**מבוא לבינה מלאכותית**

**סמסטר חורף תשפ"ב**

**מטלה 3 - Domain-Independent Planning**תאריך הגשה: 5.12.21 23:55

# הנחיות

* שאלות בנושא מטלה זו יש לשאול דרך המודל, בפורום "מטלה 3".
* הוראות להגשת המטלה מופיעים בסוף מסמך זה.
* הקבצים הנדרשים להרצת הקוד הינם:
  + one\_passenger.py
  + multiple\_passengers.py
  + main.py
  + palnner.py
  + pyddl.py
  + init\_\_.py\_\_
* העבודה להגשה ביחידים בלבד אלא אם כן המגישים קיבלו אישור להגשה שאינה ביחידים.
* לפני שניגשים לממש את המטלה מומלץ לעיין רבות בהסברים וכן בקוד הקיים.
* פתרון המטלה שתגישו ייבדק מול שאר ההגשות על ידי תוכנת העתקות.
* **מי שימצא כי העתיק יכשל בקורס וכן יועבר לוועדת משמעת אוניברסיטאית**.
* הפרויקט נכתב וייבדק בשפת התכנות python, גרסה 3.6 או 3.7.
* מסמך זה בנוי באופן הבא: תיאור המטלה, בעיית תכנון, מרחב הבעיה, שאלות המטלה, הנחיות לביצוע השאלות, והסבר על הגשת המטלה.

# תיאור המטלה

# חברת המוניות "אוטומה" החליטה שהיא מעוניינת לרכוש מכוניות אוטונומיות. מכיוון שהחברה מעוניינת לחסוך בכסף, החברה רכשה מוניות שאינן מסוגלות לנווט בעצמן, אך מאפשרות לכתוב תכנית שמבצעת זאת. במטלה זו יש עליכם לעזור לחברת "אוטומה" ליצור תכנית לאיסוף והורדת נוסעים. בהינתן מצב התחלתי של מונית, עלינו למצוא תוכנית עבור המונית האוטונומית כך שהמונית תסיע נוסעים ממקום למקום ותשתמש בכללי פריקה וטעינה. לשם כך עליכם להיעזר בשפת STRIPS להגדרת אופרטורים לבעיה.

# בעיית תכנון

# בהינתן מיקום המונית (רכב אוטונומי) ומיקומים התחלתיים ויעדם של מספר נוסעים על גבי מפה, עלינו לתכנן תוכנית כך שהמונית תוכל להסיע כל אחד מהנוסעים, מהמיקום ההתחלתי אל מיקום היעד.

# מרחב הבעיה

המפה של האזור מוגדרת על גבי מטריצה בגודל . המונית האוטונומית שלנו נמצאת באחד התאים במטריצה והנוסעים נמצאים גם כן בתאים כלשהם במטריצה. בנוסף, לכל נוסע מוגדר תא במטריצה שאליו הוא רוצה להגיע בעזרת המונית שלנו. נגדיר את אלו בעזרת הפרדיקטים הבאים:

* at(p, x, y)

המונית או נוסע p נמצאים במיקום x,y.

* free(t)

מונית t פנויה לאיסוף נוסע.

* inc(v1, v2)

מותר לעבור משורה\עמודה v1 לשורה\עמודה v2 (בהתאמה, כך ש- v1+1=v2).

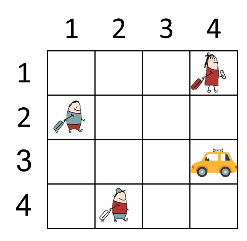
* dec(v1, v2)

מותר לעבור משורה\עמודה v1 לשורה\עמודה v2 (בהתאמה, כך ש- v1-1=v2). פרדיקט זה וזה המופיע מעל (inc) נועדו לדאוג לכך שהמונית תוכל לנוע למשבצות סמוכות בלבד.

* on\_taxi(p)

נוסע p נמצא על המונית.

בדוגמה הבאה המונית נמצאת במיקום (4,3) ושלושה נוסעים נמצאים כרגע במיקומים (4,1), (1,2) ו-(2,4).



אופרטורים:

● התקדמות למשבצת שכנה (מעלה, מטה, ימינה ושמאלה) של המונית.

● העלאה של נוסע.

● הורדה של נוסע.

מצב: מיקום המכונית האוטונומית, האם המונית פנויה, מיקום נוכחי ומיקום יעדם של נוסעים פוטנציאלים.

# שאלות המטלה

1. בשלב הראשון, המונית מאפשרת שיהיה נוסע אחד בלבד לכל היותר על המונית בכל זמן. עזרו לחברת "אוטומה" להגדיר בשפת STRIPS את האופרטורים, **המפורטים מטה,** הנדרשים לטובת שימוש ב-planner בקובץ one\_passenger.py.
2. בחברת "אוטומה" חושבים ירוק, ולכן ביקשו שתהיה קיימת אפשרות להעלות נוסעים נוספים אם יצא מסלול קצר וחסכוני יותר. בשל כך, השלימו את האופרטורים **המפורטים מטה,** בקובץ multiple\_passengers.py כך שהמונית תוכל להעלות ולהוריד מספר נוסעים ככל שנדרש. מומלץ תחילה להעתיק את המימוש שכתבתם בשאלה 1 ולהתאים אותו לביצוע שאלה זו.
3. הריצו מה-main את שני הפתרונות לבעיה ובדקו שחברת אוטומה אכן הצליחה לחסוך וליצור מסלול קצר יותר עבור המונית. שימו לב, הזמן הממוצע להחזרת פתרון הינו כחצי דקה עד דקה.

# הנחיות לביצוע השאלות

* עבור כל אופרטור יש להשלים את שתי הרשימות:

**preconditions** -

הפרדיקטים הנדרשים עבור הפעלת אופרטור זה.

**effects** -

הפרדיקטים שיתווספו או יוסרו בעקבות הפעלת אופרטור זה. לפני פרדיקטים שמסירים יש לשים neg.

* כלל הפרדיקטים מוגדרים בקוד כאוסף ערכים. לדוגמה, הפרדיקט at עבור מונית t הנמצאת במיקום px,py נרשום (‘at’,’t’,’px’,’py’). בקובץ main ניתן לראות כיצד, בעזרת הפרדיקטים, מוגדרים המצב ההתחלתי והמצב הסופי של הבעיות אותן נפתור במטלה. הבעיה משלבת מונית אחד המסומנת ב-T ושלושה נוסעים המסומנים ב-P1, P2 ו-P3. כפי שלמדנו, כאשר מגדירים את האופרטורים, נשתמש במשתנים המייצגים נוסעים ומוניות ולא בקבועים אלו.
* האופרטור **’move-up’** (הזזת המונית על הגריד שורה אחת למעלה) ממומש ומוכן להפעלה, תוכלו להיעזר בו להשלמת האופרטורים האחרים עבור הבנת מבנה הקוד שיש להשלים.

האופרטורים שיש להשלים הם:

* 'move down’

הזז את המונית על הגריד שורה אחת למטה.

פרמטרים:

* מונית - t
* מיקום נוכחי בציר x - px
* מיקום נוכחי בציר py - y
* מיקום חדש בציר by - y
* ‘move-left’

הזז את המונית על הגריד עמודה אחת שמאלה.

פרמטרים:

* מונית - t
* מיקום נוכחי בציר px - x
* מיקום נוכחי בציר py - y
* מיקום חדש בציר bx - x
* ‘move-right’

הזז את המונית על הגריד עמודה אחת ימינה.

פרמטרים:

* מונית - t
* מיקום נוכחי בציר px - x
* מיקום נוכחי בציר py - y
* מיקום חדש בציר bx - x
* ‘pick-up’

אסוף נוסע.

פרמטרים:

* מונית - t
* מיקום נוכחי בציר px - x
* מיקום נוכחי בציר py - y
* נוסע לאיסוף - p
* ‘put-down’

הורד נוסע.

פרמטרים:

* מונית - t
* מיקום נוכחי בציר px - x
* מיקום נוכחי בציר py - y
* נוסע להורדה - p

# הגשת המטלה

* יש להגיש **שני קבצים בלבד** לאתר המודל בדומה להגשת המטלה הקודמת. הקובץ הנדרש:
  + one\_passenger.py
  + multiple\_passengers.py

ניתן לראות את הפידבק להרצה (קומפילציה, מספר טסטים שעברו, שגיאות זמן ריצה וכו'...). ניתן להעלות קובץ נוסף ולהריץ אם היו טעויות שאתם מעוניינים לתקן ולהשלים.

שימו לב, הבדיקה האוטומטית עלולה לקחת מעל דקה. לאחר סיום ההרצה יתקבלו התוצאות. ישנם 2 טסטים הבודקים את הפתרון המוגש למטלה. אם התקבלה שגיאת קומפליציה, עליכם להעלות קובץ חדש בכדי לקבל ציון לאחר התיקונים. הקבצים המקוריים שקיבלתם עם המטלה הינם ללא שגיאות זמן ריצה.

בהצלחה 😊