הטכניון – מכון טכנולוגי לישראל

Introduction to Artificial Intelligence

תרגיל בית 2

מגישים :

שי יחזקאל - 205917883

דורין רואינסקי – 315691410

17.05.22

שי יחזקאל
דורין רואינסקי

תוכן עניינים

שי יחזקאל
דורין רואינסקי

חלק א'- סוכן חמדן משופר

1. שלושת הפרמטרים החדשים עליהם נסתמך הם: מיקום הנוסע, יעד הנסיעה, דלק במונית.
נניח שאנחנו מריצים את היוריסטיקה על מונית 0: $H(T_0)$:

$$H(T_0) = B - C + D$$

משתנים שישמשו אותנו להיוריסטיקה:

$$I = \left(gasT_0 > MD(p, D_1) + MD(T_0, p)\right)$$

$$P^* = \arg\max_{p \in \{p_1, p_2\}} \left((MD(p, D_1) - MD(T_0, p)\,) \cdot I\,\right)$$

$$K = is\ passenger\ on$$

$$G^* = \arg\min_{g \in \{g_1, g_2\}} \left(MD(T_0, g)\right)$$

הפרמטרים להיוריסטיקה:

$$B = MD(P^*, D_1) \cdot K$$

$$C = MD(T_0, X(P^*) \cdot K \cdot I + MD(T_0, P^*) \cdot (1 - K) \cdot I + \ MD(T_0, G^*) \cdot (1 - I) \cdot (1 - K)$$

$$D = Cash(T_0) - Cash(T_1)$$

2. המוטיבציה לשינוי היא שקלול הפרמטרים המרכזיים במשחק, ערך הכסף שאנו עתידים לקבל ביחס לכמות הדלק שנשארה.
בבחירת הנוסע אותו נעלה, נעריך את כמות הכסף המקסימאלית שנקבל עבור הסעת הנוסע וזאת אך ורק במידה ויש מספיק דלק להסיע אותו. במידה שלא, היוריסטיקה השולטת תהיה המרחק מתחנת הדלק.

הסוכן החמדן הנתון בהשוואה לסוכן המבוסס היוריסטיקה
נשים לב לרצף הפעולות של הסוכן,
$O \in \{north, south, west, east, park, refuel, dropoff, pickup\}$
מאחר והעלאת הנוסע והסעתו ליעד אינן בעלות ערך לסוכן החמדן הנתון ערכן ההיוריסטי זהה לביצוע $o \in O$ ועל כן, יבחר לבצע את הפעולות הראשונות ולא יעלה נוסע.
ולכן הסוכן החמדן שייצרנו עבור היוריסטיקה ינצח תמיד את הסוכן החמדן הנתון.

3. מימוש ההיוריסטיקה.
   א. כצפוי הסוכן החמדן המשופר מביס את החמדן הרגיל לאור שיפור היורסטיקה, כך שהסוכן החמדן המשופר מצליח לאסוף ולהוריד 4 נוסעים ולהשיג סה"כ 16 יח' כסף:

שי יחזקאל
דורין רואינסקי

```
[0, 16]
taxi 1 wins!
```

ב. כפי שציינו, הסוכן החמדן הרגיל אינו מבצע מהלכים בצורה מיודעות כלל, ובמקרה היחיד בו כן יבצע מהלך מיודע הוא כאשר הוא ירוויח כסף במיידית במהלך הבא, אך זה אפשרי רק אם העלאה נוסע ונמצא במשבצת היעד שלו, דבר שלא מתבצע במיודע ע"י הסוכן החמדן הרגיל. מכאן שפעולותיו אקראיות (ביצוע North/South)כמו הסוכן הרנדומאלי ונקבל תיקו:

```
[0, 0]
draw
```

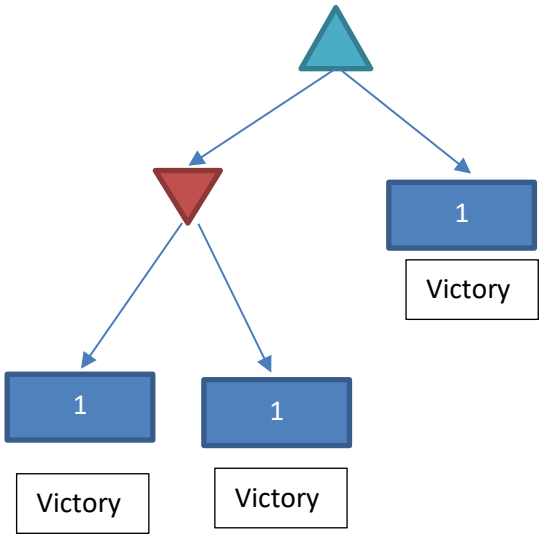
ג. כמו במשחק של חמדן נגד חמדן משופר, נצפה שהחמדן המשופר ינצח. גם כאן המשופר העלאה 4 נוסעים וצבר 16 נק':

```
[16, 0]
taxi 0 wins!
```

שי יחזקאל
דורין רואינסקי

חלק ב' – סוכן minimax

1. האלגוריתם תקין,
המקרה המתואר יתכן כאשר בעץ המשחק יש ענף ראשון שמוביל לנצחון לאחר פעולת היריב לעומת ענף שבו אנו נבצע את מהלך הניצחון, שהוא שני בסדר. ומאחר ושני הענפים בעלי אותו ערך, האלגוריתם יבחר בענף הראשון אשר יותר ארוך. זה נובע מאופן סריקה שהיא $post\ order$.



שינוי: נשנה את פונקצית התועלת (utility) כך שתבחר לתת ערך גבוהה יותר למסלולים קצרים יותר.

2. יתרון אלגוריתם אלפא בטא:
   א. עשוי לשפר את זמן הריצה, האלגוריתם מאפשר להימנע מחקירת תתי עצים שלא ישפיעו על ערך המינמקס של שורש תת העץ שלהם.
   ב. שומר על אופטימליות ה-minmax, שקול בתוצאה אותה הוא מחזיר לminmax

שינוי שאפשר לבצע באלגוריתם אלפא בטא הוא סידור בנים כך שבצמתי המקסימום ערך המינמקס הגבוהה ביותר נמצא בילד הראשון ובצמתי המינימום הערך הנמוך ביותר נמצא בילד הראשון. וכך הגיזום מתגלה בשלב מוקדם יותר בכל תת עץ.

יתרון אלגוריתם ההעמקה ההדרגתית:

א. העמקה הדרגתית מאפשרת לנו להתמודד עם מגבלות זמן שלא מתאפשר בהרצת מינימקס רגיל.
ב. באמצעות ההעמקה ההדרגתית נבטיח פתרון מוגבל בעומק, מאשר פתרון לא טוב שיכל להתקבל במינימקס רגיל מוגבל בזמן, כי ייתכן והמינימקס הרגיל היה ממצה את הזמן בענף פתרונות פחות טוב.

שיפור האלגוריתם יהיה באמצעות טבלאות מצבים, נשמור את ערכי צמתי הminmax ואת העומק של הערך ובמידה ונמצא מצב זהה באותו העומק נחסוך את פיתוח תת העץ שלו.

שי יחזקאל
דורין רואינסקי

חלק ג' – סוכן Alpha – Beta

1.
2. בהנחה שבסביבת הריצה היו K מוניות, כעת נהפוך את הפונקציה להיות המקסימאלית עבור כל סוכן בנפרד, שכן כעת בשונה משני השחקנים, האינטרסים אינם מנוגדים לחלוטין, ולכן לא ניתן להתייחס לפעולה של שחקן יריב בתור פעולת מינימום של השחקן הנוכחי.
ככל שנגדיל את K זמן הריצה יהיה ארוך יותר מאחר והאלגוריתם יצטרך לפתוח יותר צמתים.
אם נסתכל על עץ המשחק, כל סוכן נוסף מוסיף שכבת עומק **נוספת** ועל כן העומק יגדל.

$$O\left(B^{D\cdot\left(\frac{K}{2}\right)}\right)$$

במקרה של סביבת המונית הצעדים מוגדרים לכל מונית בנפרד ולכן משך הזמן יגדל כתלות בכמות הסוכנים, נציין כי קיימים משחקים בהם הגדלת כמות השחקנים הייתה מקטינה את משך המשחק, לדוגמא איקס עיגול.

נציין בנוסף כי אם עדיין נריץ את האלגוריתם עם זמן מוגבל אז במקרה הגרוע האלגוריתם ירוץ באותו זמן לכל מספר של סוכנים.

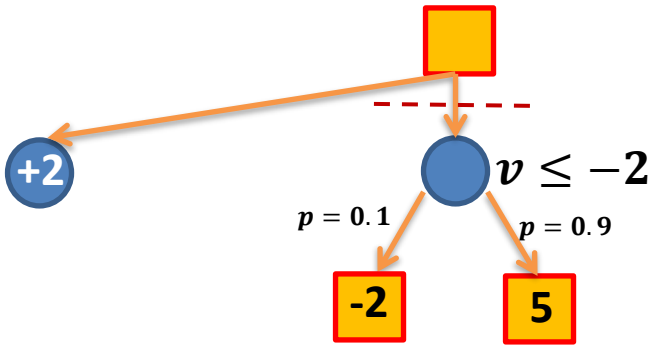
3. נתייחס לשני מקרים:
ריצה מוגבלת בזמן
במקרה זה, ותחת הנחה שהאלגוריתם לא מספיק להגיע אל סוף עץ המשחק, הרי ששני האלגוריתמים ירוצו באותו זמן, אך סביר להניח כי אלפא-בטא יספיק לראות מהלכים טובים יותר משהספיק לראות מינימקס שסורק ענפים שנגזמו ע"י אלפא בטא, ולכן בחירת המהלכים עשויה להיות שונה.
במקרה ששני האלגוריתמים רואים את סוף עץ המשחק, נוכל להתייחס למקרה זה כאל ריצה שאינה מוגבלת בזמן.

ריצה שאינה מוגבלת בזמן
במקרה זה אלפא-בטא עשוי לרוץ מהר יותר מאשר מינימקס לאור גיזום הענפים ופסילת נתיבי משחק שאינם משפיעים על ערכי ה-minimax. אך במקרה זה המהלך שיבחר אלפא-בטא יהיה זהה לזה שבחר מינימקס, שכן שניהם סורקים את כל עץ המשחק, וערכי המינימקס של השורש זהים בשניהם.

שי יחזקאל
דורין רואינסקי

חלק ד' – סוכן Expectimax

1. אכן מדובר בבאג,
במידה וידוע לנו כי ענף פתרונות מסוים מכיל ערך תוחלת, ייתכן ואלגוריתם אלפא בטא יעדיף לגזום ענף זה על פי ערך של אחד מבניו, אך אלגוריתם ה Expectimax יבחר בענף זה מכיוון שערך התוחלת בפועל תהיה יותר גבוהה מערך האלפא/בטא שגרם לגיזום.
דוגמא לאלגוריתם:



לפי מימוש גיזום אלפא בטא, הענף האמצעי ייגזם מכיוון שהערך של הבן הראשון שנקבל יהיה $0.1 \times (-2) = -\mathbf{0.2} < 2$, ועל כן האלגוריתם יחשוב שהערך שיבחר לצומת אמור להיות לכל היותר -0.2, למרות שאנחנו מבצעים פעולות תוחלת ובמקרה זה ערך הצומת עם הבן הימני הוא: $E = 0.1 \times (-2) + 0.9 \times 5 = 4.3$, ואז ערך השורש יהיה $\max\{2,4.3\} = 4.3$.

עם הגיזום, ערך השורש יהיה 2, וזה כמובן לא נכון.

שי יחזקאל
דורין רואינסקי

שאלה פתוחה

א. נוכל להגדיר את פונקציית התועלת בתור הפרש הכסף שנצבר ע"י נהג ספציפי ממוצע הכסף שנצבר ע"י נהגים אחרים. כלומר:

$$U(T_i) = Cash(T_i) - \frac{1}{(n-1)}\sum_{j\neq i}^{n} Cash(T_j)$$

ואז מתקיים סכום אפס בין כל הנהגים:

$$\sum_{i}^{n} U(T_i) = Cash(T_1) - \frac{1}{(n-1)}\sum_{j\neq 1}^{n} Cash(T_j) + \cdots + Cash(T_n) - \frac{1}{(n-1)}\sum_{j\neq n}^{n} Cash(T_j) = 0$$

ב. הפונקציה מונוטונית עולה, ועל כן בחירת מינימום ומקסימום אינה משתנה. כלומר:

$$\log(\max(a,b)) = \max\left(\log(a), \log(b)\right)$$

$$\log(\min(a,b)) = \min\left(\log(a), \log(b)\right)$$

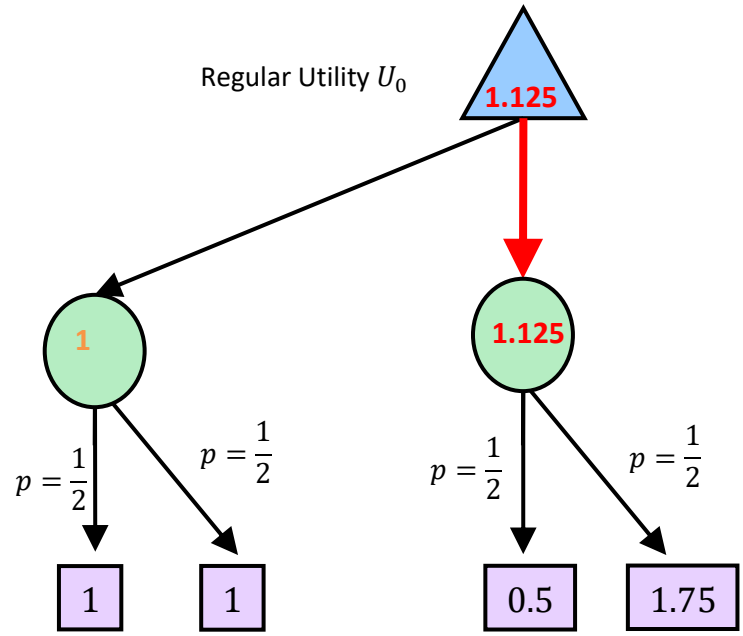
ולכן ערכי הצמתים אמנם ישתנו, אך בחירתם לא תשנה, ונבחר אותם מהלכים, ולכן, לפונקצית התועלת כל עוד חיובית לא תהיה השפעה על מהלכי המשחק.

עם זאת, עבור ערכים שלילים אינה מוגדרת, ולכן גם פונקציית התועלת לא תהיה מוגדרת, ואז במקרה זה כתלות במימוש – ייתכן והאלגוריתם יקרוס, או שמע ידלג על ענפים בהם הארגומנטים של ה- $\log$ שליליים, ואז לא ייבחרו בהכרח אותם מהלכים.

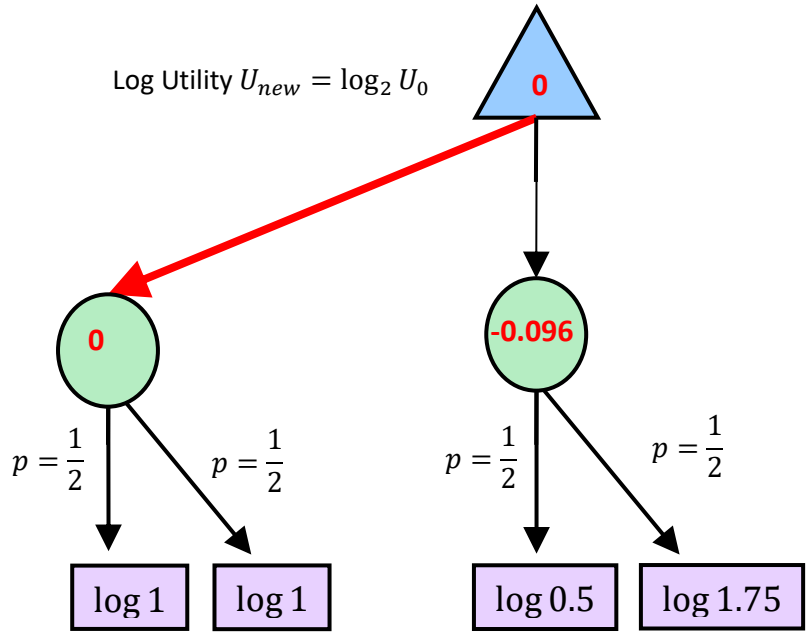
ג. בדומה להסבר בסעיף קודם, פונקציה זו מונוטונית עולה עבור כל ערכי x חיוביים ושליליים. ולכן, לפונקצית התועלת לא תהיה השפעה על מהלכי המשחק, שכן יבחרו אותם צמתים.

ד. תשובתנו משתנה כך שבשני המקרים, גם פונקציית הלוג וגם במעלה השביעית, כעת ייתכן ונקבל בחירות שונות בעץ המשחקים, וזה נובע מהעובדה שכעת בExpectimax אנו לוקחים ערכים תוחלת, וזה דורש פעולת חיבור.
אך פעולת הלוג והמעלה השביעית אינן לינאריות, כלומר $\log(a+b) \neq \log a + \log b$.
להלן דוגמאות שבהן הבחירות בעץ המשחק משתנות לאור השינויים בסעיף ב' וג' עם מימוש הExpectimax:

כך למשל דוגמה שבה באופן פעולת פונקציית תועלת רגילה נבחר בענף הימני:

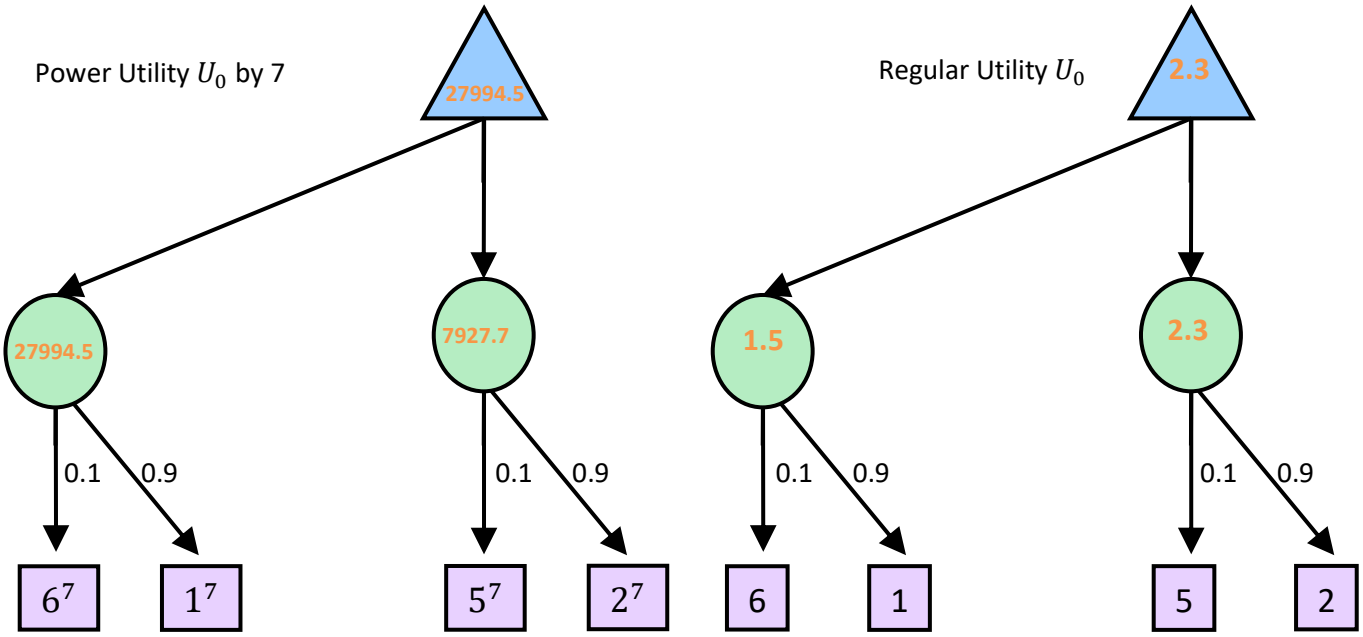


ולאחר פונקציית התועלת עם הלוג, אנו בוחרים בענף השמאלי:



שי יחזקאל
דורין רואינסקי

הדוגמה המוצגת ממחישה זאת,



שי יחזקאל
דורין רואינסקי

ה. 1. **אלגוריתם אלפא בטא ללא הגבלת משאבים**

במקרה זה שידוע שנגיע לעלים, השימוש בפונקציה f מקטין את מספר הצמתים שיפותחו על ידי האלגוריתם מבלי לפגוע בנכונות שלו, מכיוון שנוכל להפעיל את f ולקבל חסם עליון/תחתון על ערך הMM של העץ, וכך נוכל לדעת האם ניתן כבר לגזום אותו. הנכונות נובעת מהעובדה שחסם זה הוא על ערך הMM על העץ כולו, ומכיוון שאנחנו רצים ללא הגבלת משאבים, ערכי הMM יהיו אלה שf נותן עליהם חסם.

כלומר ערך f נותן לנו אינדיקציה נוספת על חסמים שיכולים להיות על ערכי הMM, ונוכל לעדכן כך את α, β בהתאם.

כלומר בכל צומת נוכל לשלוף את הערך $f(s)$.

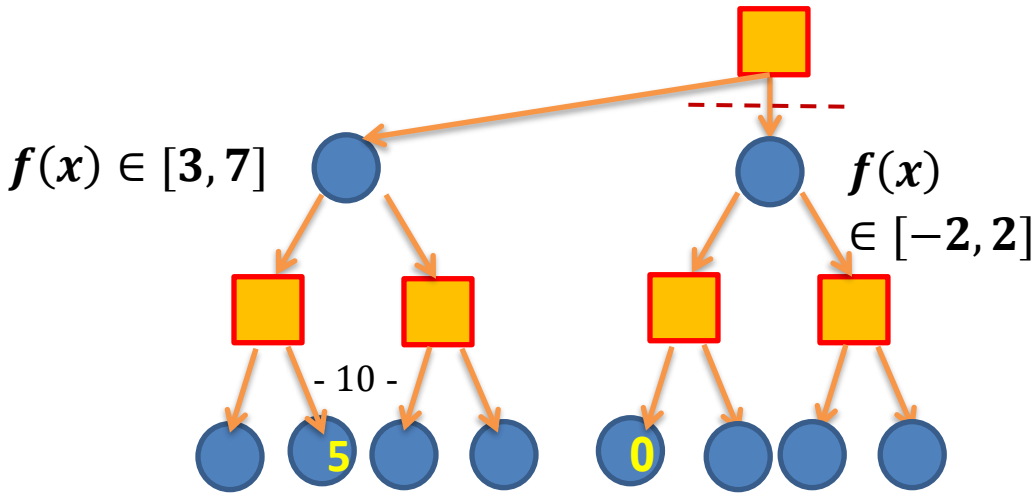
אם אנחנו בצומת של **מקסימום**, נוכל לעדכן את חסם β להיות **המקסימלי** מבין β ו-$f(s)+2$.

אם אנחנו בצומת של **מינימום**, נוכל לעדכן את חסם α להיות **המינימלי** מבין α ו-$f(s)-2$.

השינוי באלגוריתם נראה כך (pseudo-code נלקח מהתרגול):

```
def AlphaBeta(State, Agent,Alpha,Beta)
    if G(State) return U(State,Agent)
    turn = turn(State)
    Children = Succ(State)
    f_val = f(s)
    if Turn = Agent then
        CurMax = -inf
        loop for c in Children
            v =AlphaBeta(c,Agent,Alpha,Beta)
            CurMax = max(v,CurMax)
            Alpha = max(CurMax,Alpha)
            Beta = min(Beta,f_val+2)
            if CurMax >= Beta then return inf
        return CurMax
    else
        CurMin = inf
        loop
        for c in Children
            v = AlphaBeta(c, Agent, Alpha, Beta)
            CurMin = min(v, CurMin)
            Beta = min(CurMin, Beta)
            Alpha = max(Alpha, f_val - 2)
            if CurMin <= Alpha then return -inf
        return CurMin
```

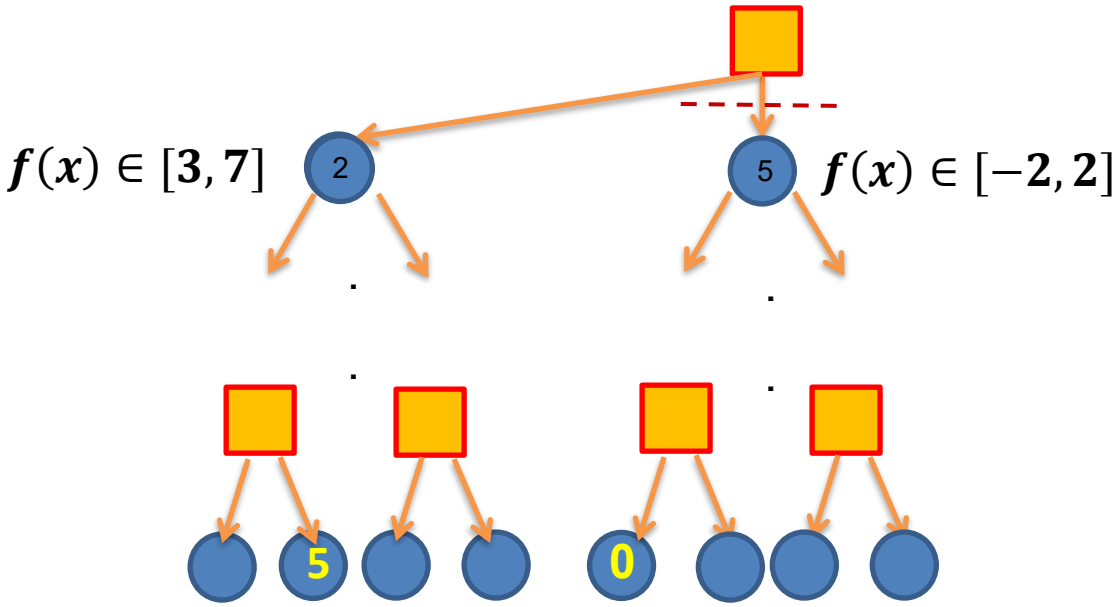
דוגמה לשימוש:



בדוגמה הנ"ל, נניח והבן השמאלי הראשון מקבל את הערך 5 (ולכן ערך f עליו בין 3 ל7), וכאשר נגיע אל הבן האמצעי, נוכל להפעיל עליו את f ולקבל חסם על הערך *MM* של הענף, במקרה זה הערך יהיה לכל היותר 2, ולכן כבר נוכל לגזום את הענף.

2. עבור אלגוריתם אלפא בטא מוגבל משאבים

תמי אינה יכולה להשתמש באלגוריתם להקטין את מספר הצמתים שיפותחו.
במקרה זה כאשר ישנה הגבלת משאבים לא בטוח שנגיע לעלים, כעת האלגוריתם נעצר בעומקים שונים ולכן יתכן שהפתרון אינו האופטימלי עבור העומק הנתון שבו האלגוריתם עוצר (או בעומק אחד לפני העומק בו הוא עוצר באמצע איטרציה)
דוגמה לשימוש:



נזכר כי הערך f נותן לנו חסם על ערך המינימקס של עץ המשחק **הכולל**, וכאשר אנו נריץ עם הגבלת משאבים (הגבלת עומק) אז נקבל שf נותן חסם על בסיס העץ המלא, ולא על הערכים שנכונים עד לאותו עומק אותו אנו רוצים לגלות במשאבים הנתונים.