National And Kapodistrian University of Athens
Department of Informatics and Telecommunications

# Project #2

Student name: *Stylianos Bitzas (sma1100202)*

Course: *Artificial Intelligence (ΥΣ02)* – Professor: *Manolis Koubarakis*
Due date: *December 3rd, 2019*

**Πρόβλημα 1**

Θεωρήστε το παιχνίδι της τρίλιζας. Υποθέστε ότι η μεταβλητή X n μας δίνει τον αριθμό των γραμμών, στηλών ή διαγωνίων που έχουν ακριβώς n σύμβολα X και κανένα σύμβολο O. Ομοίως υποθέστε ότι η μεταβλητή O n μας δίνει τον αριθμό των γραμμών, στηλών ή διαγωνίων που έχουν ακριβώς n σύμβολα O και κανένα σύμβολο X. Η συνάρτηση χρησιμότητας αναθέτει +1 σε κάθε κατάσταση με $X_3 = 1$ και −1 σε κάθε κατάσταση με $O_3 = 1$. Όλες οι άλλες τερματικές καταστάσεις έχουν χρησιμότητα 0. Για μη τερματικές καταστάσεις s χρησιμοποιούμε τη γραμμική συνάρτηση αποτίμησης $Eval(s) = 3X_2(s) + X_1(s) - (3O_2(s) + O_1(s))$. Να απαντήσετε τις ακόλουθες ερωτήσεις:

1. Πόσα διαφορετικά παιχνίδια τρίλιζας μπορούν να παιχτούν;
2. Σχεδιάστε όλο το δένδρο παιχνιδιού μέχρι βάθος 2 (η ρίζα του δένδρου είναι σε βάθος 0, άρα σε βάθος 2 έχουν παίξει από μία φορά οι παίκτες).
3. Σημειώστε πάνω στο προηγούμενο δένδρο παιχνιδιού όλες τις αποτιμήσεις καταστάσεων (κόμβων).
4. Υποθέστε ότι εκτελούμε τον αλγόριθμο MINIMAX -DECISION των διαφανειών. Ποια θα είναι η minimax απόφαση στη ρίζα του δένδρου; Γιατί;
5. Υποθέστε ότι εκτελούμε τον αλγόριθμο ALPHA-BETA-SEARCH των διαφανειών (δηλαδή, κάνουμε κλάδεμα άλφα-βήτα). Ποιους κόμβους θα κλαδέψει στον βάθος 2 αυτός ο αλγόριθμος αν υποθέσουμε ότι οι κόμβοι παράγονται με την σειρά που τους έχετε σχεδιάσει (από τα αριστερά προς τα δεξιά); Τι θα συμβεί αν οι κόμβοι παράγονται με την αντίστροφη σειρά; Ποια είναι η βέλτιστη σειρά παραγωγής των κόμβων;

**Answer.**

1. Μπόρουν να παιχτούν 9! διαφορετικά παιχνίδια τρίλιζας. Άφου, για την πρώτη κίνηση υπάρχουν 9 επιλόγες, για τη δεύτερη κίνηση 8 επιλογές κ.ο.κ. Από πολλαπλασιαστική αρχή έχουμε $9 * 8 * 7 * 6 * 5 * 4 * 3 * 2 * 1 = 9! = 362.880$ παιχνίδια. Φυσικά αυτό είναι ενά χοντροκομμένο άνω φράγμα αφού, για κάθε παιχνίδι που τελείωνει σε λιγότερο από 9 κινήσεις , υπάρχουν πολλά αλλά ισοδύναμα όπου οι υπόλοιπες κινήσεις δεν παίζουν ρόλο. Επιπλέον αν σε αυτή τη μείωση υπολογίσει κανείς και τις συμμετρίες που δημιουργούνται από περιστροφές και ανακλάσεις ο αριθμός μειώνεται κι άλλο.

2. Οτάν το βαθός είναι 2, δηλαδή οι παίκτες έχουν πάιξει μία φορά, υπάρχουν 72 καταστάσεις οπότε για απλότητα θα αφαιρέσω τις συμμετρικές καταστάσεις. Με τον όρο συμμετρικές καταστάσεις εννοώ, συμμετρικές ως προς τις περιστροφές και τις ανακλάσεις(κάθετα και οριζόντια). Έχοντας αυτό στο νου ο πρώτος παίκτης έχει την επιλογή να παίξει σε μία γωνία. Έστω ότι αυτή είναι η πάνω αριστερή γώνια, τότε περιστρέφοντας το ταμπλό κατά 90° για μία ή περισσοτέρες φορές, βλέπουμε ότι όλες οι γωνίες παράγουν το ίδιο παιχνίδι, δηλάδη η πάνω αριστέρη γωνία είναι η μόνη επιλογή που χρειαζόμαστε. Ακριβώς ανάλογα μπορουμέ να καταλήξουμε ότι η δεύτερη επιλόγη που έχει ο πρώτος παίκτης είναι να παίξει στο μεσαίο κουτάκι στην πρώτη γραμμή(με περιστροφές ολά τα άλλα εξωτερικά μεσαία κουτάκια είναι ισοδύναμα). Τέλος μένει μόνο μία επιλογή που δεν έχει εξεταστεί, όπου ο πρώτος παίκτης διαλέγει το κεντρικό κουτάκι και αυτή η κίνηση είναι προφανώς μοναδίκη ως προς συμμετρίες. Άρα από τις 9 επιλογές για τον πρώτο παίκτη, οι ουσιαστικά διαφορετικές είναι 3. Παρόμοια ελέγχει κανείς για τον δεύτερο παίκτη και από τα $24(3*8)$ παιχνίδια που δημιουργούνται μετά την κίνηση του δεύτερου παίκτη μένουν μόνο 12 διαφόρετικα ως προς τη συμμετρία που έχουμε ορίσει. Το δένδρο που παράγεται είναι το παρακάτω. (τις καταστάσεις τις έχω δημιουργήσει με Python και το tkinter ενώ το διάγραμμα με την ιστοσελίδα draw.io και δεν αποτελούν προΐον αντιγραφής ή κλοπής)

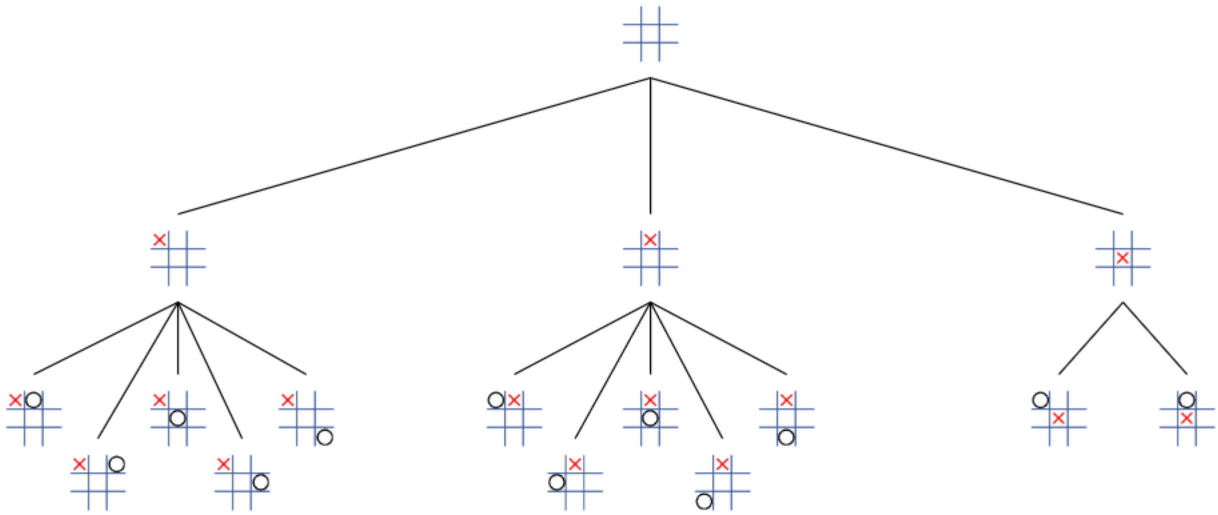


Figure 1: Tic Tac Toe Tree

3. Επειδή οι δύο παίκτες έχουν παίξει από μία φορά τα $X_2(s), 0_2(s) = 0, \forall s$, η συνάρτηση αποτίμησης είναι η $Eval(s) = X_1(s) - O_1(s)$. Στο παρακάτω δένδρο είναι υπολογισμένες όλες οι αποτιμήσεις των καταστάσεων θεωρώντας ότι η ρίζα δεν έχει τίμη αποτίμησης. Ενδεικτίκα για τον υπολογισμό των τιμών αυτών η τιμή αποτίμησης για το πρώτο φύλλο, από αρίστερα, είναι $Eval(s) = X_1(s) - O_1(s) = 3 - 2 = 1$ και για το τρίτο φύλλο $Eval(s) = X_1(s) - O_1(s) = 3 - 4 = -1$ παρόμοια και πολύ εύκολα υπολογίζονται και οι υπόλοιπες τιμές που αναγράφονται στο δένδρο.

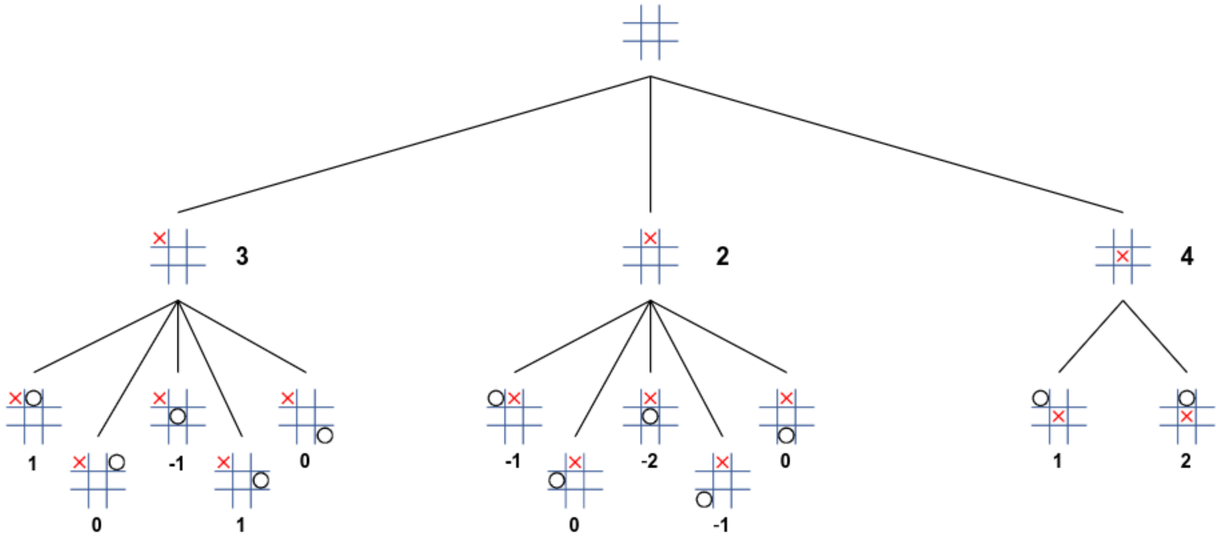



Figure 2: Evaluation

4. Με βάση τον αλγόριθμο, ο πρώτος παίκτης είναι ο max και παίζει X και ο δεύτερος παίκτης είναι ο min και παίζει O, έπειτα ο αλγόριθμος υπολογίζει την τιμή αποτίμησης για κάθε κατάσταση και τις επιστρέφει στο γονέα . Αυτός με τη σειρά του διαλέγει το ελάχιστο στοιχείο και αφού ο αλγόριθμος έχει επισκεφθεί όλες τις καταστάσεις του κόμβου min και έχει κάνει τις απαραίτητες επιστροφές των τιμών στον max, αυτός με τη σειρά του διαλέγει τη μέγιστη τιμή και την επιστρέφει στην ρίζα. Στη γλώσσα της θεωρίας παιγνίων η τιμή της ρίζας θα είναι η τιμή maximin που αποτελεί και την τιμή του παιγνιδιού που είναι μοναδική και ξέρουμε ότι υπάρχει, καθώς η τρίλιζα σε οποιοδήποτε βάθος μαζί με μία συνάρτηση αποτίμησης ή τιμή τερματικής κορυφής είναι ένα παιχνίδι μηδενικού αθροίσματος.Ακολουθεί το δένδρο και στη συνέχεια μια περιληψή του τρεξίματος με τις τιμές.

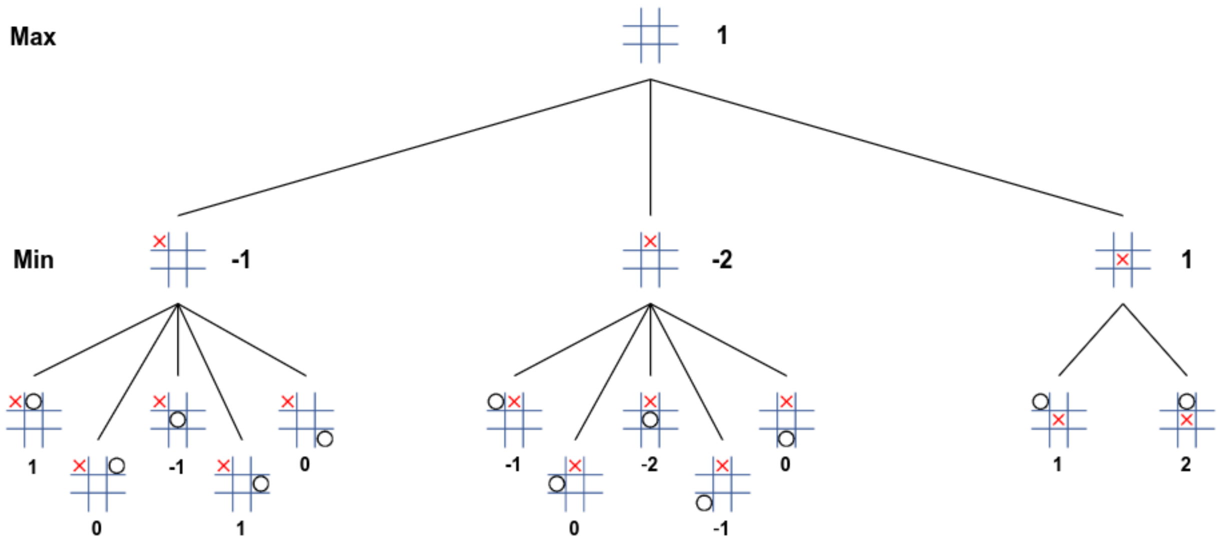



Figure 3: Values

Με πιο απλά λογία ο min διαλέγει την ελαχιστή τιμή στο αριστερό υποδένδρο που είναι το -1, μέτα διαλέγει το -2 από το μεσαίο υποδένδρο και τελός διαλέγει το 1 από το δεξί υποδένδρο. Στη συνέχεια ο max διαλέγει το max απο τα $(-1, -2, 1)$, δηλαδή το 1, και επιστρέφει στη ρίζα την τίμη 1, που αποτελεί την τιμή του παιχνιδιού.

5. Επείδη ο max κομβός δεν έχει προγόνους κάθε φορά θα τρέχει και θα αλλαζεί τις τιμές του $\alpha$ που αρχικά θα είναι $-inf$, η τιμή $\beta$ θα μείνει πάντα $+inf$ στο scope του max. Αυτό έχει σαν αποτέλεσμα όποιο κλάδεμα γίνεται,να το κάνει ο min. Όμως O min κόμβος κλαδεύει μόνο αν κάποιο παιδί επιστρέψει τιμή μικρότερη από το $\alpha$. Αφού τότε ο min θα πρέπει να διαλέξει το πολύ αυτήν την τιμή που είναι μικρότερη απο το $\alpha$ και ο max δεν προκείται να προτιμήσει αυτή την κατάσταση,διότι κάποια αλλή κατάσταση του εξασφαλίζει τουλάχιστον $\alpha$ το οποίο είναι μεγαλύτερο. Ακολουθώντας αυτή τη λογική το πρώτο δεντρό θα παραχθεί όλο καθώς ο max δεν αλλάζει το $\alpha$ παρά μόνο μετά την πρώτη επιστροφή του min. Αυτός θα επιστρέψει -1 και το $\alpha$ θα γίνει -1. Στο δευτέρο υποδένδρο θα παραχθούν ολοί οι κόμβοι, μέχρις ώτου βρεθεί η τιμή -2 που έιναι μικρότερη από $\alpha$ και τοτέ θα κλαδευτουν οι υπόλοιποι κόμβοι. Θα επιστραφεί το -2 και το $\alpha$ θα παραμείνει -1. Τέλος στο τελευταίο υποδένδρο θα παραχθούν ολές οι καταστάσεις. Σημαντικό είναι να προσέξει κανείς ότι η τιμή του παιχνιδιού είναι 1 που συμφώνει προφανώς με την τιμή του αλγορίθμου minimax. Ακολουθεί το διαγραμμά όπου έχουν αφαιρεθεί οι κόμβοι που έχουν κλαδευτεί.

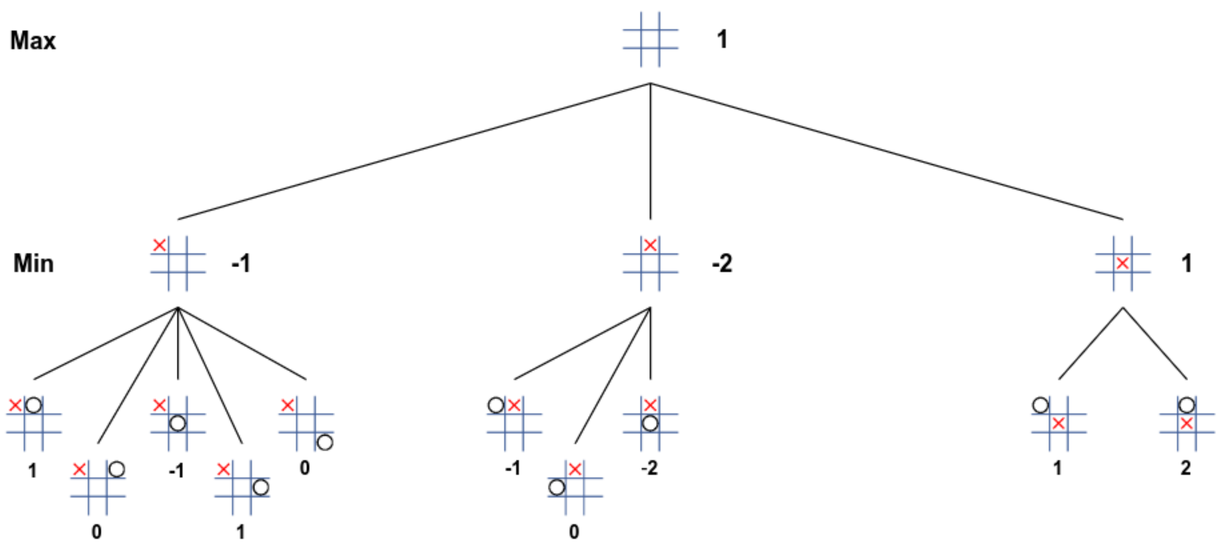



Figure 4: Normal Pruning

Με ακριβώς παρόμοια επιχειρήματα γίνεται και το κλάδεμα όπου οι κόμβοι παράγονται με την αντιστρόφη σείρα(από δεξιά προς τα αριστερά). Ακολουθεί το διαγραμμά όπου έχουν αφαιρεθεί οι κόμβοι που έχουν κλαδευτεί καθώς και οι τιμές που επιστρέφονται σε κάθε κατάσταση. Παρατηρώ πάλι ότι η τιμή είναι η ίδια με πριν.

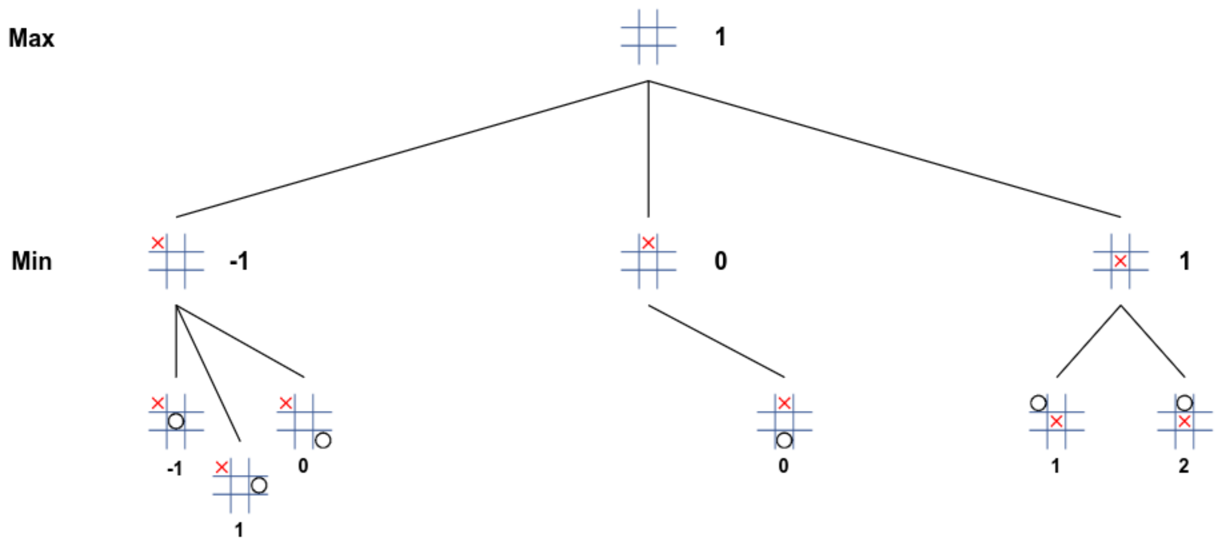



Figure 5: Reversed Pruning

Τέλος η καλύτερη σειρά παραγωγής κόμβων θα ήταν, από αριστερά προς τα δεξία ξεκινώντας από την μεγαλυτερή τιμή αποτίμησης και συνεχίζοντας με φθίνουσα σειρά. Καθώς έτσι η τίμη maximin θα βρισκέται σίγουρα στο αριστερό υποδένδρο(και μάλιστα στο τελευταίο δεξιό παιδί), με αποτέλεσμα ο max να έχει σταθεροποιήσει ένα $\alpha$ μέγιστο που καμία τιμή πιο δεξιά δεν θα είναι μεγαλύτερη. Τελικά σε κάθε επισκέψη οποιουδήποτε υποδένδρου το πρώτο αριστερό παίδι θα είναι αρκετό για να κλαδευτούν όλα τα επόμενα παιδιά. Αν δεν μπορεί να επιτευχθεί πλήρης ταξινόμηση, η παραγωγή κόμβων με μεγάλες τίμες όσο πιο αριστερά γίνεται είναι αρκετά καλή.

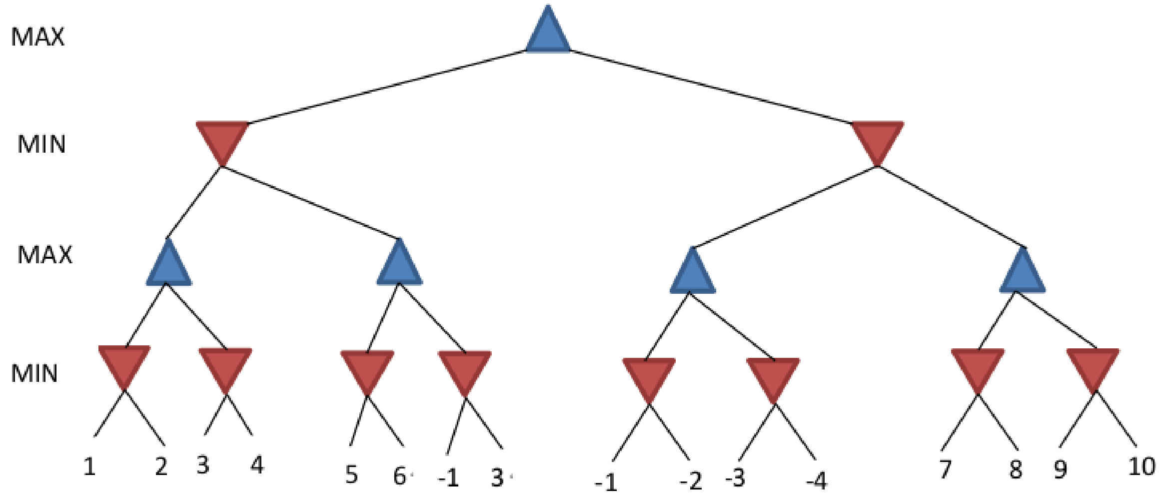
## Πρόβλημα 2

Θεωρήστε ένα δένδρο παιχνιδιού με βάθος δύο δηλ. με μία κίνηση για τον κάθε παίκτη. Ποια ιδιότητα πρέπει να έχουν οι τιμές χρησιμότητας στα φύλλα του δένδρου ώστε ο αριθμός των κόμβων που κλαδεύεται με την τεχνική άλφα-βήτα είναι μέγιστος; Ελάχιστος;

**Answer.** Από το τελευταίο μέρος του ερωτήματος 5 στο πρόβλημα 1 έπεται το ζητούμενο για μέγιστο κλάδεμα. Για ελάχιστο κλάδεμα αρκεί να παράγονται οι κόμβοι με αύξουσα σειρά, οπότε και πότε δεν θα ικανοποιείται η συνθήκη $v < \alpha$ και άρα δεν θα έχουμε καθόλου κλάδεμα.

## Πρόβλημα 3

Θεωρήστε το παρακάτω δένδρο παιχνιδιού.




(a) Για κάθε κόμβο που δεν είναι φύλλο, να υπολογίσετε την minimax τιμή του.

(b) Ποια είναι η minimax απόφαση στη ρίζα του δένδρου;

(c) Να δώσετε όλους τους κόμβους οι οποίοι κλαδεύονται από τον αλγόριθμο ALPHA-BETA-SEARCH όταν αυτός εκτελεστεί για το παραπάνω δένδρο.Να δείξετε αναλυτικά την εκτέλεση του αλγόριθμου και να υποθέσετε ότι τα παιδιά ενός κόμβου επισκέπτονται από τα αριστερά προς τα δεξιά.

**Answer.**

(a) Ακολουθεί το διάγραμμα με τις τιμές minimax.

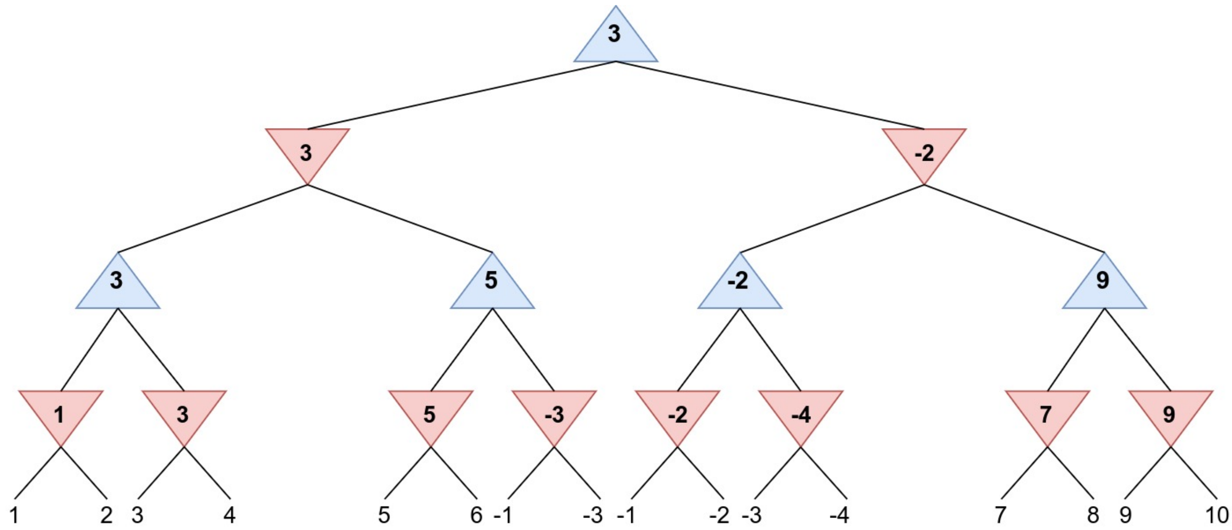



Figure 6: Minimax Tree

(b) Από το (a) έπεται ότι η τιμή maximin(που είναι ίση με τη minimax από θεωρία παιγνίων) στην ρίζα είναι 3.

(c) Επειδή έχω αναλύσει και σε προηγουμένο ερώτημα τον αλγόριθμο Alpha-Beta θα κάνω μία περηλιπτική και γρήγορη εκτέλεση για το δένδρο αυτό. Ξεκινώντας με τον πρώτο κόμβο max, από αριστερά στο επίπεδο 2, το υποδενδρό θα δημιουργηθεί ολόκληρο και θα επιστρέψει στον κόμβο τιμή 3 Αυτό με τη σειρά του θα επιστραφεί στον πρώτο κόμβο min (βάθος 1) και η τιμή του $\beta$ θα γίνει 3 και άρα $[\alpha, \beta] = [-\infty, 3]$. Έπειτα στον δεύτερο κόμβο max(βαθός 2), το αριστερό παιδί είναι κόμβος min και καθώς το $\alpha = -\infty$ θα παραχθούν όλοι οι κόμβοι και ο min θα επιστρέψει την τιμή 5. Σε αυτό το σημείο ο max θα κλαδέψει το δεξίο παιδί, καθώς $5 > \beta = 3$. Στη συνέχεια ο κόμβος max επιστρέφει 5 και ο min γονιός διαλέγει την τιμή 3 και την επιστρέφει στον max στην ρίζα, όπου το $[\alpha, \beta] = [3, +\infty]$. Προχωρώντας στο δευτέρο κόμβο min(βαθός 1) και με παρομοία λογική , το αριστερό υπόδενδρο θα παραχθεί όλο και θα επιστρέψει τιμή -2. Σε αυτό το σημείο ο κόμβος min(βαθος 1) θα κλαδέψει το δεξίο παιδί του καθώς το $-2 < \alpha = 3$. Τελικά προκύπτει το παρακάτω δένδρο όπου έχουν αφαιρεθεί οι κόμβοι που έχουν κλαδευτεί.

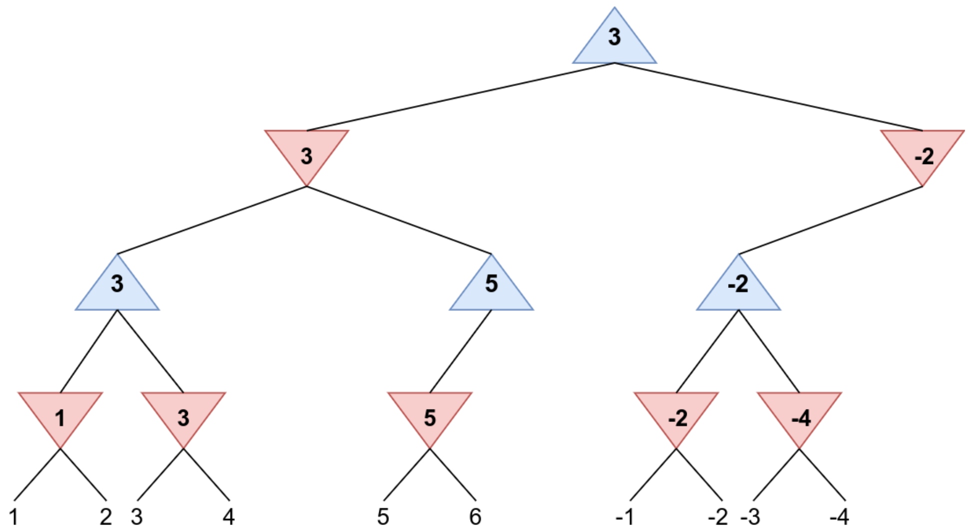



Figure 7: AlpbaBeta Tree

## Πρόβλημα 4

Ορίζουμε ένα δένδρο max να είναι ένα δένδρο που έχει μόνο MAX κόμβους. Ορίζουμε επίσης ένα δένδρο expectimax να είναι ένα δένδρο το οποίο έχει ένα κόμβο MAX στη ρίζα του και μετά μια ακολουθία επιπέδων από κόμβους CHANCE και κόμβους MAX που εναλλάσσονται. Στους κόμβους CHANCE οι πιθανότητες των αποτελεσμάτων είναι μη μηδενικές. Για κάθε μία από τις παρακάτω περιπτώσεις (a)-(h), δώστε ένα παράδειγμα ή εξηγήστε ότι αυτό που ρωτάμε είναι αδύνατο.

(a) Αν υποθέσουμε ότι η τιμή κάθε φύλλου ενός δένδρου max είναι οποιοσδήποτε πραγματικός αριθμός, μπορείτε να προτείνετε ένα αλγόριθμο κλαδέματος όπως το κλάδεμα άλφα-βήτα;

(b) Απαντήστε την ερώτηση του (a) για την περίπτωση ενός δένδρου expectimax

(c) Αν οι τιμές των φύλλων σε ένα δένδρο max είναι όλες αρνητικές ή μηδέν, υπάρχουν περιπτώσεις που μπορούμε να κλαδέψουμε κάποια κλαδιά του δένδρου;

(d) Αν οι τιμές των φύλλων σε ένα δένδρο expectimax είναι όλες αρνητικές ή μηδέν, υπάρχουν περιπτώσεις που μπορούμε να κλαδέψουμε κάποια κλαδιά του δένδρου;

(e) Απαντήστε το ερώτημα (c) για την περίπτωση που οι τιμές των φύλλων είναι θετικές ή μηδέν.

(f) Απαντήστε το ερώτημα (d) για την περίπτωση που οι τιμές των φύλλων είναι θετικές ή μηδέν.

(g) Αν οι τιμές των φύλλων σε ένα δένδρο max είναι στο διάστημα [0,1], υπάρχουν περιπτώσεις που μπορούμε να κλαδέψουμε κάποια κλαδιά του δένδρου;

(h) Αν οι τιμές των φύλλων σε ένα δένδρο expectimax είναι στο διάστημα [0,1], υπάρχουν περιπτώσεις που μπορούμε να κλαδέψουμε κάποια κλαδιά του δένδρου;

**Answer.**

(a) Όχι είναι αδύνατο. Επειδή στον Alpha-Beta αλγόριθμο η max μέθοδος για να κάνει pruning ελέγχει την υπόθεση $v > \beta$ και το $\beta$ δεν αλλάζει πότε μέσα σε κόμβους max(μόνο σε κόμβους min), άρα η τίμη του $\beta$ θα παραμένει για πάντα $+\infty$. Επιπλέον οι τίμες του δένδρου δεν είναι άνω φραγμένες, οπότε σε κανένα σημείο δεν μπορεί να αποφανθεί η max, ότι έχει βρει τη μέγιστη τιμή και γι'αυτό αναγκαστικά θα παράξει όλο το δένδρο χωρίς να κλαδέψει κανένα κόμβο.

(b) Όχι είναι αδύνατο. Η μέθοδος chance πρέπει να παράξει πάντα όλους τους κόμβους για να δώσει το μέσο όρο, αφού ότι τιμή και να επιστρέψει το παιδί max, δεν μπορεί ο chance να αποφανθεί για τις επόμενες καθώς εξαρτώνται από το βάρος που τους δίνει η συνάρτηση πιθανότητας. Απο την αλλή πλευρά επειδή ο chance επιστρέφει την μέση τιμή όλων τον κόμβων, ακομά και ο τελευταίος κομβός μπορεί να επηρεάσει την συνολική τιμή (και απο την ίδια την τιμή και από το βάρος της). Αυτό έχει σαν αποτέλεσμα , να μην μπορεί πότε να αποφανθεί ο max αν πρέπει να κλαδέψει. Σε κάθε περίπτωση λοιπόν δεν θα μπορέσει να γίνει κλάδεμα.

(c) Είναι δυνατό να κλαδέψουμε καποιά κλαδιά. Αν σε κάποιο σημείο κάποιο παίδι max επιστρέψει 0, τότε δεν υπαρχει μεγαλύτερη τιμή και μπορούμε να επιστρέφουμε την τιμή κατευθείαν εως ότου φτάσουμε στη ρίζα και να τερματίσουμε τον αλγόριθμο. Αυτό συμβαίνει έπειδη όλοι οι κόμβοι max ψαχνούν το μέγιστο και έδω οι τιμές είναι άνω φραγμένες από το 0.

(d) Είναι δυνατό να κλαδέψουμε καποιά κλαδιά.Από το (b) αν βρεθούμε σε κομβό chance πρεπει να παράξουμε όλα τα παιδιά του (η περίπτωση με μηδενίκη πιθανότητα παντού θεωρείται τετριμμένη και παραλείπεται). Σε περίπτωση που είμαστε σε max κόμβο από το (c), αν κάποια στιγμή βρεθεί τιμή 0 απευθείας κλαδεύουμε τα παιδία του, καθώς δεν μπορεί να ελπίζει για κάτι καλύτερο απο 0. Τέλος επιστρέφει την τίμη 0 στον γονέα chance και ο αλγόριθμος συνεχίζει και αντιμετωπιζεί τις καταστάσεις που θα προκύψουν ανάλογα.

(e) Παρόμοια με το (a) και επειδή οι τιμές δεν είναι άνω φραγμένες, οι κόμβοι max δεν μπορούν να ξέρουν a priori αν οι επόμενες τιμές θα είναι μεγαλύτερες και έτσι δεν μπορούμε να κλαδέψουμε.

(f) Όχι είναι αδύνατο. Από το (b) αν βρεθούμε σε κομβό chance πρεπει να παράξουμε όλα τα παιδιά του (η περίπτωση με μηδενίκη πιθανότητα παντού θεωρείται τετριμμένη και παραλείπεται). Στην περιπτώση του max οπώς έχω αναφέρει στο (a) οι τιμές δεν είναι άνω φραγμένες και ποτέ δεν ξέρει αν οι επομένες τιμές θα είναι μεγαλύτερες και έτσι δεν μπορούμε να κλαδέψουμε.

(g) Είναι δυνατό να κλαδέψουμε καποιά κλαδιά . Ακριβώς ανάλογο με το (c) μόνο που έδω το φράγμα, άρα και η τιμή που μετά κλαδέυουμε, είναι το 1 αντί για το 0.

(h) Είναι δυνατό να κλαδέψουμε καποιά κλαδιά . Ακριβώς ανάλογο με το (d) μόνο που έδω το φράγμα, άρα και η τιμή που μετά κλαδέυουμε, είναι το 1 αντί για το 0. Από το (b) αν βρεθούμε σε κομβό chance πρεπει να παράξουμε όλα τα παιδιά του (η περίπτωση με μηδενίκη πιθανότητα παντού θεωρείται τετριμμένη και παραλείπεται).