מבוא לבינה מלאכותית

תרגיל בית 2

מגישים:
עדן לוי, ת.ז 208932335
עומר גרייף, ת.ז 208200154

חלק א'

1. נגדיר פורמלית את המשחק הנתון לנו ע"פ הנתונים שקיבלנו מהסביבה:

S = קבוצת כל המצבים האפשריים ללוח 5 על 5, לפריסת 2 עמדות טעינה, 2 סוכנים, 2 חבילות (על המשבצת או מוחזקות על ידי אחד הסוכנים), 2 נקודות הורדה\שילוח, מספר הצעדים שנותרו לסוכנים ומצב הטעינה שלהם (מספרים שלמים).
O = קבוצה הצעדים האפשריים: ימינה, שמאלה, מעלה ומטה, להרים חבילה, להניח חבילה ולהטעין. נשים לב שלא כל הצעדים אפשריים מכל משבצת, אלא כתלות במשבצת עצמה (לדוגמה משבצת של עמדת טעינה).
I = מצב השייך לקבוצה S, כאשר הסוכנים טעונים במלואם, לא מחזיקים בחבילה ועם מספר הצעדים המוגדר באתחול (דגל -C).
G = מצב השייך ל-S כך שמספר הצעדים של שני הסוכנים הוא 0 או כאשר הסוללה של אחד הסוכנים נגמרה.

2. נגדיר יוריסטיקה משלנו להערכת מצבי המשחק:

```
def smart_heuristic(env: WarehouseEnv, taxi_id: int):
    robo = env.get_robot(taxi_id)

    agent_to_package = min(manhattan_distance(robo.position, env.packages[0].position),
                           manhattan_distance(robo.position, env.packages[1].position))

    heuristic = robo.credit * 30

    if robo.package:
        agent_to_destination = manhattan_distance(robo.position, robo.package.destination)
        heuristic -= (agent_to_destination - 10)  # go to delivery destination / pick up package
    else:
        heuristic -= agent_to_package  # go to package position
    return heuristic
```

מילולית, במידה והסוכן לא מחזיק כרגע חבילה, נחזיר יוריסטיקה לפי המרחק מהחבילה הקרובה ביותר, כך נוודא שהוא מתקדם לעברה. במידה והוא כן מחזיק חבילה, באופן דומה נחזיר יוריסטיקה שמקרבת אותו אל היעד של החבילה. אם הוא נמצא בעמדת השילוח (ביעד), נתעדף פי 30 כפול כמות הניקוד המתקבלת משילוח החבילה, את פעולה זו כמתואר בפונקציה לעיל.

3. רטוב

4. החיסרון העיקרי הוא שהסוכן ה-greedy מתעלם מקיומו של הסוכן הנוסף בעולם שיכול להרעות את מצבו או להגיע לפניו למשאב ששניהם נלחמים עליו. בניגוד, minimax פועל על פי הפעולות של היריב ומתעדף תגובה מיטבית אליהן.

חלק ב'

1. ככל שהיוריסטיקה יותר קלה לחישוב, נוכל להיכנס עמוק יותר בעץ ה-minimax, כלומר להתחשב ביותר צעדים קדימה של השחקן וכך לבצע פעולה שמתבסס על עתיד רחוק יותר. מנגד, החיסרון בכך הוא שהיוריסטיקה פחות מיודעת ולכן ייתכן שנתבסס על מידע לא מדויק\נגיע לתוצאה לא מיטבית.
2. המצב ייתכן מכיוון שבאלגוריתם minimax אנחנו מסתכלים מספר שלבים קדימה במשחק ונותנים ציון לכל בחירה של השחקן לפי האופציה שתהיה הכי "גרועה" עבור היריב וחוזר חלילה. לכן ייתכן שהמצב שמוביל לניצחון מידי עבור השחקן עשוי להביא את היריב לקבל יותר נקודות גם אם לא ניצחון ביחס לבחירה במסלול אחר בו ייקח לשחקן יותר מצעד אחד לניצחון, אך היריב יסיים בתוצאה גרועה יותר. כלומר אלגוריתם minimax לא מתעדף עומק קצר לניצחון אלא הגעה מוחלטת/בטוחה יותר לניצחון.
3. רטוב
4. א. נתון כי כל סוכן רוצה לנצח ולא אכפת לו רק מאיתנו לכן כל סוכן הוא למעשה סוכן מקסימום, כלומר בוחר ביוריסטיקה הטובה ביותר עבורו. לכן נשנה את אלגוריתם מינימקס כך שבתור של כל סוכן אחר מאיתנו (K-1 סוכנים), הסוכן בוחר בצעד הכי טוב עבורו לפי היוריסטיקה שלו (מבצע MAX) וכך נבנה את עץ המינימקס תחת ההנחה הנתונה.
ב. נתון כי כל סוכן כעת רוצה רק שלא ננצח, כלומר הם סוכני מינימום. לכן בבניית עץ המינימקס נבצע K-1 הרצות של מינימום על היוריסטיקה של הסוכן שלנו ולאחרה נבצע MAX לאפשרויות שנותרו לסוכן שלנו באופן דומה לאלגוריתם מינימקס הקלאסי רק שכעת מדובר ב-K-1 סוכנים.
ג. נתון כי כל סוכן כעת רוצה שהסוכן שאחריו בתור ינצח. לכן כל סוכן יהיה סוכן מקסימום ליוריסטיקה של הסוכן הבא אחריו. הסוכן שלנו יבצע MAX ליוריסטיקה שלנו וכך יהיה לו יתרון על שאר הסוכנים (בהנחה שהמטרה שלנו היא לנצח עדיין).

חלק ג'

1. רטוב
2. הסוכן שמימשנו בחלק זה יתנהג באופן זהה לסוכן שמימשנו בחלק ב' מבחינת בחירת מהלכים. השוני בינו לבין הסוכן מחלק ב' (מינימקס) הוא שכעת הוא גוזם ענפים שמאליו לא ישפיעו על התוצאה הסופית ולכן ייתכן שמבצע את החישוב שלו מהר יותר (במקרים בהם גיזום מתבצע בפועל). עם זאת התוצאה הסופית תהייה זהה מעצם הגדרת אלפא בטא וגיזום להיות כזה שלא משפיע על בחירת היוריסטיקה בכל שלב ולכן הסוכן יתנהג באופן זהה.

חלק ד'

1. נשתמש בהסתברות בדידה יוניפורמית. כלומר ערך זהה לכל אפשרות של היריב. זאת מכיוון ששחקן שפועל באופן אקראי לחלוטין יכול לבצע כל צעד בהסתברות זהה.
2. נתון כי היורסטיקה חסומה בין 1 ל-מינוס 1, לכן ניתן לחשב חסם עבור אותו הערך כאשר מניחים ערך יוריסטי גרוע ביותר עבור כל בן. כלומר לכל צומת בן יש ערך מוגדר גרוע ביותר אותו ניתן לקבל. אם אותו חסם קטן מצומת MAX שמעל (אב). כלומר נחשב כבר עבור הבנים שקיבלנו את התוחלת ונחשב גם את המקרה הטוב ביותר שיכול להיות (שהוא 1 מכיוון שהירוסיטקה חסומה) ונבדוק האם בכלל ייתכן שמשתלם לנו לפתח את הצומת הזו כיוון שגם במקרה הטוב ביותר ייתכן ולאבא כבר יש צומת עדיפה.
3. רטוב

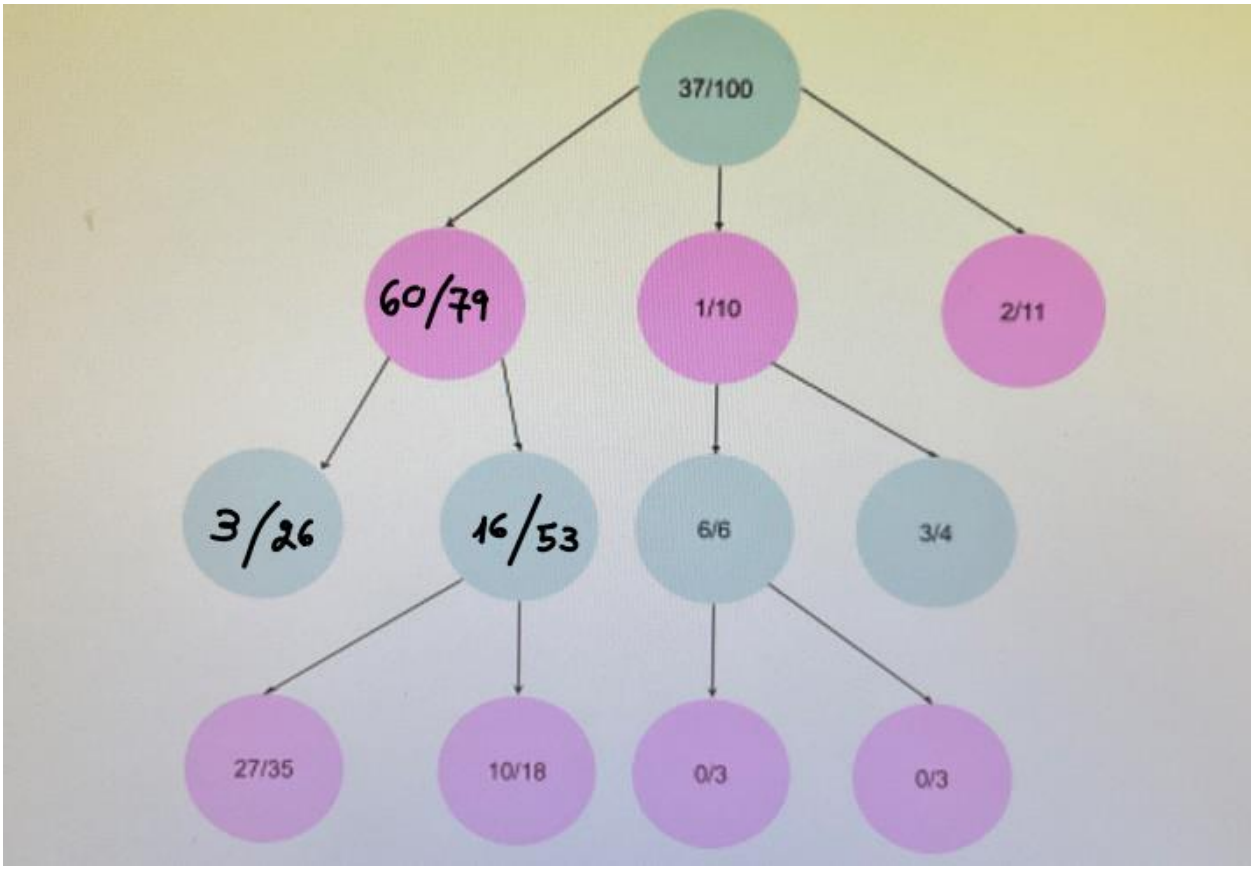
חלק ה'

1. א. מקדם הסיעוף הוא מספר הצעדים האפשריים המקסימלי מכל המשבצות בלוח. לכן, גם אם נגדיל את הלוח לכל גודל n*n שהוא, מקדם הסיעוף יישאר זהה כי הסוכן שלנו עדיין יכול לנוע רק מעלה, מטה, שמאלה, ימינה, להטעין, לאסוף חבילה ולשלח אותה. בנוסף, גם אם נוסיף מחסומים לכל היותר נוריד את מקדם הסיעוף עבור משבצות צמודות למחסום, אך בהנחה שעדיין קיימת לפחות משבצת אחת בלוח שלא קיים לידה מחסום, ומכיוון שתמיד מסתכלים על הגרוע ביותר בחישוב מקדם סיעוף, הוא יישאר זהה. מקדם הסיעוף של המשחק שלנו הוא 7 לכל הצעדים האפשריים או 5 לצעדים חוקיים (כי לא ניתן לאסוף או לשלח חבילה באותה משבצת למשל, כנ"ל עם הטענה).
ב. כעת התווספה היכולת של הרובוט לבחור משבצת ריקה כלשהיא בלוח כולו ולהניח עליה בלוק. מכיוון שרוצים לחשב את מקדם הסיעוף, נתייחס למקרה הגרוע ביותר, כלומר למקרה בו מספר המשבצות הריקות גדול ביותר. זה קורה כאשר שני הסוכנים מחזיקים את החבילות וכך מפנים את המשבצות עליהן הן עמדו ובנוסף הם שניהם עומדים על עמדת טעינה או עמדת שילוח. יש לנו 25 משבצות בלוח בסך הכל (5 על 5), מתוכן 2 משבצות הטענה ו2 משבצות שילוח, לכן סה"כ 21 משבצות ריקות במקרה הגרוע ביותר לכן מקדם הסיעוף שלנו מהסעיף הקודם מקבל תוספת של 21, כלומר מקדם הסיעוף החדש הינו 21 ועוד 4 לתנועה לכל הכיוונים ועוד 1 לביצוע פעולה נוספת במקרה הגרוע (למשל שילוח חבילה). **סה"כ מקדם סיעוף 26 לצעדים חוקיים**. אם נתייחס לכל הצעדים האפשריים כלל נקבל מקדם סיעוף של 7+25 = 32 (25 משבצות בהתעלמות מהנחה על משבצת לא חוקית).

2. א. אחד מהאלגוריתמים מהסעיפים הקודמים בו נוכל להשתמש עבור השינוי מסעיף 1ב', כך שזמן הריצה שלו לא גדול מהותית מהזמן שלוקח לו לרוץ ללא השינוי הוא Improved Greedy. זאת מכיוון שאלגוריתם Improved Greedy לא מסתכל קדימה מעבר לצעד הנוכחי ורץ פעם אחת על כל האפשרויות ובוחר בזאת עם היוריסטיקה הטובה ביותר, כלומר עדיין נחזיר את הצעד הבא בזמן סביר כי אנחנו לא נדרשים לפתח את הצעדים כמקדם הסיעוף אקספוננציאלית.
ב. ניתן להשתמש באלגוריתם Monte Carlo (MCTS) כפי שראינו בהרצאה על מנת להתגבר על מקדם הסיעוף הגדול ויחד עם זאת כן להתחשב גם בעומק צעדים. בחרנו בו מכיוון שהוא מאפשר על ידי הרצת סימולציות קבלת תמונה יחסית טובה של אילו רצפי מהלכים מיטיבים עם הסוכן ואילו פחות בשילוב ההסתברות שיקרו. על ידי כך ניתן לקבל החלטות מיודעות באופן יחסי ולפעול גם בתנאים של מקדם סיעוף גדול בזמן מהיר יחסית תוך התחשבות בעומק המצבים.

חלק ו'

1.



2. הצומת הבא שייבחר בשלב ה-selection הוא הצומת שערכו 60/79, זאת מכיוון שערך ה-UCB1 שלו הוא הגבוה ביותר מהאופציות האחרות. נחשב:

$$UCB\left(\frac{60}{79}\right) = \frac{60}{79} + \sqrt{2} * \sqrt{\frac{\ln 100}{79}} = 1.1$$

$$UCB\left(\frac{1}{10}\right) = \frac{1}{10} + \sqrt{2} * \sqrt{\frac{\ln 100}{10}} = 1.06$$

$$UCB\left(\frac{2}{11}\right) = \frac{2}{11} + \sqrt{2} * \sqrt{\frac{\ln 100}{11}} = 1.09$$

לכן מבין אלה נבחר בצומת השמאלית שערכה 60/79 כאמור. כעת נחשב את UCB1 עבור הצאצאים של צומת זו:

$$UCB\left(\frac{3}{26}\right) = \frac{3}{26} + \sqrt{2} * \sqrt{\frac{\ln 79}{26}} = 0.69$$

$$UCB\left(\frac{16}{53}\right) = \frac{16}{53} + \sqrt{2} * \sqrt{\frac{\ln 79}{53}} = 0.71$$

לכן מבין אלה נבחר בצומת הימנית שערכה 16/53 כאמור. כעת נחשב את UCB1 עבור הצאצאים של צומת זו:

$$UCB\left(\frac{27}{35}\right) = \frac{27}{35} + \sqrt{2} * \sqrt{\frac{\ln 53}{35}} = 1.24$$

$$UCB\left(\frac{10}{18}\right) = \frac{10}{18} + \sqrt{2} * \sqrt{\frac{\ln 53}{18}} = 1.21$$

לכן מבין אלה נבחר בצומת השמאלית שערכה 27/35. לכן הצומת ש-selection יבחר תהייה זו.

3. מסעיפים קודמים ראינו כי ערך ה-UCB של הצאצא הקרוב ביותר לערכו של ערך ה-UCB של הצאצא 60/79 הינו עבור הצומת הימנית – 2/11. לכן נחשב עבורו את כמות הנצחונות K הדרוש כדי שערך ה-UCB שלו יהיה גדול יותר וכך למעשה נוודא שהוא ייבחר בתהליך ה-selection, נחשב:

$$UCB\left(\frac{60}{79}\right) < UCB\left(\frac{2}{11}\right)$$

$$\frac{60}{79+k} + \sqrt{2} * \sqrt{\frac{\ln(100+k)}{79+k}} < \frac{2}{11} + \sqrt{2} * \sqrt{\frac{\ln\,(100+k)}{11}}$$

$$k > 0.3303$$

כלומר עבור ניצחון אחד (K=1), נקבל כי צאצא של אב קדום ביותר אחר יבחר בשלב ה-selection.

4. כדי לתעדף exploration על exploitation נרצה שחלק זה בנוסחה יהיה גדול יותר וכך ייתן ערכי UCB1 גדולים יותר עבור צמתים בהם נבצע exploration. נתון כי יש לנו גישה רק ל-N(s) לכן נרצה להגדיל הערך המתקבל בשורש ביחס לחישוב הרגיל. לכן נגדיר את N כך:

$$N'(s) = N(s) * 100$$

$$\frac{U(s)}{N(s)} = exploitation \rightarrow exploitation' = \frac{exploitation}{100}$$

$$exploration = \sqrt{\frac{\ln\left(N(s.parent)\right)}{N(s)}} \rightarrow exploration'$$

$$= \sqrt{\frac{\ln\left(N(s.parent * 100)\right)}{N(s) * 100}}$$

$$= \frac{\sqrt{\frac{\ln\left(N(s.parent)\right) + \ln(100)}{N(s)}}}{10} \geq \frac{\sqrt{\frac{\ln\left(N(s.parent)\right)}{N(s)}}}{10}$$

$$= \frac{exploration}{10} \rightarrow \ exploration' \geq \frac{exploration}{10}$$

לכן, הקטנו את ה-exploitation פי 100 ואת ה- exploration לכל היותר פי 10.
מכאן שביחס ל-exploitation, כעת ה-exploration מתועדף יותר מאשר לפני השינוי.

$$\mathrm{UCB1(s)} = \underbrace{\frac{U(s)}{N(s)}}_{\text{explotation}} + C \times \underbrace{\sqrt{\frac{\ln(N(s.\mathrm{parent}))}{N(s)}}}_{\text{exploration}}$$