

חלק ב': היסק לוגי

תכנון מבוסס על ידע חלקי

מבט על

בחלק זה של הפרויקט משימתכם דומה לזו של החלק הראשון: מטרתכם לתכנן פעילות לאיסוף מודיעין הכוללת הפעלה של מגוון מכשירים שברשותכם על מטרות המוגדרות מראש. ההבדל היחיד (והחשוב) הוא שעתה עליכם לקחת בחשבון שהמרחב בו החלליות פועלת עלול להכיל שידורי לייזר המסכנים את החלליות: כשחללית חוצה שידור לייזר היא מתפוצצת (ומפסיקה לתפקד). למרבה המזל, יש לחללית אפשרות מוגבלת לחוש נוכחות של גלי לייזר שהיא נמצאת בסמיכות אליהם. יכולת זאת מוגבלת כי לחללית יש רק אפשרות לדעת כמה שדורי לייזר יש בתאים סביבה אך לא לדעת את מקורם. בנוסף, מספר שדורי הלייזר (מכונות השידור) ידוע מראש. משימתכם היא להשתמש בתצפיות (שנקראות observations) שמתקבלות אחרי הפעלת כל פעולה כדי להשלים את משימתכם. לרשותכם יעמדו כלים להיסק לוגי ותכנון המתבססים על העקרונות שנלמדו במהלך הקורס.

הבעיה

הדומיין (העולם) של הבעיה זהה לזה של החלק הראשון מלבד התוספת של שדורי הלייזר. גלי הלייזר משודרים מעמדות קבועות (שמספרם ניתן כחלק מהקלט) הממוקמות לאורכם של קווי הגריד. כך, למשל, בגריד בגודל 3 (עם ערכי קורדינטות בין 0 ל 2) שדור לייזר בעל ערך $x=1, y=0$ ישדר לכל אורכה של עמודה זו (לכל ערכי z).

כאמור, לחללית יש אפשרות לחוש את מספר שידורי הלייזר בתאים הסמוכים למיקומה הנוכחי (מלבד התאים שנמצאים באלכסון אליה) אך לא את מקור השידור. תצפית ממומשת על ידי dictionary שממפה לכל חללית מספר (מסוג int) שמתאר את מספר שדורי הלייזר סביב למיקום החללית. במידה והחללית מתפוצצת כתוצאה מתזוזה, התצפית המתקבלת עבור חללית זו הוא -1.

שימו לב שהתצפית הנ"ל (עבור חללית שהתפוצצה) תתקבל רק פעם אחת. בהפעלות הבאות החללית שהתפוצצה לא תיכלל בתצפית (והפעלת החללית תחשב כפקודה לא חוקית שתביא לכישלון ההרצה).

דוגמא:

נתון גריד בגודל 3X3X3 שבו יש שתי חלליות spaceship_a ו spaceship_b שנמצאות במיקום (0,0,0) ומיקום (2,0,2) בהתאמה. שידורי לייזר קיימים במיקום $x=1, y=0$ (לכל ערכי z) ומיקום $y=1, z=0$ (לכל ערכי x). במקרה כזה תתקבל התצפית הבאה:

{spaceship_a: 2, spaceship_b: 1}

פעולות מותרות

הפעולות הניתנות לביצוע זהות לאלה שתוארו עבור החלק הראשון. ההבדל היחיד הוא שפעולות תזוזה עלולות להסתיים בפיצוץ כאשר חללית פוגעת בשידור לייזר. שימו לב שמותר לכם להחליט להזיז חללית לתא שבו קיים

שידור לייזר (ולוותר על החללית לטובת המידע שייאסף) - אך עליכם לוודא שהמשימה תושלם. כמו כן - הפעלת חללית שהתפוצצה היא פעולה לא חוקית.

הפתרון

בניגוד לחלק הראשון שבו נדרשתם להחזיר תכנית מלאה שפותרת את הבעיה - בחלק זה של הפרויקט עליכם לממש בקר (controller) לתפעול המערכת. בהתאם לכך, בעוד שהפותר (solver) שמימשתם בחלק הראשון החזיר פתרון שלם, הבקר הנה תכנית שמקבלת בכל שלב תצפית (observation) מהסביבה על המצב הנוכחי (שהתקבל בעקבות הפעלת הפקודה האחרונה), ומחזירה את הפעולה הבאה לביצוע (או None במקרה שאין פעולה לביצוע).

הדרישה

לתרגיל מצורף קוד המממש מבני נתונים ואלגוריתמים שמבצעים היסק לוגי כפי שלמדתם במהלך הקורס. כמו כן, מומלץ לעשות שימוש בקבצי העזר שניתנו לכם בחלק 1. עליכם למדל את הדומיין ואת הבקר על פי הדרישות.

ע"מ לעמוד בדרישות עליכם להשלים את מימושו של class ששמו SpaceshipController ושניתן לכם באופן חלקי ולהשלים בו את הפונקציות הבאות:

1. פונקציית אתחול (init) שמקבלת את תאור הבעיה (problem) ומספר מכונות שדור לייזר שקיימות במרחב (num_of_transmitters).
2. פונקציית get_next_action שמקבלת תצפית (observation) שמתארת את התצפית האחרונה שהתקבלה (התצפית הראשונית או זאת שהתקבלה אחרי הפעלת הפעולה האחרונה). הפונקציה מחזירה רשומה בודדת המפרטת את הפעולה הבאה לביצוע (לפי הפורמט שהוגדר עבור החלק הראשון). במקרה והמערכת לא מחזירה פעולה לביצוע עליכם להחזיר None

לדוגמא: אם הבקר מחזיר את הפעולה

(move,spaceship_a, (0, 0, 0), (0, 1, 0))

על המערכת להזיז את החללית spaceship_a ממיקום (0, 0, 0) למיקום (0, 1, 0)

אתם יכולים להוסיף לקובץ פונקציות ומשתנים כרצונכם ומלבד שמימושכם יהיה מוכל כולו בקובץ ex2.py.

בפרט, אתם יכולים להשתמש בספריית itertools שמממשת פונקציות שונות על קבוצות כגון בחירה של פרמוטציות בעלות גודל מסוים מתוך קבוצה וכו'. את תיעוד הספרייה ניתן למצוא בקישור הבא (קיימות דוגמאות רבות ברשת):

<https://docs.python.org/2/library/itertools.html>

שימו לב כי מידול מצב המערכת הוא על אחריותכם, אך אנו ממליצים להשתמש באובייקט מסוג logic.PropKB (שמסופק לכם בקובץ logic.py המצורף) כדי לתחזק את המידע הרלוונטי למיקומי הלייזר ובטיחות התאים.

במקרים בהם הפעלת הפקודה get_next_action מחזירה פעולת move, מערכת הבדיקה תבדוק אם יעד התזוזה הנו אכן בטוח (לפי מיקום הלייזרים בפועל). אם יש שידור לייזר ביעד, החללית תתפוצץ ותקבלו על כך חיזוי אך לא תוכלו להפעילה בהמשך.

הבדיקה

מאחר והפתרון תלוי בתצפית שמתקבלת בעת ההפעלה, הבדיקה תעשה בצורת סימולציות, הפעלה של המערכת לפי פקודות הבקר ודיווח לבקר על התצפית שהתקבלה עד להשלמת המשימה או מיצוי המשאבים.

כל פתרון ייבדק על כ-20 בעיות בגדלים שונים. לכל סימולציה יינתן פרק זמן מוגבל של חמש דקות להגיע ליעד.

טיב הפתרון שהבקר מייצר עבור בעיה יקבע ע"י העלות שלו (ביחידות דלק-פעולות, כמה שפחות – יותר טוב), בתנאי שהפתרון אכן פותר את הבעיה.

הסבר קצר על הקוד המצורף

הקוד מורכב מקבצי הפיתון הבאים:

1. Ex2.py – קובץ שבו נעשית העבודה העיקרית שלכם, והקובץ היחיד שעליכם לשנות. אמור להכיל את ה-class של spaceship_controller כפי שמתואר בסעיף "דרישה". הקובץ כולל את חתימות של הפונקציות שעליכם לממש. **התרגיל ייבדק עם קובץ ex2.py שלכם, ושאר הקבצים כפי שהם מופיעים במצבם המקורי (כולל אלה שניתנו בחלק א) ולכן אין טעם לשנות קבצים אחרים ולהגישם כחלק מהפתרון.**
שימו לב שאתם משנים את הערך של המשתנה ids בראש הקובץ לערכם של תעודות הזהות שלכם!
2. logic.py – מכיל את אלגוריתמי היסק, ומכיל את מחלקת PropKB. שימו לב - בקובץ עצמו תוכלו למצוא דוגמאות לאתחול ושימוש בבסיס הידע wumpus_kb. אנו ממליצים לעשות שימוש בכלים בתורת הפסוקים כפי שראינו בשיעור.
3. שאר הקבצים – קבצי עזר (שכוללים את הקובץ search.py שניתן לכם בחלק א).

הקלט

הקלט הוא ייצוג של הבעיה (problem) שעליכם למדל (בדיוק כמו בחלק א) ומשתנה num_of_transmitters

שמייצג את מספר תחנות השידור.

דגשים

- 1) שימו לב שהבדיקה של התרגיל היא אוטומטית, ולכן **חשוב מאוד** להקפיד על שמות וסינטקס מדויק של קבצים, פונקציות, מחלקות ופעולות (אותיות גדולות\קטנות, רווחים שלא במקום, קווים תחתונים וכיו"ב). חבל אם הפתרון לא יתקבל על שגיאת דפוס
- 2) אתם יכולים להניח שהקלט ובכלל זה התצפיות תקינות ומתארות מצב עולם אפשרי (חללית יכולה להיות רק במיקום אחד וכו'). לעומת זאת אין ערובה לכך שקיים פתרון.
- 3) כאמור, צורת ייצוג של המצב הפנימי של המערכת היא לשיקולכם, אך אנו ממליצים לממש את בסיס הידע שלכם כאובייקט מסוג PropKB מקובץ logic.py. הקובץ נלקח מהמאגר הבא <https://github.com/yeesian/aima-python> ומכיל מימושים שונים של היסק לוגי.
- 4) מותר לכם להזיז חללית למקום עם שדור לייזר (ובכך להקריב אותה לטובת התצפית שמתקבלת) אך עליכם לקחת בחשבון שבמקרה כזה התצפית עבור חללית זו יהיה 1- פעם אחת בלבד. מאותה נקודה והלאה החללית איננה קיימת והפעלתה אינה חוקית.
- 5) ע"מ להצליח בתרגיל כדאי לעשות את הדברים הבאים:
 - a) אתחול יעיל של בסיס הידע
 - b) בחירה יעילה של הפעולה לביצוע תוך תחזוקה של בסיס הידע
 - c) איזון בין הרצון להתקדם למטרה במהירות לבין ביצוע פעולות שמקנות ידע חשוב לביצוע של המשימה.
- 6) למרות שתאריך ההגשה רחוק, כדאי להתחיל לעבוד על הפרויקט בהקדם

אפשרויות לבדיקה עצמית

אחרי שבניתם את הקובץ `ex2.py` כרצונכם, תוכלו להריץ את פתרוןכם בעזרת שרת שיאפשר לכם לבצע בדיקה אוטומטית של הפתרון שלכם ויפורסם בשבועות הקרובים. שימו לב שהבדיקה אמורה לקבל כקלט גם את הבעיה וגם את מספר עמדות השידור.

הגשה, בדיקת התרגיל וציונים

הבדיקה של כל החלקים יחד תתבצע בסוף הסמסטר. הגשה בזוגות או יחידים בלבד

לחלק הזה:

- חלק מהציון ינתן על עמידה בדרישות הבסיס, וחלק ינתן על בסיס תחרות בין המגישים
- התחרות תתבצע על ביצועים של הפתרון על מגוון קלטים. על הבקר להביא את המערכת לפתרון מהר ככל האפשר תוך שמירה על בסיס ידע מעודכן.
- משקל הציון של החלק הזה הוא $1/3$ מהציון של הפרויקט

תאריך ההגשה: היום האחרון של הסמסטר (25.1.2018 בשעה 23:55)