

# Einführung in die Künstliche Intelligenz

## SoSe 19



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

Prof. Dr. J. Fürnkranz, Prof. Dr. K. Kersting

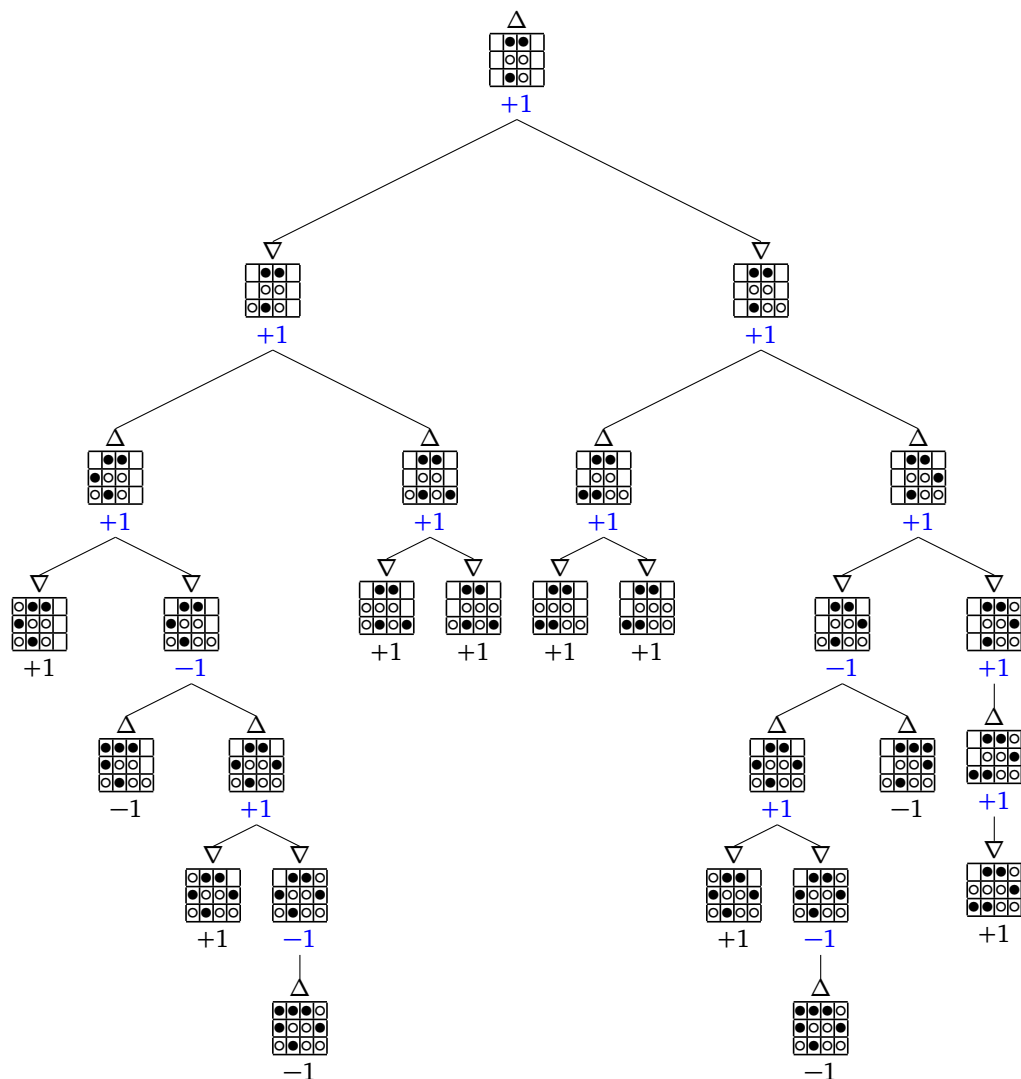
### Beispiellösung für das 3. Übungsblatt

#### Aufgabe 1 Minimax, Alpha-Beta Suche

a) siehe b)

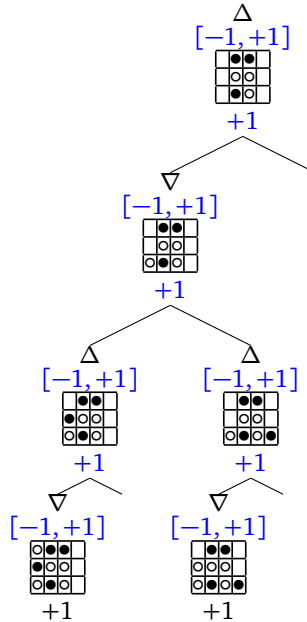
b) Der Minimax-Wert eines Zustandes ist unterhalb des entsprechenden Knoten dargestellt. Die blauen Werte entsprechen dabei Bewertungen, die im Laufe des Algorithmus ermittelt werden.

Minimax Algorithmus:

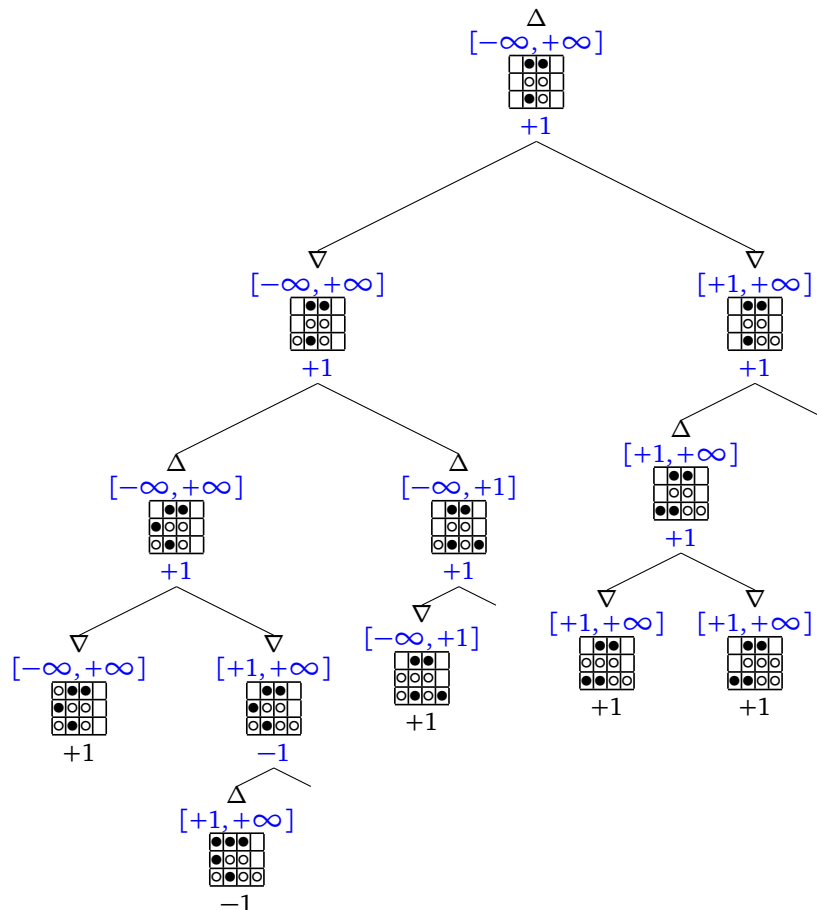


c) Für c) und d) stellt der Wert unterhalb eines Knotens nun den Rückgabewert der MaxValue bzw. MinValue Methode dar. Das heisst, entweder handelt es sich um den Minimax-Wert oder es wurde geprunt und stellt die letzte Bewertung dar. Oberhalb des Knotens sind die Schranken  $[\alpha, \beta]$  beschrieben, mit denen der Knoten besucht wird.

AlphaBeta-Algorithmus mit den Schranken  $[-1, +1]$  und Gewinne für Weiß bzw. für Schwarz wurden mit  $+1$  bzw.  $-1$  bewertet:



d) AlphaBeta-Algorithmus mit den Schranken  $[-\infty, +\infty]$  und Gewinne für Weiß bzw. für Schwarz wurden mit  $+1$  bzw.  $-1$  bewertet:



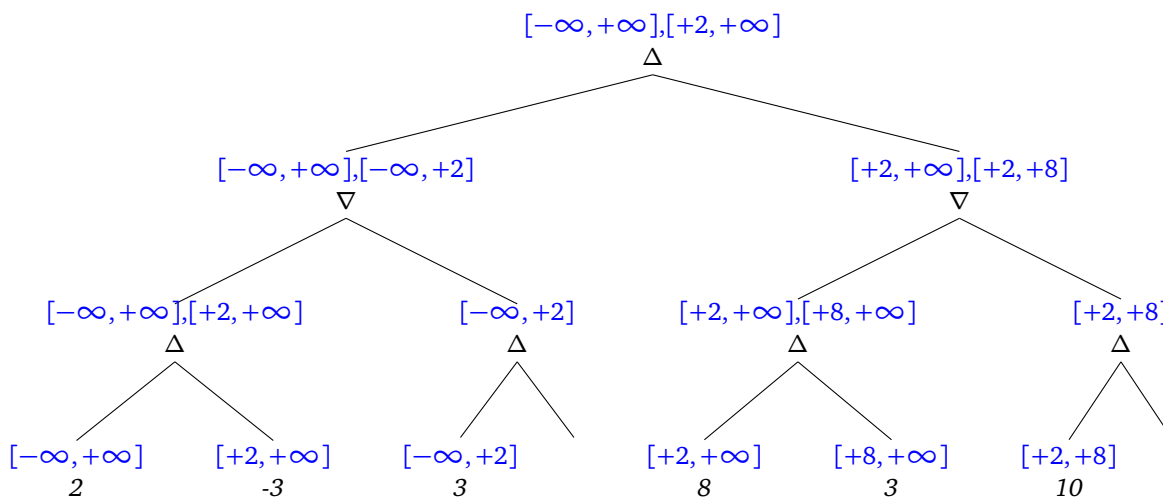
In Teilaufgabe c) konnten mehr Zustände geprunt werden, da im Gegensatz zu d) die Ausgangs-Schranken bezüglich der Bewertungsfunktion optimal waren. Es lässt sich keine größere untere Schranke und keine kleinere obere Schranke angeben. Im Allgemeinen führen schärfere Schranken zu einem gleichwertigen oder erhöhtem Pruning.

In diesem konkreten Fall wiederholte sich auf ähnliche Weise folgende Situation: nach der Evaluation des vierten Knotens, das von einem Max-Knoten aufgerufen wurde, war mithilfe der  $\beta$ -Schranke klar, dass es sich um den maximalen Wert handelt (hier war es sogar der maximal mögliche Wert, und nicht nur der maximale Wert, den der Min-Spieler „noch erlaubt“). Somit ist eine weitere Evaluierung der Nachbarknoten unnötig. In Teilaufgabe d) ist diese Information nicht bekannt, so dass die Nachbarknoten weiter nach potentiell besseren Bewertungen durchsucht werden.

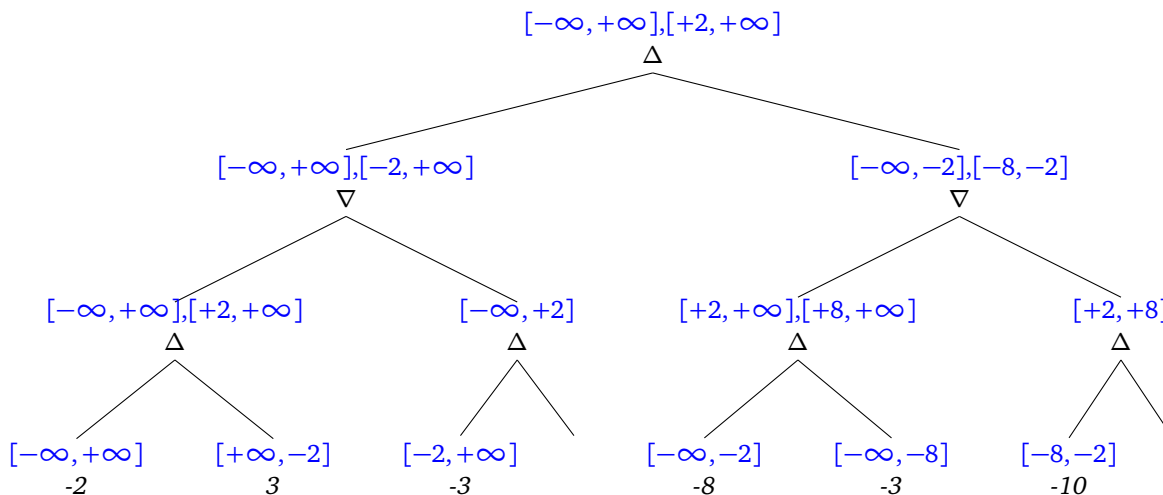
## Aufgabe 2 NegaMax Formulierung, Minimal Window

In den folgenden Bäumen sind zu jedem besuchten Knoten die Anfangs-Schranken und eventuelle Anpassungen (der lokalen Schranken) angegeben. Die Werte sind dabei von links nach rechts zu lesen.

a) Alpha-Beta Algorithmus:



b) Alpha-Beta Algorithmus in der NegaMax-Formulierung:



c) Alpha-Beta mit Minimal Window Search:

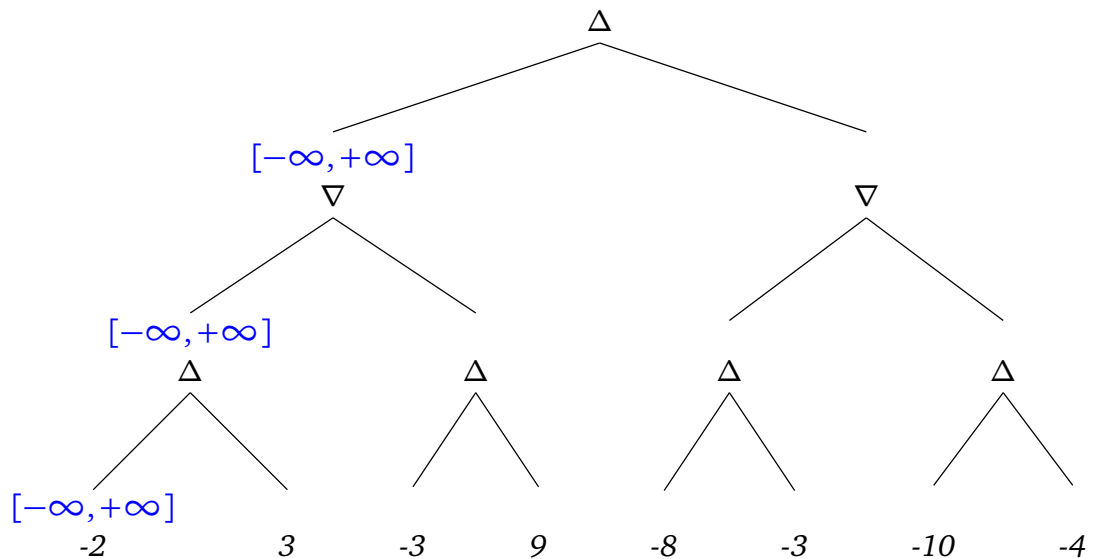
Siehe Aufgabe d), wobei alle Vorzeichenwechsel und Wechsel zwischen  $\alpha$  und  $\beta$ -Werten nicht durchgeführt werden müssen, dafür aber abwechselnd maximiert (ungerade Suchtiefen) und minimiert (gerade Suchtiefen) werden muß.

d) Alpha-Beta in der NegaMax-Formulierung mit Minimal Window Search:

Abarbeitung des Algorithmus NegaScout (Folie 75):

Die Blattwerte werden in der NegaMax-Formulierung aus der Sicht des Ziehenden dargestellt (im konventionellen Setting immer aus der Sicht des Max-Spielers).

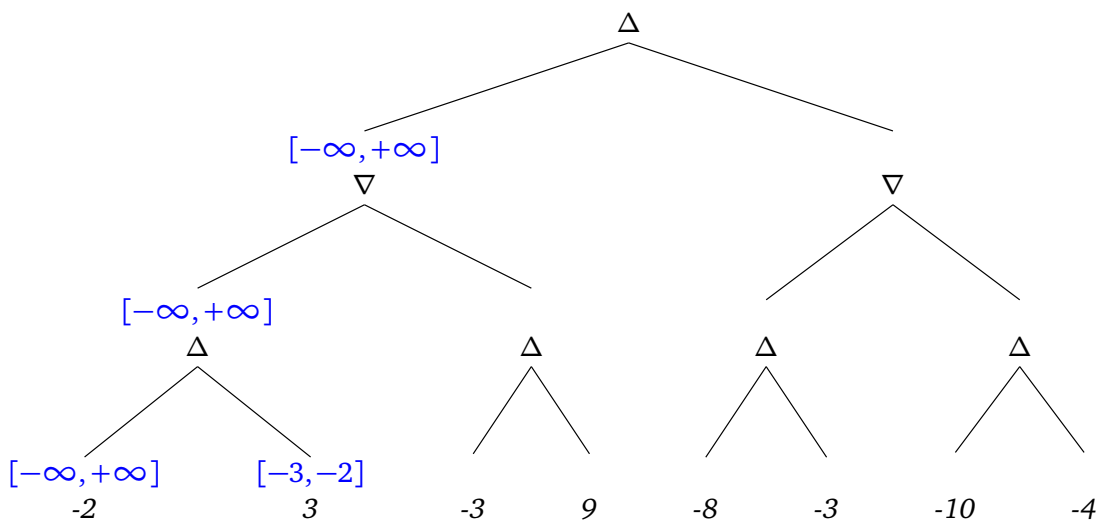
Aufruf mit  $[-\infty, +\infty]$ , die Werte werden bis zum Knoten  $-2$  links unten durchgereicht (wobei bei der jeweils nächsten Ebene  $\beta = -\alpha$  und  $\alpha = -\beta$  gesetzt wird, was sich aber nicht bemerkbar macht).



An dieser Stelle wird jetzt Wert  $-2$  an die aufrufende Instanz von NegaScout zurückgeliefert, und somit die Variable  $t$  auf  $-(-2)$  also auf  $+2$  gesetzt.

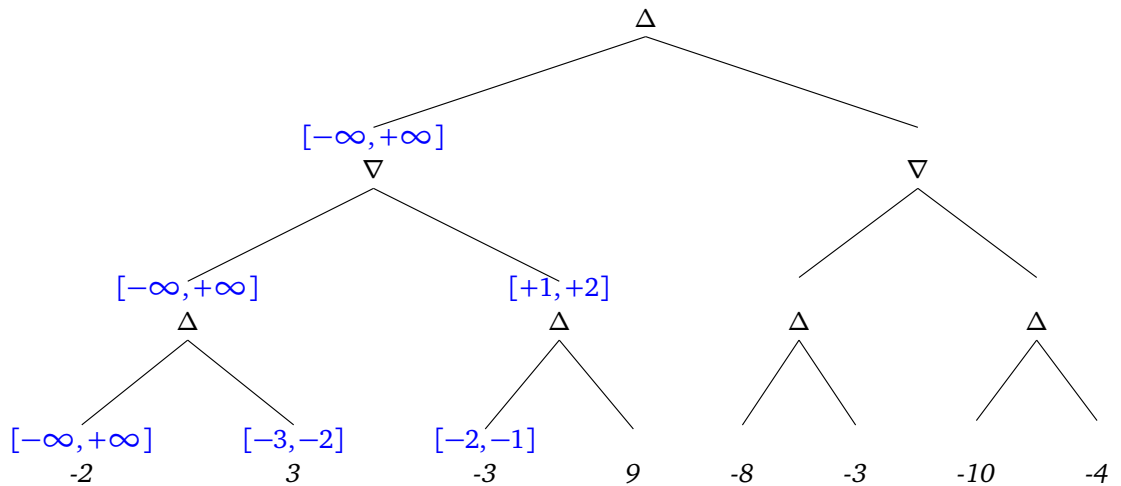
$t$  is nun größer als  $a$  (wurde mit dem Wert von  $\alpha$ , also mit  $-\infty$  initialisiert), und kleiner als  $\beta$ , wir sind aber noch beim ersten Zug, d.h.  $i = 1$ , daher passiert noch keine Re-Search, die untere Schranke  $a$  wird auf das Maximum des alten Wertes von  $a$  ( $-\infty$ ) und des Wertes von  $t$ , also auf  $2$  gesetzt.  $a$  is somit nach wie vor kleiner als  $\beta$  ( $\infty$ ) und der nächste Zug muss durchsucht werden.

Abweichend von der Aufgabe b) wird jetzt jedoch nicht das Intervall  $[a, \beta]$  zum Durchsuchen des nächsten Zugs verwendet, sondern das Intervall  $[a, a + 1]$ , also  $[2, 3]$ . Im rekursiven Aufruf von NegaScout wird das dann zu  $[-3, -2]$ .



Der Aufruf liefert als Ergebnis 3 zurück,  $t$  wird also auf den Wert  $-3$  gesetzt. Damit tritt kein Fail-High auf. Da es keine weiteren Züge mehr gibt, wird der Wert  $a = 2$  an den aufrufenden Min-Knoten zurückgemeldet.

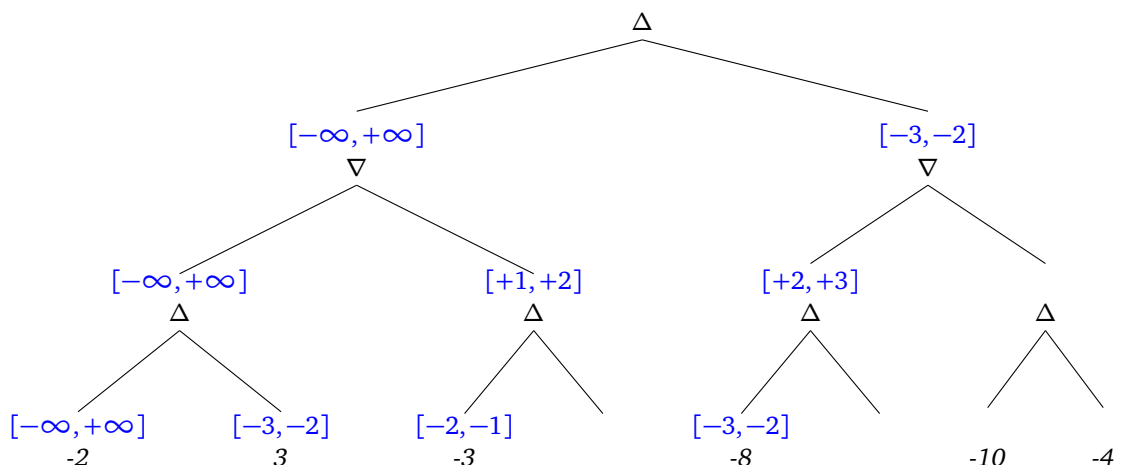
Dieser erhält also das Resultat  $t = -2$  als Ergebnis der Suche seines linken Teilbaums, und durchsucht nun den rechten Teilbaum (analog zu oben) mit dem Minimal Window  $[-2, -1]$ , was im rekursiven NegaScout-Aufruf zu  $[+1, +2]$  wird. Diese Werte werden (wiederum invertiert) an das Blatt weitergegeben.



Der Max-Knoten erhält also den Wert  $t = -(-3) = +3$ . Der Wert ist damit nicht kleiner als  $b (= 2)$  und somit tritt kein Fail-High auf. Allerdings wird  $a$  nun auf 3 gesetzt und es gilt, dass  $a \geq \beta (3 \geq 2)$ . Das bedeutet, dass dieser Zug "zu gut" ist, d.h., Min kann im Knoten darüber durch das Spielen des anderen Zuges dafür sorgen, dass Max nicht Gelegenheit erhält, den Zug mit 3 Punkten zu spielen. Somit werden keine weiteren Züge durchsucht, sondern sofort  $a = 3$  als Ergebnis an den Min-Knoten weitergeleitet (*cut-off*).

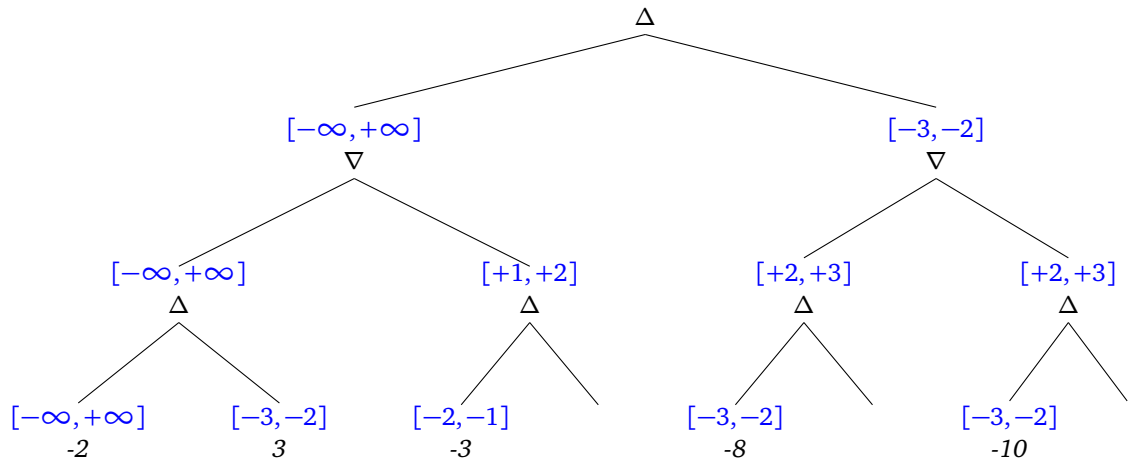
Der Min Knoten erhält also  $t = -3$  als Ergebnis, bildet das Maximum von  $a = -2$  (aus der Suche im linken Teilbaum) und  $t = -3$ .  $t$  ist hier kleiner als  $a$ , d.h. das Re-Search wird nicht ausgeführt und der Wert von  $a$  bleibt unverändert. Da es keine weiteren Züge mehr gibt, wird nun  $-2$  als Resultat an den Wurzelknoten weitergegeben, der also einen Wert von  $t = -(-2) = +2$  erhält.

Der Wurzelknoten setzt nun  $a$  von  $-\infty$  auf 2. Das ist immer noch kleiner als das  $\beta$  in diesem Knoten ( $\infty$ ), weshalb kein cut-off passieren kann. Alpha-Beta würde dann den rechten Zweig mit  $[+2, +\infty]$  durchsuchen, aber die Minimal Window Search setzt die obere Schranke  $b$  auf  $a + 1$  und ruft den nächsten Knoten mit  $[-3, -2]$  auf. Dieser Wert wird wieder bis an das Blatt durchgegeben.



Hier wird nun der Wert  $-8$  an den darüber liegenden Max-Knoten zurückgegeben, der erhält also den Wert  $t = -(-8) = +8$ , der über seinem  $\beta$ -Wert von  $+3$  liegt. Es passiert also ein cut-off, der bei einer normalen Alpha-Beta-Suche nicht passiert wäre, und der rechte Zweig wird nicht weiter durchsucht. Als Ergebnis wird  $+8$  nach oben durchgereicht, was dann als  $t = -8$  im Min-Knoten ankommt. Dieses  $t$  ist kleiner als das  $a$ , welches in diesem

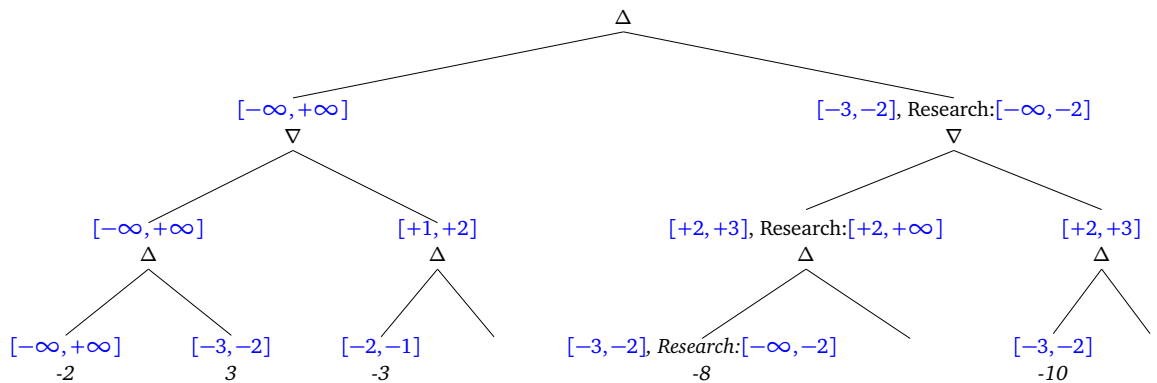
Knoten noch  $-3$  ist, der  $\alpha$ -Wert bleibt also unverändert, und der rechte Zweig wird wiederum mit  $[-3, -2]$  bzw.  $[+2, +3]$  durchsucht.



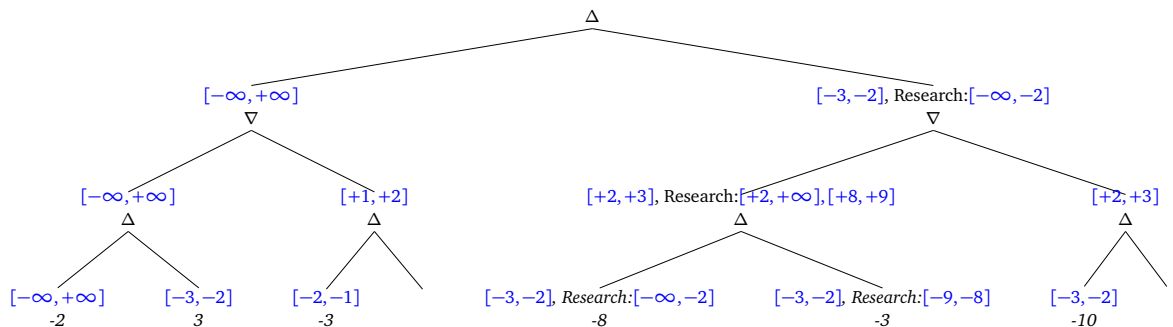
Der Max-Knoten erhält nun  $t = +10$  als negierten Wert aus dem Min-Knoten. Da  $t > \beta (10 > 3)$  ist, passiert ein cut-off wie im klassischen alpha-beta. Somit wird der letzte Arm nicht durchsucht und  $-10$  retourniert.

Der Min-Knoten bildet jetzt das Maximum von  $t = -10$  und dem  $\alpha$ -Wert  $a = -3$ . Der Wert von  $-3$  zeigt, an, dass Max im linken Teilbaum des Min-Knotens einen Wert von  $3$  oder besser hat (wir hatten ja oben schon  $+8$  gefunden, und dann einen Ast geprunt). Der Min-Spieler würde hier seinen rechten Zug wählen, wodurch sich das Prunen des Astes neben dem  $+8$ -Knoten nachträglich als falsch herausgestellt hat (Wird gleich repariert).

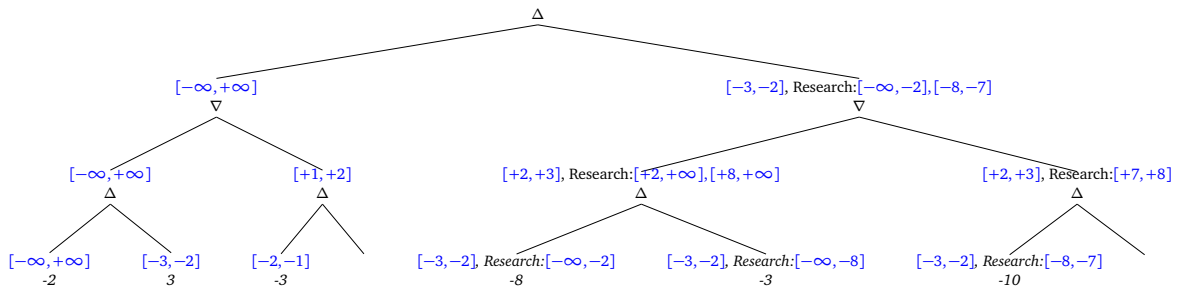
Der Min-Knoten retourniert nun  $-3$  an den aufrufenden Max-Knoten, der es als  $t = +3$  erhält. Es wurde also im rechten Teilbaum eine Variante gefunden, die einen Wert von mindestens  $3$  liefern würde. Damit ist die Minimal-Window Annahme verletzt ( $3 > 2$  und  $3 < \infty$ ) und der rechte Teilbaum muss noch einmal mit dem korrekten beta Wert durchsucht werden:



Nun trifft der fälschliche cut-off nicht mehr auf, und der rechte Teil des Max-Knotens wird durchsucht. Dort wird allerdings nur eine  $3$  gefunden und somit  $8$  retourniert.



Nachdem nun  $-8$  an den Min-Knoten zurückgeliefert wurde, wird der rechte Teilbaum auch erneuert durchsucht. Dort tritt wie beim klassischen alpha-beta ein cut-off auf und  $8$  zum Wurzelknoten zurückgeliefert.



### Aufgabe 3 Expectiminimax

a)

