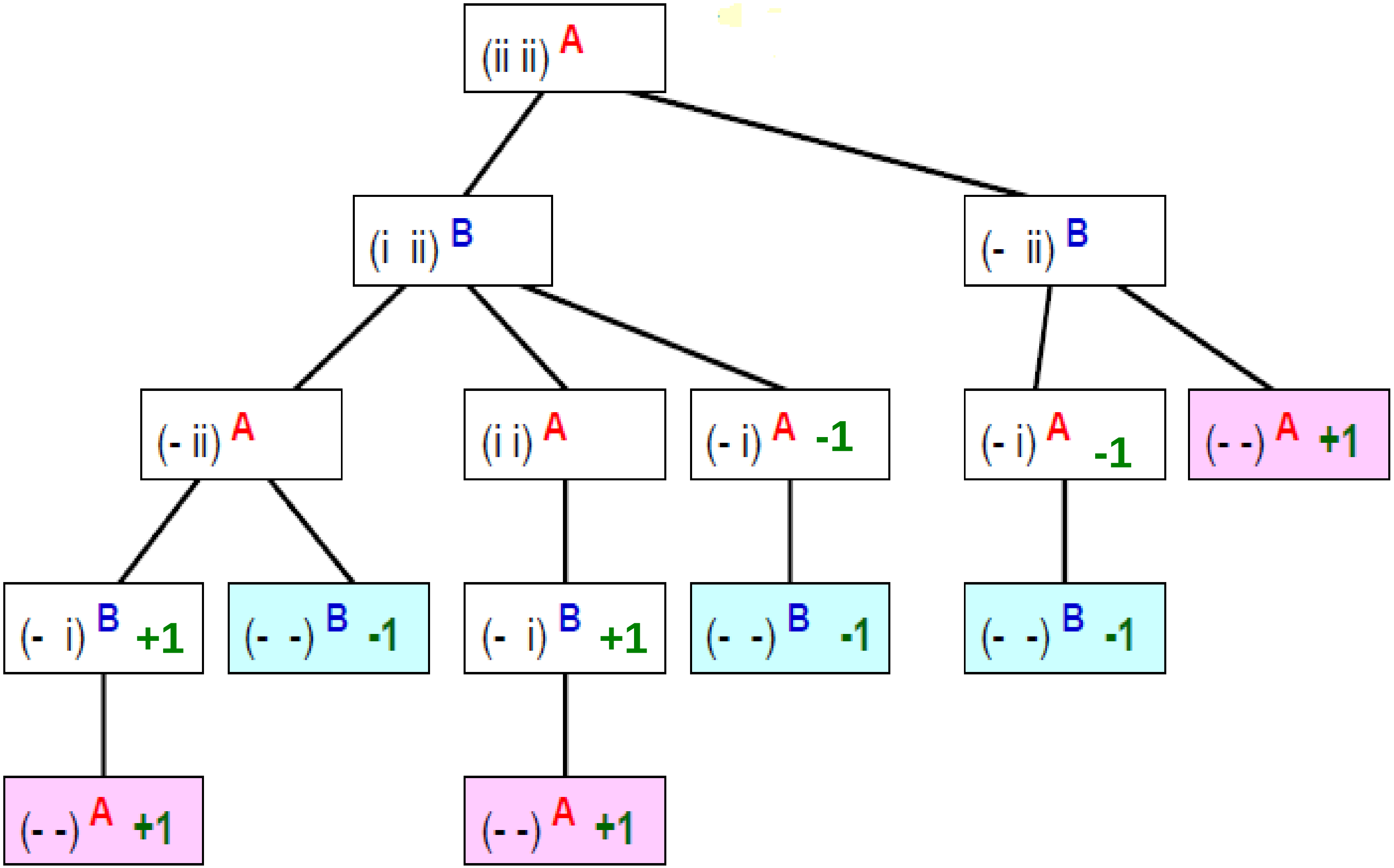

A: MAX
B: MIN
(ii ii) A
(i ii) B
(- ii) B
(- ii) A
(i i) A
(- i) A
(- i) A
(- -) A +1
(- i) B
(- -) B -1
(- i) B
(- -) B -1
(- -) B -1
(- -) A +1
(- -) A +1


(ii ii) A
(i ii) B
(- ii) B
(- ii) A
(i i) A
(- i) A
(- i) A
(- -) A +1
(- i) B +1
(- -) B -1
(- i) B
(- -) B -1
(- -) B -1
(- -) A +1
(- -) A +1

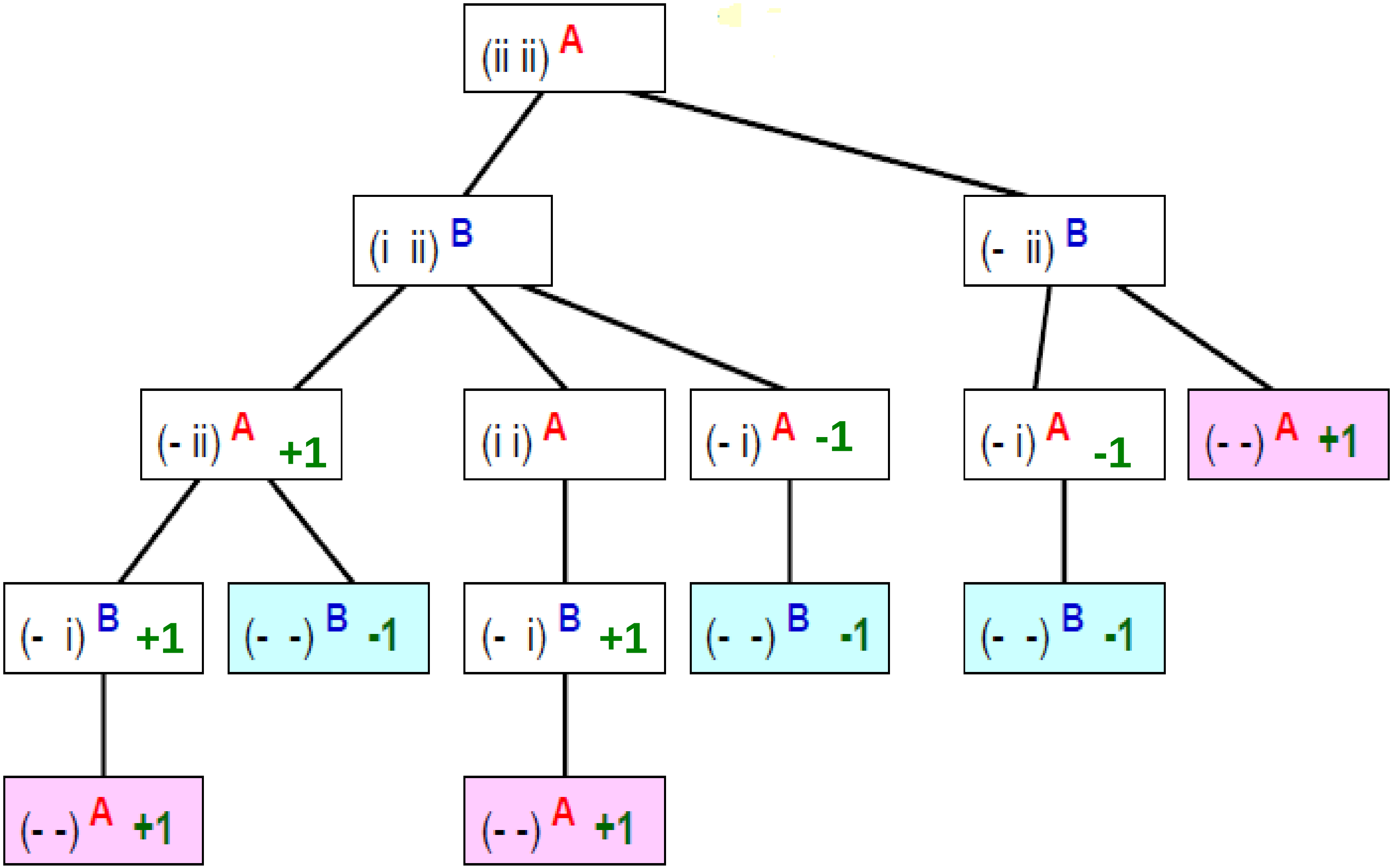


(ii ii) A
(i ii) B
(- ii) B
(- ii) A
(i i) A
(- i) A
(- i) A
(- -) A +1
(- i) B +1
(- -) B -1
(- i) B +1
(- -) B -1
(- -) B -1
(- -) A +1
(- -) A +1

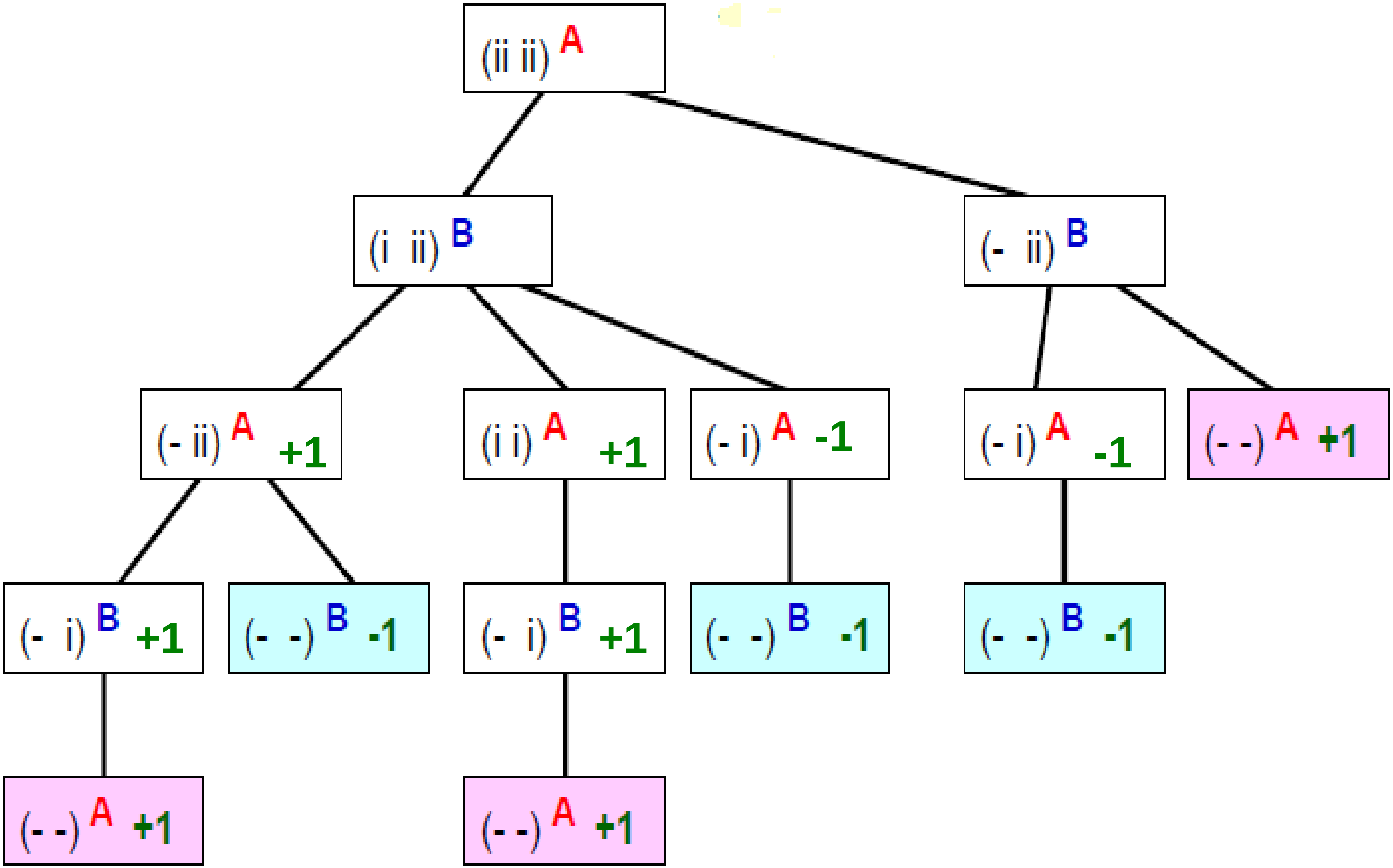

(ii ii) A
(i ii) B
(- ii) B
(- ii) A
(i i) A
(- i) A
(- i) A -1
(- -) A +1
(- i) B +1
(- -) B -1
(- i) B +1
(- -) B -1
(- -) B -1
(- -) A +1
(- -) A +1

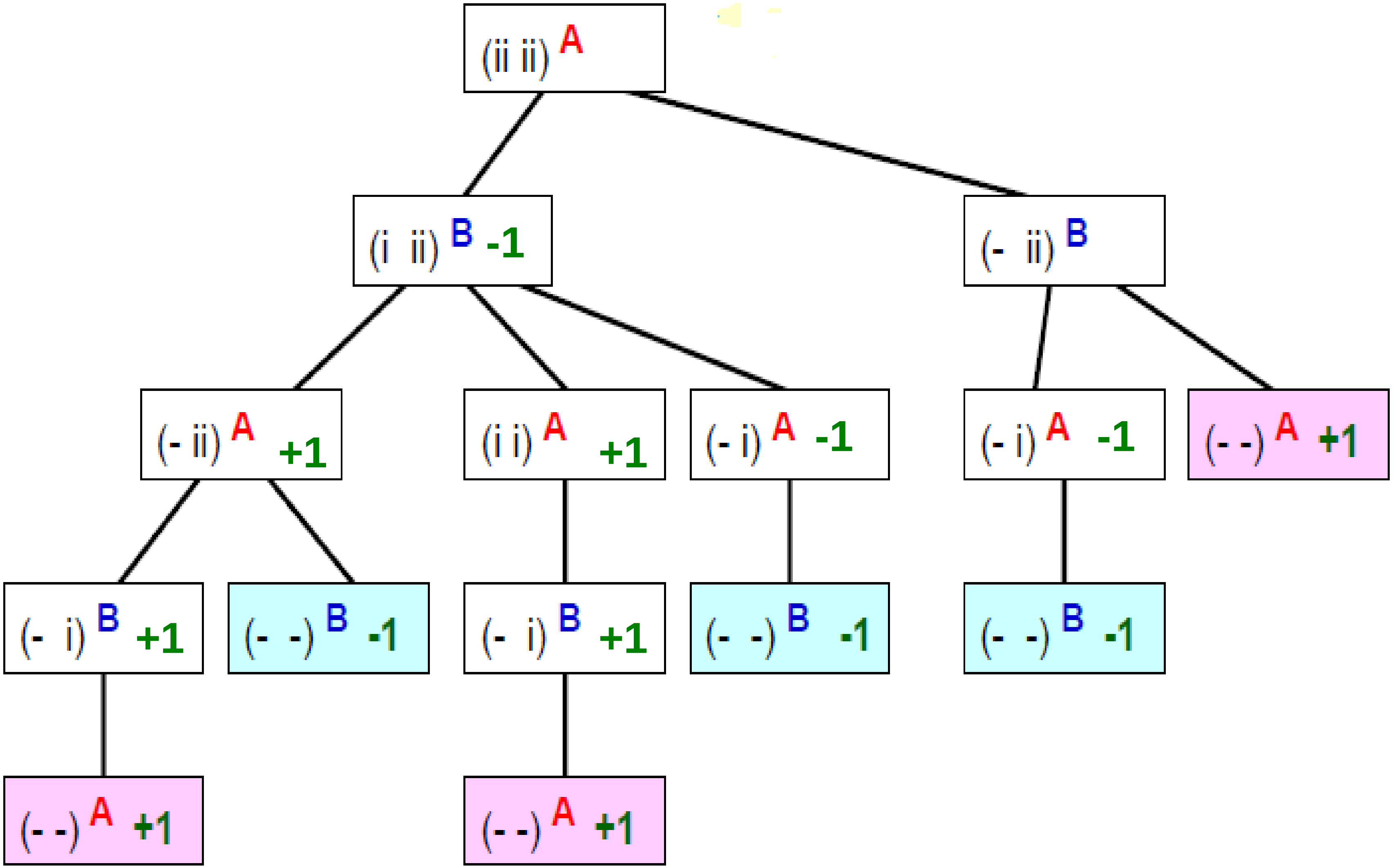


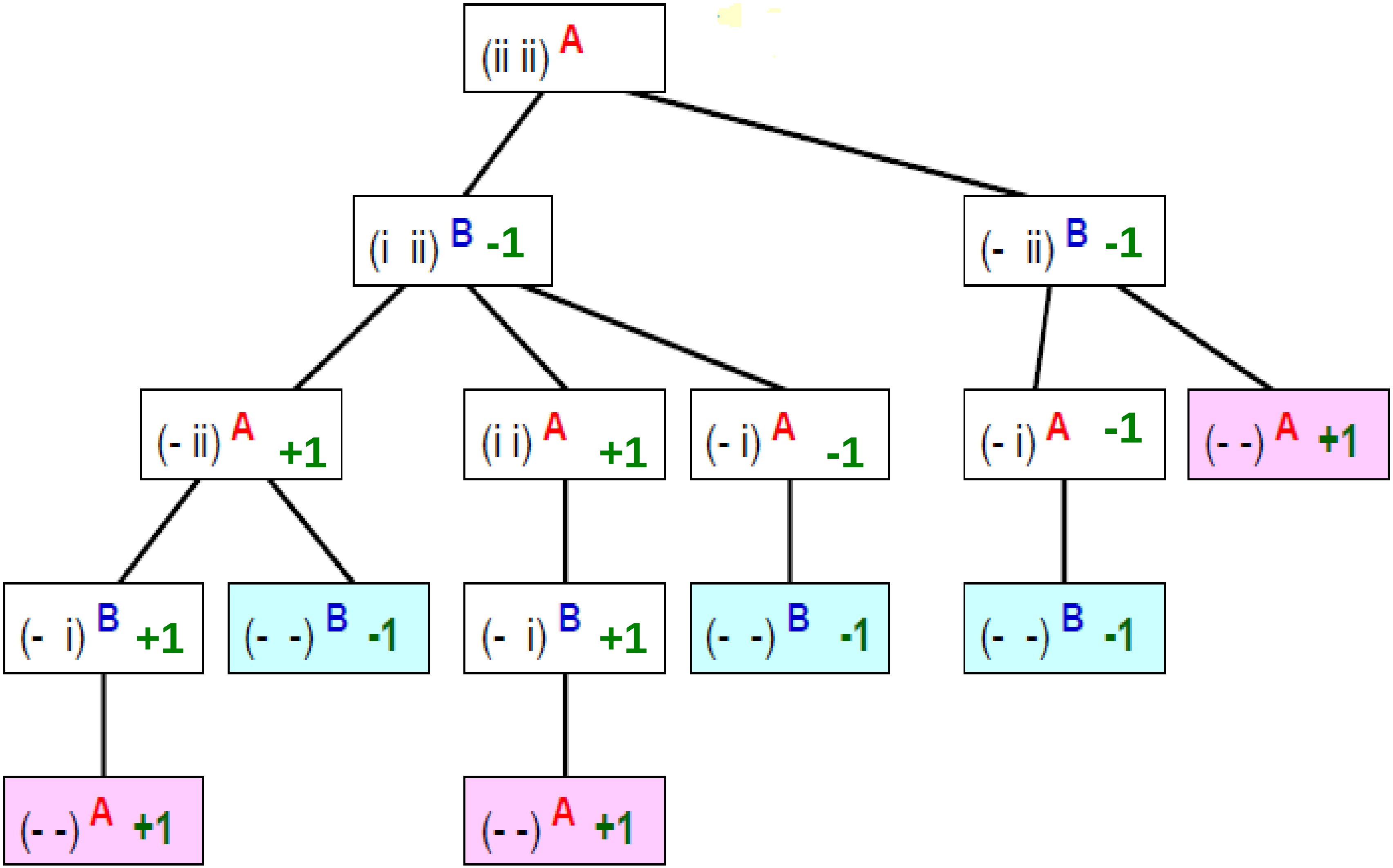
(ii ii) A
(i ii) B
(- ii) B
(- ii) A
(i i) A
(- i) A -1
(- i) A -1
(- -) A +1
(- i) B +1
(- -) B -1
(- i) B +1
(- -) B -1
(- -) B -1
(- -) A +1
(- -) A +1



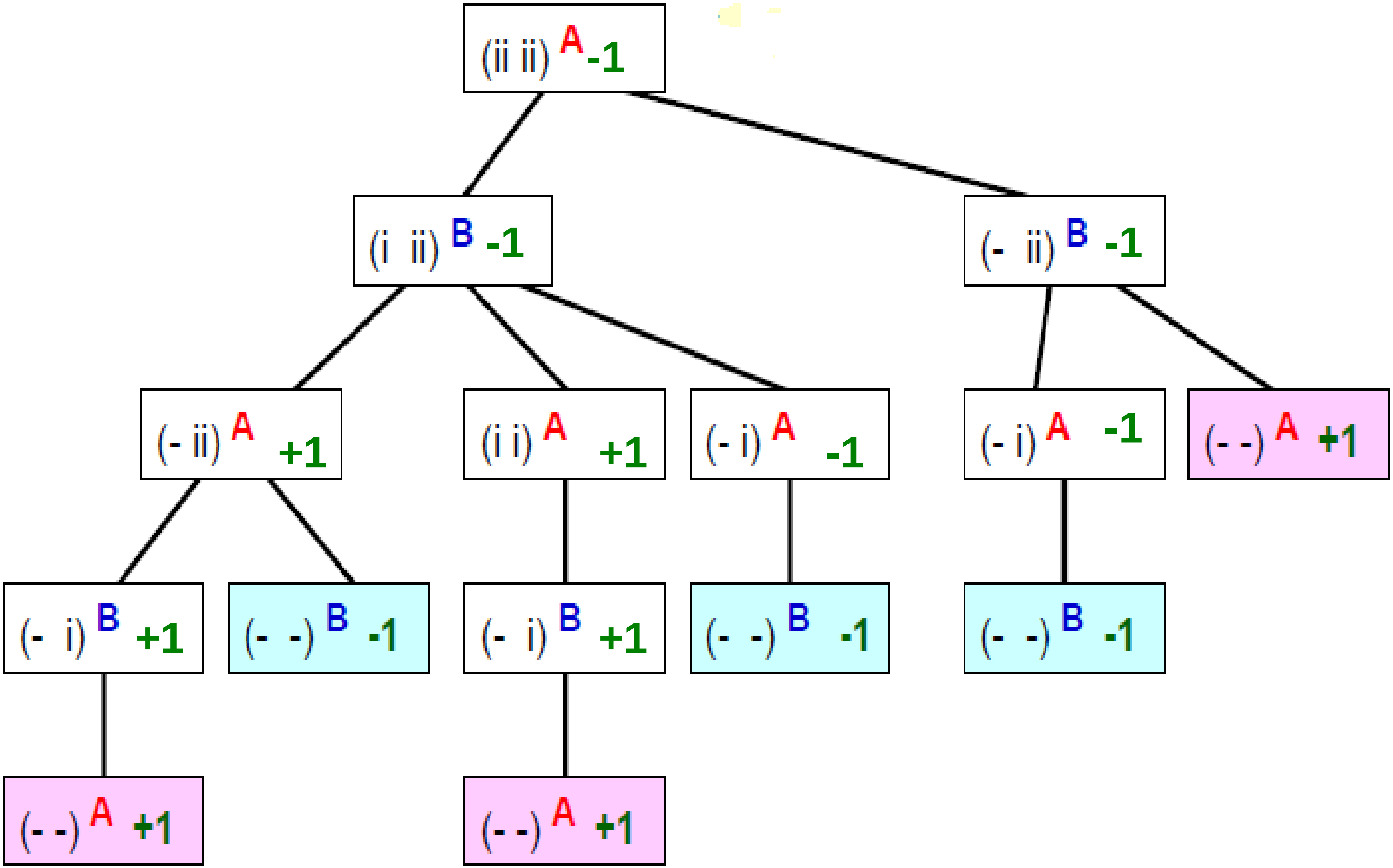
(ii ii) A
(i ii) B
(- ii) B
(- ii) A +1
(i i) A
(- i) A -1
(- i) A -1
(- -) A +1
(- i) B +1
(- -) B -1
(- i) B +1
(- -) B -1
(- -) B -1
(- -) A +1
(- -) A +1


(ii ii) A
(i ii) B
(- ii) B
(- ii) A +1
(i i) A +1
(- i) A -1
(- i) A -1
(- -) A +1
(- i) B +1
(- -) B -1
(- i) B +1
(- -) B -1
(- -) B -1
(- -) A +1
(- -) A +1


(ii ii) A
(i ii) B -1
(- ii) B
(- ii) A +1
(i i) A +1
(- i) A -1
(- i) A -1
(- -) A +1
(- i) B +1
(- -) B -1
(- i) B +1
(- -) B -1
(- -) B -1
(- -) A +1
(- -) A +1



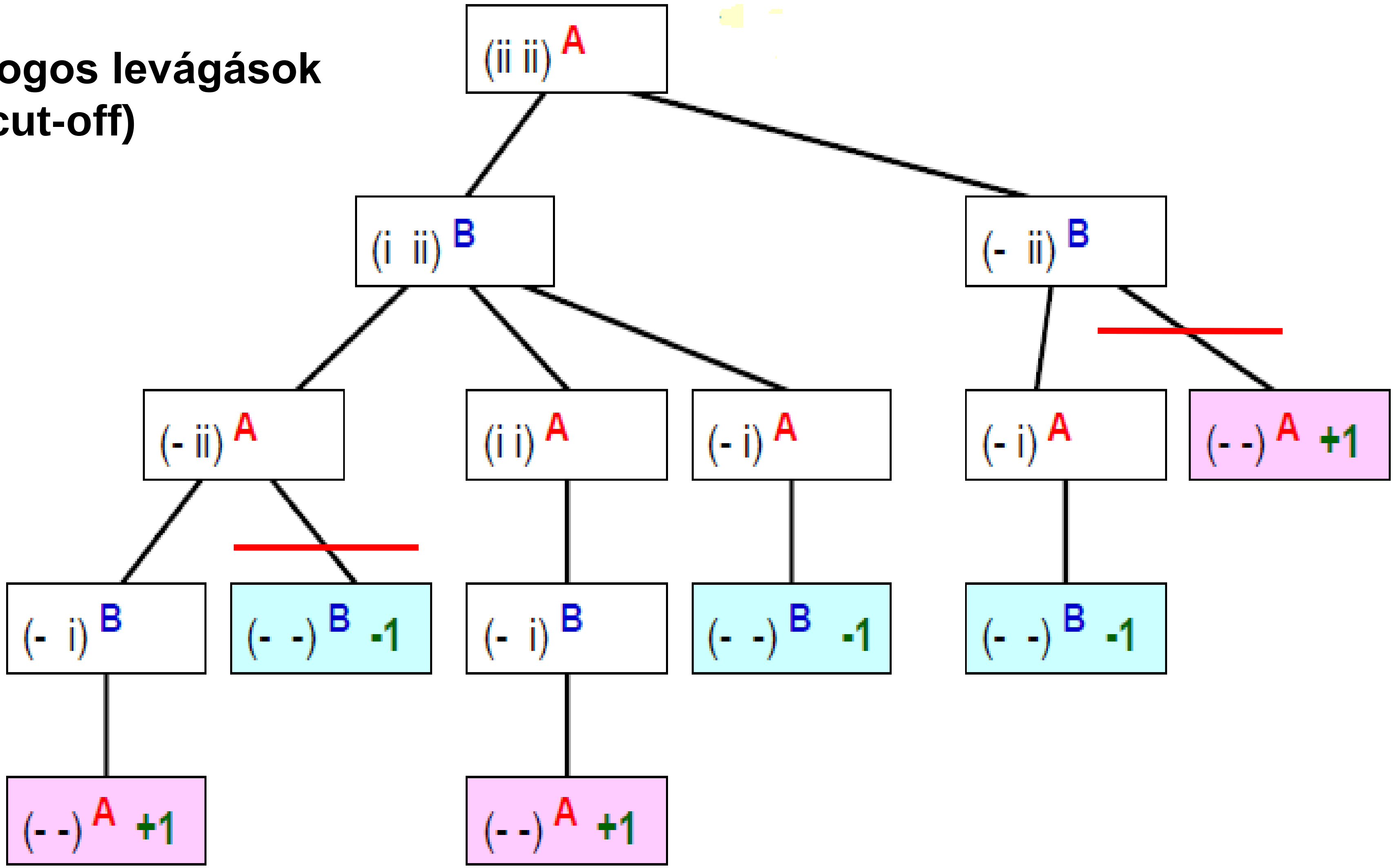
(ii ii) A
(i ii) B -1
(- ii) B -1
(- ii) A +1
(i i) A +1
(- i) A -1
(- i) A -1
(- -) A +1
(- i) B +1
(- -) B -1
(- i) B +1
(- -) B -1
(- -) B -1
(- -) A +1
(- -) A +1



(A játék megoldása: Aki kezd, az veszít, ha az ellenfél optimálisan játszik)

Minimax algoritmus *Spórolhatunk-e a számításokból?*

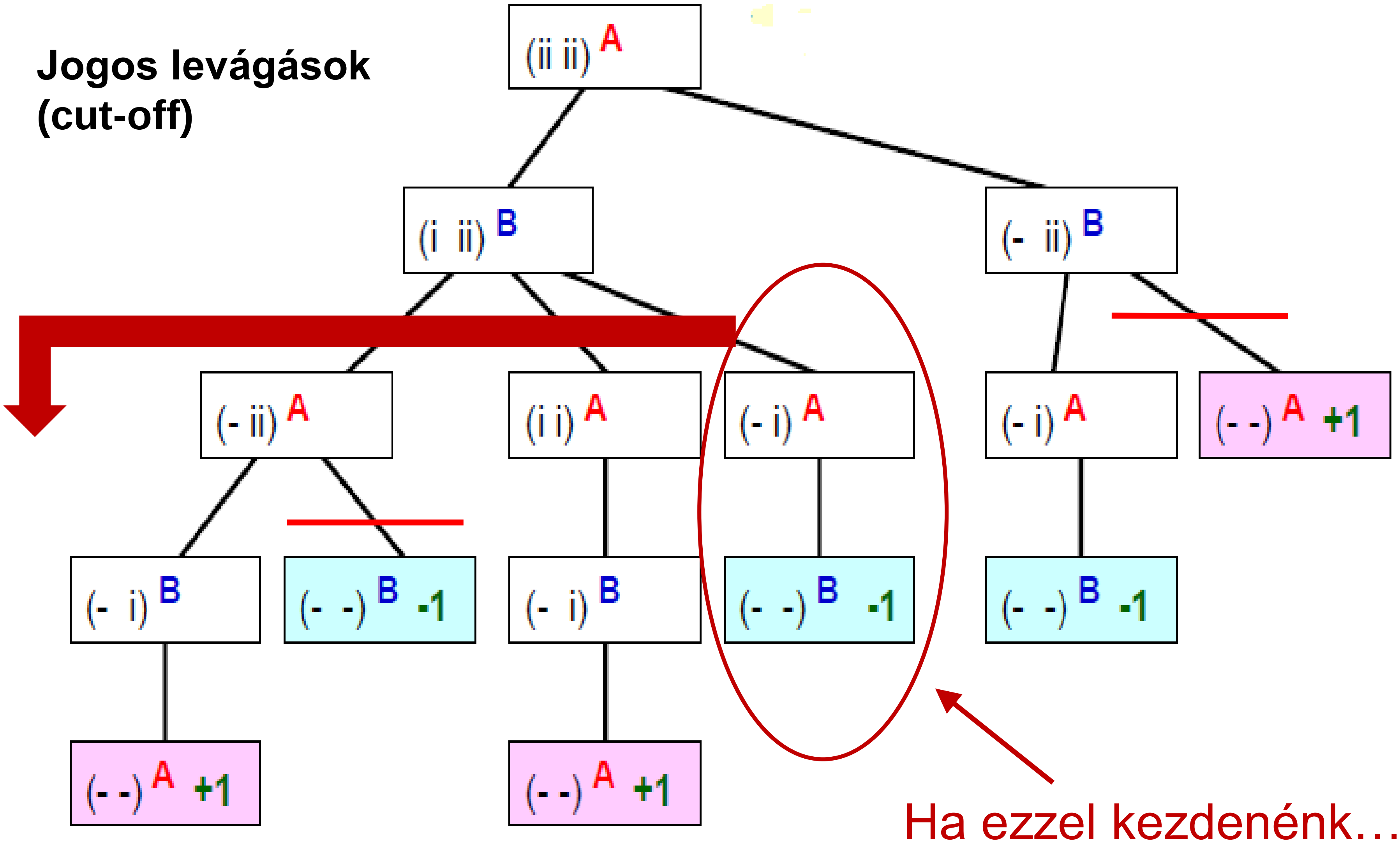
Jogos levágások (cut-off)



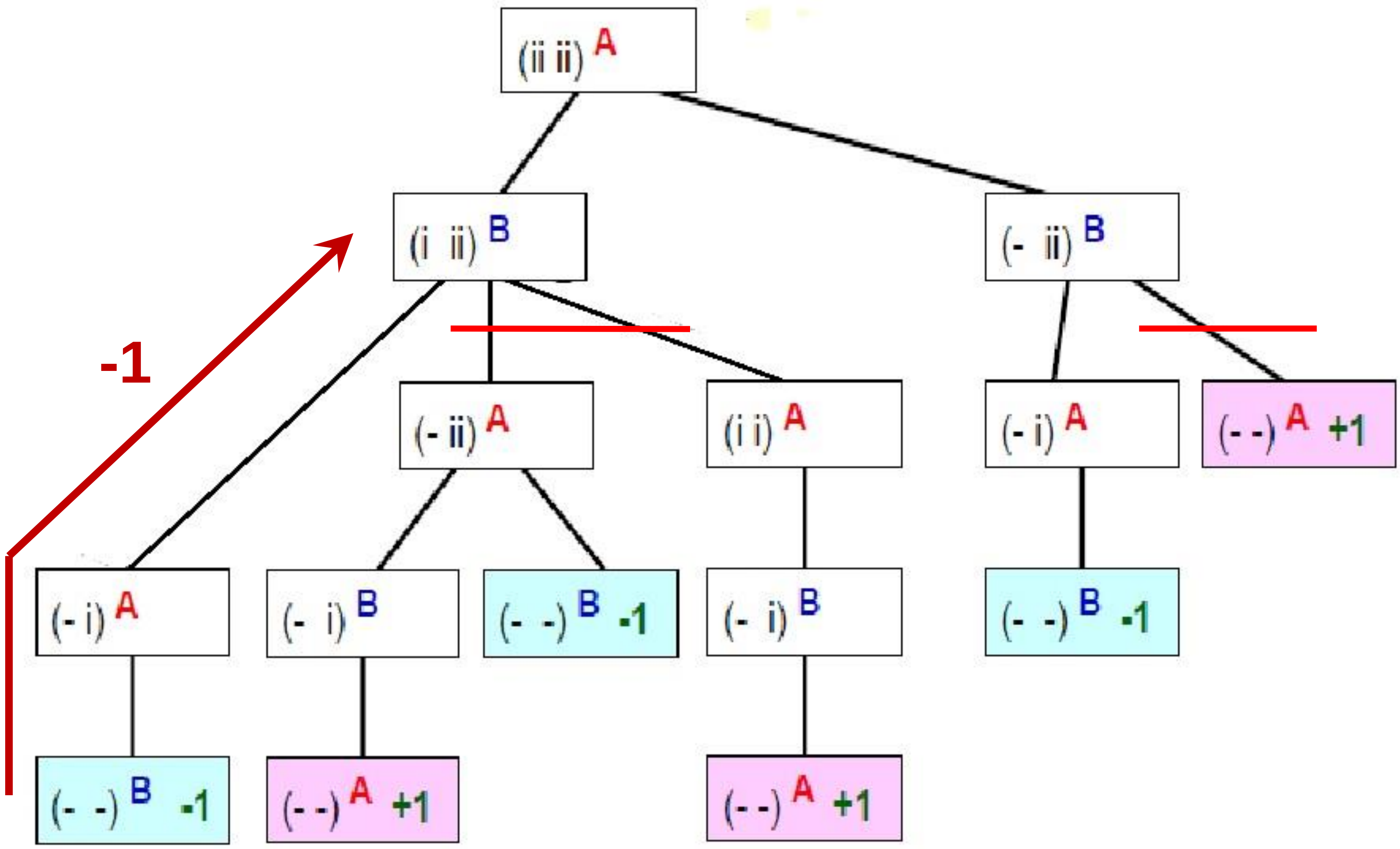
Minimax algoritmus

Tegyük fel, hogy tudjuk, hogy a játék kimenetele csakis -1 és 1. Spórolhatunk-e a számításokból?

Jogos levágások (cut-off)

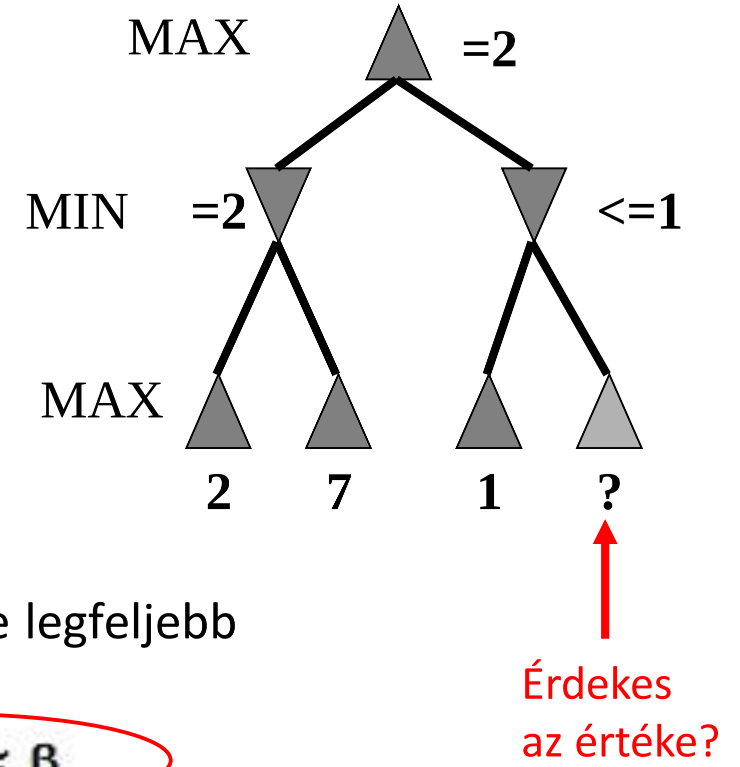


Minimax algoritmus

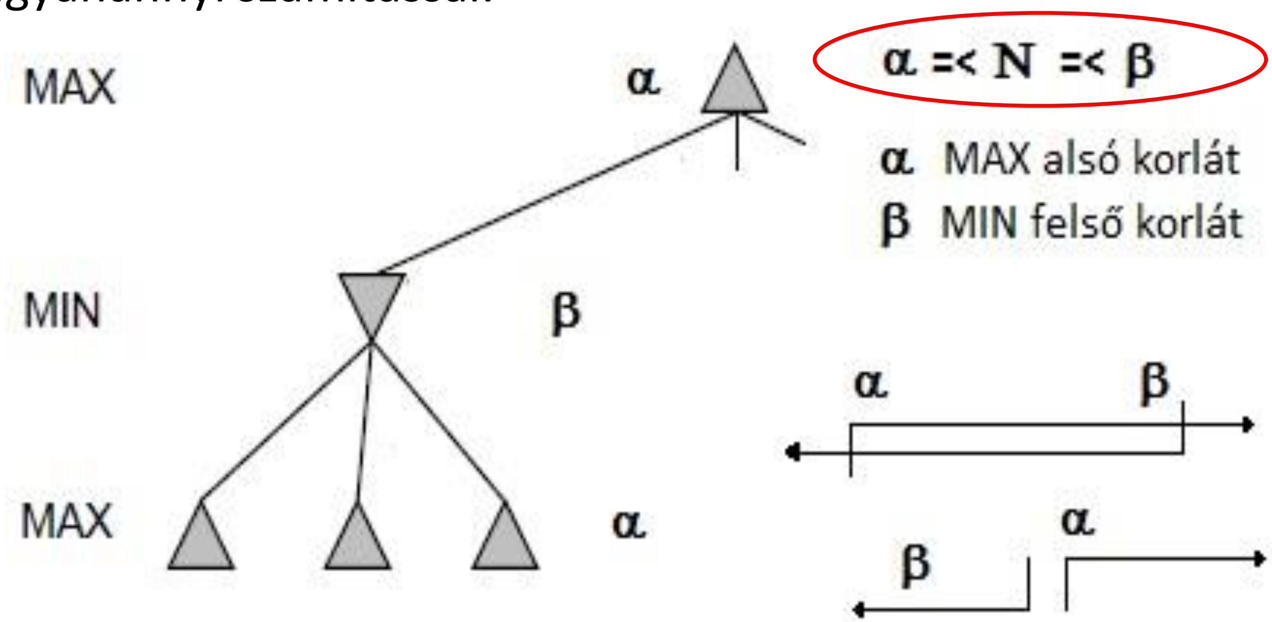


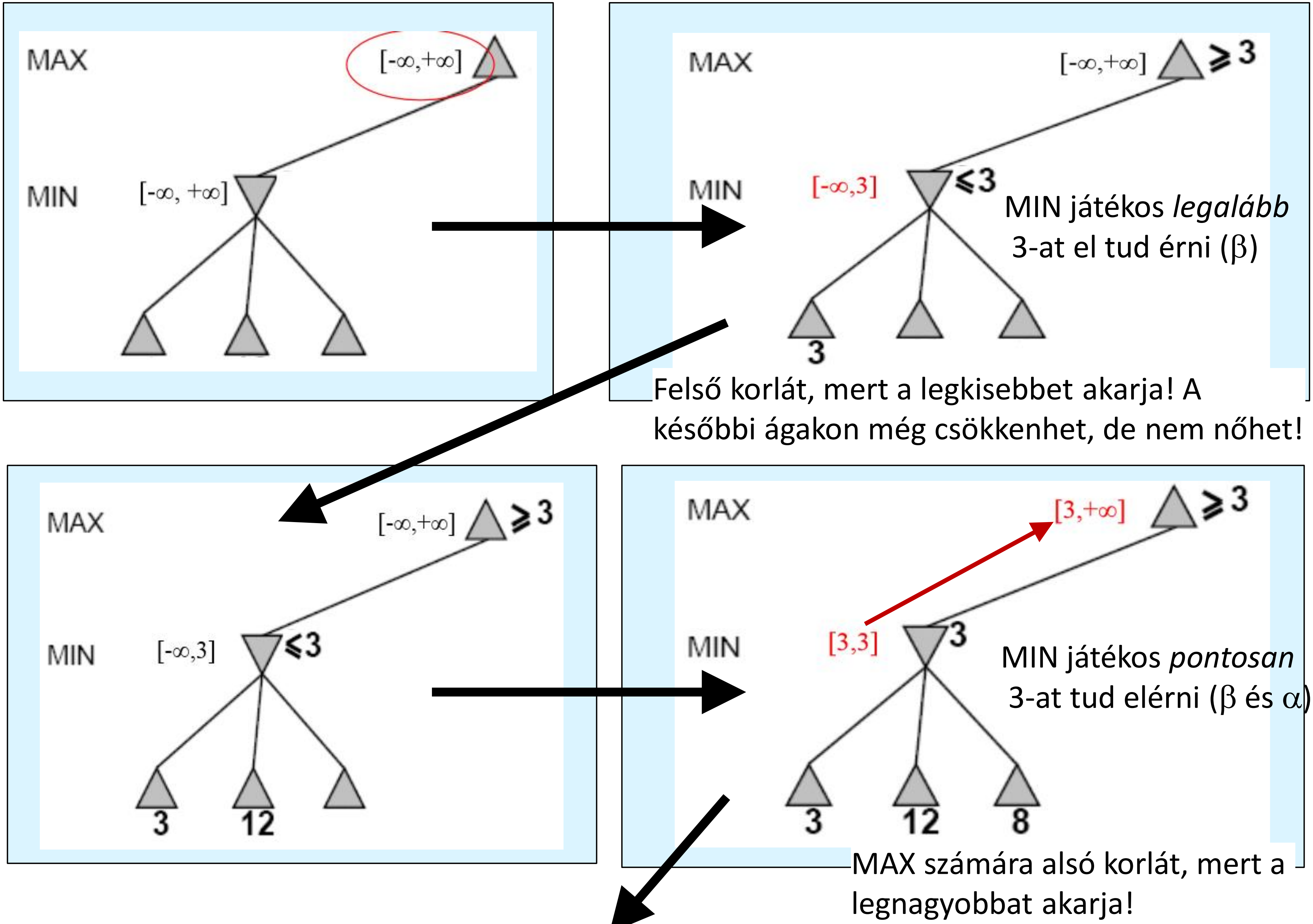
Minimax és alfa-béta nyesés

- Játékfa generálása mélységi keresés jelleggel, L mélységig.
- Ha lehetséges, a kiértékelő függvény-értékek becsléseinek (alfa és béta értékek) visszaterjesztése .
- A végleges döntést befolyásolni nem képes ágak nyesése.

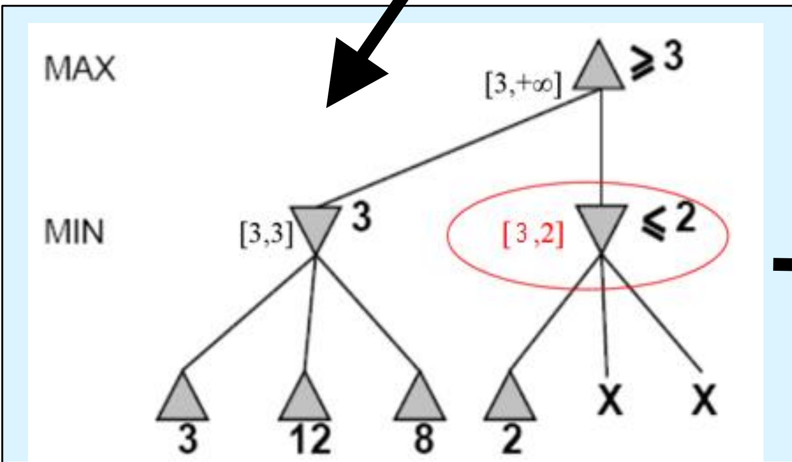


Garantált a minimax eredmény, de kevesebb, illetve legfeljebb ugyanannyi számítással.

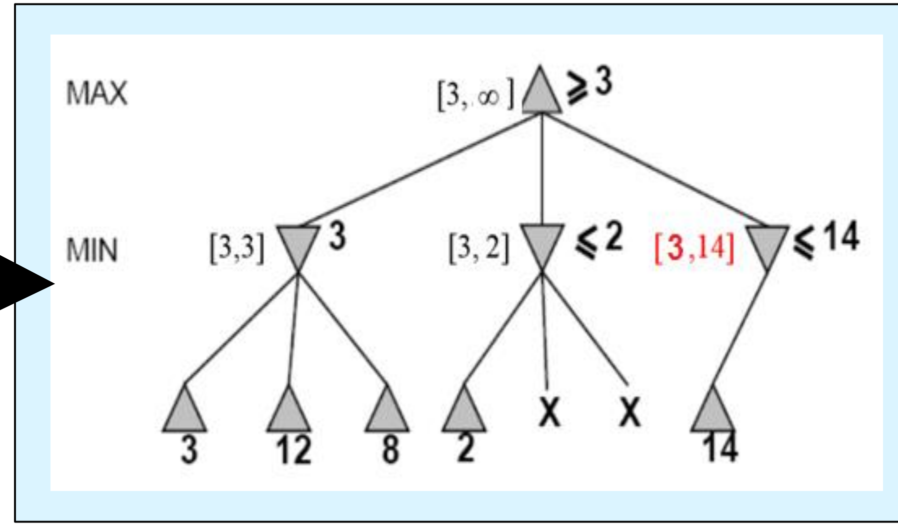


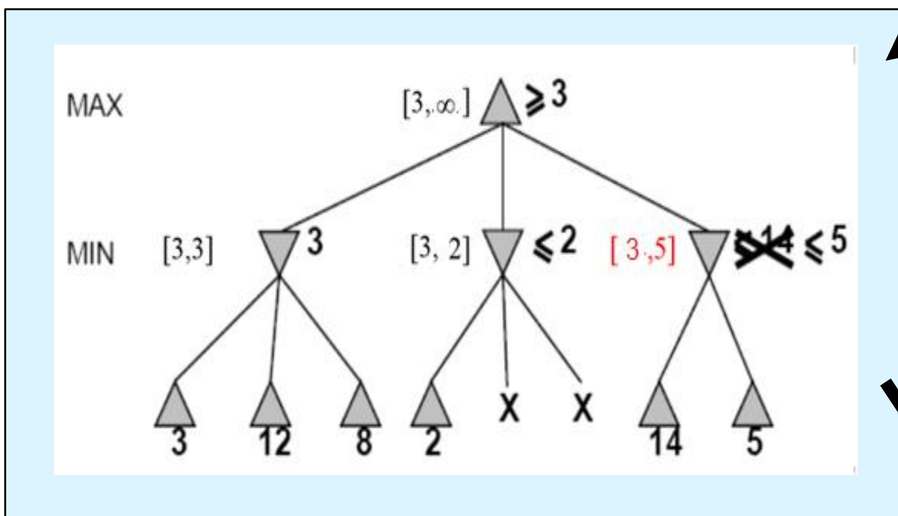


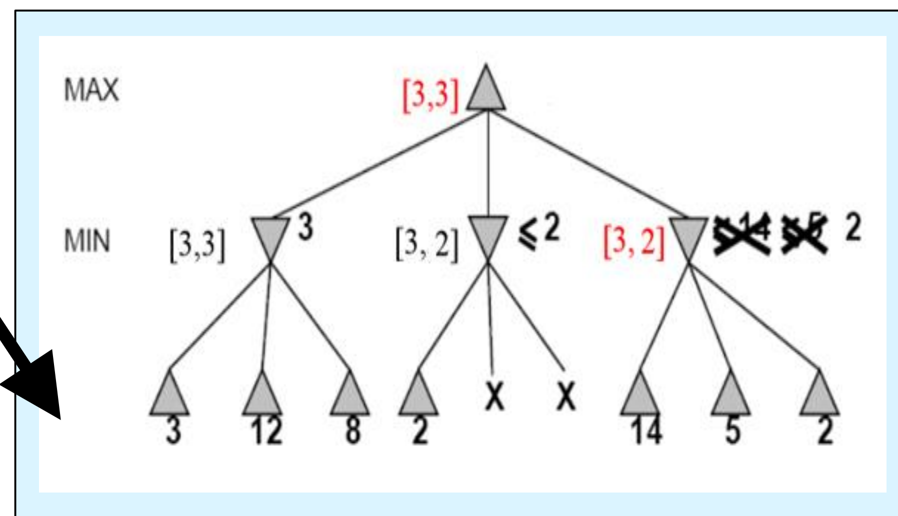
MAX
[-∞,+∞]
MIN
[-∞, +∞]
MAX
[-∞,+∞]
≥3
MIN
[-∞,3]
≤3
3
MIN játékos *legalább* 3-at el tud érni (β)
Felső korlát, mert a legkisebbet akarja! A későbbi ágakon még csökkenhet, de nem nőhet!
MAX
[-∞,+∞]
≥3
MIN
[-∞,3]
≤3
3
12
MAX
[3,+∞]
≥3
MIN
[3,3]
3
3
12
8
MIN játékos *pontosan* 3-at tud elérni (β és α)
MAX számára alsó korlát, mert a legnagyobbat akarja!



Biztosan nem fogunk ebbe az irányba menni, mert MIN 2-t (vagy ha később van kisebb, akkor azt) választaná, ezért MAX az első (baloldali) irányba indul! Nem kell ezzel a részfával foglalkoznunk!



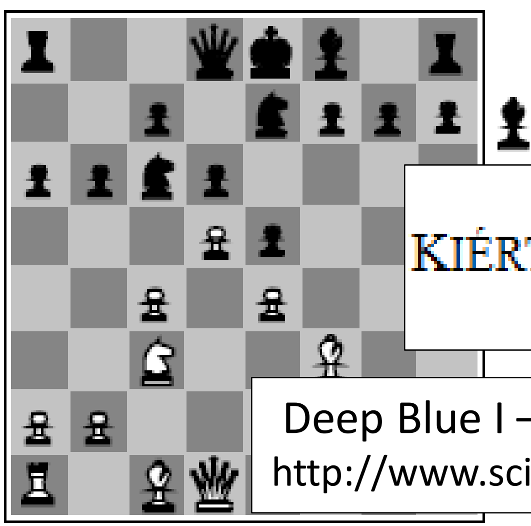




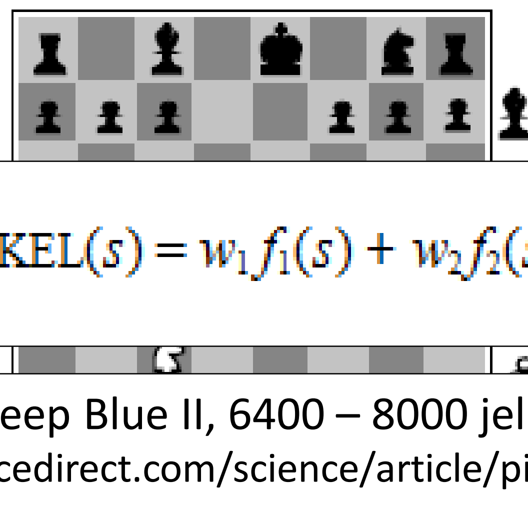
http://inst.eecs.berkeley.edu/~cs61b/fa14
/ta-materials/apps/ab_tree_practice/

Kiértékelő függvények : Állapot → mérőszám (érték) leképezés.

- Minél nagyobb a mérőszám, annál értékesebb az állapot (pozíció). Adott állapot minősítése – egy bizonyos mélységű keresési fa bejárása, ahol *a levelek nem a játék végállapotai, hanem a kiértékelő függvénnyel minősített közbülső állapotok*.
- Játék elején (a célállapottól messze a kiértékelő függvény pontatlan, valamilyen mélységű kereséssel kell párosítani.
- Játék végén (a célállapothoz közel) a kiértékelő függvény olyan pontos lehet, hogy az állapot jellemzésére akár közvetlenül is alkalmazható.



Black to move
White slightly better



White to move
Black winning

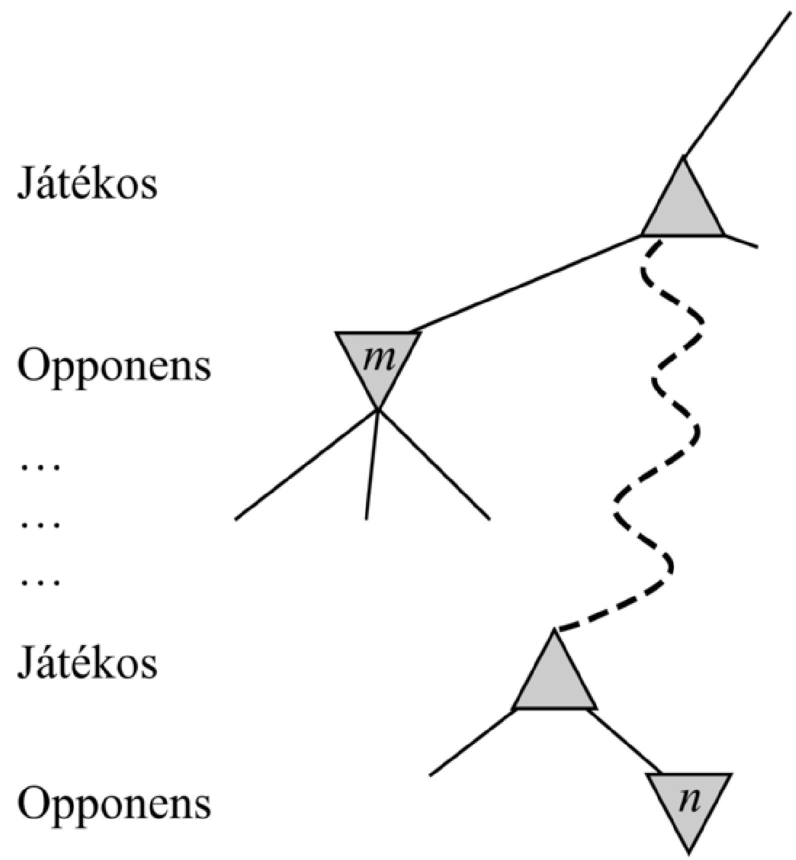
$$\text{KIÉRTÉKEL}(s) = w_1 f_1(s) + w_2 f_2(s) + \ldots + w_n f_n(s) = \sum_{i-1}^{n} w_i f_i(s)$$

Deep Blue I – Deep Blue II, 6400 – 8000 jellegfüggvény (chess chip)
http://www.sciencedirect.com/science/article/pii/S0004370201001291

Minimax és alfa-béta nyesés

- A nyesés a végeredményt nem befolyásolja.
- Nem mindegy, hogy milyen sorrendben értékeljük ki a csomópontokat!
- A jó lépésrendezés fokozza a nyesés hatékonyságát
- „Tökéletes rendezéssel" az idő-komplexitás = $O(b^{d/2})$

A keresés mélységét duplázza: ha eddig *d*-t tudtunk elérni az erőforrásainkkal, akkor most 2**d*-t érhetünk el.



Ha *m* a Játékos számára jobb, mint *n*, akkor a játék során sosem fogunk elérni *n*-be.

Két megjegyzés:

„Tökéletes rendezést” tipikusan nem tudunk elérni
Átlagosan = $O(b^{3d/4})$ az elérhető.

A sakknál tökéletes rendezés esetén $O(b^{d/2}) = O((\sqrt{b})^{d})$,
ez másképpen megfogalmazva olyan, mintha az effektív *b* az
eredeti 35 helyett: $b = \sqrt{35} \approx 6$ lenne.