Ekspertski sistemi
Vežbe

Teorija igara

Mart 2013.

Program vežbi

- Algoritmi pretraživanja
- Teorija igara
- Formalna logika
- Produkcioni sistemi
- Strategije rešavanja problema
- Uvod u mašinsko učenje
- Rad u neizvesnom okruženju

Uvod

- Jedna od prvih primena veštačke inteligencije
- Prve igre koje su dobile AI igrače:
  šah, X-O, Go, Gomoku, Othello
- Danas: Fifa, Call of Duty, Warcraft, ...

Teorija igara

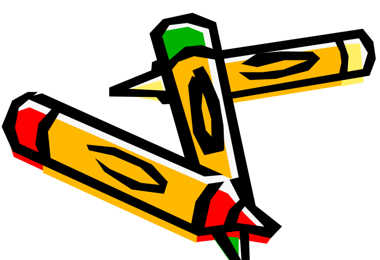
- Grana primenjene matematike
- Pokušava da matematički opiše ponašanje u strateškim situacijama, u kojima uspeh pojedinca u donošenju odluke zavisi od delanja drugih učesnika
- Ne koristi se samo u igrama
  - ekonomija, biologija, politika...
- Primer: **Nash equilibrium**
  - John Forbes Nash, Jr. (film Blistavi um)
  - Dobitnik Nobelove nagrade za ekonomiju 1994.

Nash-ova teorema

- Skup strategija je Nešov ekvilibrijum ako nijedan igrač ne može da napreduje ako jednostrano promeni svoju strategiju.
- „Ako znam strategije drugih igrača, i znam da se one neće menjati, da li ću imati koristi od promene moje strategije?"

Nash-ova teorema

- Ako je odgovor „Da" onda takav skup NIJE strategija Nešov ekvilibrijum.
- Ali, ako svaki igrač ne želi da promeni strategiju (ili mu je svejedno), takav skup JESTE Nešov ekvilibrijum
- Svaka strategija u ekvilibrijumu je najbolji odgovor na ostale strategije u ekvilibijrumu



Strategija

- Strategija igrača u teoriji igara predstavlja jednu od mogućnosti koju može da izabere u situaciji koja ne zavisi samo od njegovih akcija, već i od akcija drugih učesnika u igri.
- Strategija odlučuje koje će akcije igrač izvesti u svakom trenutku igre

Strategija

- Strategija nije **potez**.
- Potez predstavlja akciju igrača u nekom trenutku u toku igranja igre.
  - na primer: izabrati polje u X-O igri
- **Strategija** je *algoritam* za igranje cele igre, koja govori igraču koji **potez** da izabere u svakoj mogućoj situaciji.

Vrste strategija

- Čista strategija (**pure strategy**) daje kompletne podatke kako će igrač igrati igru.
- Određuje potez koji će igrač izabrati u kakvoj god situaciji da se nađe.
- **Skup strategija** jednog igrača je skup čistih strategija koje su mu dostupne.

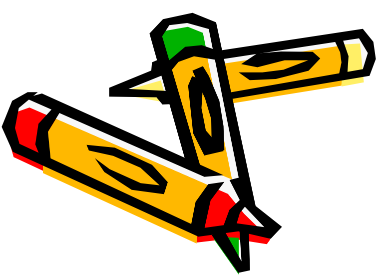
Vrste strategija

- Mešovita strategija (**mixed strategy)** predstavlja dodelu verovatnoće svakoj čistoj strategiji.
- Ovo omogućava igraču da nasumično izabere čistu strategiju.
- Pošto je verovatnoća kontinualna veličina, postoji beskonačno mnogo mešovitih strategija

Matrica isplativosti

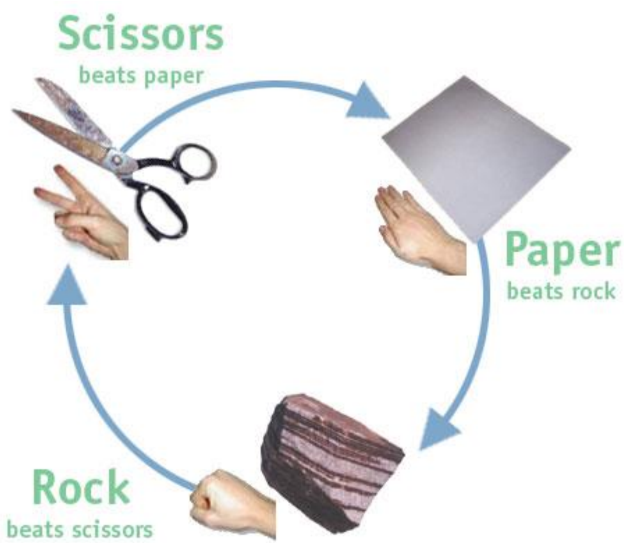
- **Payoff matrix** predstavlja reprezentaciju igre na osnovu strategija i funkcija isplativosti
- Funkcija isplativosti je dobitak igrača ukoliko su izabrane strategije u čijem se preseku dobitak nalazi

	Strategija 1	Strategija 2
Strategija 1	(2,2)	(4,1)
Strategija 2	(1,4)	(3,3)



Simultane jednopotezne igre

- Igrači moraju izabrati poteze istovremeno, ne znajući poteze ostalih igrača



Igrač 1

Igrač 2

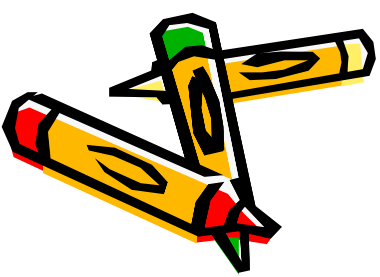
			
	0,0	1,-1	-1,1
	-1,1	0,0	1,-1
	1,-1	-1,1	0,0

Payoff matrica

Zatvorenikova dilema

- Dva kriminalca su uhapšena i policija ih ispituje odvojeno.
- Ako jedan svedoči protiv drugog, a drugi odbije da svedoči, onaj ko je svedočio odlazi slobodan, a onaj ko je odbio biva osuđen na 10 godina zatvora.
- Ako obojica svedoče jedan protiv drugog, obojica dobijaju po 5 godina zavtora.
- Ako odbojica odbiju da svedoče, dobijaju po jednu godinu zatvora

	Krimos 1: Svedoči	Krimos 2: Odbije
Krimos 1: Svedoči	-5,-5	-10,0
Krimos 2: Odbije	0,-10	-1,-1



Zatvorenikova dilema

- Razmišljanje Krimosa 1:
  - Recimo da Krimos 2 svedoči protiv mene. Onda ja dobijam 5 godina zatvora ako svedočim protiv njega, a 10 godina ako odbijem da svedočim. Bolje mi je da svedočim.
  - Recimo da Krimos 2 odbije da svedoči protiv mene. Ako ja svedočim protiv njega, onda odlazim slobodan, a ako i ja odbijem da svedočim, dobijam 1 godinu zatvora. Bolje mi je da svedočim.
- **Dominantna strategija:** Strategija koja je najbolja za igrača bez obzira na strategiju koju je izabrao drugi igrač

	Krimos 1: Svedoči	Krimos 2: Odbije
Krimos 1: Svedoči	-5,-5	-10,0
Krimos 2: Odbije	0,-10	-1,-1

Dominantna strategija

- Ako postoji striktno dominantna strategija za jednog igrača, taj igrač će je koristiti u svim Nešovim ekvilibrijumima.
- Ako oba igrača imaju striktno dominantnu strategiju, igra ima jedinstven Nešov ekvilibrijum.

Dominantna strategija

- Takav Nešov ekvilibrijum nije uvek Pareto optimalan
  - možda postoji izbor u igri koji nije u ekvilibrijumu, a koji bi bio bolji za oba igrača
- Pareto optimalnost
  - niko ne može da napreduje,
    a da bar nekom od učesnika ne bude lošije

Prisoner's dilemma

- **Nešov ekvilibrijum:** par strategija takav da nijedan igrač neće imati veći dobitak ako promeni strategiju, ako se drugi igrač drži svoje strategije
  - (Svedoči, svedoči) je ekvilibrijum dominantne strategije
- **Pareto optimalni ishod: nije**, jer ako obojica odbiju da svedoče, zajedno će proći bolje

	Krimos 1: Svedoči	**Krimos 2: Odbije**
Krimos 1: Svedoči	-5,-5	-10,0
Krimos 2: Odbije	0,-10	-1,-1

Zatvorenikova dilema u realnom životu

- Rat cena
- Trka oružja
- Upotreba steroida
- Kontrola zagađenja

	Sarađuj	Odbij
Sarađuj	pobeda – pobeda	velika pobeda – veliki poraz
Odbij	veliki poraz – velika pobeda	poraz – poraz

Trka oružja

- Recimo da dve zemlje potpišu sporazum da se više neće naoružavati. Svačiji interes je da to poštuju, pod pretpostavkom da i drugi poštuju sporazum, jer bi taj novac mogli daulože u zdravstveni sistem, ekologiju, ...
- Međutim, svako može da strahuje da druga zemlja krši sporazum i da nastavlja sa naoružavanjem ☹

		Zemlja #2	
		Nastavi naoružavanje	Poštuj sporazum
Zemlja #1	Nastavi naoružavanje	(2,2)	(4,1)
	Poštuj sporazum	(1,4)	(3,3)

Trka oružja

- Recimo da Zemlja #1 izabere da poštuje sporazum, dok Zemlja #2 nastavi da se naoružava.
- Onda bi Zemlja #2 bila u prednosti, jer nastavlja da jača svoju vojsku.
- Nešov ekvilibrijum za ovu igru je da obe zemlje nastave sa naoružavanjem.
- Rešenje: međusobna inspekcija.

Zemlja #2

Zemlja #1

	Nastavi naoružavanje	Poštuj sporazum
Nastavi naoružavanje	(2,2)	(4,1)
Poštuj sporazum	(1,4)	(3,3)

Lov na jelena

	Lovac 1: jelen	Lovac 2: zec
Lovac 2: jelen	2,2	1,0
Lovac 2: zec	0,1	1,1

- Postoji li dominantna strategija?
- Postoji li Nešov ekvilibrijum?
  - (jelen, jelen) **payoff dominant** strategija
  - (zec, zec) **risk dominant** strategija
- Model kooperativnog delanja

Zatvorenikova dilema naspram Lova na jelena

Prisoner' dilemma

	Sarađuj	Odbij
Sarađuj	pobeda, pobeda	velika pobeda, veliki poraz
Odbij	veliki poraz, velika pobeda	poraz, poraz

Igrači dobijaju ako jednostrano odbiju

Stag hunt

	Lovac 1: jelen	Lovac 2: zec
Lovac 2: jelen	v. pobeda, v. pobeda	pobeda, poraz
Lovac 2: zec	poraz, pobeda	pobeda, pobeda

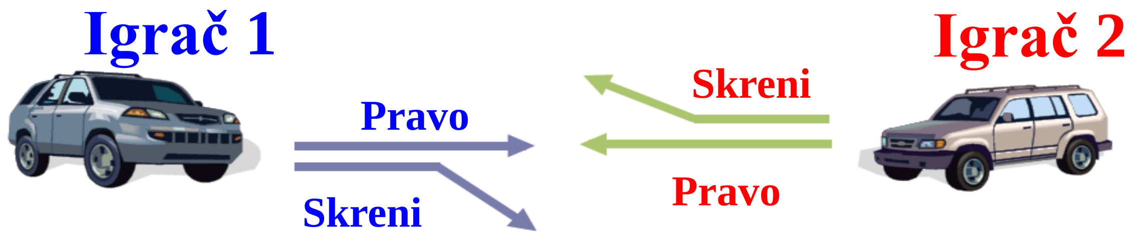
Igrači gube ako jednostrano odbiju

Coordination game (Bitka polova)

	Žena: Balet	Žena: Fudbal
Muž: Balet	3, 2	0, 0
Muž: Fudbal	0, 0	2, 3

- Nešov ekvilibrijum?
  - (Balet, balet) ili (fudbal, fudbal)
- Kako izabrati ekvilibrijum?
- Poželjno je imati istu strategiju.

Igra kukavica (chicken)

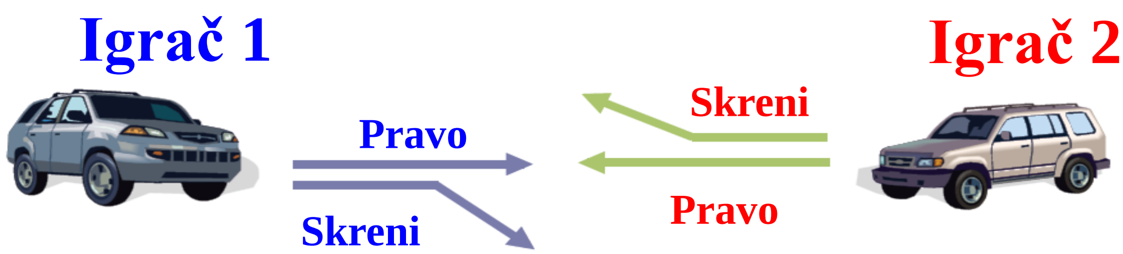


- Postoji li dominantna strategija za svakog igrača? - Ne
- Nešov ekvilibrijum

  (pravo, skreni) ili (skreni, pravo)
- *Anti-coordination* igra: obostrano korisno je da se izabere različita strategija
- Kako izabrati strategiju?
  - Unapred se obavezati ili koristiti pretnju ☺

	Pravo	Skreni
Pravo	-10, -10	-1, 1
Skreni	1, -1	0, 0

Mešovita strategija ekvilibrijuma



	Pravo	Skreni
Pravo	-10, -10	-1, 1
Skreni	1, -1	0, 0

- **Mešovita strategija:** igrač bira strategiju prema verovatnoći
- Recimo da svako bira **Pravo** sa verovatnoćom 1/10.
- Razmotrimo dobitak Igrača 1 dok Igrač 2 ne menja strategiju
  - Ako izabere **Pravo** dobitak je: (1/10)(−10) + (9/10)1 = −1/10
  - Ako izabere **Skreni** dobitak je (1/10)(−1) + (9/10)0 = −1/10

Pronalaženje mešovite strategije

	I1: Pravo sa verovat. p	**I1**: Skreni sa verovat. $1-p$
I2: Pravo sa verovat. q	-10, -10	-1, 1
I2: Skreni sa verovat. $1-q$	1, -1	0, 0

- Očekivani dobitak I1, ako je poznata strategija I2:

  I1 bira Pravo: $q(-10) + (1-q)1 = -11q + 1$

  I1 bira Skreni: $q(-1) + (1-q)0 = -q$

- Da bi strategija I2 bila deo Nešovog ekvilibrijuma, I1 mora da bude ravnodušan prema izboru svoji akcija:

  $-11q + 1 = -q$ or $q = 1/10$

  Slično, dobijamo i $p = 1/10$

Nešov ekvilibrijum i racionalno donošenje odluka

- Ako igra ima jedinstveni Nešov ekvilibrijum, on će biti prihvaćen, ako svaki igrač:
  - je racionalan i payoff matrix je tačna
  - ne greši u izvršavanju
  - je sposoban za računanje Nešovog ekvilibrijuma
  - veruje da odstupanje u njegovoj strategiji neće izazvati promene u strategijama drugih igrača
  - opšte je poznato da svi igrači zadovoljavaju ove uslove

Nešov ekvilibrijum i racionalno donošenje odluka

Da li imaš dominantnu strategiju?

da

ne

Primeni dominantnu strategiju

Da li znaš šta će protivnik da uradi?

da

ne

Maksimizuj korist

Da li je protivnik racionalan?

da

ne

Da li možemo da se dogovorimo oko Nešovog ekvilibrijuma?

Maksimizuj najgori mogući ishod

da

ne

Primeni ekvilibrijum strategiju

Maksimizuj najgori mogući ishod

Primer igre Kukavica

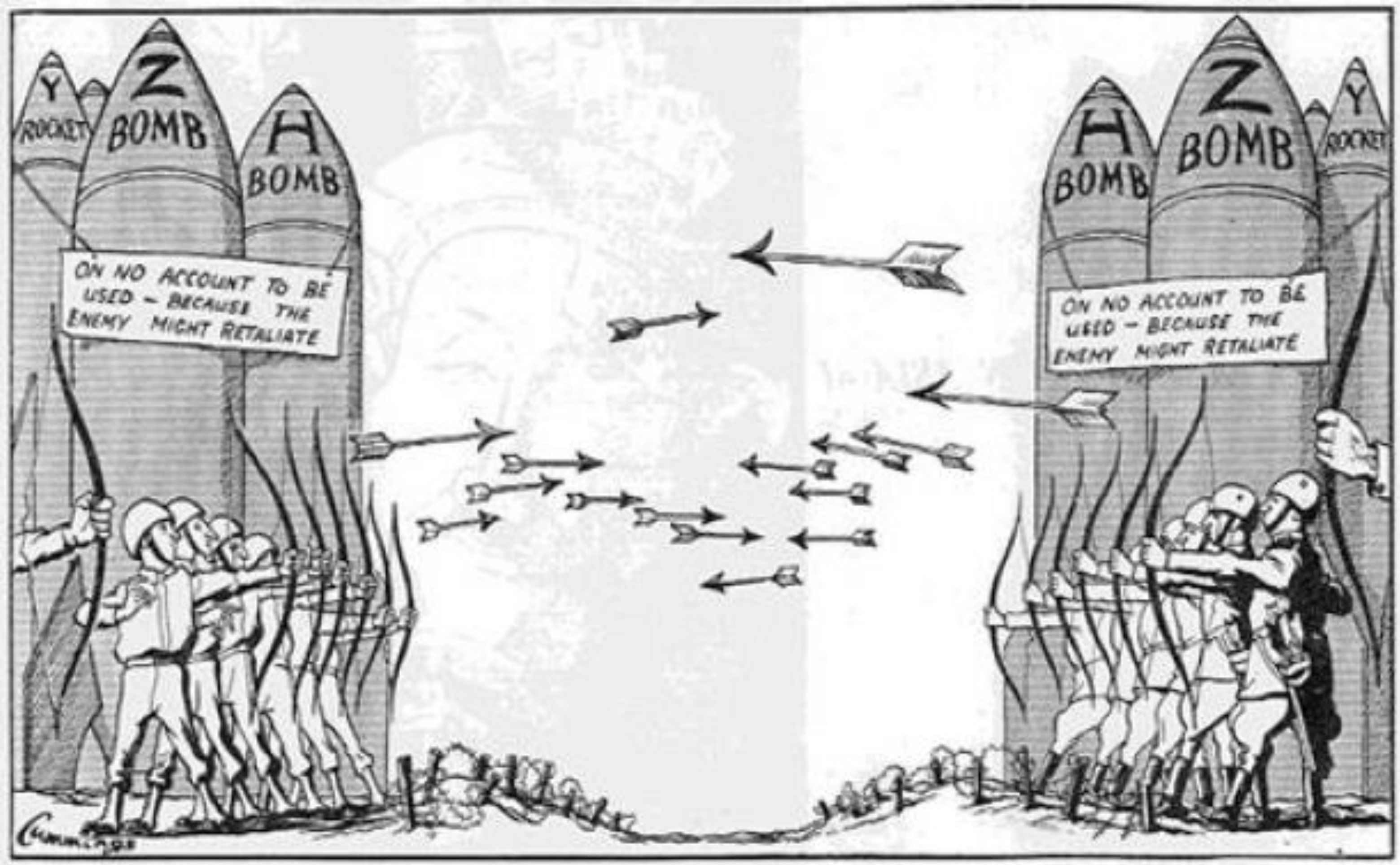
- Kubanska raketna kriza može da se modeluje kao igra
- 1960. SSSR je počeo da snabdeva Kubu raketama.
- SAD je započeo blokadu da bi zaustavio SSSR
- Kako se kriza razvijala, da je svako nastavio sa svojim akcijama, posledice bi bile katastrofalne
- Na svu sreću, pregovori u poslednjem trenutku su izbegli takav ishod. SAD nisu morale da se povuku,
  a SSSR je uspeo da postigne deo svojih interesa
    - SAD nisu napale Kubu

Kompromis
Nije ekvilibrijum

SSSR

SAD		Povlačenje	Nastavak
	Blokada	(3,3)	(4,2)
	Vazdušni napad	(2,4)	(0,0)

Nuklearni rat


Y ROCKET
Z BOMB
H BOMB
ON NO ACCOUNT TO BE USED — BECAUSE THE ENEMY MIGHT RETALIATE
H BOMB
Z BOMB
Y ROCKET
ON NO ACCOUNT TO BE USED — BECAUSE THE ENEMY MIGHT RETALIATE

Problemi teorije igara

- Da li je primenjiva na stvarni život?
  - Ljudi nisu uvek racionalni
  - Nisu svi uslovi unapred poznati
  - Političke poteškoće mogu da spreče primenu teoretski optimalnih mehanizama
- Da li može bolje da se primeni na AI nego na stvarni život?
  - Računanje ekvilibirjuma u komplikovanim igrama je teško
  - Veza između Nešovog ekvilibrijuma i racionalnog donošenja odluka je suptilna

Teorija igara

- Ograničena primena za igre u "realnom vremenu"
  - koristi se samo za izradu strategija
  - kretanje, donošenje odluka, iscrtavanje
    - posebni algoritmi
- Idealna za "igre na tabli"
  - potrebna samo strategija

Igre na tabli

- Plodno polje za AI
- Razvijeni algoritmi različog kvaliteta:
  - amaterski nivo - Go
  - nivo velemajstora - poker
  - nivo šampiona - šah
  - bolji od šampiona – otello (reversi)
  - rešene igre – gomoku, mice, X-O, awari

Tipovi igara

- Klasifikacija na osnovu:
  - broja igrača
  - cilja igre
  - informacije

Broj igrača

- Najčešće je broj igrača dva
- U tom obliku se uglavnom i prezentuju
- Dodavanje igrača komplikuje algoritam
- Mnoge optimizacije su primenjive samo na dva igrača

Cilj igre

- POBEDA! ☺
- Jedan pobeđuje a svi gube -> zero-sum
  - pobeda vredi 1 poen, nerešeno 0, poraz -1
  - poker, šah, gomoku, X-O
- Ne gube svi -> non-zero-sum
  - neki ishodi dovode do situacije da svi dobijaju "nešto"
    - dilema zatvorenika

Informacije

- Potpune informacije (IN)
  - potpuno poznato trenutno stanje igre
  - igrač ne zna šta će protivnik odigrati, ali zna koja je posledica svakog poteza
  - šah, gomoku, X-O, mice...
  - lakše za analizu
- Nepotpune informacije (OUT)
  - faktor slučajnosti - kockice
  - faktor neizvesnosti - kartaške igre

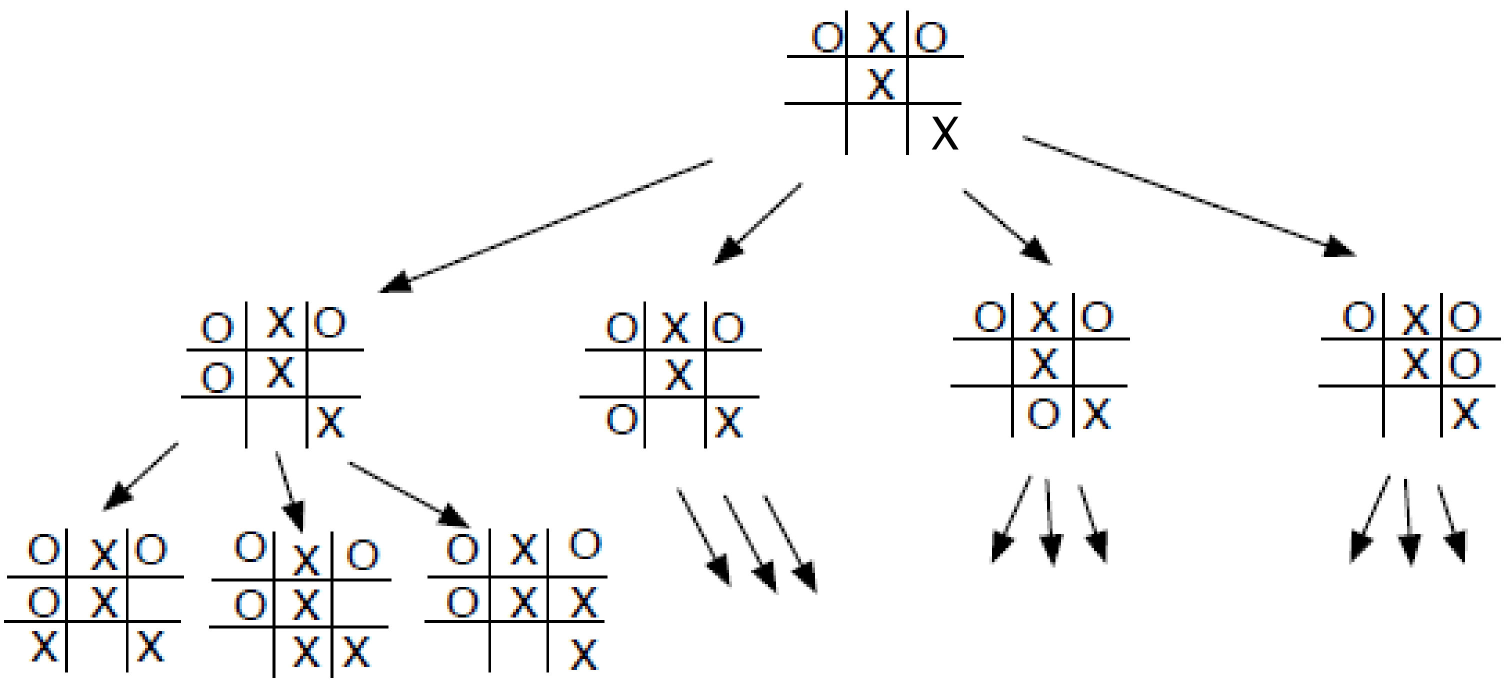
Najpoznatiji algoritmi

- Razvijeni su za igre
  - sa dva igrača,
  - dostupne su im potpune informacije i
  - igra je zero-sum
- Uz određene modifikacije, moguća primena i na druge vrsta igara

Stablo igre

- Potezna igra može da se predstavi stablom igre
- Svaki čvor je jedna pozicija u igri
- Svako grananje je jedan mogući potez
  - odigran po pravilima igre
- Kako igrači igraju jedan za drugim, svakom odgovara jedan nivo u stablu

Primer stabla igre



Nivo 0:
O na potezu

Nivo 1:
X na potezu

Nivo 2:
O na potezu

Faktor grananja

- Broj grananja je jednak mogućem broju poteza tekućeg igrača
  - za X-O u prvom potezu 9, zatim 8,...
  - Gomoku – tabla 19x19 -> prvi potez - 361 mogućnost; drugi – 360,...
- Broj grana – faktor grananja
  - odličan indikator “težine” igre

Dubina stabla

- Broj ukupno odigranih poteza do završetka igre
- Konačan
  - X-O – 9 poteza
  - pravilima ograničen broj poteza
- Beskonačan
  - Šah - skoro beskonačan broj poteza

Transpozicija

- Dolazak do istog stanja različitim sekvencama poteza
  - više puta u toku igre
  - samo jednom, ali na više različitih načina
- Stablo?
  - Ne, grane se mogu stapati
- Olakšava procenu poteza

Terminalne pozicije

- Određena stanja nemaju naslednike
- Takva stanja nemaju moguće poteze
- To su terminalne pozicije
  - predstavljaju kraj igre
- Za svakog igrača se određuje konačni rezultat
  - zero-sum - zbir rezultata je jednak nuli
  - non-zero-sum - rezultati predstavljaju kvalitet pobede/poraza

Minimax ideja

- Cilj – odabrati najbolji potez
  - izabrati svoj najbolji potez,
    zatim izabrati najbolji odgovor na taj potez,
    zatim odgovor na odgovor na taj potez...
- Razmišljanje unapred
- Procena "kvaliteta" mogućih poteza

Funkcija procene

- Kako odrediti šta je “najbolji” potez?
- Koristiti - statičku funkciju procene
- Treba da pokaže trenutno stanje
  – koliko je igrač “blizu” pobede
- Iskazuje se brojem poena iz opsega koji igrač može da dobije na kraju igre

Funkcija procene (2)

- Lako – ako je terminalna pozicija
  - ako je pobednik, rezultat f-je procene je 1, ako je nerešeno 0, ako je poraz 1, ili kako god je raspoređen broj poena
- Šta ako se nalazimo u sredini igre?
  - teže izračunavanje
  - treba ispravno proceniti ko je u prednosti

Funkcija procene (3)

- Ako je igrač jasno u prednosti, njegov broj poena treba da bude blizu broju poena koji se dobija za pobedu
- Pozicija igrača koji je trenutno nadjačan, treba da se vrednuje brojem poena koji označava da je poraz blizu
- Nije uvek lako odrediti ko je u prednosti
- Tada “znanje” igranja dolazi do izražaja

Funkcija procene (4)

- Pogrešno je vrednovati trenutnu poziciju, koja nije terminalna, većim brojem poena od onog koji se dobija za pobedu
- Ako bi u istom potezu mogla da se izabere i pozicija koja dovodi do pobede, jer bi (pogrešno) bila manje poželjna

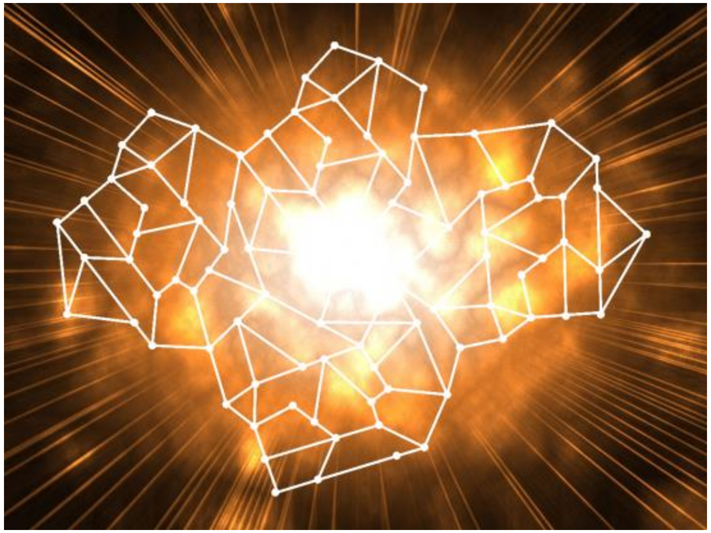
Funkcija procene (5)

- Idelna funkcija procene?
- Nemoguća misija
- Da postoji, sam rezultat funkcije bi bio dovoljan za odlučivanje šta je najbolji sledeći potez
- Zato je potrebno uzeti u obzir i protivnikov odgovor na izabrani potez

Minimax ideja

- Birati mogući potez pretragom unapred
- Izabrati jedan od mogućih poteza, zatim izabrati protivnikov odgovor, pa naš odgovor na protivnikov odgovor...
- Kada biramo potez za sebe, biramo najbolji mogući (najgori po protivnika)
- Kada naš protivnik bira potez, bira najbolji za njega (najgori za nas)

```
minimax(trenutnoStanje, maxDubina, trenutnaDubina):
    ukoliko je terminalno stanje ili je trenutna dubina jednaka maksimalnoj
        return vrednost statičke funkcije procene za trenutno stanje
    ukoliko je trenutni igrač MAX
        najboljaVrednost = - BESKONAČNO
    inče ukoliko je trenutni igrač MIN
        najboljaVrednost = BESKONAČNO
    za svaki mogući potez trenutnog igrača određujemo novo stanje i njegovu vrednost
        novoStanje = Kreiraj izgled novog stanja u koje bi se prešlo
        trenutnaVrednost = minimax(novoStanje,maxDubina,trenutnaDubina+1)
        ukoliko je trenutni igrač MAX i trenutnaVrednost je veća od najboljeVrednosti
            najboljaVrednost = trenutnaVrednost
        ukoliko je trenutni igrač MIN i trenutnaVrednost je manja od najboljeVrednosti
            najboljaVrednost = trenutnaVrednost
    return najboljaVrednost
# kraj minimax algoritma
```

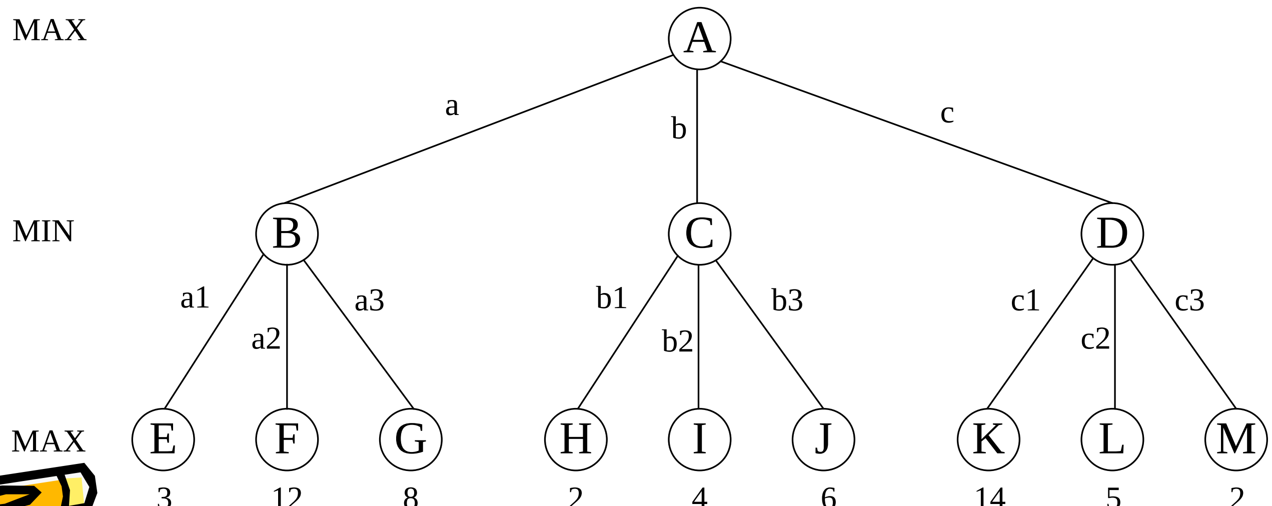


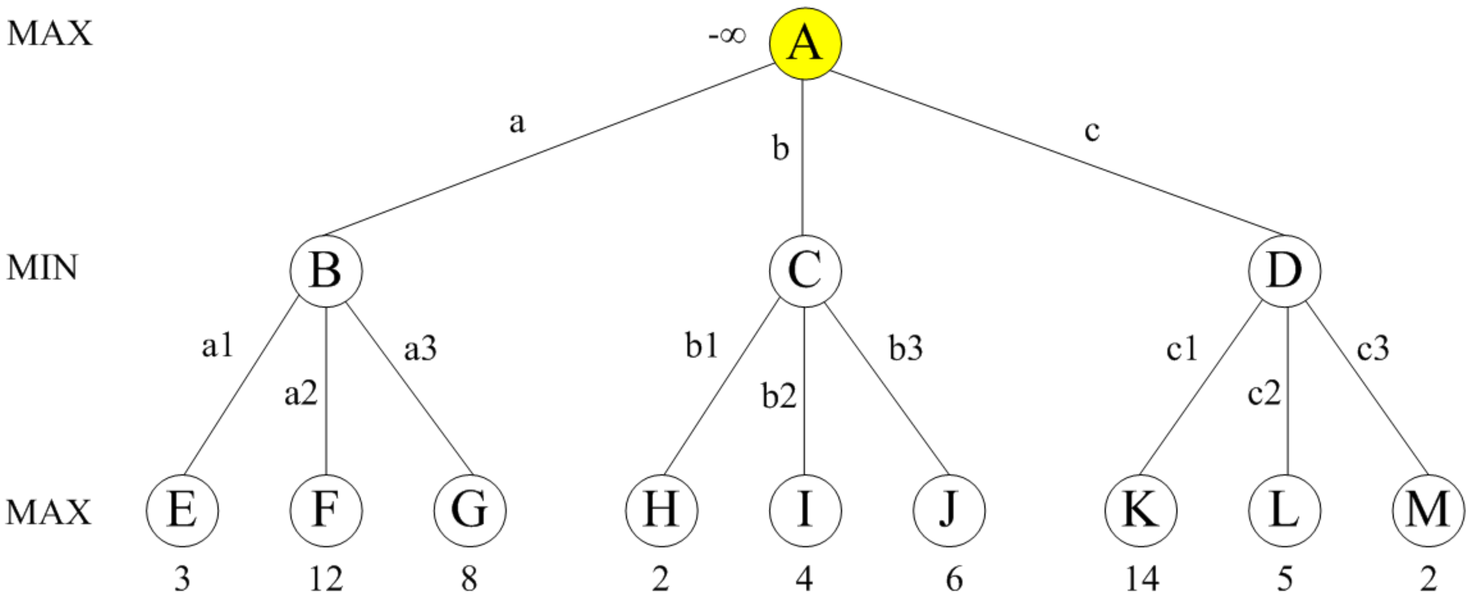
Minimax

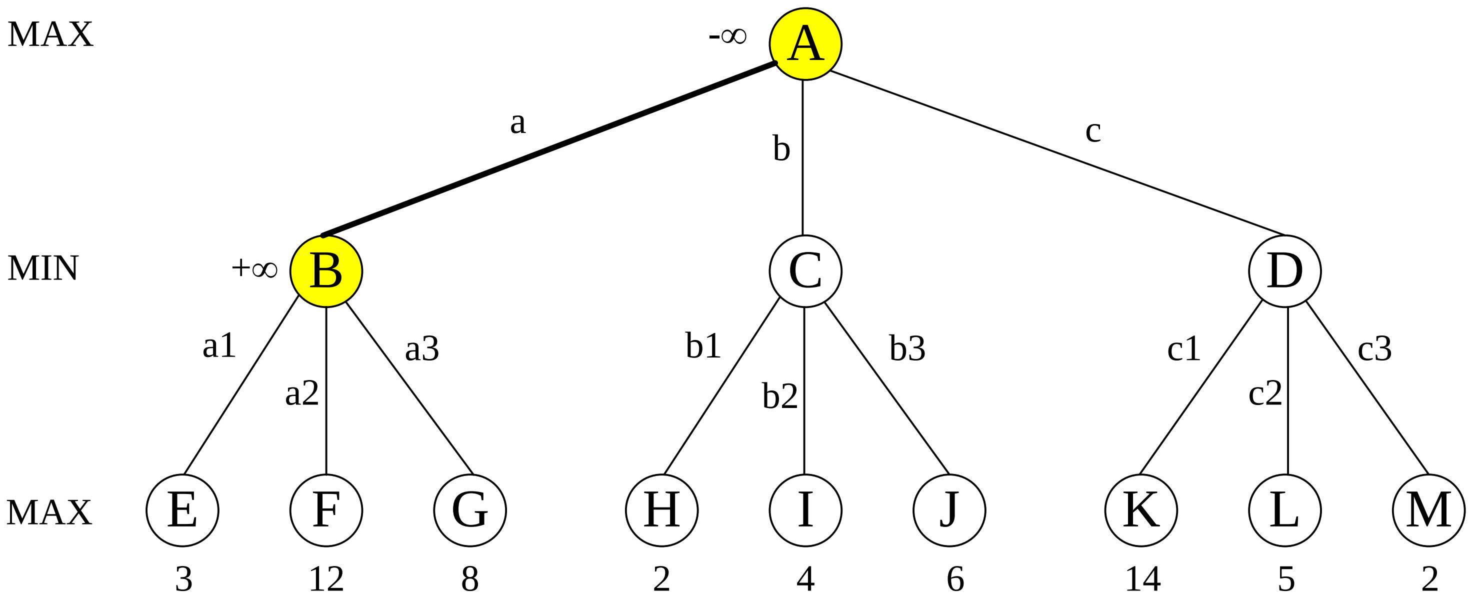
- Koliko poteza unapred treba proveriti?
  - dubina pretrage
- Prostor nije problem
  - algoritam je rekurzivan, pa je slozenost O(d), gde je *d* maksimalna dubina pretrage
- Problem je vreme
  - složenost je O(n^d), gde je *n* mogući broj poteza za svaku poziciju
  - Užasno mnogo poteza za veliku dubinu

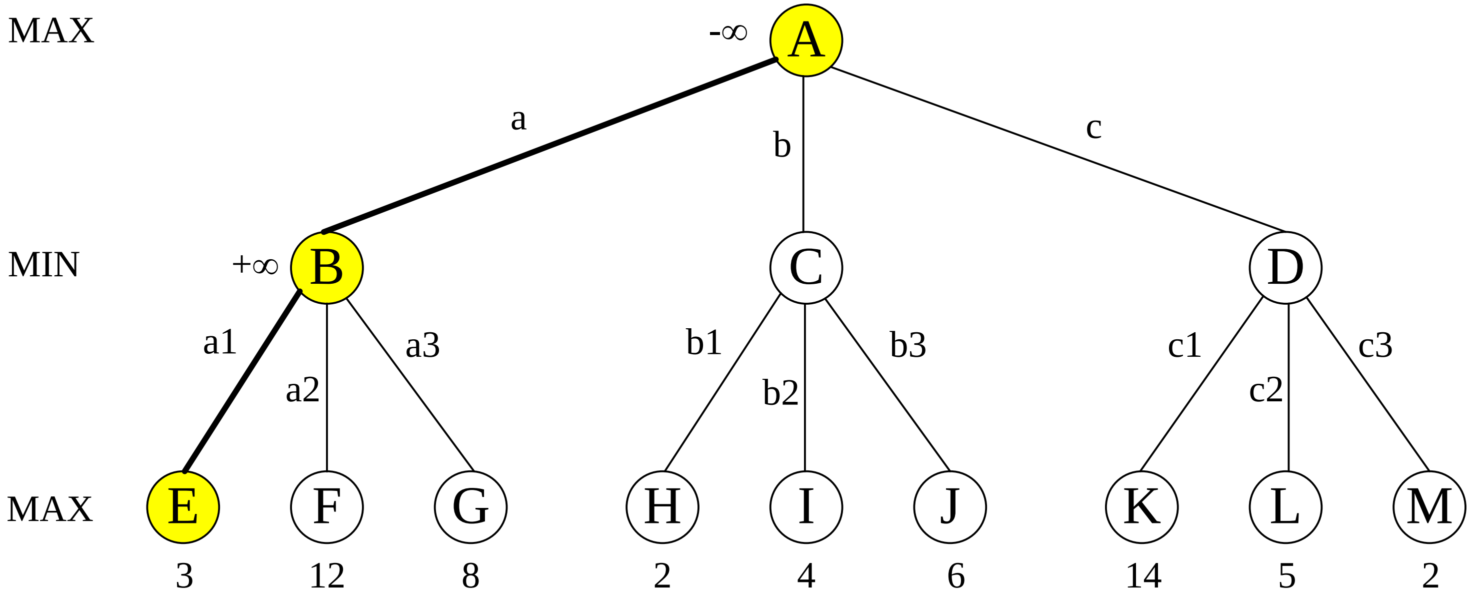
Zadatak 1: Minimax metoda

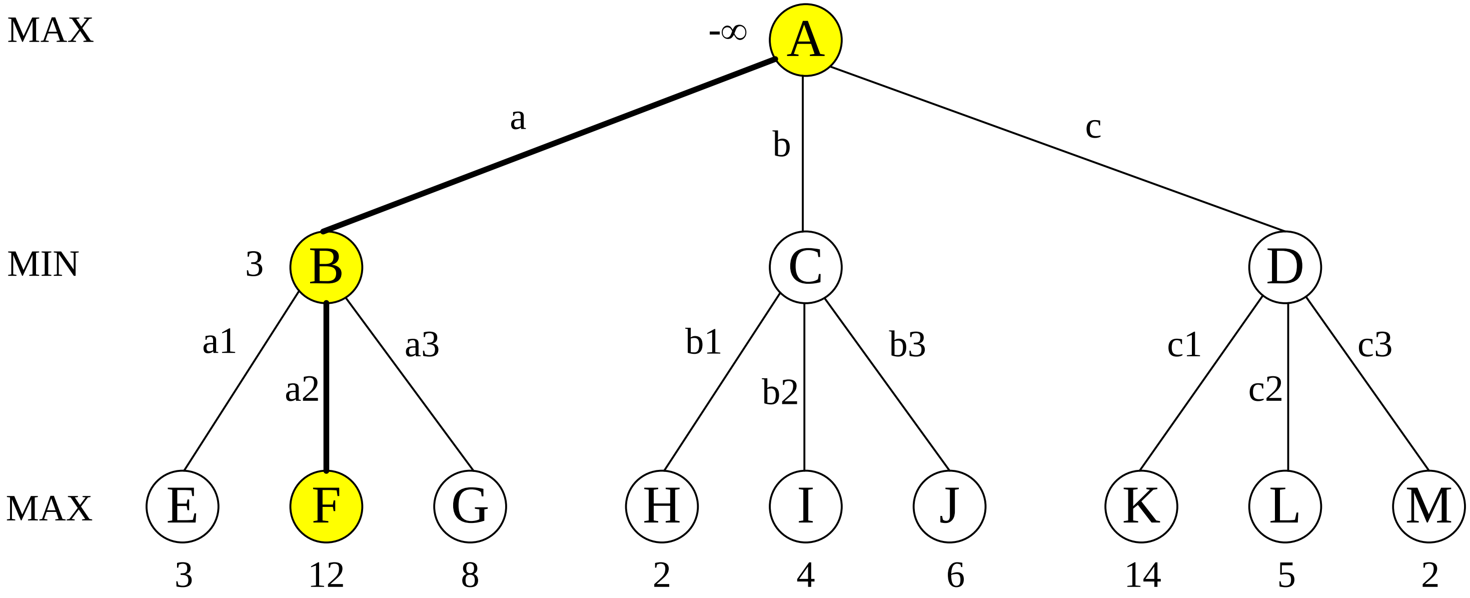
- Upotrebom minimax algoritma, za dato stablo igre, pronaći naredni potez koji će biti odigran

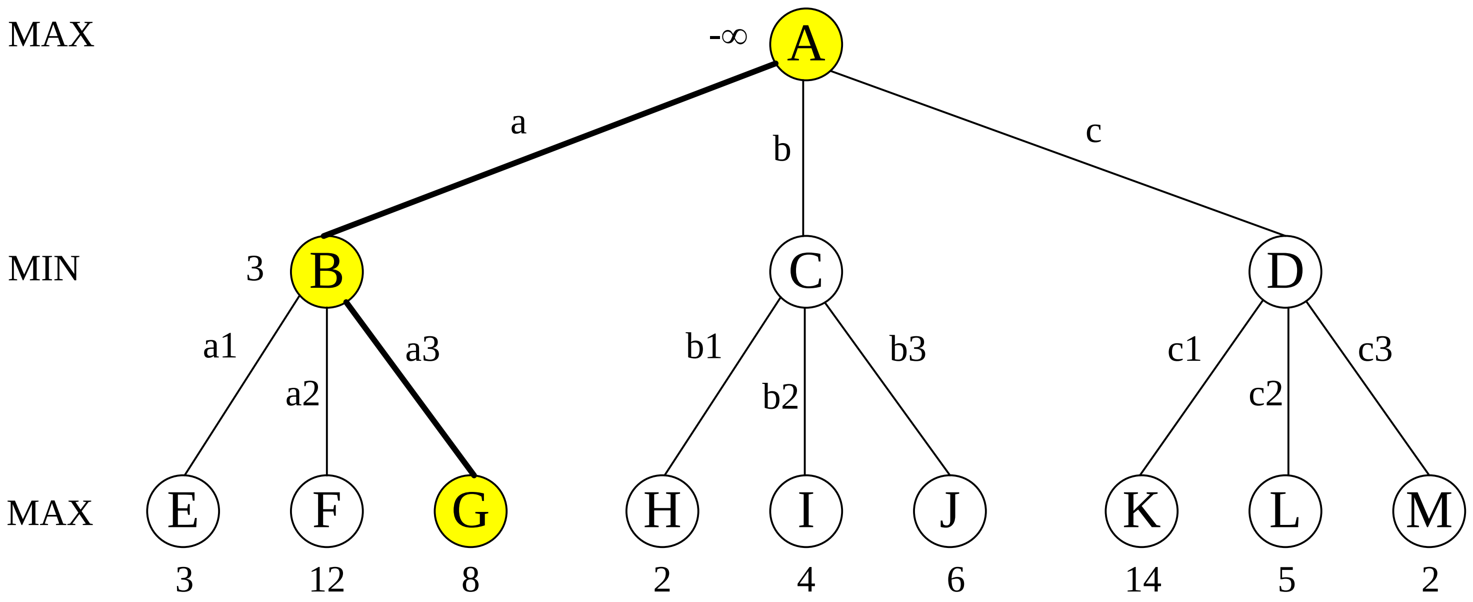


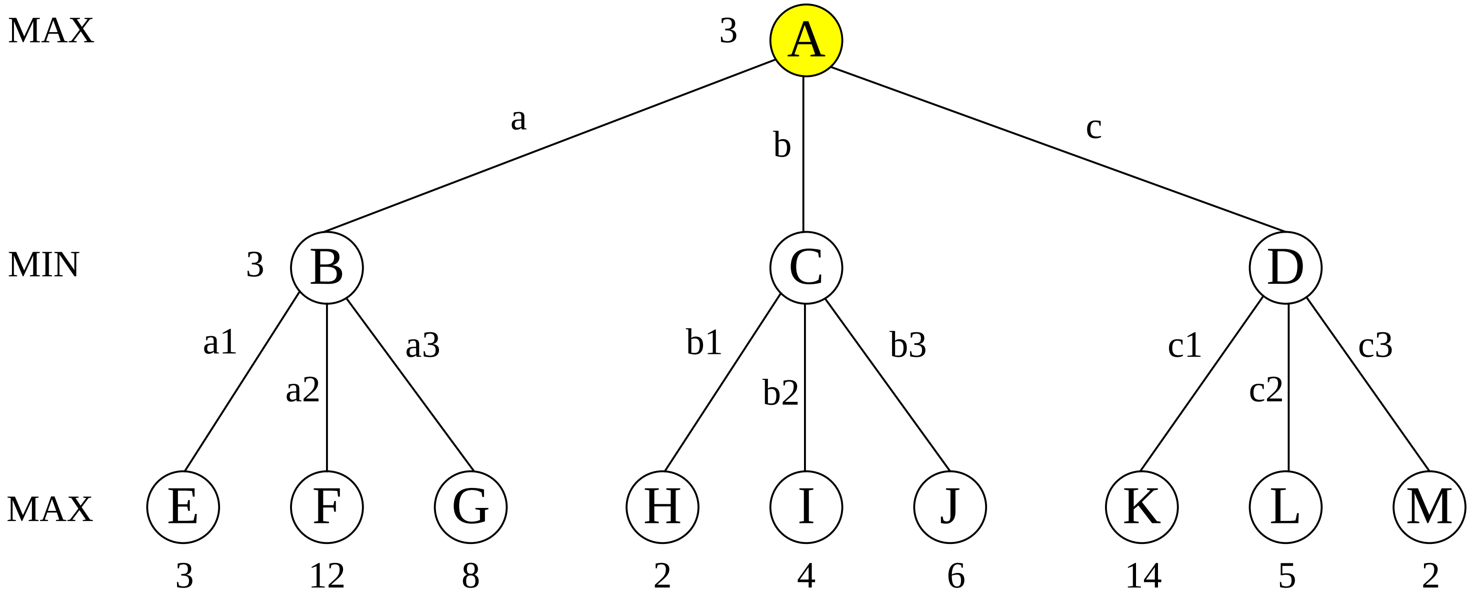

MAX
-∞
A
a
b
c
MIN
B
C
D
a1
a2
a3
b1
b2
b3
c1
c2
c3
MAX
E
F
G
H
I
J
K
L
M
3
12
8
2
4
6
14
5
2

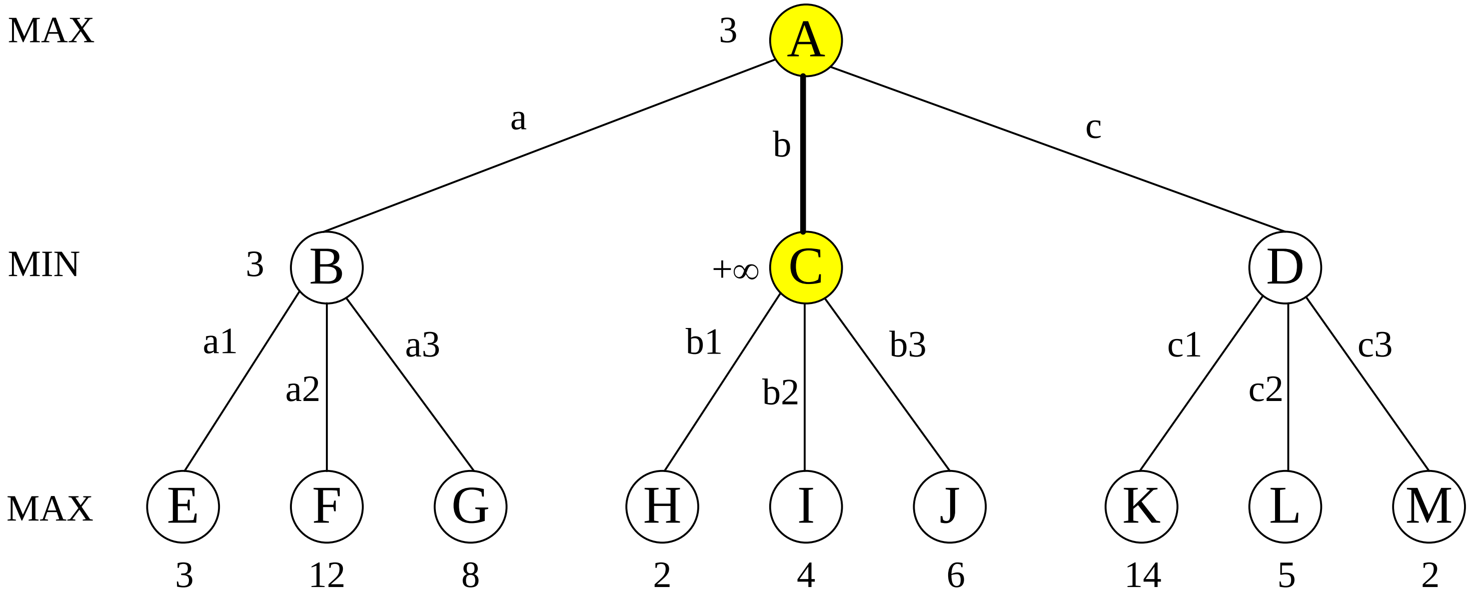

MAX
MIN
MAX
-∞
A
+∞
B
C
D
a
b
c
a1
a2
a3
b1
b2
b3
c1
c2
c3
E
F
G
H
I
J
K
L
M
3
12
8
2
4
6
14
5
2

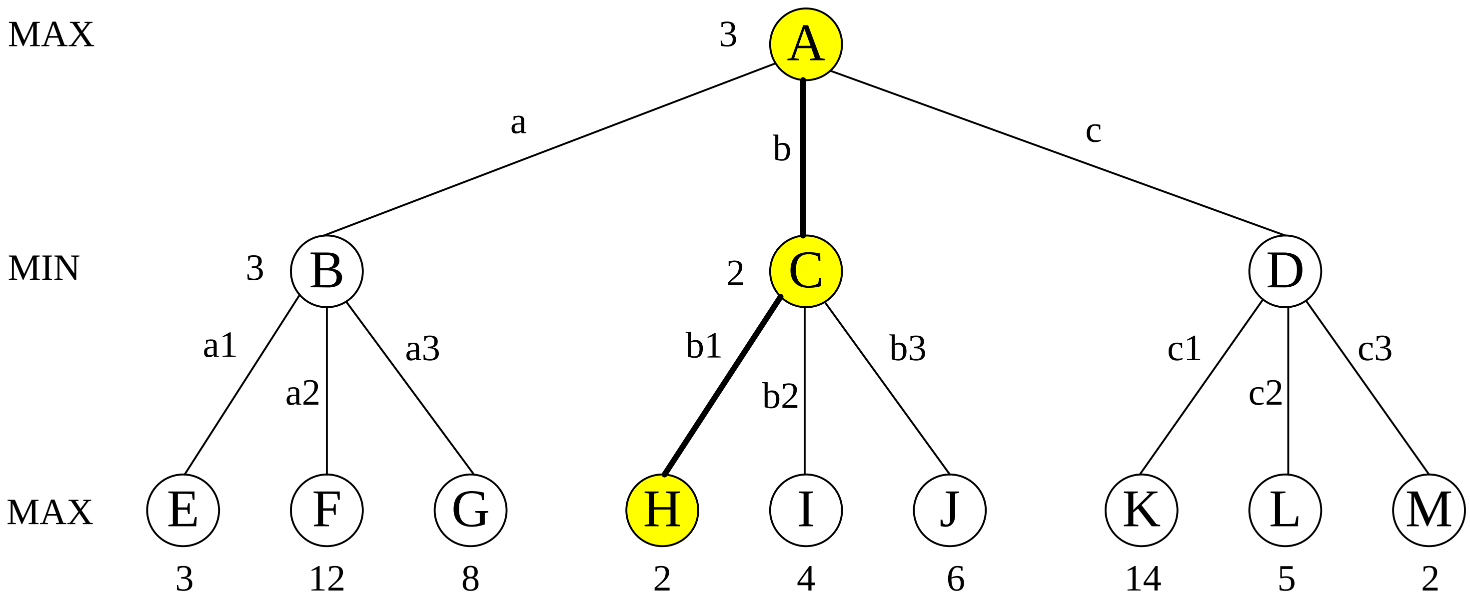

MAX
MIN
MAX
-∞
A
+∞
B
C
D
a
b
c
a1
a2
a3
b1
b2
b3
c1
c2
c3
E
F
G
H
I
J
K
L
M
3
12
8
2
4
6
14
5
2

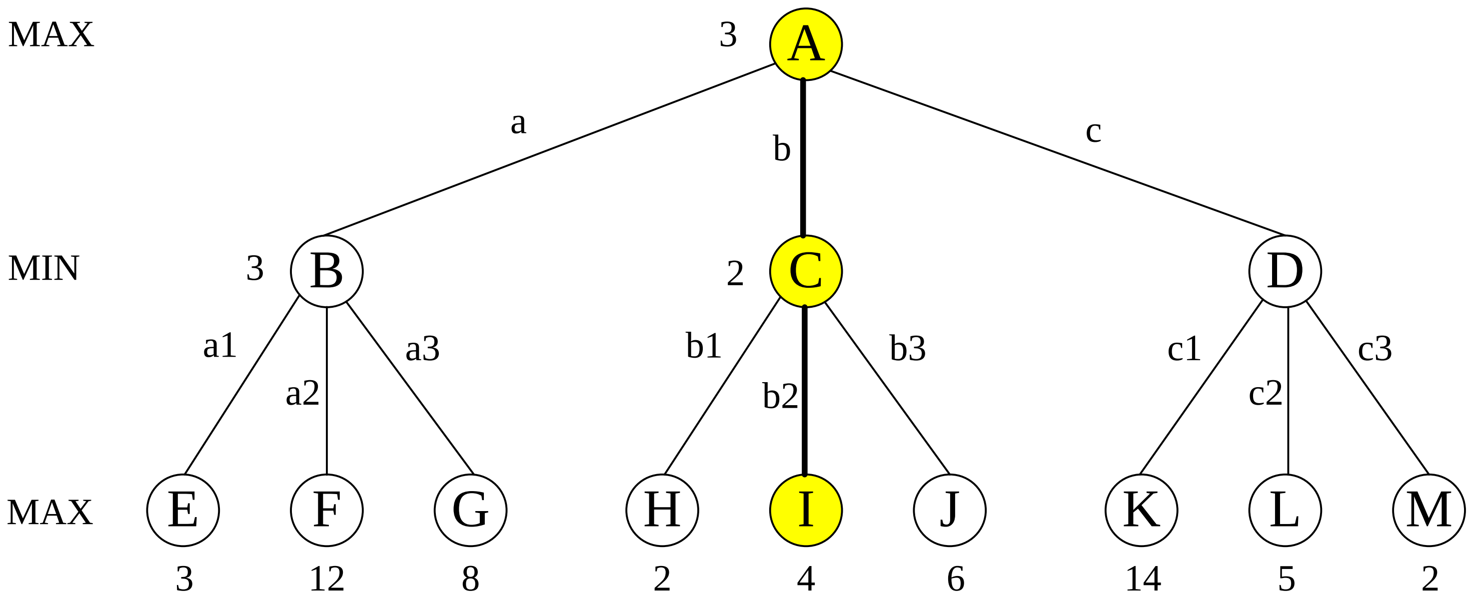

MAX
MIN
MAX
-∞
A
a
b
c
3
B
C
D
a1
a2
a3
b1
b2
b3
c1
c2
c3
E
F
G
H
I
J
K
L
M
3
12
8
2
4
6
14
5
2

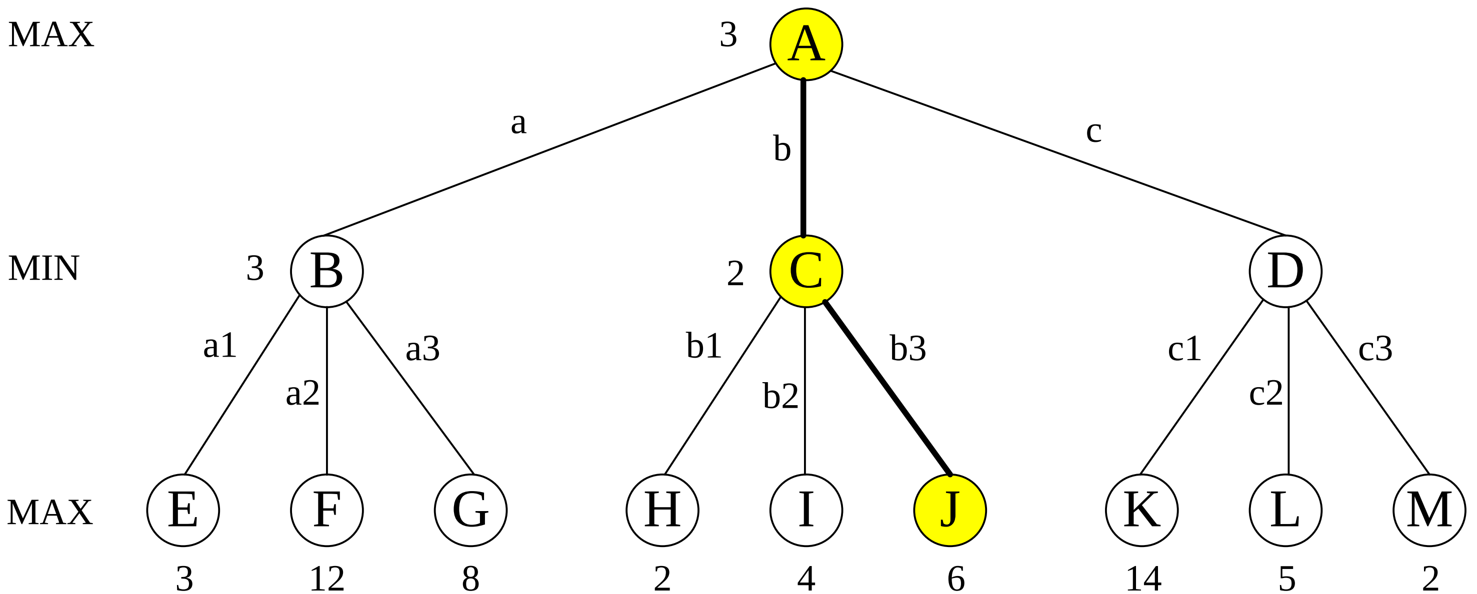

MAX
MIN
MAX
-∞
A
a
b
c
3
B
C
D
a1
a2
a3
b1
b2
b3
c1
c2
c3
E
F
G
H
I
J
K
L
M
3
12
8
2
4
6
14
5
2

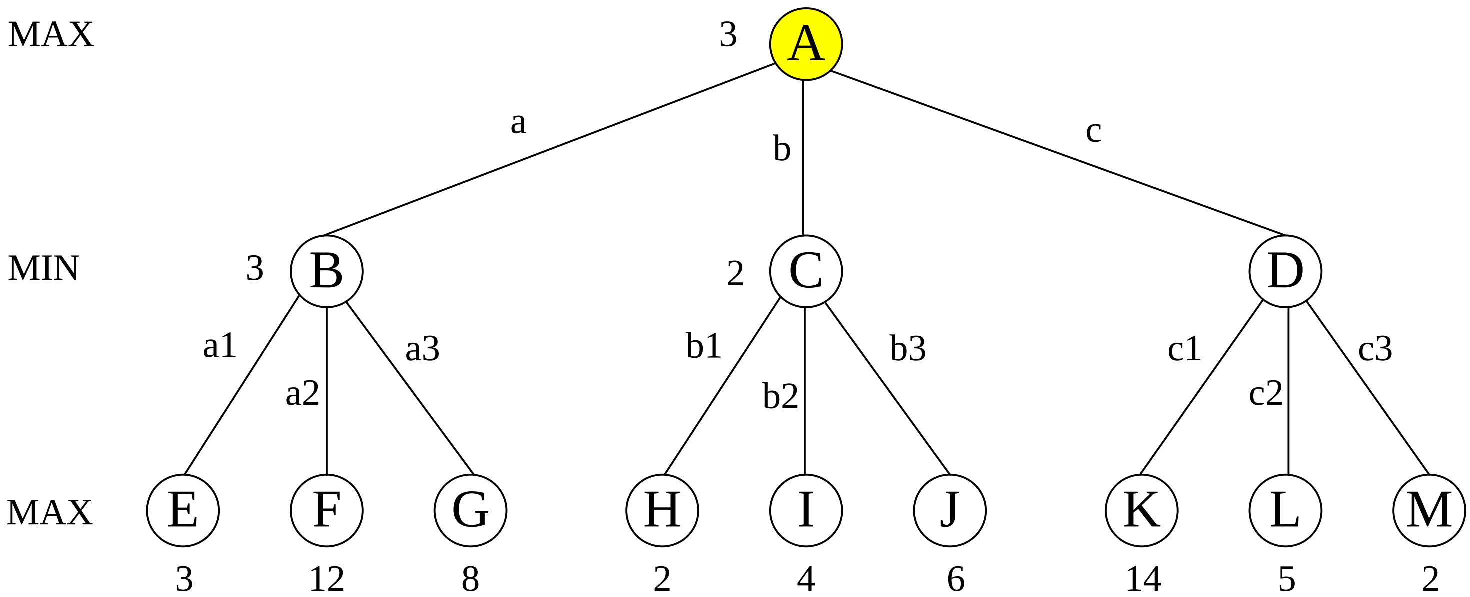

MAX
3
A
a
b
c
MIN
3
B
C
D
a1
a2
a3
b1
b2
b3
c1
c2
c3
MAX
E
F
G
H
I
J
K
L
M
3
12
8
2
4
6
14
5
2

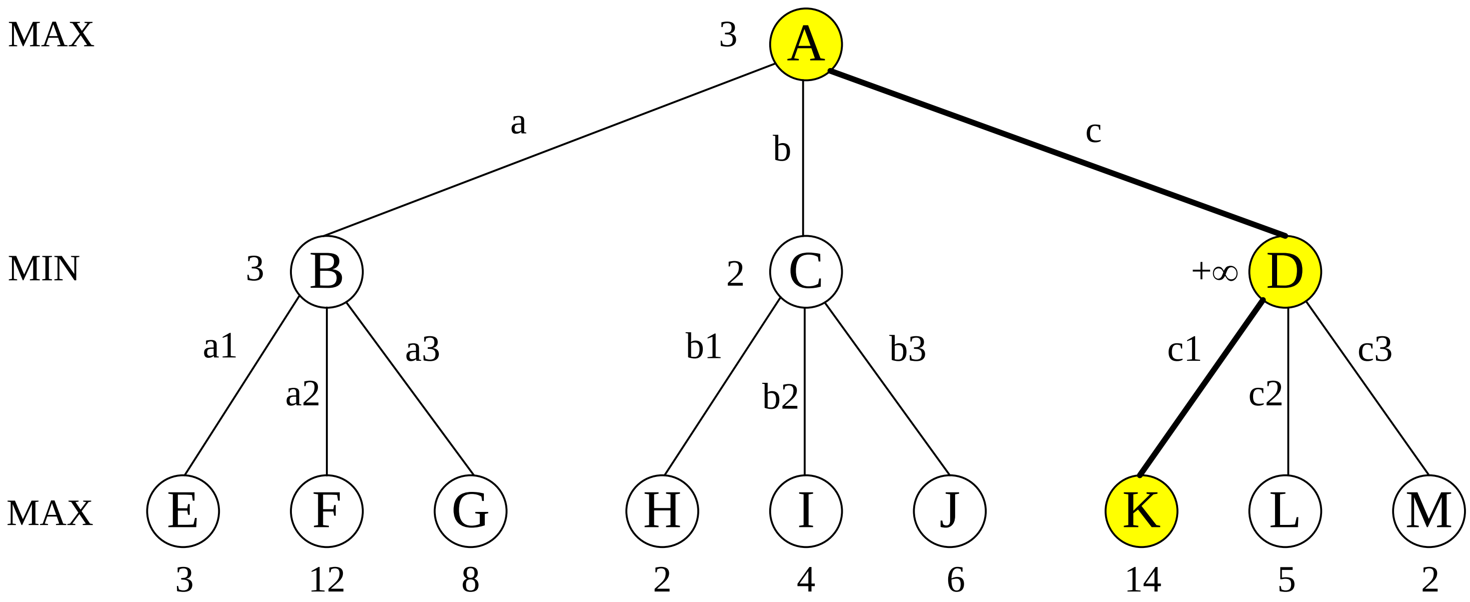

MAX
MIN
MAX
3
A
a
b
c
3
B
+∞
C
D
a1
a2
a3
b1
b2
b3
c1
c2
c3
E
F
G
H
I
J
K
L
M
3
12
8
2
4
6
14
5
2

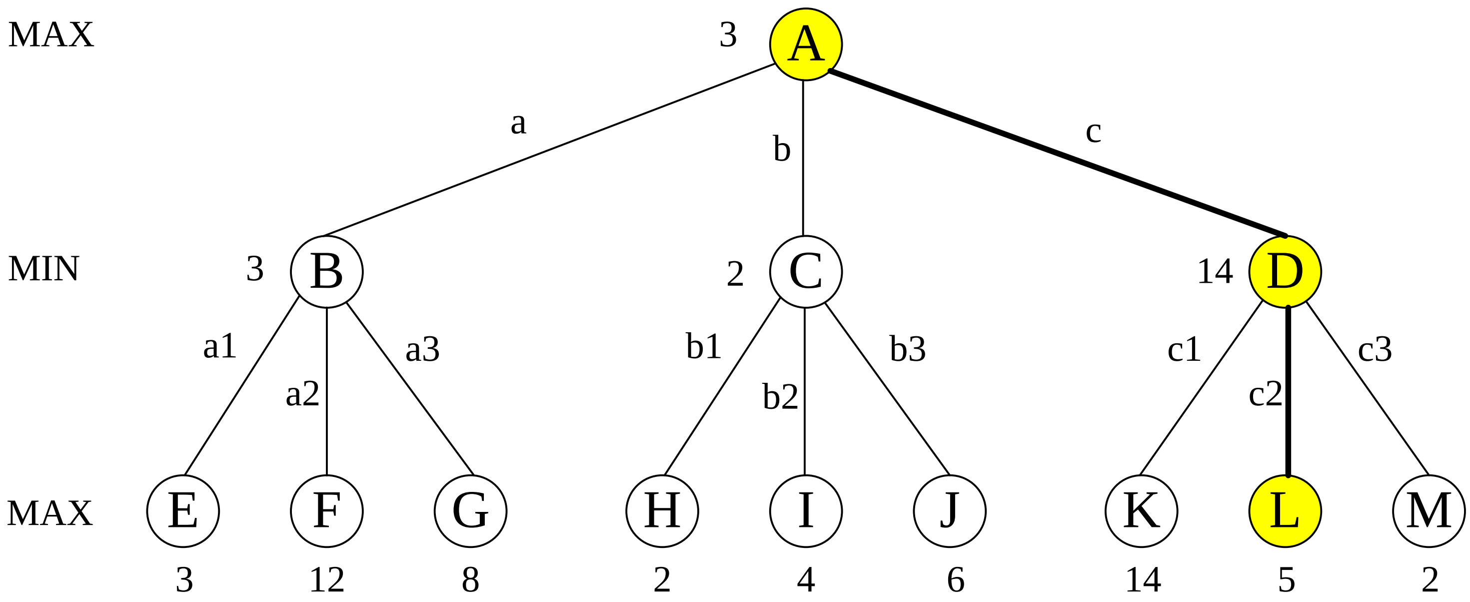

MAX
MIN
MAX
3
A
a
b
c
3
B
2
C
D
a1
a2
a3
b1
b2
b3
c1
c2
c3
E
F
G
H
I
J
K
L
M
3
12
8
2
4
6
14
5
2

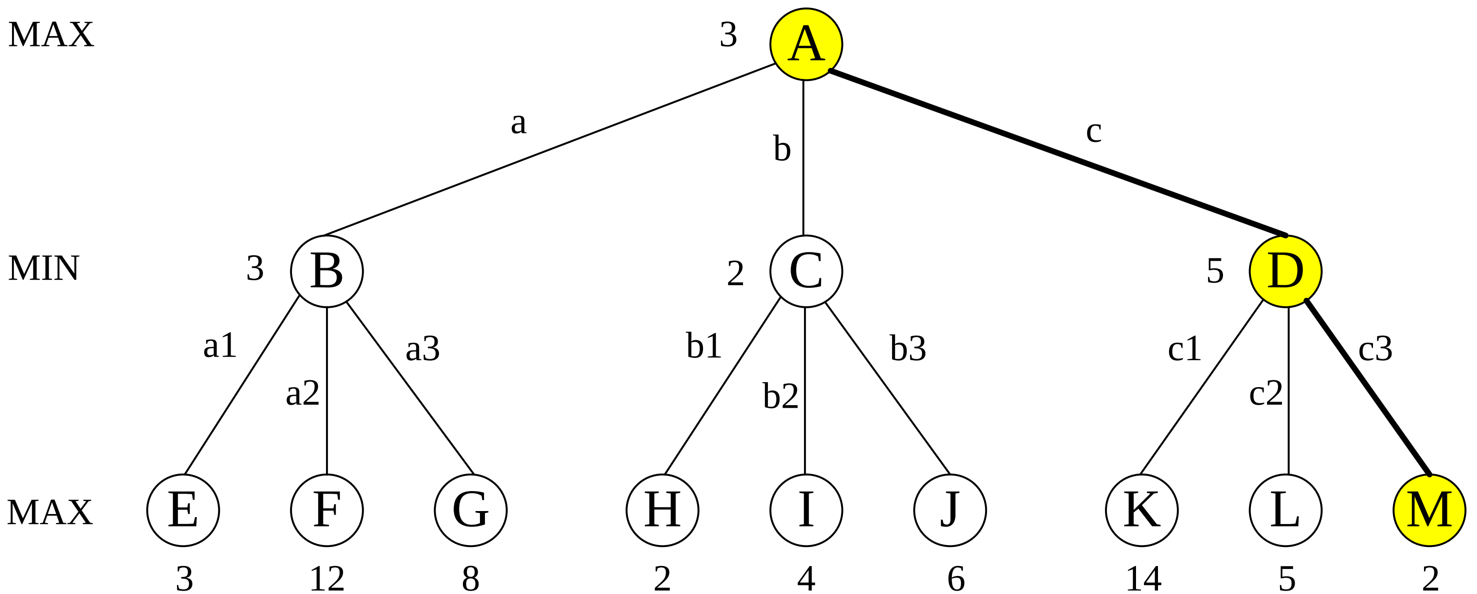

MAX
MIN
MAX
3
A
a
b
c
3
B
2
C
D
a1
a2
a3
b1
b2
b3
c1
c2
c3
E
F
G
H
I
J
K
L
M
3
12
8
2
4
6
14
5
2

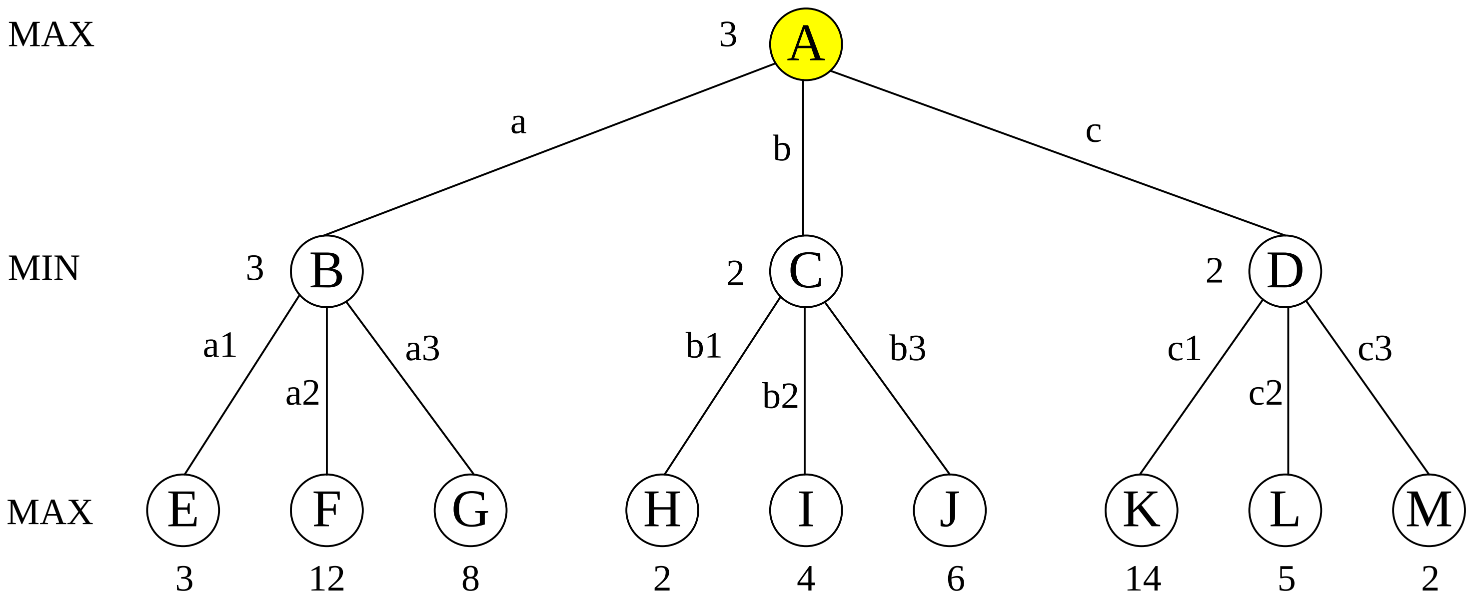

MAX
MIN
MAX
3
A
a
b
c
3
B
2
C
D
a1
a2
a3
b1
b2
b3
c1
c2
c3
E
F
G
H
I
J
K
L
M
3
12
8
2
4
6
14
5
2


MAX
MIN
MAX
3
A
a
b
c
3
B
2
C
D
a1
a2
a3
b1
b2
b3
c1
c2
c3
E
F
G
H
I
J
K
L
M
3
12
8
2
4
6
14
5
2


MAX
MIN
MAX
3
A
a
b
c
3
B
2
C
+∞
D
a1
a2
a3
b1
b2
b3
c1
c2
c3
E
F
G
H
I
J
K
L
M
3
12
8
2
4
6
14
5
2


MAX
3
A
a
b
c
MIN
3
B
2
C
14
D
a1
a2
a3
b1
b2
b3
c1
c2
c3
MAX
E
F
G
H
I
J
K
L
M
3
12
8
2
4
6
14
5
2

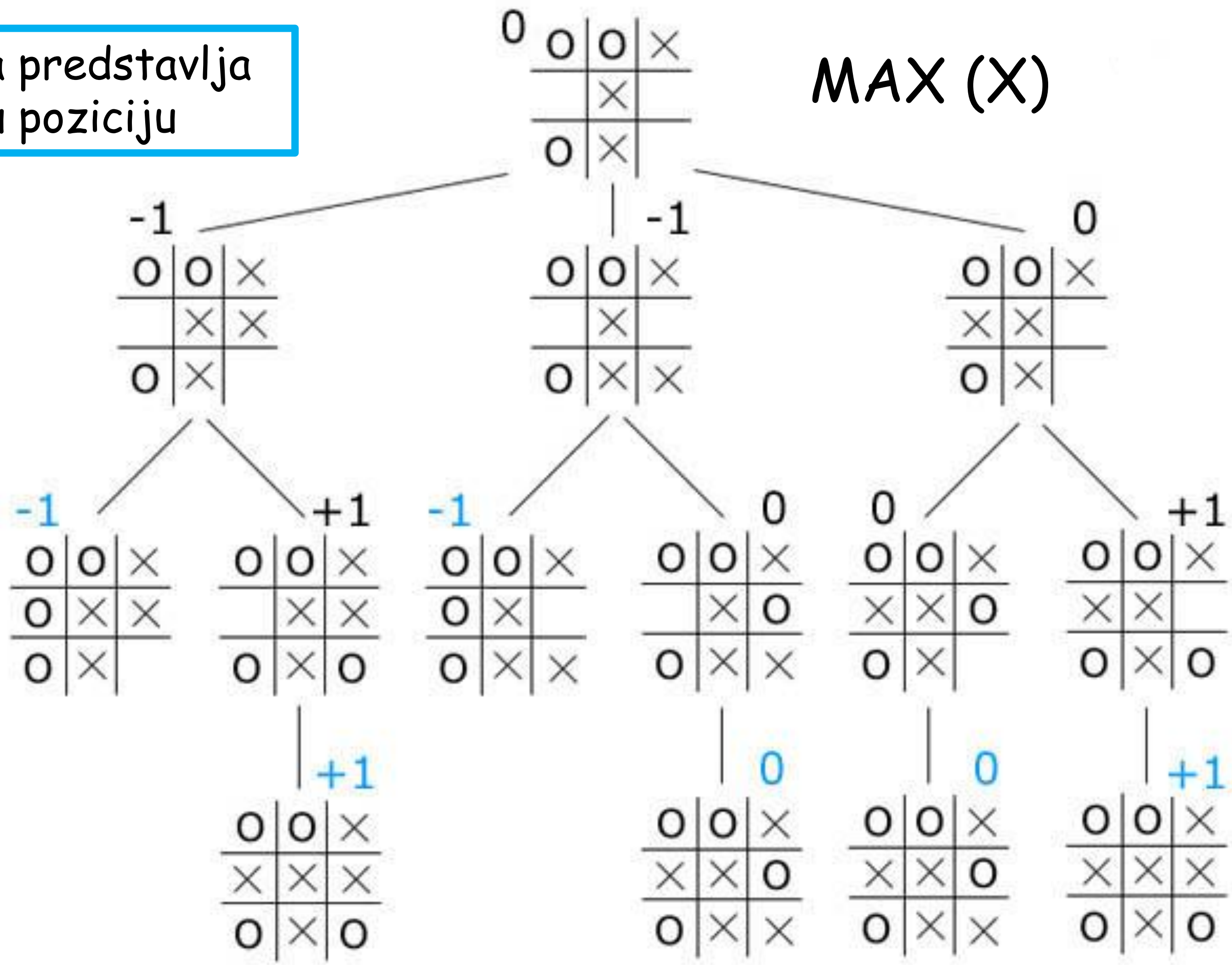

MAX
MIN
MAX
3
A
a
b
c
3
B
2
C
5
D
a1
a2
a3
b1
b2
b3
c1
c2
c3
E
F
G
H
I
J
K
L
M
3
12
8
2
4
6
14
5
2



MAX
MIN
MAX
3
A
a
b
c
3
B
2
C
2
D
a1
a2
a3
b1
b2
b3
c1
c2
c3
E
F
G
H
I
J
K
L
M
3
12
8
2
4
6
14
5
2

Minimax primer u X-O

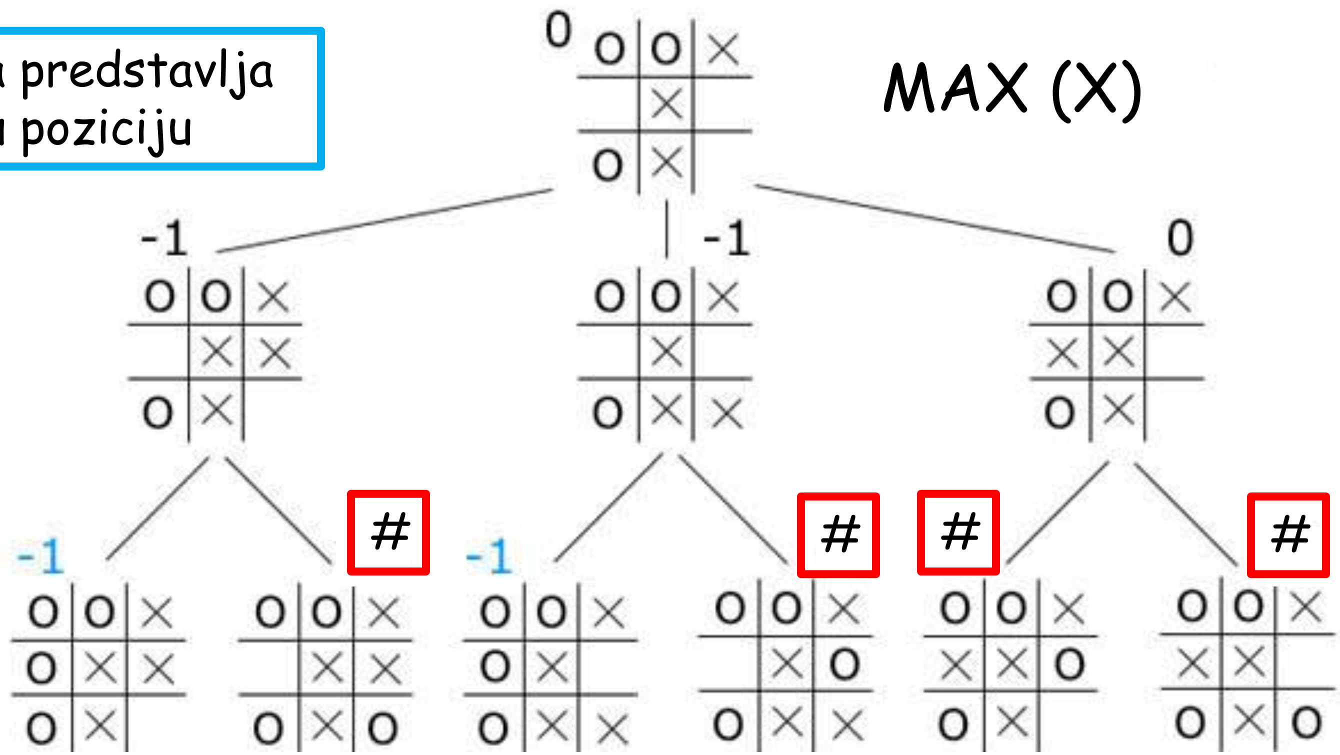
Plava cifra predstavlja terminalnu poziciju



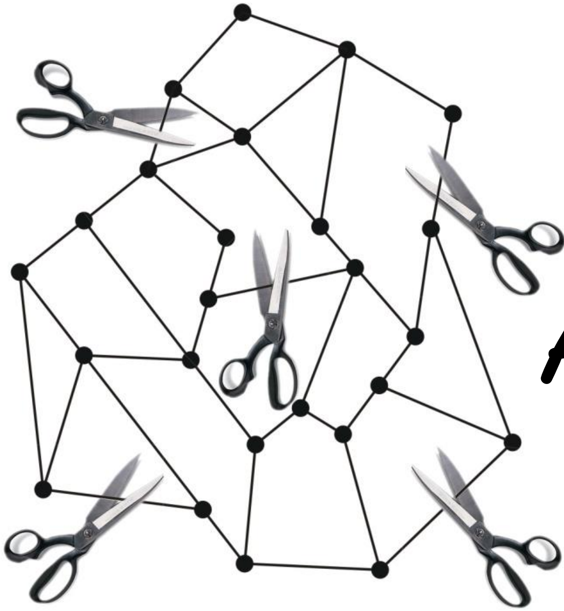
Minimax primer u X-O

Plava cifra predstavlja terminalnu poziciju

MAX (X)



Koja je vrednost f-je procene za čvorove obeležene sa “#”?



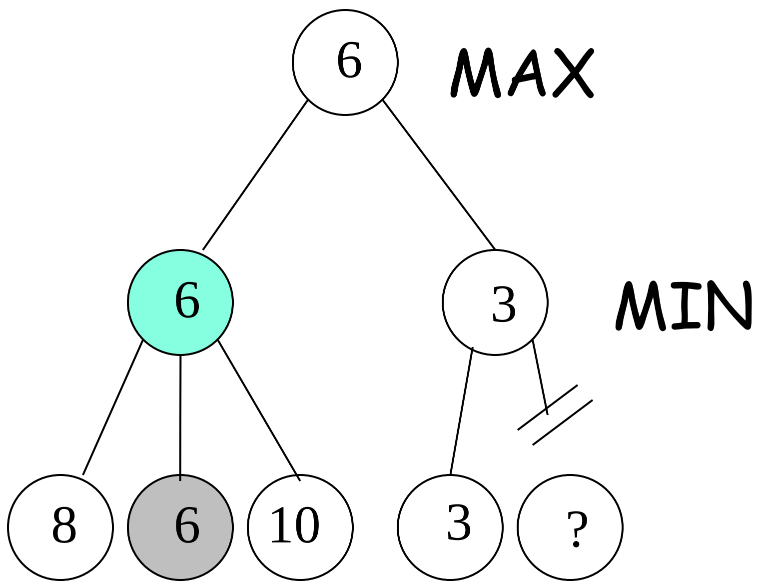
Alfa-beta odsecanje

- Ogroman broj čvorova – smanjiti graf
- Inače, mala dubina – loša efikasnost
- Rešenje – ograničiti vrednost procene donjom i gornjom granicom
- Mi i dalje igramo najbolji potez, a protivnik najgori (po nas)

Alfa odsecanje

- *Alfa* je vrednost najpovoljnijeg poteza po nas koji je **do sada** pronađen
- Predstavlja donju granicu procene poteza koji možemo da prihvatimo
- Nikada nećemo odigrati lošiji potez, od najboljeg do tada pronađenog (alfa)
- Kasnije možemo pronaći bolji potez
- Ako nas protivnik natera na lošiji potez od alfa, nećemo ga izabrati, odbacićemo taj deo stabla

Alfa odsecanje - primer



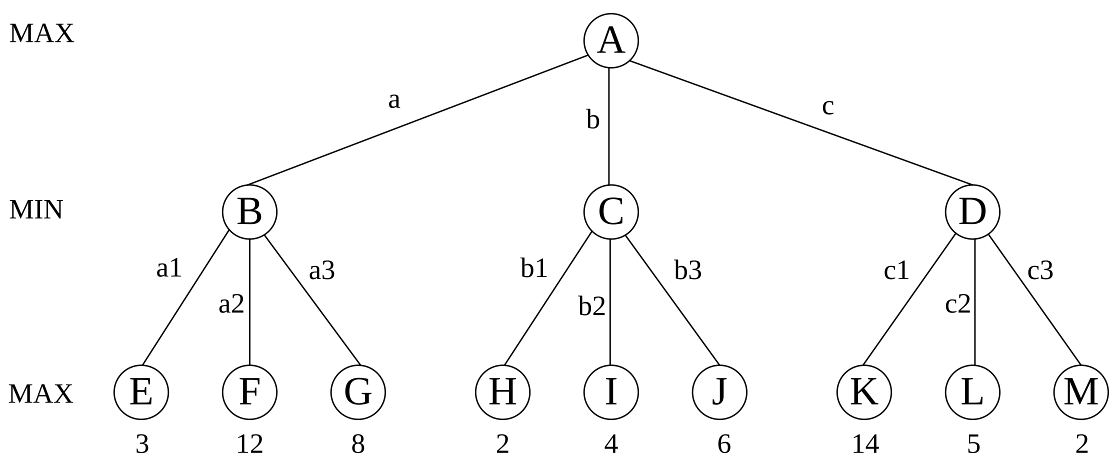
Beta odsecanje

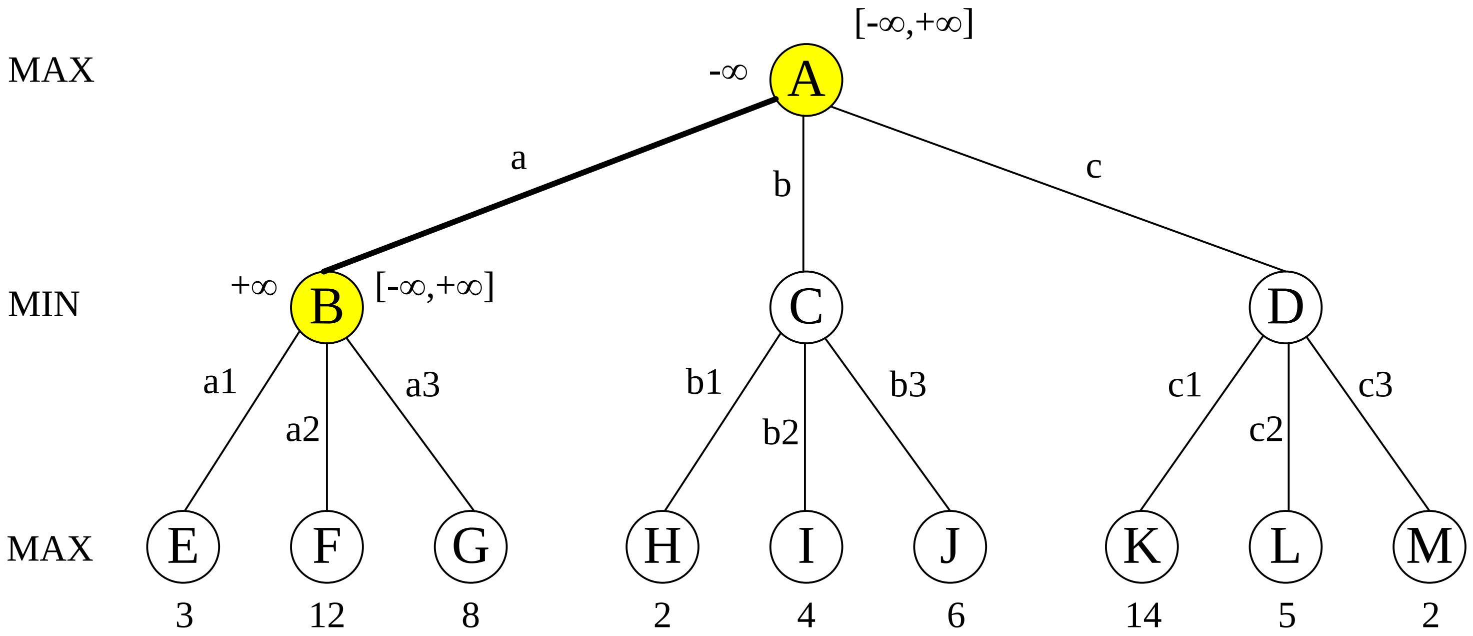
- Predstavlja gornju granicu koju **možda** možemo da dostignemo (β)
- Sigurno ne možemo više od toga
- Postoji šansa da nas protivnik natera da odigramo i lošije
- Eliminišemo podstablo gde je vrednost mogućeg poteza veća od β

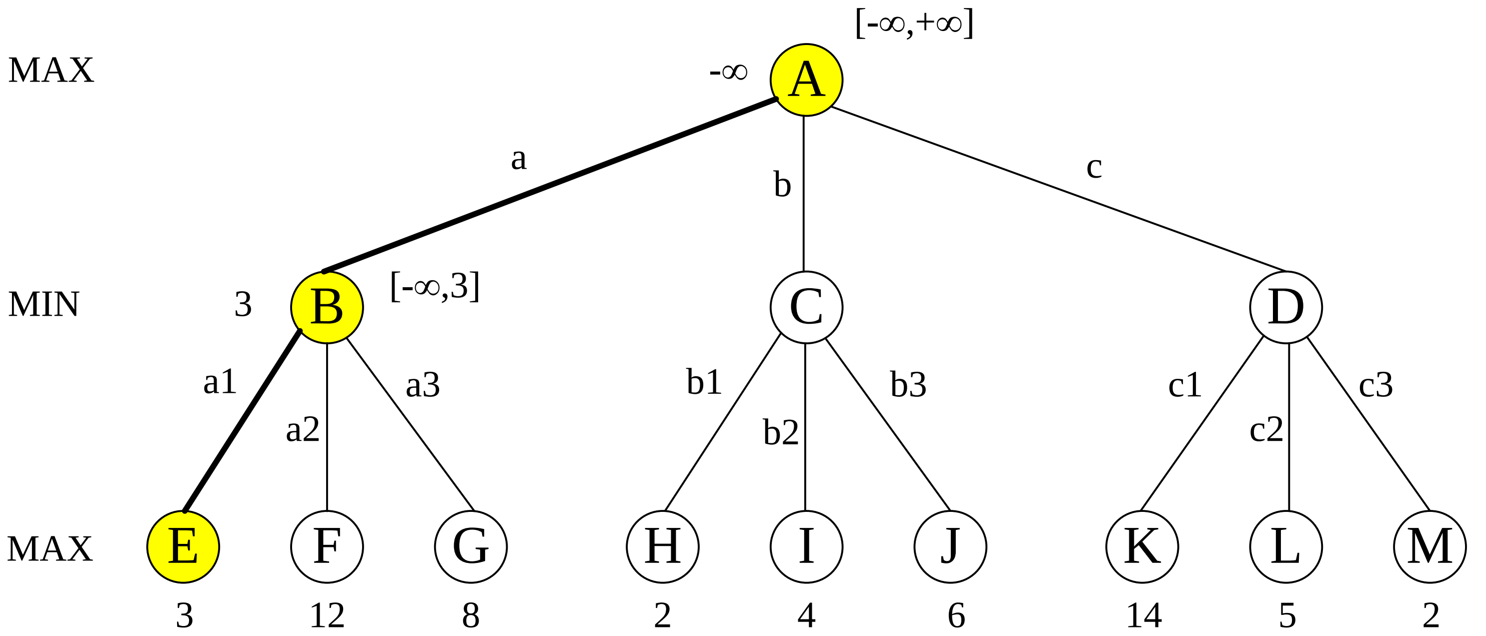
```
minimax(trenutnoStanje, maxDubina, trenutnaDubina, alfa, beta):
    ukoliko je terminalno stanje ili je trenutna dubina jednaka maksimalnoj
        return vrednost statičke funkcije procene za trenutno stanje
    ukoliko je trenutni igrač MAX
        najboljaVrednost = - BESKONAČNO
    inče ukoliko je trenutni igrač MIN
        najboljaVrednost = BESKONAČNO
    za svaki mogući potez trenutnog igrača određujemo novo stanje i njegovu vrednost
        novoStanje = Kreiraj izgled novog stanja u koje bi se prešlo
        trenutnaVrednost = minimax(novoStanje,maxDubina,trenutnaDubina+1, alfa, beta)

        ukoliko je trenutni igrač MAX i trenutnaVrednost je veća od najboljeVrednosti
            najboljaVrednost = trenutnaVrednost
            ukoliko je najboljaVrednost veća ili jednaka beta vrši se odsecanje
                return najboljaVrednost
            alfa = max (alfa, najboljaVrednost)
        ukoliko je trenutni igrač MIN i trenutnaVrednost je manja od najboljeVrednosti
            najboljaVrednost = trenutnaVrednost
            ukoliko je najboljaVrednost manja ili jednaka alfa vrši se odsecanje
                return najboljaVrednost
            beta = min (beta, najboljaVrednost)
    return najboljaVrednost
    # kraj minimax algoritma
```

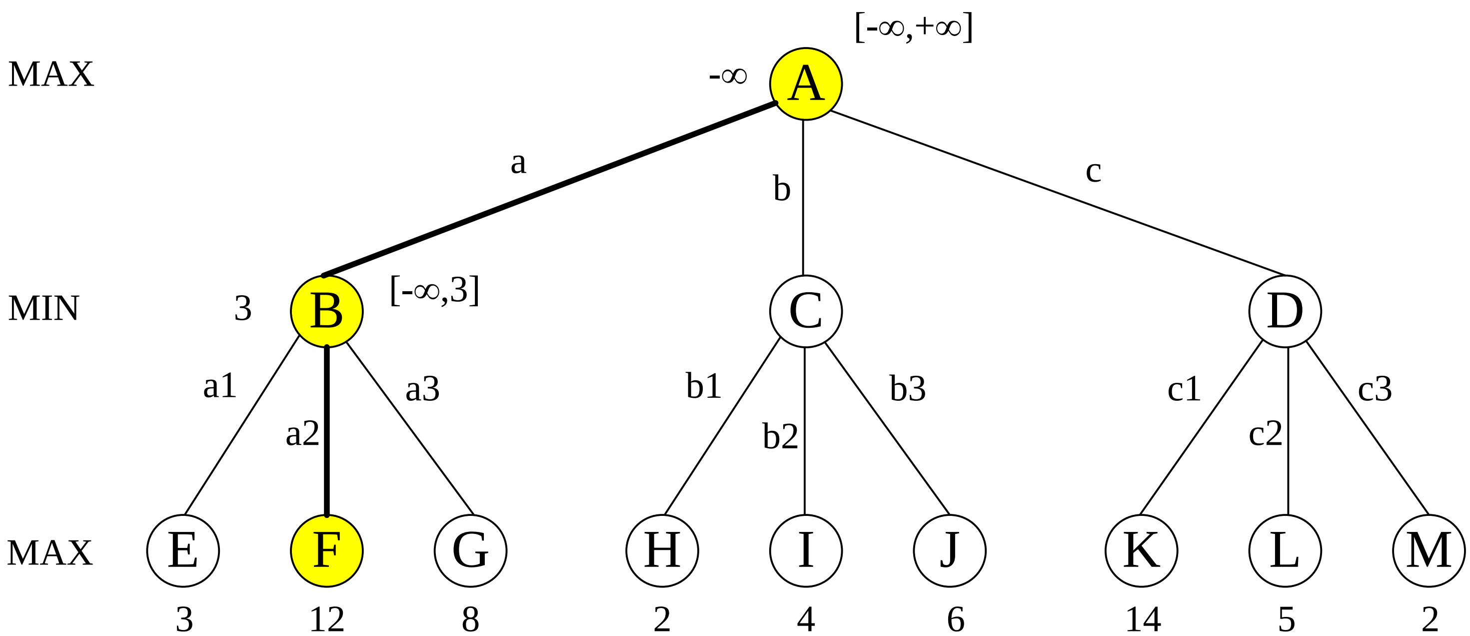
Zadatak 2: Alfa-beta odsecanje

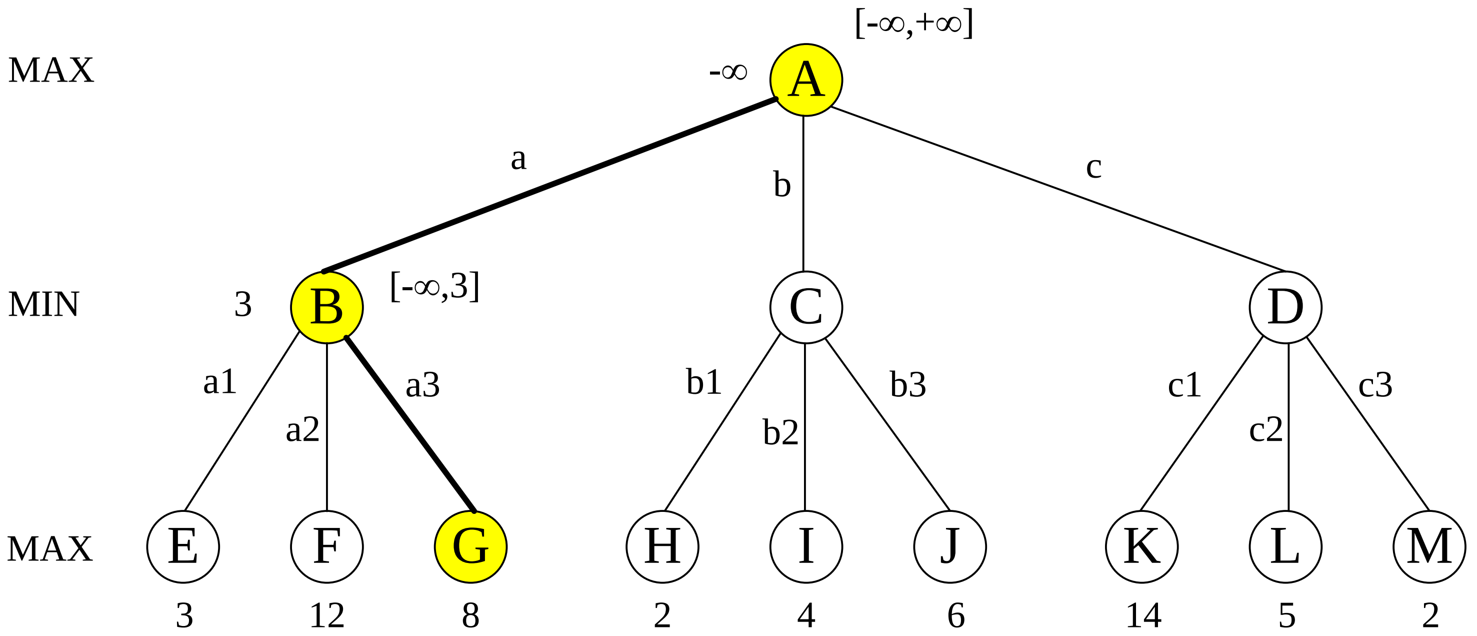
- Upotrebom minimax algoritma uz primenu alfa-beta odsecanja za dato stablo igre pronaći naredni potez koji će biti odigran. Naznačiti koji čvorovi stabla neće biti obidjeni

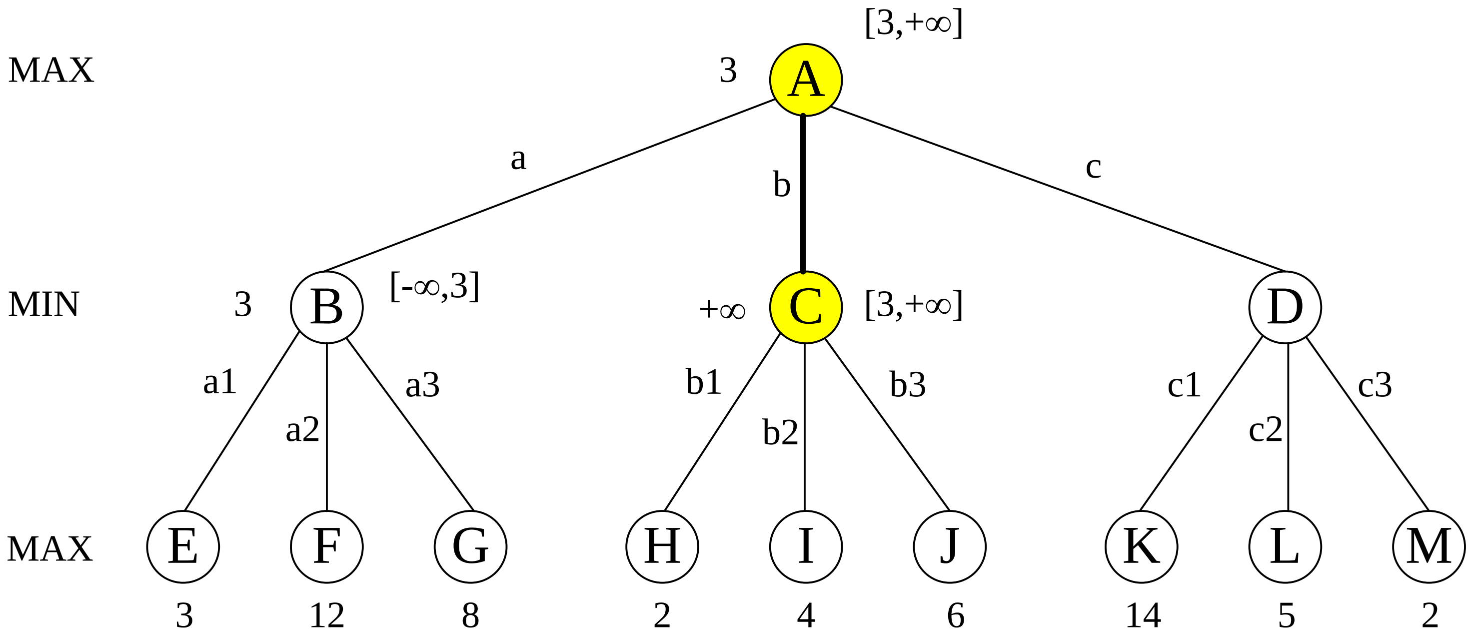


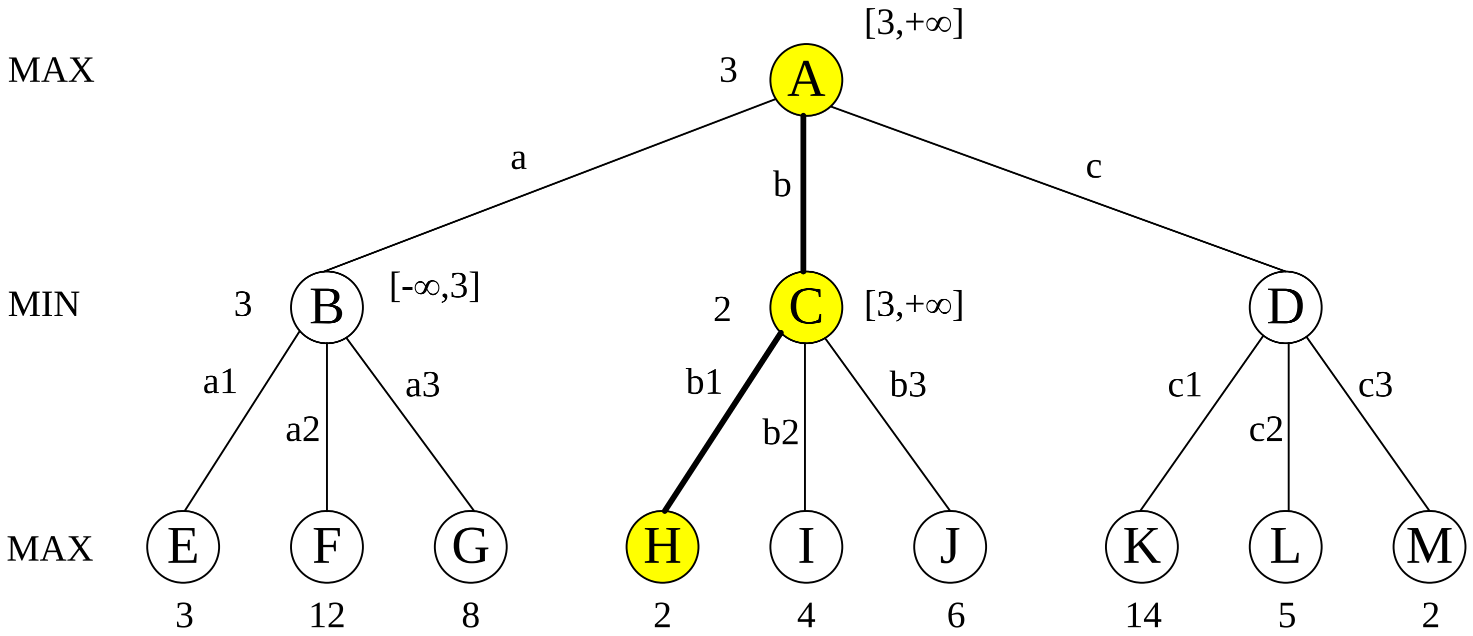

[-∞,+∞]
MAX
-∞
A
a
b
c
MIN
+∞
B
[-∞,+∞]
C
D
a1
a2
a3
b1
b2
b3
c1
c2
c3
MAX
E
F
G
H
I
J
K
L
M
3
12
8
2
4
6
14
5
2

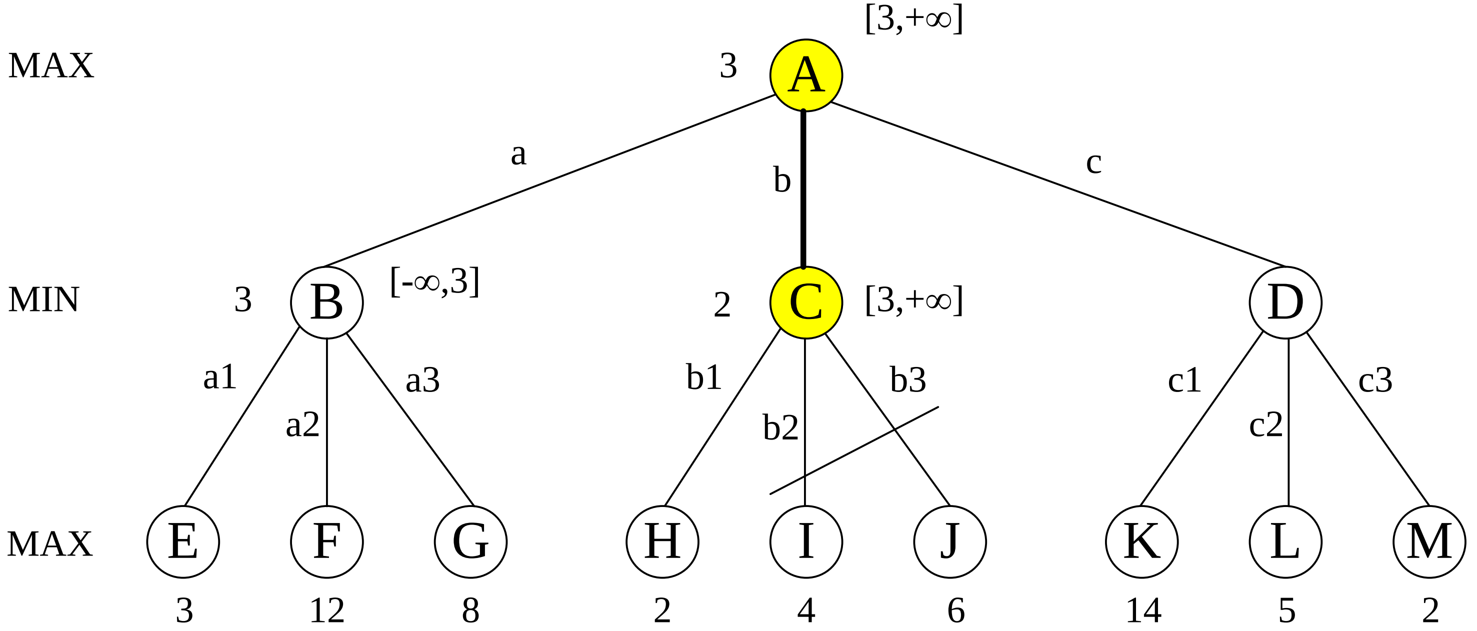

[-∞,+∞]
MAX
-∞
A
a
b
c
MIN
3
B
[-∞,3]
C
D
a1
a2
a3
b1
b2
b3
c1
c2
c3
MAX
E
F
G
H
I
J
K
L
M
3
12
8
2
4
6
14
5
2

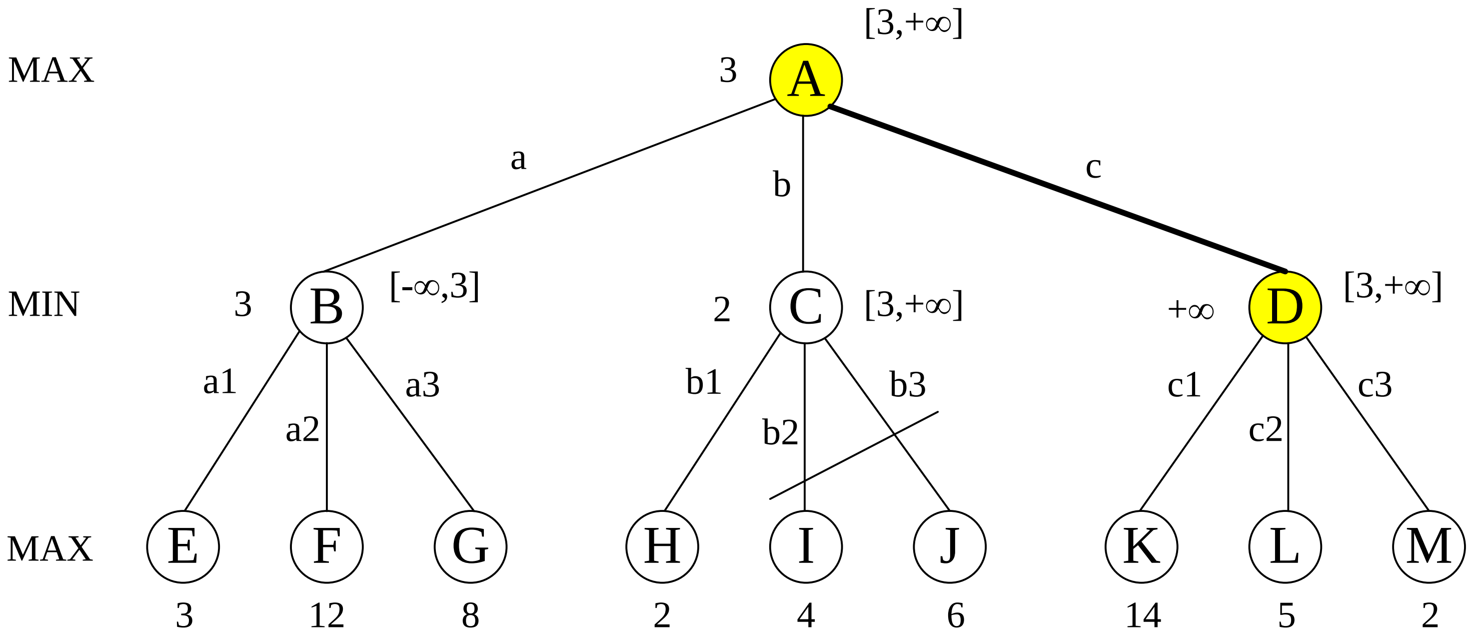

[-∞,+∞]
MAX
-∞
A
a
b
c
MIN
3
B
[-∞,3]
C
D
a1
a2
a3
b1
b2
b3
c1
c2
c3
MAX
E
F
G
H
I
J
K
L
M
3
12
8
2
4
6
14
5
2

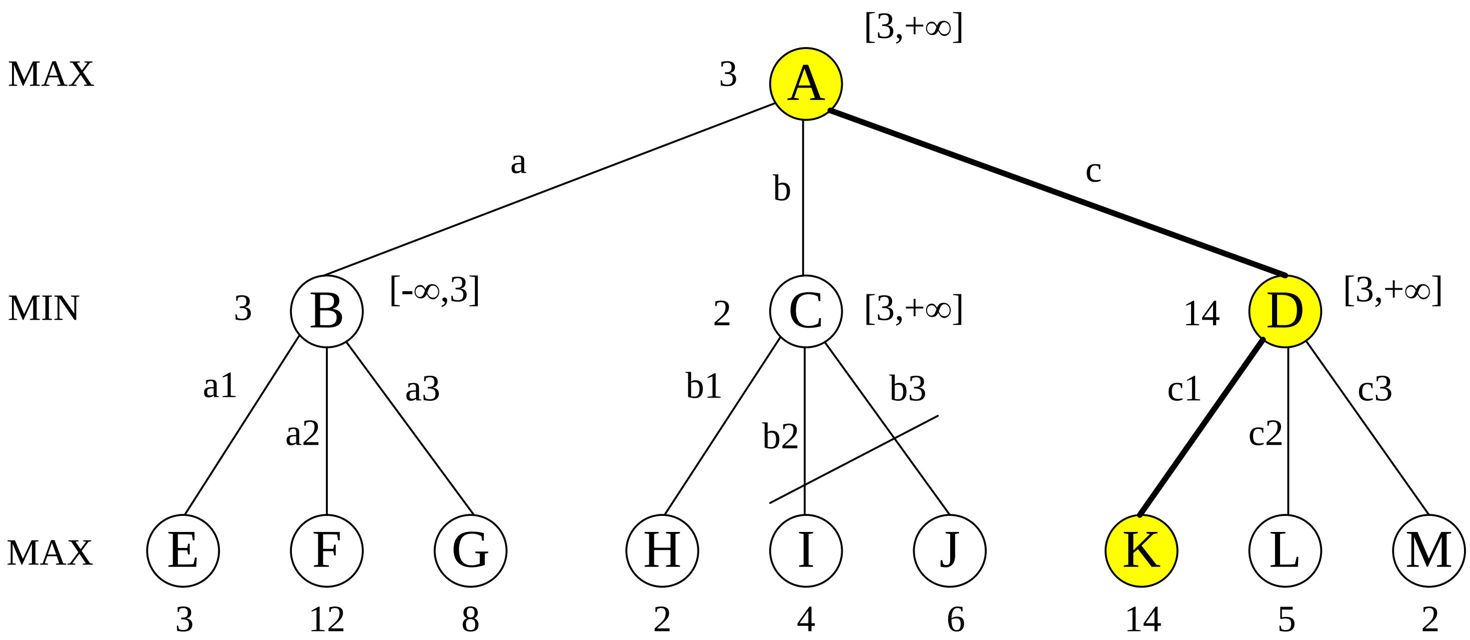

[-∞,+∞]
MAX
-∞
A
a
b
c
MIN
3
B
[-∞,3]
C
D
a1
a2
a3
b1
b2
b3
c1
c2
c3
MAX
E
F
G
H
I
J
K
L
M
3
12
8
2
4
6
14
5
2

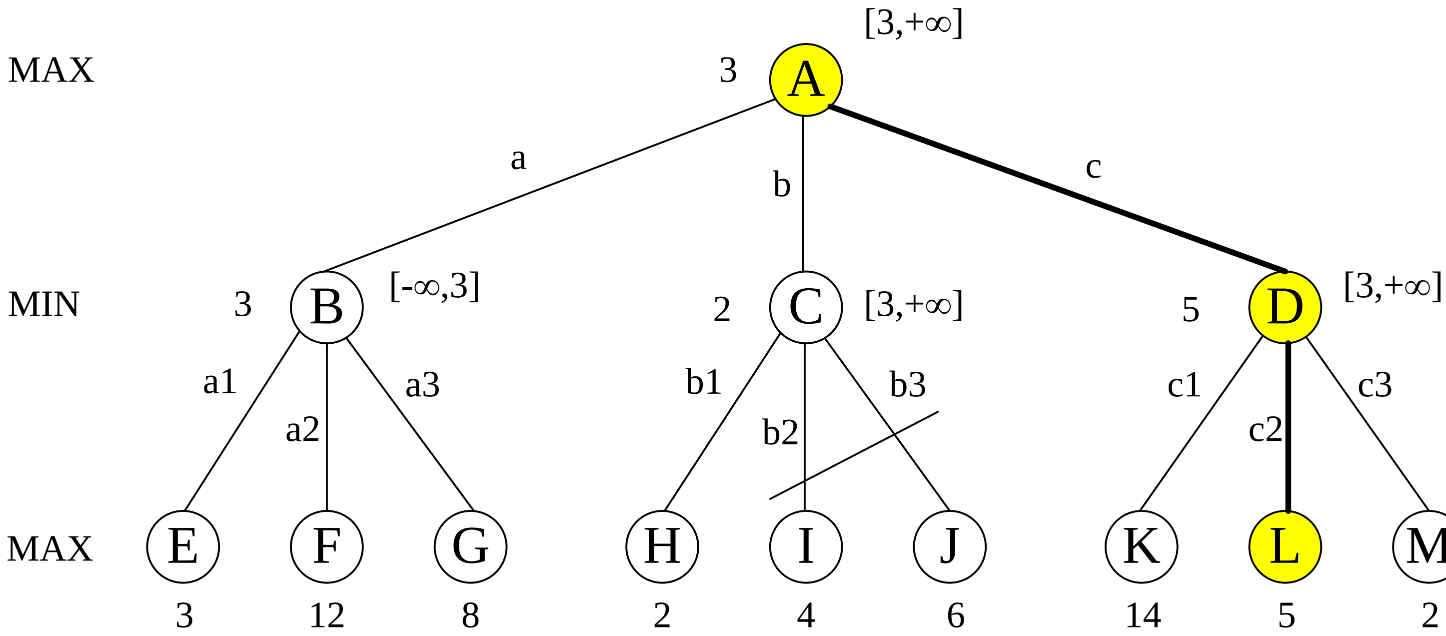

[3,+∞]
MAX
3
A
a
b
c
MIN
3
B
[-∞,3]
+∞
C
[3,+∞]
D
a1
a2
a3
b1
b2
b3
c1
c2
c3
MAX
E
F
G
H
I
J
K
L
M
3
12
8
2
4
6
14
5
2

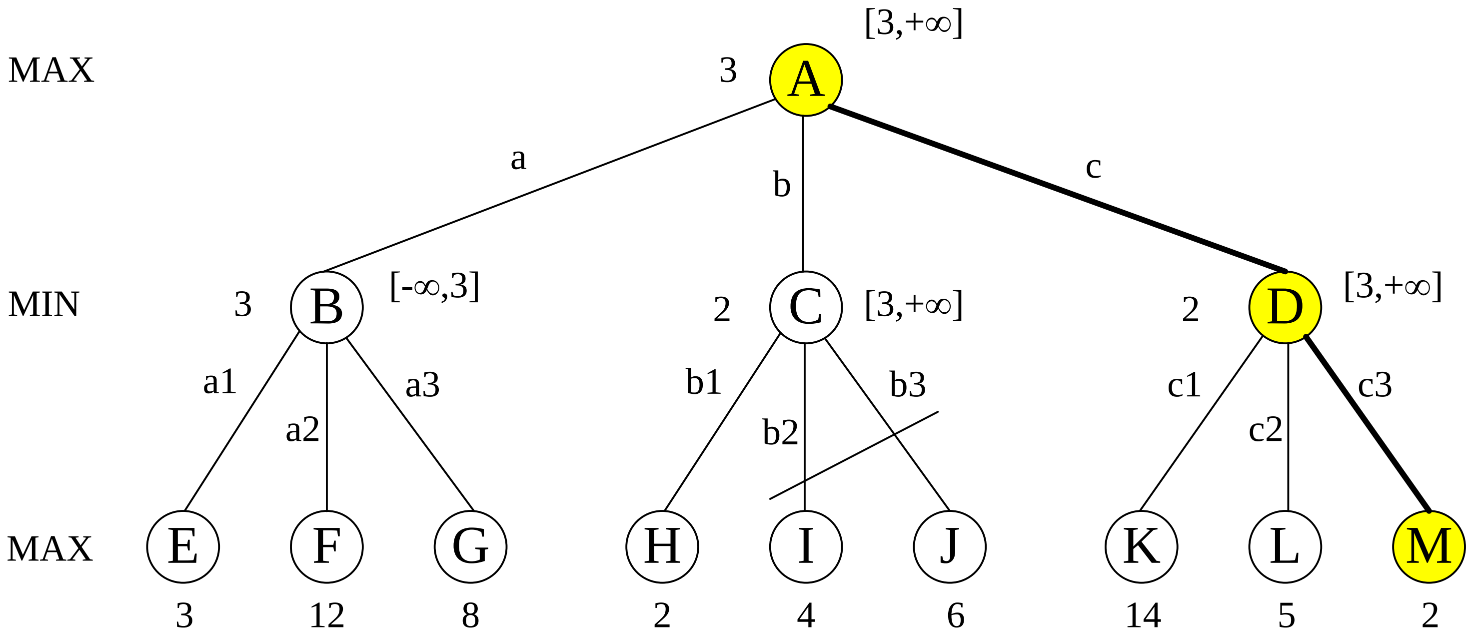

[3,+∞]
MAX
3
A
a
b
c
MIN
3
B
[-∞,3]
2
C
[3,+∞]
D
a1
a2
a3
b1
b2
b3
c1
c2
c3
MAX
E
F
G
H
I
J
K
L
M
3
12
8
2
4
6
14
5
2


[3,+∞]
MAX
3
A
a
b
c
MIN
3
B
[-∞,3]
2
C
[3,+∞]
D
a1
a2
a3
b1
b2
b3
c1
c2
c3
MAX
E
F
G
H
I
J
K
L
M
3
12
8
2
4
6
14
5
2

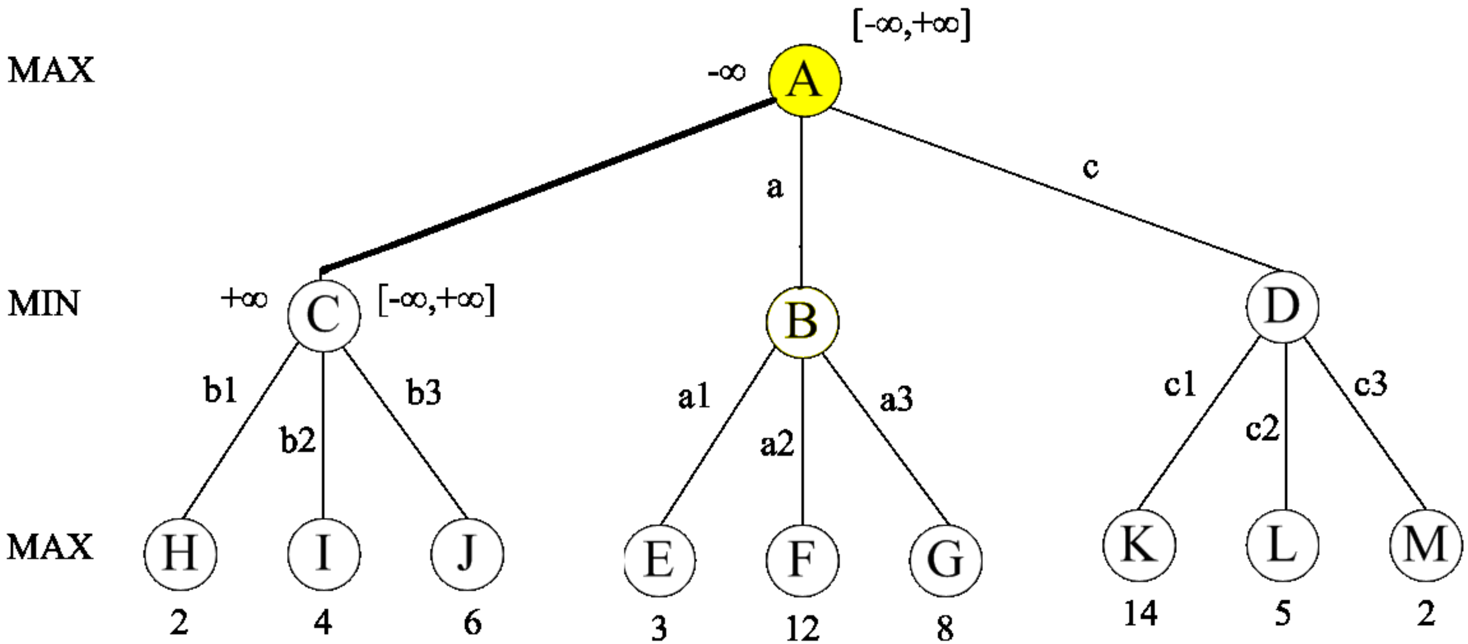

[3,+∞]
MAX
3
A
a
b
c
MIN
3
B
[-∞,3]
2
C
[3,+∞]
+∞
D
[3,+∞]
a1
a2
a3
b1
b2
b3
c1
c2
c3
MAX
E
F
G
H
I
J
K
L
M
3
12
8
2
4
6
14
5
2


[3,+∞]
MAX
3
A
a
b
c
MIN
3
B
[-∞,3]
2
C
[3,+∞]
14
D
[3,+∞]
a1
a2
a3
b1
b2
b3
c1
c2
c3
MAX
E
F
G
H
I
J
K
L
M
3
12
8
2
4
6
14
5
2


[3,+∞]
MAX
3
A
a
b
c
MIN
3
B
[-∞,3]
2
C
[3,+∞]
5
D
[3,+∞]
a1
a2
a3
b1
b2
b3
c1
c2
c3
MAX
E
F
G
H
I
J
K
L
M
3
12
8
2
4
6
14
5
2

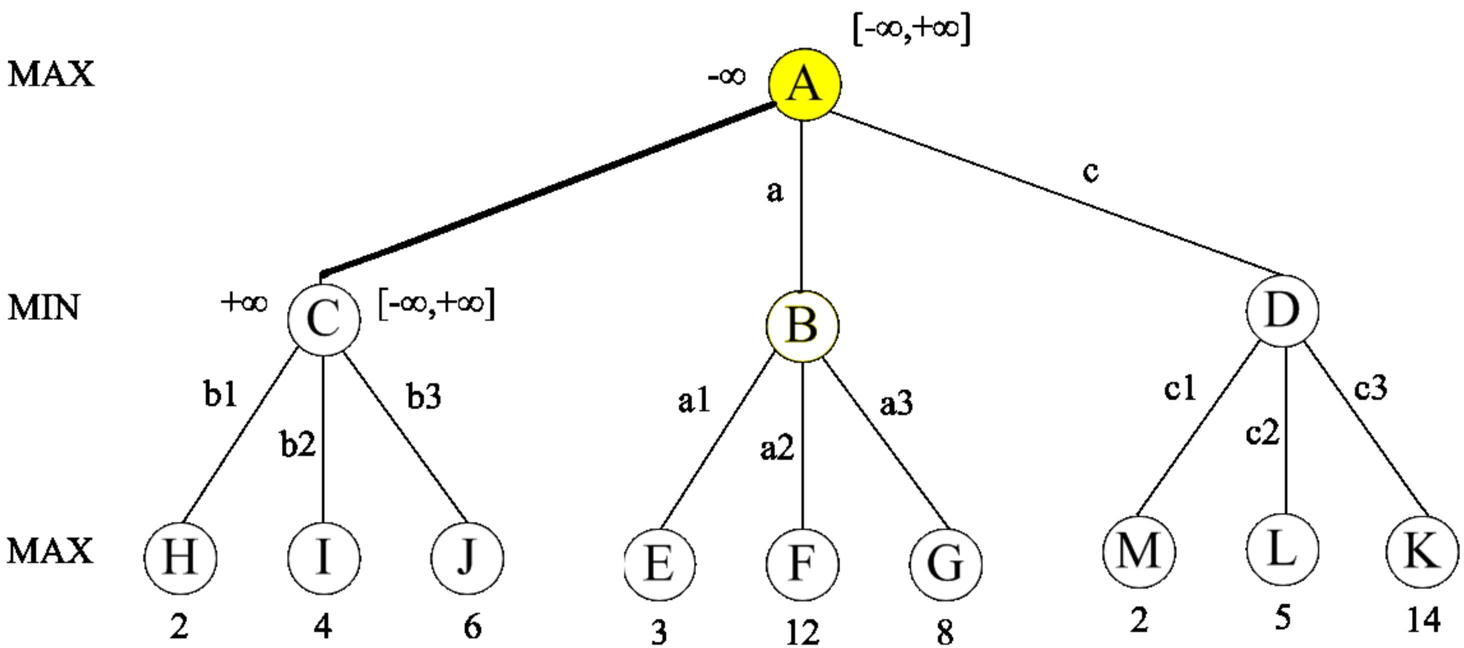

[3,+∞]
MAX
3
A
a
b
c
MIN
3
B
[-∞,3]
2
C
[3,+∞]
2
D
[3,+∞]
a1
a2
a3
b1
b2
b3
c1
c2
c3
MAX
E
F
G
H
I
J
K
L
M
3
12
8
2
4
6
14
5
2

C ⟺ B?



[-∞,+∞]
MAX
-∞
A
a
c
MIN
+∞
C
[-∞,+∞]
B
D
b1
b2
b3
a1
a2
a3
c1
c2
c3
MAX
H
I
J
E
F
G
K
L
M
2
4
6
3
12
8
14
5
2

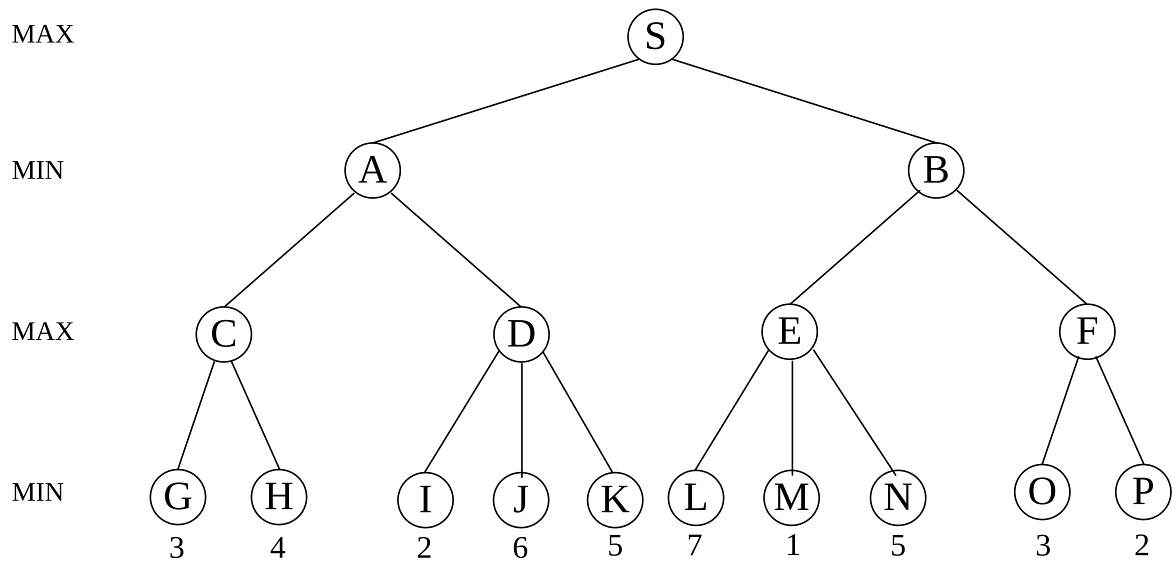
K ⟺ M?



[-∞,+∞]
MAX
-∞
A
a
c
MIN
+∞
C
[-∞,+∞]
B
D
b1
b2
b3
a1
a2
a3
c1
c2
c3
MAX
H
I
J
E
F
G
M
L
K
2
4
6
3
12
8
2
5
14

Zadatak 3: Progresivno produbljivanje

- Za svaki list je naznačena statička vrednost. Uobičajeno, što je statička vrednost veća, bolja je situacija iz prespektive MAX igrača



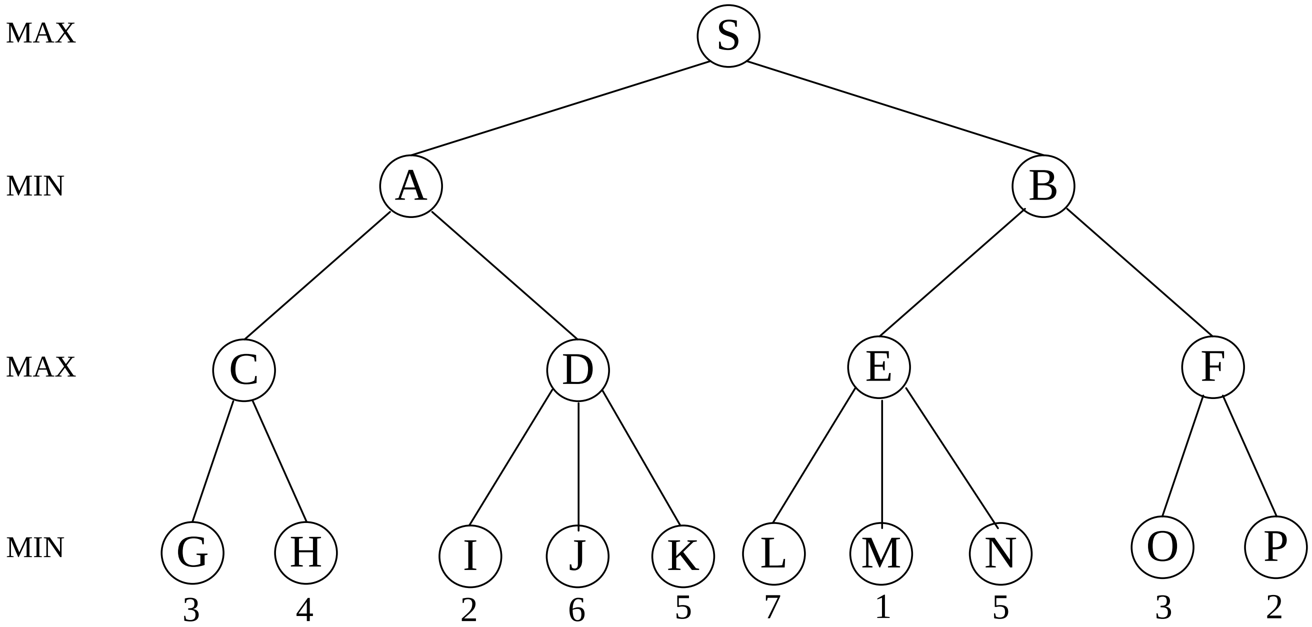
a) Ukoliko se upotrebljava metoda progresivnog produbljivanja, koliko statičkih vrednosti će biti izračunato?

b) Pretpostaviti da je moguće izvršiti razmeštanje grana koje izlaze iz istog čvora na dubini 2 (C, D, E i F). Da li bi to smanjilo broj izračunavanja statičkih vrednosti prilikom alfa-beta pretrage?

Progressive deepening

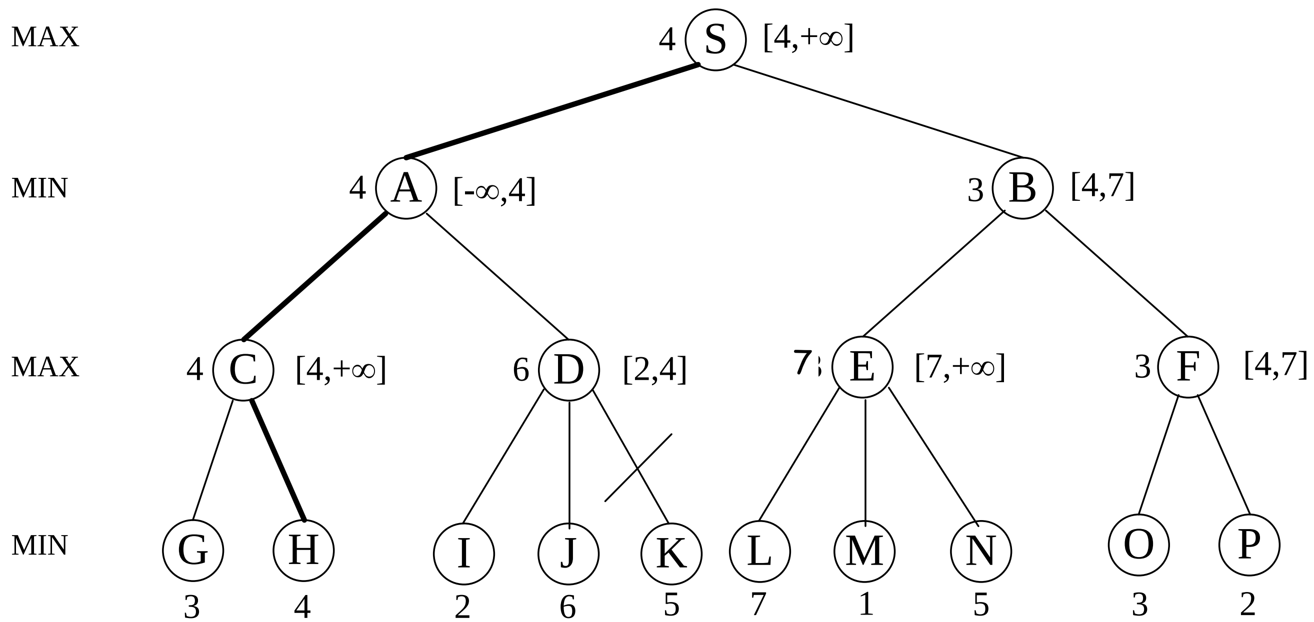
- Na turnirima od igrača se očekuje da napravi odredjeni broj poteza u zadatom vremenu ograničenom od strane nemilosrdnog sata
- Česta je primena metode progresivnog produbljivanja
- Najpre se analizira ishod za dubinu 1, zatim za dubinu 2, pa dubinu 3, ...
- Kada vreme istekne, rezultat je najbolji potez sa nivoa koji je za jedan manji od tekućeg
- Uvek postoji potez spreman da se odigra!

Rešenje (a)



2 + 4 + 10

Rešenje (b)

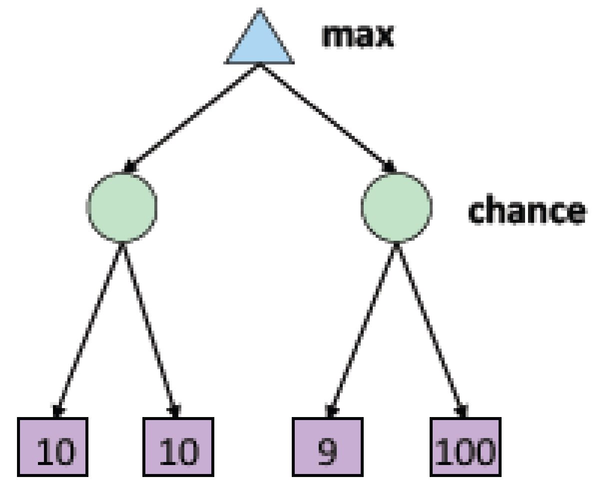


2 + 4 + 9

Šta ako zamenimo E i F? 2 + 3

Stohastički igrač

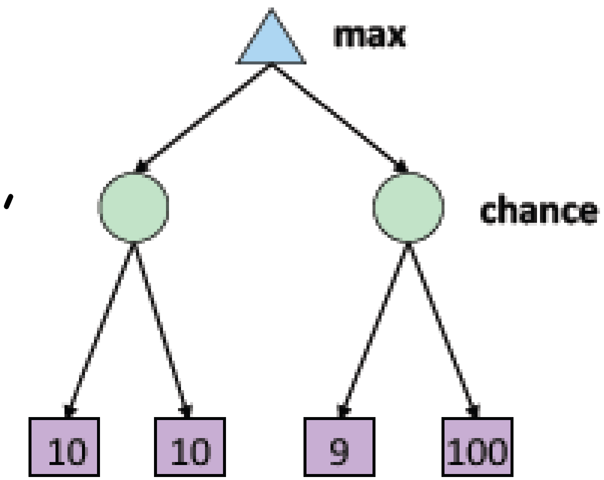
- Šta ako ne znamo šta će biti rezultat akcije?
  - podela karata u pokeru
  - bacanje kockica
- Vrednosti treba da predstavljaju prosek ishoda, expectimax, a ne najgori slučaj (minimax)



*(stohastički sistem je onaj čije stanje je nedeterminističko)

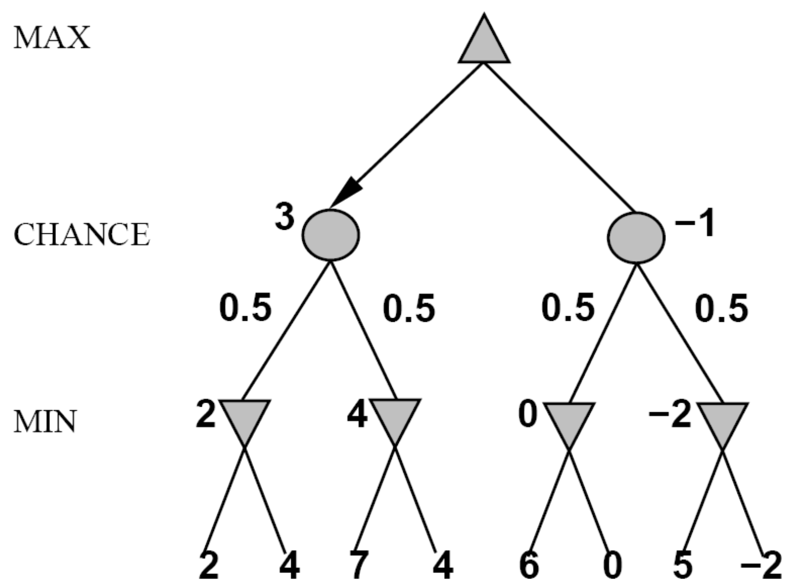
Expectimax pretraga

- Izračunaj prosečan rezultat pod optimalnom igrom
  - *max* čvorovi kao kod minimax
  - *chance* čvorovi kao *min* čvorovi, ali je ishod neizvestan
  - izračunati njihove očekivane vrednosti, odnosno izračunati prosek sa težinama njihove dece (eng. weighted average)



Expectimax pretraga

if *state* is a MAX node **then**
  return the highest EXPECTIMINIMAX-VALUE of SUCCESSORS(*state*)
if *state* is a MIN node **then**
  return the lowest EXPECTIMINIMAX-VALUE of SUCCESSORS(*state*)
if *state* is a chance node **then**
  return average of EXPECTIMINIMAX-VALUE of SUCCESSORS(*state*)



Verovatnoća

- ***Slučajna promenljiva*** predstavlja događaj čiji je ishod nepoznat
- ***Raspodela verovatnoće*** je dodela težina ishodima
- Primer: kiša
  - slučajna promenljiva: K = ako ima gužve
  - ishodi: K € {nema, blaga, jaka}
  - distribucija: P(K=nema) = 0.25, P(K=blaga) = 0.55, P(K=jaka) = 0.20

Verovatnoća

- Vremenom, kako se dobijaju nove informacije, verovatnoća može da se promeni.
- Primer: kiša
  - P(K=jaka) = 0.20,
    P(K=jaka| oblačno=da) = 0.60

Očekivanje

- Vrednosti koje nas zanimaju vrlo često zavise od slučajne promenljive
- ***Očekivana vrednost*** (expected value) diskretne slučajne promenljive je zbor verovatnoća za svaki ishod pomnožen vrednošću tog ishoda.
- Predstavlja očekivanu vrednost ako se slučajni eksperiment ponovi veliki broj puta.
- Ta vrednost ne mora biti među vrednostima koje uzima slučajna promenljiva

Očekivanje

- Očekivanje funkcije slučajne promenljive:

$$E_{P(X)}[f(X)] = \sum_x f(x)P(x)$$

- Primer: očekivane vrednost bacanje fer kockice:

$$1 \times \frac{1}{6} + 2 \times \frac{1}{6} + 3 \times \frac{1}{6} + 4 \times \frac{1}{6} + 5 \times \frac{1}{6} + 6 \times \frac{1}{6} = 3.5$$

X	P	f
1	1/6	1
2	1/6	2
3	1/6	3
4	1/6	4
5	1/6	5
6	1/6	6

Expectimax pretraga

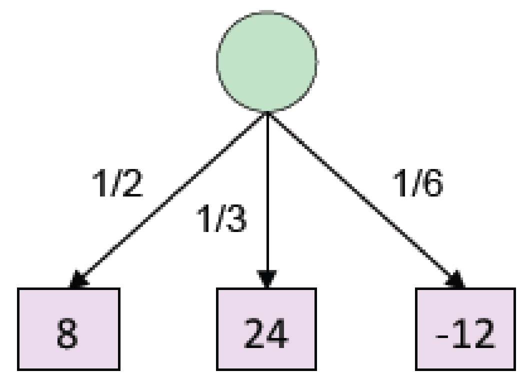
```
def value(state):
    if the state is a terminal state: return the state's utility
    if the next agent is MAX: return max-value(state)
    if the next agent is EXP: return exp-value(state)
```

```
def max-value(state):
    initialize v = -∞
    for each successor of state:
        v = max(v, value(successor))
    return v
```

```
def exp-value(state):
    initialize v = 0
    for each successor of state:
        p = probability(successor)
        v += p * value(successor)
    return v
```

Expectimax pretraga

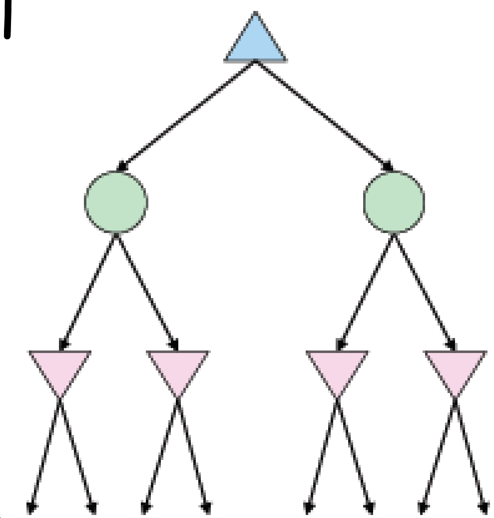
```
def exp-value(state):
    initialize v = 0
    for each successor of state:
        p = probability(successor)
        v += p * value(successor)
    return v
```



v = (1/2) (8) + (1/3) (24) + (1/6) (-12) = 10

Expectimax pretraga

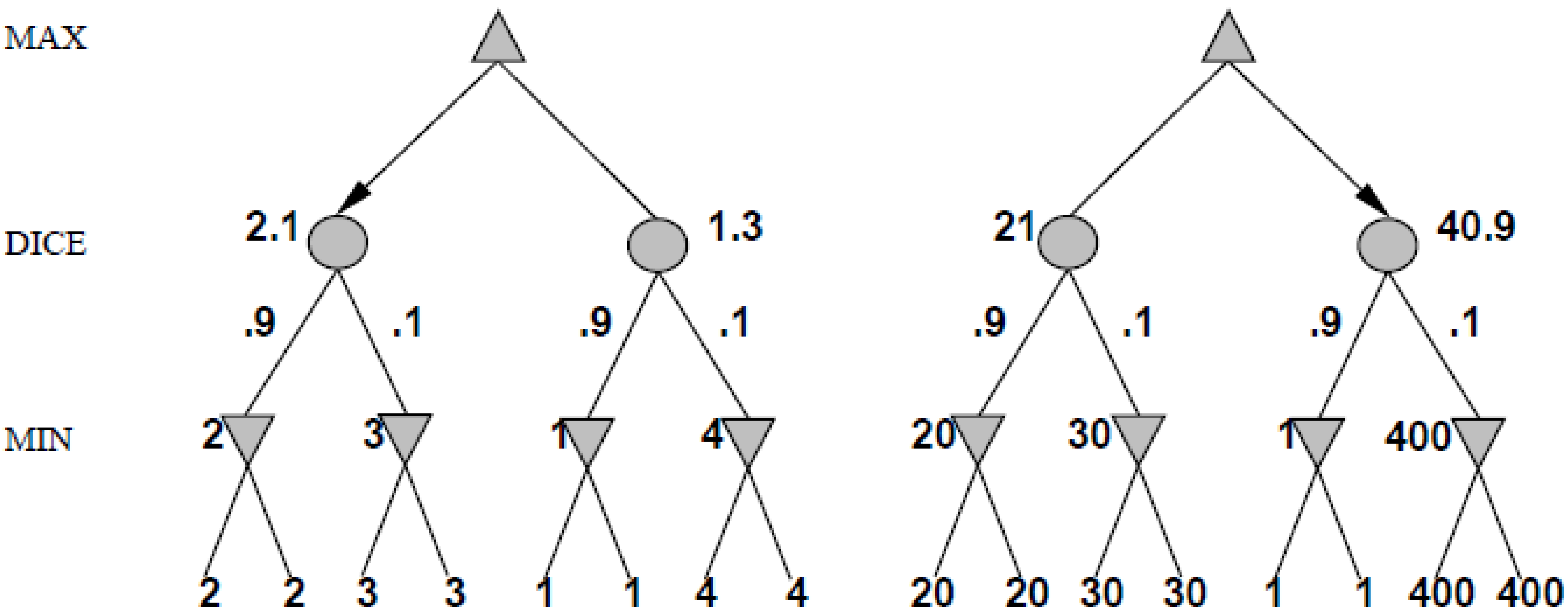
- U expectimax pretrazi imamo model verovatnoće kako će se protivnik, ili okruženje, ponašati
- Model može biti:
  - jednostavna uniformna distribucija npr. bacanje kockica
  - sofisticiran i zahtevati mnogo računanja
- Postoji čvor za svaki ishod koji je van naše kontrole: protivnik ili okruženje



Expectimax evaluacija

- Za minimax pretragu je potrebno da bolja stanja dobiju veće vrednosti funkcije procene, da bismo mogli ispravno da ih uredimo
- Za expectimax želimo da veličine budu smislene:
  - da li je 50% / 50% šansa između A i B bolja od 100% šanse za C
  - 10 ili -100 naspram 0 je različito od -10 ili 100 naspram 0

Expectimax evaluacija



Neizvesnost

- Nije prisutna samo igrama
  - kašljem, da li sam bolestan?
  - email sadrži „FREE!!!“, da li je spam?
  - zub boli, da li je pokvaren?
  - pola sata je dovoljno do posla?
  - ...