

האוניברסיטה הפתוחה

20551

# **מבוא לבינה מלאכותית**

חוברת הקורס סתיו 2017א

כתבה : אילנה בס

אוקטובר 2016 – סמסטר סתיו – תשע"ז

**פנימי – לא להפצה.**

© כל הזכויות שמורות לאוניברסיטה הפתוחה.

## תוכן העניינים

א	אל הסטודנט	
ב	1. לוח זמנים ופעילויות	
ד	2. תיאור המטלות	
ד	2.1 מבנה המטלות	
ה	2.2 חומר הלימוד הדרוש לפתרון המטלות	
ה	2.3 ניקוד המטלות	
ו	3. התנאים לקבלת נקודות זכות	
1	ממ"ן 11	
11	ממ"ן 12	
15	ממ"ן 13	
21	ממ"ן 14	
25	ממ"ן 15	
27	ממ"ן 16	
	ממ"ן 17 יפורסם בהמשך	
	ממ"ן 18 יפורסם בהמשך	



## אל הסטודנט,

אנו מקדמים את פניך בברכה עם הצטרפותך אל הלומדים בקורס "מבוא לבינה מלאכותית".

בחוברת זו תמצא את לוח הזמנים של הקורס, תנאים לקבלת נקודות זכות וחלק מהמטלות.

לקורס קיים אתר באינטרנט בו תמצאו חומרי למידה נוספים, אותם מפרסם/מת מרכז/ת ההוראה. בנוסף, האתר מהווה עבורכם ערוץ תקשורת עם צוות ההוראה ועם סטודנטים אחרים בקורס. פרטים על למידה מתוקשבת ואתר הקורס, תמצאו באתר שה"ס בכתובת:

<http://telem.openu.ac.il>

מידע על שירותי ספרייה ומקורות מידע שהאוניברסיטה מעמידה לרשותכם, תמצאו באתר הספרייה באינטרנט [www.openu.ac.il/Library](http://www.openu.ac.il/Library).

צוות הקורס ישמח לעמוד לרשותך בכל שאלה שתתעורר.

ניתן לפנות למנחים בשעות ההנחייה הטלפונית שלהם או אלי בכל יום ד' בשעות 11:00-13:00 בטלפון 09-7781239. כמו כן ניתן לפנות אלי ב- e-mail כתובתי: [ilana@openu.ac.il](mailto:ilana@openu.ac.il)

### לתשומת לב הסטודנטים הלומדים בחו"ל:

למרות הריחוק הפיסי הגדול, נשתדל לשמור אתכם על קשרים הדוקים ולעמוד לרשותכם ככל האפשר.

הפרטים החיוניים על הקורס נכללים בחוברת הקורס וכן באתר הקורס. מומלץ מאוד להשתמש באתר הקורס ובכל אמצעי העזר שבו וכמובן לפנות אלינו במידת הצורך.

אני מאחלת לך לימוד פורה ומהנה.

ב ב ר כ ה,

אילנה בס

מרכזת ההוראה בקורס

**1. לוח זמנים ופעילויות (20551/א/2017)**

שבוע הלימוד	תאריכי שבוע הלימוד	יחידת הלימוד המומלצת	מפגשי ההנחיה*	תאריך אחרון למשלוח הממ"ן (למנחה)
1	4.11.2016-30.10.2016	פרקים 1,2	מפגש 1	
2	11.11.2016-6.11.2016	פרק 3		
3	18.11.2016-13.11.2016	פרק 4	מפגש 2	ממ"ן 11 (להרצה) 19.11.2016
4	25.11.2016-20.11.2016	פרק 5		
5	2.12.2016-27.11.2016	פרק 6	מפגש 3	ממ"ן 12 (תיאורטי) 3.12.2016
6	9.12.2016-4.12.2016	פרק 7		
7	16.12.2016-11.12.2016	פרק 8	מפגש 4	ממ"ן 13 (להרצה) 16.12.2016
8	23.12.2016-18.12.2016	פרק 9		

\* התאריכים המדויקים של המפגשים הקבוצתיים מופיעים ב"לוח מפגשים ומנחים".

לוח זמנים ופעילויות - המשך

שבוע הלימוד	תאריכי שבוע הלימוד	יחידת הלימוד המומלצת	מפגשי ההנחיה*	תאריך אחרון למשלוח הממ"ן (למנחה)
9	30.12.2016-25.12.2016 (א-ו חנוכה)	פרק 10	מפגש 5	ממ"ן 14 (תיאורטי) 30.12.2016
10	6.1.2017-1.1.2017 (א חנוכה)	פרק 13		
11	13.1.2017-8.1.2017	פרק 14	מפגש 6	ממ"ן 15 (להרצה) 14.1.2017
12	20.1.2017-15.1.2017	פרק 17		
13	27.1.2017-22.1.2017	פרק 18		ממ"ן 16 (תיאורטי) 28.1.2017
14	3.2.2017-29.1.2017	פרק 18	מפגש 7	
15	10.2.2017-5.2.2017	חזרה		ממ"ן 17 (תיאורטי) 11.2.2017

מועדי בחינות הגמר יפורסמו בנפרד

**תאריך אחרון למשלוח ממ"ן 18: 10.3.17**

\* התאריכים המדויקים של המפגשים הקבוצתיים מופיעים ב"לוח מפגשים ומנחים".

## 2. תיאור המטלות

קרא היטב עמודים אלו לפני שתתחיל לענות על השאלות

בקורס זה 8 מטלות, 4 מטלות תיאורטיות ו-4 מטלות להרצה. פתרון המטלות הוא חלק בלתי נפרד מלימוד הקורס, שכן הבנה מעמיקה של חומר הלימוד דורשת תרגול רב. יש להגיש לפחות 2 מטלות מבין המטלות התיאורטיות (12,14,16,17) (במשקל כולל של 5 נק' לפחות) ו-2 מטלות לפחות מבין מטלות הרצה (11,13,15,18) (במשקל כולל של 10 נק' לפחות). אם שאלה מסוימת בממ"ן אינה ברורה לך, אל תהסס להתקשר אל המנחה (בשעות הייעוץ הטלפוני שלו) או להיעזר בקבוצת הדיון של הקורס. להלן תמצא הסבר על אופן הפתרון הנדרש וכיצד לשלוח את המטלה למנחה.

### 2.1 מבנה המטלות וצורת הגשתן

בקורס ישנן כאמור מטלות משני סוגים:

#### מטלות רגילות:

מטלה כזו מורכבת מכמה שאלות. בראש כל שאלה מצוין משקלה היחסי בקביעת ציון המטלה. פתרון השאלות במטלה כזו אינו דורש הרצת תכניות במחשב. הן נועדו לבדוק את הבנתך בחומר הלימוד. את הפתרונות למטלה כזו יש לכתוב בצורה ברורה ומסודרת.

#### מטלות הרצה:

במטלות אלה עליך לכתוב תכניות ולהריץ אותן במחשב. את התכניות במטלות 11,13,15 יש לכתוב בשפת Python. את התכנית במטלה 18 ניתן לכתוב ב-Python או ב-C/C++ או ב-Java.

#### תיעוד:

בכל תכנית הוסף תיעוד בגוף התכנית המסביר מהו תפקידו של כל משתנה, מה מבצעת כל שורה וכל הסבר נוסף החשוב להבנת מהלך פעולתה של התכנית. יש לתת שמות משמעותיים למשתנים ולשגרות המופיעים בתכניות. יש להקפיד על קריאות ובהירות תוך שימוש בהיסח (אינדטציה) מסודרת ואחידה.

#### במטלת הרצה עליך לשלוח למנחה:

- את התכנית לאחר שבדקת שהיא מבצעת את הנדרש ממנה ללא טעויות.
- יש להגיש את קובץ המקור של התכנית (source code).
- יש לצרף מספר דוגמאות ריצה של התכנית. תכנית שתשלח ללא דוגמאות ריצה (דוגמאות קלט והפלט שהופק עבורן) לא תיבדק!

תכניות שתוגשנה בכתב-יד או ללא תיעוד או ללא קובץ המקור - לא תבדקנה!



## 2.2 חומר הלימוד הדרוש לפתרון המטלות

בטבלה שלהלן תמצא מהו חומר הלימוד הנדרש (לפי פרקי הספר) לפתרון כל אחת מהמטלות.

### שים לב!

אין להשתמש לפתרון המטלות בידע הנרכש בפרקי לימוד מתקדמים יותר מהפרקים בהם עוסקת המטלה.

מטלה	חומר הלימוד הנדרש לפתרונה
ממ"ן 11	פרקים 1-4
ממ"ן 12	פרקים 1-6
ממ"ן 13	פרקים 1-6
ממ"ן 14	פרקים 7-9
ממ"ן 15	פרקים 1-10
ממ"ן 16	פרק 10, פרקים 13-14
ממ"ן 17	פרקים 17-18
ממ"ן 18	פרקים 1-18

## 2.3 ניקוד המטלות

המשקל הכולל של ממ"נים 11-18 הוא 30 נקודות. עליך לצבור לפחות 15 נקודות.

**ללא עמידה בדרישות המטלות לא ניתן יהיה לגשת לבחינת הגמר**

הכנת המטלות 11-18 חייבת להיעשות ע"י כל סטודנט בנפרד.

מטלות שלא יבוצעו באופן עצמאי – ייפסלו!!!

להלן פירוט הניקוד לכל מטלה:

ממ"ן	ניקוד
11	6
12	3
13	4
14	3
15	5
16	2
17	2
18	5

### **לתשומת לבכם:**

מדיניות קורס זה היא לאשר הזנת ציון אפס במטלות שלא הוגשו כנדרש בקורס. סטודנטים אשר לא הגישו את מכסת המטלות המינימאלית לעמידה בדרישות הקורס ולקבלת זכאות להיבחן, ומבקשים שמטלות חסרות יוזנו בציון אפס, יפנו למוקד הפניות והמידע בטלפון **09-7782222** או **יעדכנו בעצמם** באתר שאילתא <http://www.openu.ac.il/sheilta>

**קורסים → ציוני מטלות ובחינות → הזנת ציון 0 למטלות רשות שלא הוגשו.**

יש לקחת בחשבון כי מטלות אשר יוזן להן ציון אפס ישוקללו בחישוב הציון הסופי ובכך יורידו ציון זה ולא ניתן יהיה להמירן במטלות חלופיות במועד מאוחר יותר. על כן קיימת אפשרות שסטודנט אשר יעבור את הבחינה בהצלחה ייכשל בקורס (כשהמוצע המשוקלל של המטלות והבחינה יהיה נמוך מ-60).

**כלל זה איננו חל על מטלות חובה או על מטלות שנקבע עבורן ציון מינימום.**

### **לתשומת לבכם!**

כדי לעודדכם להגיש לבדיקה מספר רב של מטלות הנהגנו את ההקלה שלהלן:

אם הגשתם מטלות מעל למשקל המינימלי הנדרש בקורס, **המטלות** בציון הנמוך ביותר, שציוניהן נמוכים מציון הבחינה (**עד שתי מטלות**), לא יילקחו בחשבון בעת שקלול הציון הסופי.

זאת בתנאי שמטלות אלה **אינן חלק מדרישות החובה בקורס** ושהמשקל הצבור של המטלות האחרות שהוגשו, מגיע למינימום הנדרש.

**זכרו!** ציון סופי מחושב רק לסטודנטים שעברו את בחינת הגמר בציון 60 ומעלה והגישו מטלות כנדרש באותו קורס.

### **3. התנאים לקבלת נקודות זכות בקורס**

- א. הגשת 2 מטלות לפחות מבין המטלות התיאורטיות (12,14,16,17) תוך צבירת 5 נק' לפחות.
- ב. יש להגיש לפחות 2 מתוך המטלות להרצה (11,13,15,18) וצבירת 10 נק' לפחות.
- ג. ציון 60 לפחות בכל מטלת הרצה.
- ד. ציון 60 לפחות בבחינת הגמר.
- ה. ציון סופי בקורס 60 לפחות.

# מטלת מנחה (ממ"ן) 11 - להרצה

הקורס: 20551 – מבוא לבינה מלאכותית

חומר הלימוד למטלה: שפת Python ופרקים 1-3

משקל המטלה: 6

מספר השאלות: 8

מועד אחרון להגשה: 19.11.2016

סמסטר: 2017א

(אב)

## קיימות שתי חלופות להגשת מטלות:

- שליחת מטלות באמצעות מערכת המטלות המקוונת באתר הבית של הקורס
  - שליחת מטלות באמצעות הדואר או הגשה ישירה למנחה במפגשי ההנחה
- הסבר מפורט ב"נוהל הגשת מטלות מנחה"

מטלות הריצה בקורס ייכתבו בשפת [Python\(2.7\)](#), שפת תכנות מונחה עצמים.

אנו לא מניחים כלל שיש לכם נסיון בתכנות בשפה זו, אך מצפים שתלמדו את הבסיס שלה די מהר, (במיוחד לאור הכרותכם ונסיונכם עם שפת Java).

מצורף [כאן](#) קישור למדריך בסיסי ב-Python(3) ובו משולבים תרגילים שעליכם לפתור כדי להכיר השפה. [אין צורך להגישם.](#)

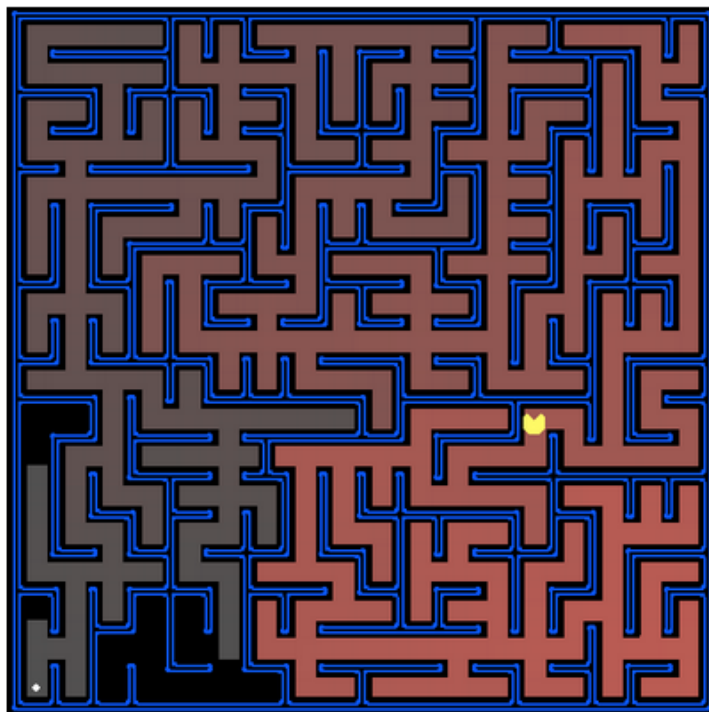
מצורף [כאן](#) קישור למדריך בסיסי ב-Python(2). הקובץ שבקישור מתייחס גם לעבודה בסביבת Unix וניתן להתעלם מההתייחסויות ל-Unix.

שתי מטלות הריצה הראשונות פותחו באוניברסיטת ברקלי על-ידי John DeNero ו-Dan Klein.

מדריך לPython בעברית ניתן למצוא [כאן](#).

מדריך מפורט ניתן למצוא [כאן](#).

# חיפוש ב-Pac-Man



All those colored walls,  
Mazes give Pacman the blues,  
So teach him to search.

## הקדמה

במטלה זו, סוכן פקמן שלכם ימצא מסלולים בעולם המבוך שלו, הן כדי להגיע למיקום מסוים והן כדי ללקט מזון בצורה יעילה. עליכם לכתוב אלגוריתמי חיפוש כלליים ולממשם לתרחישים של פקמן.

הקוד עבור מטלה זו מורכב ממספר קבצי Python. לצורך ביצוע המטלה יהיה עליכם לקרוא ולהבין את חלקם ומהאחרים תוכלו להתעלם. את כל הקוד והקבצים הנלווים ניתן להוריד

[כקובץ zip](#).

הקבצים שתערכו ולתוכם תוסיפו את הקוד שלכם :

[search.py](#)

כל אלגוריתמי החיפוש שכם ייצאו שם

[searchAgents.py](#)

כל סוכני החיפוש שלכם ייצאו שם

## קבצים שכדאי לקרוא :

[pacman.py](#)

הקובץ העיקרי שמריץ משחקי פקמן. הקובץ מתאר את הטיפוס GameState שתשתמשו בו במטלה זו

[game.py](#)

הלוגיקה שמאחורי אופן פעולת עולם הפקמן. קובץ זה מתאר מספר טיפוסים עזר כגון AgentState, Agent, Direction, Grid.

[util.py](#)

מבני נתונים שימושיים למימוש אלגוריתמי חיפוש

## קבצים שאינו צורך להתעמק בקוד שלהם :

[GraphicsDisplay.py](#)

גרפיקה עבור פקמן

[GraphicsUtils.py](#)

תמיכה לגרפיקה של פקמן

[TextDisplay.py](#)

גרפיקת אסקי לפקמן

[GhostAgents.py](#)

סוכנים לבקרה על רוחות רפאים

[KeyboardAgents.py](#)

ממשקי מקלדת לבקרה על פקמן

[Layout.py](#)

קוד לקריאת קבצי layout ואחסון תוכנם

## עליכם להגיש

יש למלא חלקים חסרים בקוד [search.py](#) ו-[searchAgents.py](#) במהלך ביצוע המטלה. עליכם לשלוח למנחה את שני הקבצים הללו (בלבד) וכן את README.txt (שם, ת.ז ותיעוד) כקובץ zip. כלומר יש לשלוח בדיוק קובץ אחד. הנכם מתבקשים לא לשנות שמות של פונקציות או מחלקות הנתונות בקוד.

## **ברוכים הבאים לפקמן**

לאחר שהורדתם את הקוד ([search.zip](#)), עשיתם לו unzip, תוכלו לשחק משחק פקמן. לצורך כך הקלידו בשורת הפקודה (command line):

```
python pacman.py
```

שימו לב: אם קיבלתם הודעת שגיאה לגבי python-tk, התקינו python-tk או ראו הוראות מפורטות יותר [בעמוד זה](#).

פקמן חי בעולם כחול בוהק של מסדרונות פתלתלים וממתקים עגולים טעימים. ניווט בעולם זה ביעילות תיהיה המשימה הראשונה של פקמן.

הסוכן הפשוט ביותר בקובץ [searchAgents.py](#) נקרא GoWestAgent, שתמיד זז מערבה (סוכן של תגובה פשוטה). סוכן זה יכול לעתים לנצח:

```
python pacman.py --layout testMaze --pacman GoWestAgent
```

אבל, העניינים מסתבכים עבור סוכן זה כאשר עליו להסתובב :

```
python pacman.py --layout tinyMaze --pacman GoWestAgent
```

אם פקמן נתקע, תוכלו לצאת מהמשחק על ידי הקשת CTRL-c.

במהרה הסוכן שלכם יפתור לא רק tinyMaze אלא כל מבוך שתמצאו. שימו לב ש- [pacman.py](http://pacman.py) תומך במספר אופציות. כל אחת מהן ניתנת לביטוי באריכות (למשל, --layout) או בקצרה (למשל, -l). תוכלו לראות את רשימת כל האפשרויות וערכי ברירת המחדל שלהן על ידי :

```
python pacman.py -h
```

כמו כן, כל הפקודות המופיעות במטלה זו, מופיעות גם ב- [commands.txt](http://commands.txt), לצורך נוחיות העתקתן.

### שימוש באלגוריתמי חיפוש למציאת נקודת מזון קבועה

בקובץ [searchAgents.py](http://searchAgents.py), תמצאו searchAgent ממומש במלואו, המתכן מסלול דרך עולם פקמן ולאחר מכן מבצע מסלול זה צעד אחר צעד. אלגוריתמי החיפוש עבור התכנית (plan) אינם ממומשים. משימה זו מוטלת עליכם. במהלך פתרון השאלות שיופיעו להלן, תצטרכו כנראה להתייחס לרשימת תיאור סוגי האובייקטים שבקוד המופיעה בסוף מטלה זו. תחילה, בידקו שה- searchAgent פועל היטב על ידי כך שתריצו :

```
python pacman.py -l tinyMaze -p SearchAgent -a fn=tinyMazeSearch
```

הפקודה שלעיל מורה ל- searchAgent להשתמש ב- tinyMazeSearch (הממומש בקובץ

[search.py](http://search.py)) כאלגוריתם החיפוש שלו. פקמן אמור לנווט במבוך בהצלחה.

כעת ניתן לכתוב פונקציות חיפוש כדי לעזור לפקמן לתכנן מסלולים. הפסאודו-קוד לאלגוריתמי החיפוש נמצא בספר הלימוד. זכרו כי צומת במרחב החיפוש צריך להכיל בנוסף למצב עצמו, את כל המידע הדרוש לבנייה מחדש של המסלול (plan) המוביל למצב זה.

*הערה חשובה :* כל פונקציות החיפוש שלכם צריכות להחזיר רשימת פעולות שתוביל את הסוכן מהמצב ההתחלתי אל מצב המטרה. פעולות אלה צריכות להיות (כולן) מהלכים חוקיים (כיוונים מותרים, לא להיכנס בתוך קירות).

רמזים :

1. האלגוריתמים דומים אחד לשני. אלגוריתמים עבור DFS, BFS, A\*, Uniform-Cost search נבדלים זה מזה רק בפרטי אופן ניהול החזית. כלומר, לאחר שתכתבו את DFS כנדרש, יהיה לכם קל מאד לכתוב את יתר האלגוריתמים. אכן, אחד המימושים האפשריים (אינו חובה) דורש יצירת מתודה לחיפוש כללי שמקבלת את מבנה הנתונים לאחזקת המצבים הממתנינים להיסרק.

2. בדקו את טיפוס Queue, Stack ו-PriorityQueue הנתונים לכם בקובץ [util.py](http://util.py) !

## שאלה 1 (10%)

ממשו את אלגוריתם חיפוש לעומק (DFS) בפונקציה `depthFirstSearch` אשר בקובץ [search.py](https://www.cs.princeton.edu/~asw/6.034/assignments/assignment3.py). כדי שהאלגוריתם שלכם יהיה שלם, כתבו את גירסת חיפוש-גרף של חיפוש לעומק, המונעת פיתוח של מצבים שנסרקו כבר. (ראו סעיף 3.3 בספר הלימוד). הקוד שתכתבו אמור למצוא פתרון עבור:

```
python pacman.py -l tinyMaze -p SearchAgent
```

```
python pacman.py -l mediumMaze -p SearchAgent
```

```
python pacman.py -l bigMaze -z .5 -p SearchAgent
```

הלוח של פקמן יראה את המצבים שנסרקו ואת הסדר על פיו נסרקו (אדום בהיר יותר פירושו שנסרקו מוקדם יותר). האם סדר הסריקה תואם לסדר שהייתם מצפים?  
רמז: אם תשתמשו ב-Stack כבמבנה הנתונים שלכם, הפתרון שימצא אלגוריתם חיפוש לעומק שכתבתם עבור `mediumMaze` יהיה באורך 130 (בהנחה שהוספתם עוקבים לחזית על פי הסדר הנתון על ידי `getSuccessors`; אתם אמורים לקבל 244 אם הוספתם אותם בסדר ההפוך).  
האם זה פתרון זול ביותר? נמקו.

## שאלה 2 (10%)

ממשו אלגוריתם חיפוש לרוחב (BFS) בפונקציית `breadthFirstSearch` שבקובץ [search.py](https://www.cs.princeton.edu/~asw/6.034/assignments/assignment3.py). שוב, עליכם לכתוב אלגוריתם חיפוש-גרף המונע פיתוח של מצבים שנסרקו כבר. בדקו את הקוד שלכם באותה הדרך שבדקתם עבור חיפוש לעומק.

```
python pacman.py -l mediumMaze -p SearchAgent -a fn=bfs
```

```
python pacman.py -l bigMaze -p SearchAgent -a fn=bfs -z .5
```

האם אלגוריתם חיפוש לרוחב מוצא פתרון זול ביותר? נמקו.

רמז: אם נראה לכם שפקמן זז לאט מדי, נסו את האופציה `--frameTime 0`.  
שימו לב: אם כתבתם את קוד החיפוש באופן כללי, הוא אמור לעבוד באותה מידה גם עבור בעיית 8-puzzle, ללא כל שינוי.

```
python eightpuzzle.py
```

## שינוי פונקציית המחיר

BFS ימצא את המסלול למטרה בעל מספר פעולות מינימלי, אך יתכן שנרצה למצוא מסלולים שהם "טובים יותר" במובנים אחרים. נתייחס ל-[mediumDottedMaze](#) ול-[mediumScaryMaze](#). על ידי שינוי פונקציית המחיר, אנו יכולים לעודד את פקמן למצוא מסלולים אחרים. למשל, אנו יכולים לתת מחיר גבוה לצעדים באיזורים בהם יש הרבה רוחות רפאים או לתת מחיר נמוך לצעדים באיזורים עתירי מזון וסוכן פקמן רציונלי אמור להתאים את התנהגותו לשינויים אלה.

### שאלה 3 (10%)

ממשו אלגוריתם לחיפוש-גרף מונחה-מחיר בפונקציה `uniformCostSearch` אשר בקובץ [search.py](#). כדאי לעבור על [util.py](#) כדי לראות מבני נתונים שיכולים להיות לכם לעזר במימוש האלגוריתם. כל הסוכנים שלהלן הינם סוכני חיפוש מונחה מחיר וההבדל ביניהם הוא בפונקציית המחיר בה כל אחד משתמש (הסוכנים ופונקציות המחיר נתונים לכם).

```
python pacman.py -l mediumMaze -p SearchAgent -a fn=ucs
python pacman.py -l mediumDottedMaze -p StayEastSearchAgent
python pacman.py -l mediumScaryMaze -p StayWestSearchAgent
```

שימו לב: אתם אמורים לקבל מחירי מסלול מאד נמוכים ומאד גבוהים עבור `StayEastSearchAgent` ו-`StayWestSearchAgent` בהתאמה, עקב פונקציות המחיר האקספוננציאליות שלהם (פרטים בקובץ [searchAgents.py](#)).

## חיפוש A\*

### שאלה 4 (15%)

ממשו אלגוריתם חיפוש-גרף A\* בפונקציה `aStarSearch` (הריקה) אשר בקובץ [search.py](#). A\* מקבל כפרמטר פונקציה יוריסטית. יוריסטיקה מקבלת שני פרמטרים: מצב בבעיית חיפוש (הפרמטר הראשי) והבעיה עצמה. דוגמה טריוויאלית היא הפונקציה היוריסטית `nullHeuristic` שבקובץ [search.py](#). אתם יכולים לבדוק את מימוש A\* על הבעיה המקורית של מציאת מסלול למיקום קבוע במבוך תוך שימוש ביוריסטיקת מרחק מנהטן (הממומשת כבר כ-`manhattanHeuristic` בקובץ [searchAgents.py](#)).

```
python pacman.py -l bigMaze -z .5 -p SearchAgent -a fn=astar,
                        heuristic=manhattanHeuristic
```



תוכלו לראות ש-A\* מוצא את הפתרון האופטימלי קצת מהר יותר מחיפוש מונחה-מחיר (בסביבות 549 לעומת 629 צמתי חיפוש מפותחים במימוש שלנו אבל כשיש עדיפויות זהות, הבחירות שהאלגוריתם יבצע עלולות לשנות מעט את מספרם של המצבים.

מה קורה ב-openMaze לאסטרטגיות החיפוש השונות?

## מציאת כל הפינות

היתרון האמיתי של A\* יבוא לידי ביטוי בבעית חיפוש מאתגרת יותר. ננסח כעת בעיה חדשה ונתכנן יוריסטיקה עבורה. נתון מבוכ עם ארבע פינות ובכל אחת מהן יש נקודה. בבעית החיפוש החדשה יש למצוא את המסלול הקצר ביותר במבוכ העובר דרך ארבע הפינות. (בין אם במבוכ יש שם מזון או לא). שימו לב לכך שעבור מבוכים מסויימים כגון [tiny Corners](#), המסלול הקצר ביותר לא תמיד עובר תחילה דרך נקודת המזון הקרובה ביותר! דמיון: אורך המסלול הקצר ביותר דרך tinyCorners הוא 28 צעדים.

## שאלה 5 (10%)

ממשו את בעיית החיפוש CornersProblem בקובץ [searchAgents.py](#). עליכם לבחור יצוג למצבים אשר יקודד את כל המידע הדרוש כדי לדעת האם נסרקו כל ארבע הפינות. כעת סוכן החיפוש שלכם יפתור:

```
python pacman.py -l tinyCorners -p SearchAgent -a fn=bfs,prob=CornersProblem
```

```
python pacman.py -l mediumCorners -p SearchAgent -a fn=bfs,prob=CornersProblem
```

כדי לקבל ניקוד מלא בשאלה זו, עליכם להגדיר את ייצוג המצב כך שלא יקודד מידע לא רלבנטי (כגון מיקום רוחות הרפאים, היכן יש מזון נוסף וכד'). במיוחד, אל תשתמשו ב-GameState של פקמן כבמצב של מרחב החיפוש. אחרת, הקוד שלכם יהיה איטי מאוד וגם שגוי.

דמיון: החלקים היחידים במצב במשחק שאליהם עליכם להתייחס במימושכם הם המיקום ההתחלתי של פקמן ומיקומן של ארבע הפינות.

המימוש של breadthFirstSearch מפתח פחות מ-2000 צמתיים ב-[medium Corners](#). ואולם, יוריסטיקה (שמשתמשים בה ב-A\*) יכולה להפחית את החיפוש הנדרש.

## שאלה 6 (15%)

ממשו יוריסטיקה עבור בעיית החיפוש CornersProblem ב- CornersHeuristic. אופן מתן הציון לשאלה זו: לא ינתן ניקוד עבור פונקציה יוריסטית שאינה קבילה. תינתן נקודה אחת לכל פונקציה יוריסטית קבילה. נקודה אחת אם יפותחו פחות מ-1200 צמתים. נקודה אחת אם יפותחו פחות מ-900 צמתים. ניקוד מלא אם יפותחו פחות מ-800 צמתים.

```
python pacman.py -l mediumCorners -p AStarCornersAgent -z 0.5
```

רמז: זכרו, פונקציות יוריסטיות מחזירות רק מספרים, אשר צריכים להוות חסמים תחתונים על מחיר המסלול (האמיתי) הקצר ביותר למטרה הקרובה ביותר, כדי שהיוריסטיקות תהיינה קבילות.

שימו לב: AStarCornerAgent הוא קיצור עבור

```
-p SearchAgent -a fn=aStarSearch,prob=Cornersproblem, heuristic=CornersHeuristic
```

## "אכילת" כל הנקודות

נפתור כעת בעית חיפוש קשה: "אכילת" כל המזון של פקמן במספר צעדים קטן ככל האפשר. לצורך כך נזדקק להגדרה חדשה עבור בעית החיפוש המבטאת את בעית "אכילת" כל המזון. FoodSearchProblem בקובץ [searchAgents.py](#) (הממומשת עבורכם). נגדיר פתרון כמסלול האוסף את כל המזון שבעולם הפקמן. לצורך ביצוע מטלה זו, הפתרונות אינם מביאים בחשבון רוחות רפאים כלשהן או חטיפי (טבליות) אנרגיה; הפתרונות תלויים רק במיקום הקירות, מזון רגיל ובפקמן. (כמובן שרוחות רפאים יכולות להרוס את ביצוע הפתרון! בכך נטפל במטלת הריצה הבאה). אם כתבתם נכון את מתודות החיפוש הכלליות,  $A^*$  ללא יוריסטיקה (השקול לחיפוש מונחה-מחיר) יוכל בקלות למצוא פתרון אופטימלי ל-[test Search](#) ללא שינוי בקוד מצדכם (מחיר כולל של 7).

```
python pacman.py -l testSearch -p AStarFoodSearchAgent
```

שימו לב: AStarFoodSearchAgent הוא קיצור עבור

```
-p SearchAgent -a fn=aStar,prob=FoodSearchProblem, heuristic=FoodHeuristic
```

אתם אמורים לגלות שחיפוש מונחה מחיר מתחיל להיות איטי אפילו ה-[tiny Search](#) הפשוט. לצורך השוואה, המימוש שלנו מוצא מסלול באורך 27 ב-2.5 שניות, לאחר פיתוח של 4902 צמתים במרחב החיפוש.

## שאלה 7 (20% + בונוס 5%\*)

השלימו את foodHeuristic בקובץ [searchAgents.py](#) עם פונקציה יוריסטית עקבית עבור FoodSearchProblem. נסו את הסוכן שלכם על לוח trickySearch :

```
python pacman.py -l trickySearch -p AStarFoodSearchAgent
```

סוכן החיפוש המונחה-מחיר שלנו מוצא את הפתרון האופטימלי ב-13 שניות לערך, תוך שהוא סורק 16,000 צמתים. אם היוריסטיקה שלכם קבילה, תקבלו את הניקוד הבא, התלוי במספר הצמתים שפותחו בעזרתה.

ניקוד	מספר צמתים קטן מ-
5%	15000
10%	12000
15%	9000
5% - בונוס*	7000

שימו לב: אם היוריסטיקה שלכם אינה קבילה, לא תקבלו ניקוד כלל. חשבו על ענין הקבילות בזהירות, שכן יוריסטיקה לא קבילה יכולה להצליח ליצור חיפושים מהירים ואף מסלולים אופטימליים. האם תוכלו לפתור [medium Search](#) בזמן קצר? אם כן, הדבר מרשים ביותר או... שהיוריסטיקה שלכם אינה קבילה. \*הציון הכולל במטלה לא יעלה על 100.

### קבילות לעומת עקביות:

טכנית, קבילות אינה מספקת כדי להבטיח נכונות בחיפוש-גרף. יש צורך בקיום תנאי חזק יותר, של עקביות. כדי שיוריסטיקה תהיה עקבית, צריך להתקיים שאם מחירה של פעולה הוא  $c$ , אזי ביצוע פעולה זו יכול לגרום רק לירידה ביוריסטיקה של  $c$  לכל היותר. אם היוריסטיקה שלכם אינה רק קבילה אלא גם עקבית, תקבלו 5% נוספים על תשובתכם לשאלה 7. כמעט תמיד, יוריסטיקות קבילות הן גם עקביות, במיוחד אם הן נגזרו מבעיות מוחלשות. לכן קל יותר, כפי הנראה, להתחיל במחשבה על יוריסטיקות קבילות. לאחר שתהיה לכם יוריסטיקה קבילה שעובדת היטב, תוכלו לבדוק האם היא אכן גם עקבית. לעתים ניתן לגלות חוסר עקביות על ידי בדיקה האם ערכי ה- $f$  של הפתרונות המוחזרים שלכם אינם יורדים. יתר על כן, אם חיפוש מונחה מחיר ו- $A^*$  מחזירים מסלולים באורך שונה, היוריסטיקה שלכם אינה עקבית.

## חיפוש תת-מיטבי (Suboptimal Search)

לעתים, גם עם  $A^*$  ופונקציה יוריסטית טובה, מציאת מסלול אופטימלי דרך כל הנקודות הינה משימה קשה. במקרים כאלה, נרצה עדיין למצוא מסלול טוב, בזמן קצר.

עליכם לכתוב כעת קוד לסוכן פקמן אשר תמיד אוכל באופן חמדני את הנקודה הקרובה ביותר. ClosestDotSearchAgent ממומש עבורכם בקובץ [searchAgents.py](#), אך חסרה לו הפונקציה העיקרית המוצאת מסלול לנקודה הקרובה ביותר.

## שאלה 8 (10%)

ממשו את הפונקציה `findPathToClosestDot` שבקובץ [searchAgents.py](#). הסוכן שלנו פותר מבוך זה (באופן תת-מיטבי) בפחות משניה אחת ובמחיר מסלול של 350 :

```
python pacman.py -l bigSearch -p ClosestDotSearchAgent -z .5
```

דמז : הדרך המהירה ביותר להשלים את `findPathToClosestDot` היא להשלים את החסר ב-AnyFoodSearchProblem (שחסר בו מבחן המטרה). לאחר מכן, פתרו בעיה זו עם פונקציית חיפוש מתאימה. הפתרון אמור להיות מאד קצר!

ה-ClosestDotSearchAgent שלכם לא תמיד ימצא את המסלול האפשרי הקצר ביותר דרך המבוך. אך אם תנסו, תוכלו לשפרו.

לנוחיותכם מרוכזים ברשימה שלהלן סוגי האובייקטים העיקריים המופיעים בקוד הבסיסי הקשורים לבעיות חיפוש:

SearchProblem (search.py)

A SearchProblem is an abstract object that represents the state space, successor function, costs, and goal state of a problem. You will interact with any SearchProblem only through the methods defined at the top of [search.py](#)

PositionSearchProblem (searchAgents.py)

A specific type of SearchProblem that you will be working with --- it corresponds to searching for single pellet in a maze.

CornersProblem (searchAgents.py)

A specific type of SearchProblem that you will define --- it corresponds to searching for a path through all four corners of a maze.

FoodSearchProblem (searchAgents.py)

A specific type of SearchProblem that you will be working with --- it corresponds to searching for way to eat all the pellets in a maze.

Search Function

A search function is a function which takes an instance of SearchProblem as a parameter, runs some algorithm, and returns a sequence of actions that lead to a goal. Example of search functions are `depthFirstSearch` and `breadthFirstSearch`, which you have to write. You are provided `tinyMazeSearch` which is a very bad search function that only works correctly on `tinyMaze`

SearchAgent

SearchAgent is a class which implements an Agent (an object that interacts with the world) and does its planning through a search function. The SearchAgent first uses the search function provided to make a plan of actions to take to reach the goal state, and then executes the actions one at a time.

# מטלת מנחה (ממ"ן) 12

הקורס: 20551 – מבוא לבינה מלאכותית

חומר הלימוד למטלה: פרקים 1-5

משקל המטלה: 3 נקודות

מספר השאלות: 5

מועד אחרון להגשה: 3.12.2016

סמסטר: 2017א

(אב)

<p><b>קיימות שתי חלופות להגשת מטלות:</b></p> <ul style="list-style-type: none"> <li>שליחת מטלות באמצעות מערכת המטלות המקוונת באתר הבית של הקורס</li> <li>שליחת מטלות באמצעות הדואר או הגשה ישירה למנחה במפגשי ההנחיה</li> </ul> <p><b>הסבר מפורט ב"נוהל הגשת מטלות מנחה"</b></p>
--

**שאלה 1 (20 נק': 14 נק' לסעיף א'; 6 נק' לסעיף ב')**

נניח כי הנכם מנהלים דוכן מזון מקסיקני ואתם רוצים להכין טורטיה ממולאת במהירות המירבית.

הכנת טורטיה ממולאת דורשת ביצוע של קבוצת משימות  $T_1, \dots, T_n$ . ביצוע כל משימה  $T_i$  אורך  $t_i$  דקות. כמו כן, משימות מסויימות דורשות ביצוע מקדים של משימות אחרות. יש לחלק את המשימות בין עובדי המטבח.

**דוגמה:** יש לכם שני עובדים במטבח ונתונות המשימות הבאות עם הדרישות והזמן לביצוען כלהלן:

הפעולה	זמן ביצועה (בדקות)	פעולות מקדימות נדרשות
חימום טורטיה	1	-
הכנת (צליית) סטייק	3	-
הוספת שעועית	1	חימום
הוספת גבינה	1	חימום
הוספת סטייק	1	חימום, צליית סטייק
גלגול טורטיה	1	3 פעולות ההוספה
הכנת שקית take away	3	-
גביית הכסף מהלקוח	3	-
הכנסת טורטיה ממולאת לשקית	1	גלגול טורטיה, הכנת שקית

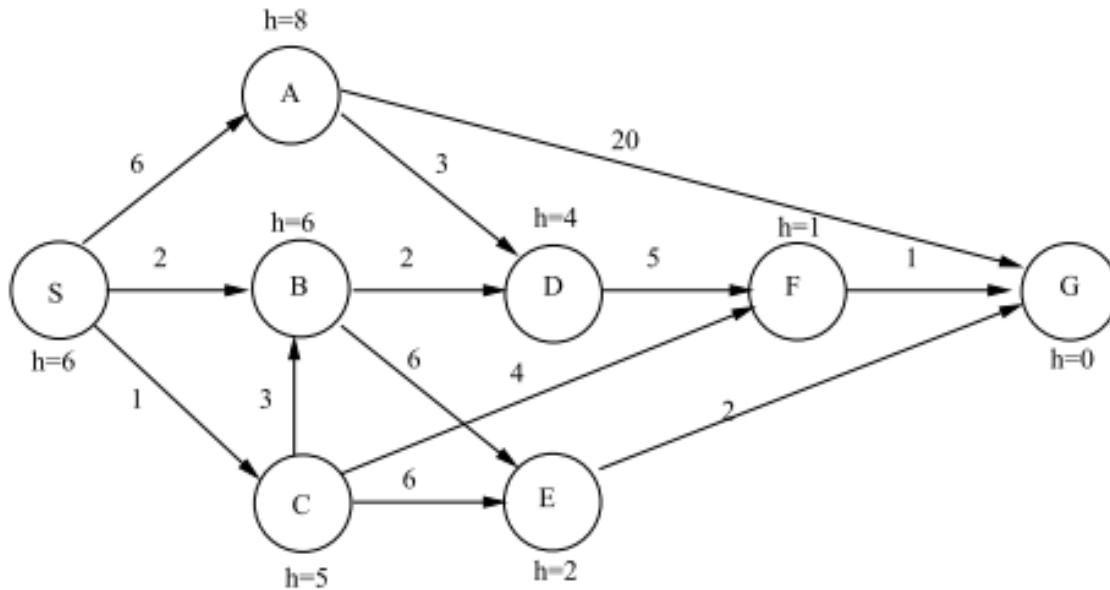
א. נסחו את בעיית הכנת טורטיה ממולאת כבעיית חיפוש כשנתון מספר כלשהו של עובדי מטבח. תארו את מרחב המצבים, המצב ההתחלתי, המצב הסופי (מבחן המטרה) ומודל המעברים עבור בעיית החיפוש זו.

אילו פעולות ניתן לבצע ומהן העלויות של ביצוע פעולה?

ב. הציעו פונקציה יוריסטית קבילה לבעיה זו והסבירו מדוע הפונקציה שהצעתם אכן קבילה.

שאלה 2 (28 נק' - 4 נק' לסעיף (i); 4 נק' לכל תת-סעיף בסעיף (ii))

נתון מרחב החיפוש שלהלן:



S הוא המצב (צומת) ההתחלתי ו-G הוא מצב (צומת) המטרה. מחירי המעבר ממצב למצב מצויינים ליד הקשתות, וערכי הפונקציה היוריסטית מצויינים ליד הצמתים.

כאשר מוסיפים צומת לרשימת הצמתים לפיתוח, הניחו כי סדר הצמתים הינו אלפביתי (כלומר אם צומת S יפותח אזי ברשימה יופיע תחילה A אח"כ B ולאחר מכן C). הניחו כי לרשימת הצמתים לפיתוח לעולם לא יכנסו אבותיהם.

(i) האם הפונקציה היוריסטית בגרף קבילה? הסבירו.

(ii) עבור כל אחד מאלגוריתמי החיפוש הבאים כתבו :

האם יגיע האלגוריתם למטרה;

כתבו על פי הסדר את הצמתים המפותחים (המוצאים מהחזית).

אם יש כמה מועמדים שווי ערך, בחרו ביניהם לפי סדר אלפביתי.

א. DFS      ד. Greedy Best First Search

ב. BFS      ה. Uniform Cost Search

ג. IDS      ו. A\* Search

### שאלה 3 (10 נק')

נתונה פונקציה יוריסטית  $h(v)$  קבילה אך לא עקבית. הראו דוגמה של גרף חיפוש שבו  $A^*$  תחזיר פתרון לא אופטימלי.

### שאלה 4 (21 נק': 12 נק' לסעיף א'; 9 נק' לסעיף ב')

א. עבור כל אחת משיטות החיפוש הבאות, הסבירו כיצד מתקדם החיפוש, וציינו לאיזו שיטת חיפוש מוכרת היא שקולה:

- חיפוש קרן מקומי (local beam search) עם  $k = 1$ .
- חיפוש קרן מקומי עם מצב התחלתי יחיד, וללא הגבלה על מספר המצבים השמורים. (כלומר, ממשיכים עם כל המצבים שנוצרים בכל שלב).
- הדמיית חיפוש (simulated annealing) עם טמפרטורה קבועה  $T = \varepsilon$ , כאשר  $\varepsilon$  הוא מספר חיובי קטן מאוד (כמעט 0).

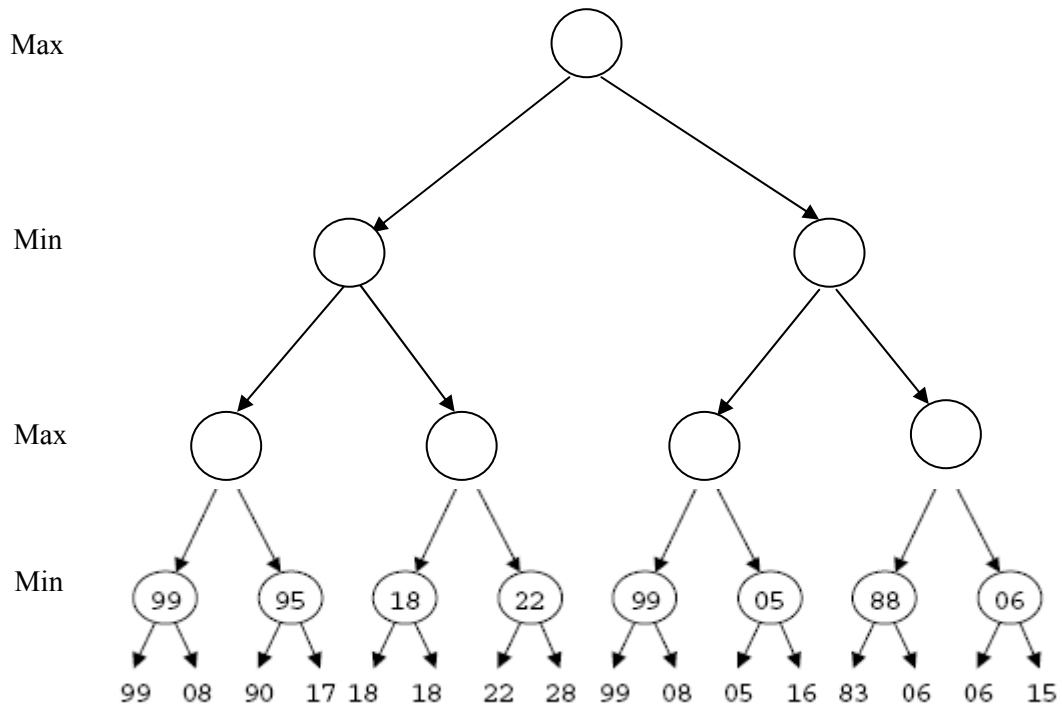
ב. נתונים 10 קלפים הממוספרים 1, ..., 10.

מצאו בעזרת אלגוריתם גנטי חלוקה שלהם לשתי תת-קבוצות של 5 קלפים כל אחת, כך ש:

- סכום הקלפים בקבוצה הראשונה יהיה קרוב ככל האפשר ל-36.
- מכפלת הקלפים בקבוצה השנייה תהיה קרובה ככל האפשר ל-360.

### שאלה 5 (21 נק': סעיפים א', ב' - 2 נק' כ"א; סעיף ג': 5 נק'; סעיף ד': 6 נק'; סעיף ה': 6 נק')

נתון עץ המשחק הבא:



המספרים למטה מייצגים את ערכי העלים לפי Max. המספרים בצמתים מייצגים את הערכת הפונקציה היוריסטית לצמתים אלו.

- (א) בהנחה שעומק החיפוש הוא 3 (כלומר Max משחק, אחריו Min, ושוב Max), איזה פעולה יבחר Max לפי אלגוריתם Minimax? מהו ערך ה-Minimax של העץ?
- (ב) בהנחה שעומק החיפוש הוא 4 (כלומר כל העץ) איזה פעולה יבחר Max לפי אלגוריתם Minimax? מהו ערך ה-Minimax של העץ?
- (ג) האם דרך הפעולה של שחקן Max יכולה להשתנות במעבר מחיפוש לעומק 3 לחיפוש לעומק 4? נמקו.
- (ד) הראו איזה צמתים לא יפותחו לפי אלגוריתם Alpha-Beta כאשר נשתמש בכל העץ וסדר החיפוש יהיה משמאל לימין.
- (ה) סידור אחר של העלים יגרום לכך שאלגוריתם Alpha-Beta יגזום יותר צמתים (כאשר משתמשים בעץ כולו וסדר החיפוש הוא משמאל לימין)?
- אם כן, סדרו מחדש את הצמתים כך שאלגוריתם Alpha-Beta יגזום כמות מקסימלית של צמתים.



# מטלת מנחה (ממ"ן) 13 - להרצה

הקורס: 20551 – מבוא לבינה מלאכותית

חומר הלימוד למטלה: פרקים 1-5

משקל המטלה: 4

מספר השאלות: 3

מועד אחרון להגשה: 3.12.2016

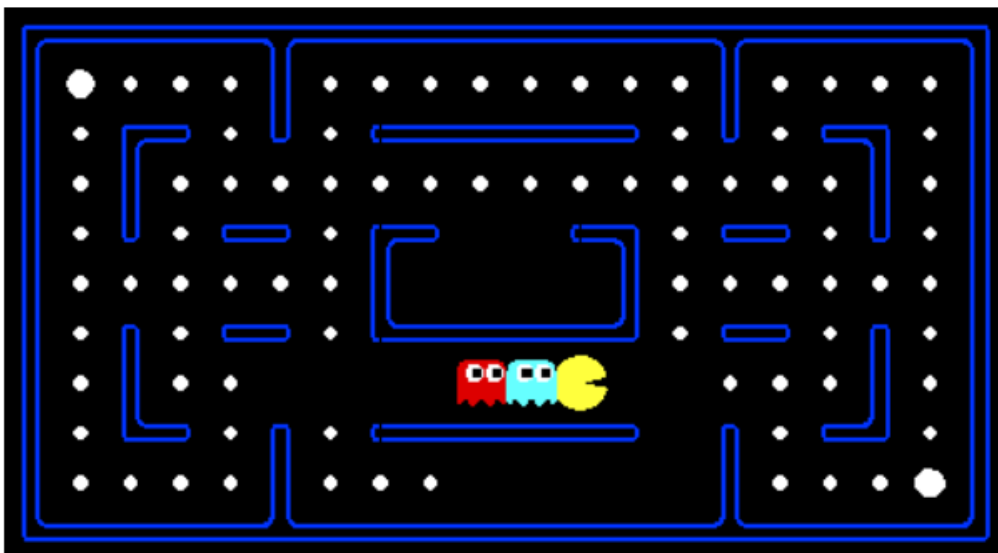
סמסטר: 2017א

(אב)

קיימות שתי חלופות להגשת מטלות:

- שליחת מטלות באמצעות מערכת המטלות המקוונת באתר הבית של הקורס
  - שליחת מטלות באמצעות הדואר או הגשה ישירה למנחה במפגשי ההנחיה
- הסבר מפורט ב"נוהל הגשת מטלות מנחה"

## Pac-Man מרובה סוכנים - Multi-Agent Pac-Man



*Pacman, now with ghosts.*

## הקדמה

במטלה זו, עליכם לתכנן סוכנים עבור הגירסה הקלאסית של Pacman עם רוחות רפאים. לצורך כך תממשו את אלגוריתם מינימקס, אלפא-ביתא ותתנסו בתכנון פונקציית הערכה. כקודמתה, גם מטלה זו פותחה באוניברסיטת ברקלי על-ידי John DeNero ו-Dan Klein.

הקוד עבור מטלה זו לא השתנה בהרבה ביחס לזה של המטלה הקודמת, אך רצוי מאד כי תתקינו אותו מחדש. בכל אופן, אתם יכולים להשתמש בקבצים search.py ו-searchAgents.py שלכם בכל דרך שתמצאו.

הקוד עבור המטלה מכיל את הקבצים הבאים :

### קבצים שיש לקרוא :

[multiAgents.py](#)

כל סוכני החיפוש (multi-agent) שלכם יימצאו שם.

[pacman.py](#)

הקובץ העיקרי שמריץ משחקי פקמן.

הקובץ מתאר את הטיפוס GameState שתשתמשו בו הרבה במטלה זו

[game.py](#)

הלוגיקה שמאחורי אופן פעולת עולם הפקמן. קובץ זה מתאר מספר טיפוסים עזר

כגון AgentState, Agent, Direction, Grid

[util.py](#)

מבני נתונים שימושיים למימוש אלגוריתמי חיפוש

### קבצים שאין צורך להתעמק בקוד שלהם :

[graphicsDisplay.py](#)

גרפיקה עבור פקמן

[graphicsUtils.py](#)

תמיכה לגרפיקה של פקמן

[textDisplay.py](#)

גרפיקת אסקי לפקמן

[ghostAgents.py](#)

סוכנים לבקרה על רוחות רפאים

[keyboardAgents.py](#)

ממשקי מקלדת לבקרה על פקמן

[layout.py](#)

קוד לקריאת קבצי פריסה ( layout ) ואחסון תוכנם

### עליכם להגיש

במהלך ביצוע המטלה יש למלא חלקים חסרים בקוד של [multiAgents.py](#). עליכם לשלוח למנחה קובץ זה (בלבד) וקובץ עם שמכם, ת.ז. והתיעוד. הנכם מתבקשים לא לשנות שמות של פונקציות או מחלקות הנתונות בקוד.

## Pac-Man מרובה סוכנים

תחילה הריצו משחק של Pac-Man הקלאסי:

```
python pacman.py
```

כעת הריצו את ReflexAgent הנתון ב-[multiAgents.py](#):

```
python pacman.py -p ReflexAgent
```

שימו לב שהוא משחק די גרוע אפילו עבור פריסות (layouts) פשוטות:

```
python pacman.py -p ReflexAgent -l testClassic
```

בדקו את הקוד (ב-[multiAgents.py](#)) וודאו שאתם מבינים את אופן פעולתו.

### שאלה 1 (27%)

שפרו את ReflexAgent ב-[multiAgents.py](#) כדי שישחק ברמה סבירה.

(בקוד הנתון עבור סוכן של תגובה-פשוטה (reflex agent) יש כמה דוגמאות שיכולות להיות לעזר של שיטות שמתשאלות את ה-GameState).

סוכן טוב של תגובה פשוטה צריך לקחת בחשבון הן מיקומים של אוכל והן מיקומים של רוחות רפאים. הסוכן שלכם אמור לאפס בקלות את פריסת ה-`testClassic`:

```
python pacman.py -p ReflexAgent -l testClassic
```

נסו את הסוכן תגובה-פשוטה שלכם על פריסת ברירת המחדל -`mediumClassic` עם רוח רפאים אחת או שתיים (וכבו את האנימציה כדי להאיץ את הריצה):

```
python pacman.py --frameTime 0 -p ReflexAgent -k 1
```

```
python pacman.py --frameTime 0 -p ReflexAgent -k 2
```

קרוב לוודאי שהסוכן שלכם ימות לעתים קרובות כאשר יש שתי רוחות רפאים על לוח ברירת המחדל, אלא אם כן פונקציית ההערכה שלכם טובה למדי.

שימו לב:

- לא יתכן שיהיו לכם יותר רוחות רפאים מאשר מאפשרת ה-[layout](#).
- פונקציית ההערכה שתכתבו מעריכה זוגות של מצב-פעולה; בהמשך המטלה תעריכו מצבים.

#### אופציות:

הרוחות שהן ברירת המחדל מתנהגות באופן אקראי. אתם יכולים בשביל הכיף לשחק עם רוחות קצת יותר חכמות בעזרת g DirectionalGhost. אם האקראיות מונעת מכם להסיק האם הסוכן שלכם השתפר, אתם יכולים להשתמש ב-f כדי להריץ עם גרעין אקראי קבוע (אותן בחירות אקראיות בכל משחק). אתם יכולים לשחק משחקים מרובים ברצף עם-n. כבו את הגרפיקה עם-q כדי להריץ הרבה משחקים במהירות.

ניקוד מלא ינתן לסוכן שיכול לאפס במהירות את פריסת ה-openClassic עשר פעמים, מבלי למות יותר מפעמיים או לנוע הלוך ושוב באופן חוזר ונשנה בין שני מיקומים מבלי להתקדם.

```
python pacman.py -p ReflexAgent -l openClassic -n 10 -q
```

#### שאלה 2 (45%)

כעת עליכם לכתוב קוד עבור סוכן חיפוש בתנאי יריבות במחלקה MinimaxAgent אשר ב-[multiAgents.py](#). סוכן המינימקס שלכם אמור לעבוד לכל מספר של רוחות רפאים, לכן עליכם לכתוב אלגוריתם שיהיה קצת יותר כללי מזה שבספר הלימוד. ובמיוחד, לעץ המינימקס שלכם יהיו מספר רמות Min (אחת לכל רוח רפאים) עבור כל רמה של שחקן Max. בנוסף, הקוד שתכתבו צריך לפתח את עץ המשחק לעומק כלשהו. ערכי העלים בעץ המינימקס שלכם ייקבעו על ידי self.evaluationFunction הנתונה, כאשר ברירת המחדל שלה היא scoreEvaluationFunction. המחלקה MinimaxAgent יורשת את MultiAgentAgent, שנותנת גישה ל-self.depth ו-self.evaluationFunction. וודאו כי קוד המינימקס שלכם משתמש במשתנים אלה כשצריך שכן משתנים אלה מקבלים ערך כתוצאה מהאפשרויות של שורת הפקודה (command line).

#### חשוב:

שלב אחד בחיפוש נחשב כמהלך אחד של Pac-Man וכל התגובות של רוחות הרפאים, לכן עומק 2 בחיפוש ירושו שני מהלכים של Pac-Man ושל כל רוח רפאים.

#### רמזים והערות

- פונקציית ההערכה עבור שאלה זו כתובה כבר (self.evaluationFunction). אינכם אמורים לשנות אותה, אך שימו לב לכך שבעת אנו מעריכים מצבים ולא פעולות, כפי שעשינו עבור הסוכן של תגובה פשוטה. סוכנים המסתכלים-קדימה מעריכים מצבים עתידיים בעוד שסוכנים של תגובה פשוטה מעריכים פעולות מהמצב הנוכחי.
- ערכי המינימקס עבור המצב ההתחלתי בסידור ה-minimaxClassic הם 9,8,7,-492 לעומקים 1,2,3,4 בהתאמה. שימו לב לכך שסוכן המינימקס שלכם ינצח לעתים קרובות

(665/1000 משחקים) למרות שהחיזוי שלו הוא שמצבים בעומק 4 הם גרועים.

```
python pacman.py -p MinimaxAgent -l minimaxClassic -a depth=4
```

- כדי להגדיל את עומק החיפוש שתוכלו להשיג באמצעות הסוכן שלכם, הסירו את הפעולה Directions.STOP מהרשימה של Pac-Man של כל הפעולות האפשריות עבורו. עומק 2 יהיה מהיר יחסית אך עומק 3 או 4 יהיה איטי. בשאלה הבאה נטפל במהירות החיפוש.
- פקמן הוא תמיד סוכן 0, והסוכנים נעים בסדר עולה של האינדקסים שלהם.
- כל המצבים במינימקס צריכים להיות GameStates המועברים ל-getAction או נוצרים באמצעות GameState.generateSuccessor.
- עבור לוחות גדולים יותר כגון openClassic ו-mediumClassic (ברירת המחדל), תמצאו לנכון ש-Pac-Man עושה חיל ב"לא למות" אך גרוע ב"לנצח". לעתים קרובות הוא יסתובב סביב עצמו מבלי להתקדם. הוא אפילו עלול להסתובב ממש ליד נקודה מבלי לאכול אותה, משום שאינו יודע לאן ללכת לאחר שיאכל אותה.
- כאשר Pac-Man מאמין שמותו בלתי נמנע, הוא ינסה לסיים את המשחק מהר ככל האפשר, בגלל העונש הקבוע עבור הישארות בחיים. לעיתים זוהי הבחירה השגויה לטיפול ברוחות רפאים אקראיות אך סוכני מינימקס מניחים תמיד את הגרוע ביותר:

```
python pacman.py -p MinimaxAgent -l trappedClassic -a depth=3
```

וודאו כי אתם מבינים מדוע Pac-Man נחפז לרוח הרפאים הקרובה ביותר במקרה זה.

### שאלה 3 (28%)

צרו ב-AlphaBetaAgent סוכן חדש המשתמש בגיזום אלפא-ביתא כדי לסרוק את עץ המינימקס בצורה יעילה. שוב, האלגוריתם שלכם יהיה מעט יותר כלי מהפסאודו קוד שבספר הלימוד, כך שחלק מהאתגר הוא להרחיב את הרעיון שבגיזום אלפא-ביתא כך שיתאים לסוכני מינימום רבים. עליכם לראות שיפור במהירות (למשל, אולי אלפא-ביתא בעומק 3 ירוץ באותו זמן של מינימקס בעומק 2). המצב האידיאלי הוא שעומק 3 ב-smallClassic ירוץ במספר שניות לכל מהלך או אף מהר יותר.

```
python pacman.py -p AlphaBetaAgent -a depth=3 -l smallClassic
```

ערכי המינימקס של AlphaBetaAgent צריכים להיות זהים לערכי המינימקס של MinimaxAgent, למרות שהסעולות שהוא בוחר יכולות להשתנות בגלל התנהלות שונה במקום של החלטות שונות לשבירת שוויון. שוב, ערכי המינימקס של המצב ההתחלתי בסידור ה-minimaxClassic הם 492, 9, 8, 7, -1. בהתאמה.



# מטלת מנחה (ממ"ן) 14

הקורס: 20551 – מבוא לבינה מלאכותית

חומר הלימוד למטלה: פרקים 6-9

משקל המטלה: 3 נקודות

מספר השאלות: 4

מועד אחרון להגשה: 30.12.2016

סמסטר: א2017

(אב)

קיימות שתי חלופות להגשת מטלות:

- שליחת מטלות באמצעות מערכת המטלות המקוונת באתר הבית של הקורס
  - שליחת מטלות באמצעות הדואר או הגשה ישירה למנחה במפגשי ההנחיה
- הסבר מפורט ב"נוהל הגשת מטלות מנחה"

השאלות בעמודים הבאים

## שאלה 1 (35 נק')

לשני המנחים דני (D) וירון (Y) מצפה בוקר עמוס והם מכינים את לוח הזמנים שלהם מראש. לוח הזמנים מורכב מיחידות זמן של שעה אחת: 8-9, 9-10, 10-11, 11-12 בבוקר.

ישנן חמש משימות שצריכות להתבצע:

- (F) – לארגן כיבוד לסמינר המחקר, משימה שלוקח שעה לבצעה.
- (H) – להכין שאלות לשיעורי בית, משימה שלצורך ביצועה נדרשות שעותיים ברצף.
- (P) – להכין את הרובוט PR2 לביקור של קבוצת תלמידי ביי"ס, משימה שלוקח שעה לבצעה.
- (S) – להנחות את סמינר המחקר, משימה שלוקח שעה לבצעה.
- (T) – ללמד את תלמידי ביי"ס על הרובוט PR2, משימה שנדרשות שעותיים ברצף כדי לבצעה.

הדרישות עבור לוח הזמנים הן כדלקמן:

- (a) בכל יחידת זמן, כל מנחה יכול לבצע משימה אחת לכל היותר (F, H, P, S, T).
- (b) הכנת PR2 (P) צריכה להתבצע לפני שמלמדים את הקבוצה של ה-preschoolers (T).
- (c) את הכיבוד יש להביא (F) לפני הסמינר (S).
- (d) הסמינר (S) צריך להסתיים עד השעה 10 בבוקר.
- (e) דני ידאג להביא את הכיבוד (F) משום שיש לו רכב.
- (f) המנחה שאינו מנחה את הסמינר (S) עדיין צריך להיות נוכח בו, ולכן אינו יכול לבצע משימה אחרת (F, T, P, H) במהלך הסמינר.
- (g) מנחה הסמינר (S) אינו מלמד את תלמידי ביה"ס (T).
- (h) המנחה שמלמד את תלמידי ביה"ס (T) צריך גם להכין את הרובוט PR2 (P).
- (i) הכנת שאלות לשיעורי בית (H) אורכת שעותיים רצופות ולכן צריכה להתחיל לכל המאוחר בשעה 10 בבוקר.
- (j) ללמד את תלמידי ביה"ס (T). משימה זו אורכת שעותיים ברצף ולכן צריכה להתחיל כל המאוחר בשעה 10 בבוקר.

כדי לנסח בעיה זו כ-CSP, השתמשו במשתנים F, H, P, S, T. הערכים שהמשתנים יכולים לקבל מציינים את שם המנחה האחראי לביצוע המשימה ואת שעת התחלת ביצועה (תחילת יחידת הזמן שבמהלכה המשימה מתבצעת). יש לזכור כי עבור משימה שהזמן הנדרש לביצועה הוא שעותיים (ברצף), המשתנה אמנם מייצג את שעת ההתחלה של ביצוע המשימה, אבל המנחה יהיה עסוק בביצועה גם ביחידת הזמן הבאה (ע"פ אילוף a). ישנם 8 ערכים אפשריים לכל משתנה ונסמנם כך:

D8, D9, D10, D11, Y8, Y9, Y10, Y11



האות מתאימה לשם המנחה והמספר ליחידת הזמן. למשל, השמת הערך D8 למשתנה פירושה שהמשימה מבוצעת על-ידי המנחה דני החל מהשעה 8 (בבוקר).

א. מהו גודל מרחב המצבים עבור CSP זו?

ב. אילו מהמשפטים a עד j (שלעיל) כוללים אילוצים אונריים?

ג. מחקו את כל הערכים כך שיתקיימו האילוצים האונריים.

ד. לאחר בחירת  $S=D9$ , מחקו את כל הערכים שהיו נמחקים על ידי בדיקה קדימה

(forward checking).

ה. בהתבסס על תוצאת הסעיף הקודם, מי הוא המשתנה הבא שתבחר יוריסטיקת MRV להציב לו ערך?

הציבו את הערך האפשרי הראשון למשתנה זה ובצעו בדיקה קדימה (מחיקת ערכים).

## שאלה 2 (23 נק')

רופא בודק ומאבחן ביחד עם המתמחים שלו פציינטים שיש להם סימפטומים מוזרים.

להלן תיאור של "ידע" רפואי שיש להשתמש בו בשאלה זו:

אם פציינט משתעל, אזי הכליות שלו ו/או הלב אינם מתפקדים כשורה.

לפציינט יהיה חום כאשר המערכת הנורולוגית אינה מתפקדת כשורה או כאשר יש בעיה בזרימת הדם.

קריש דם נוצר כאשר יש כשל בזרימת הדם או כאשר הכבד אינו מתפקד כשורה.

לפציינט יש כאב בבטן אם הכליות אינן מתפקדות כשורה.

לחץ דם גבוה נגרם כאשר הכליות אינן מתפקדות כשורה או כאשר יש בעיה בזרימת הדם.

לפציינט יש כאב ראש כאשר יש כשל באחת או יותר מהמערכות הבאות: המערכת הנורולוגית, הלב או הכליות.

הבהרה: אם נאמר במשפט: " $A$  כאשר  $B$ ", עליכם לפרש זאת כ-  $B \rightarrow A$ .

הניחו כי הפציינט מתקבל לבית החולים עם הסימפטומים הבאים (אבחנה שלכם):

שיעול, חום, קריש דם, לחץ דם גבוה וכאב ראש. לפציינט אין כאב בטן.

אבחנה היא מערכת בגוף שאינה מתפקדת כשורה.

דוגמה לאבחנה: הכליות אינן מתפקדות כשורה.

השתמשו ברזולוציה עם יוריסטיקה אחת או יותר, כדי למצוא אבחנה אפשרית עבור הפציינט.

ציינו באיזה סוג של יוריסטיקה/ות השתמשתם בתהליך הרזולוציה.

### שאלה 3 (14 נק')

ברזולוצית קלט (input resolution) מרשים שימוש בכלל הרזולוציה רק אם לפחות אחת משתי הפסוקיות המשתתפות ברזולוציה שייכת לפסוק המקורי. (כלומר, לא מרשים שימוש בכלל הרזולוציה אם שתי הפסוקיות המשתתפות בגזירת הרזולוציה אינן שייכות לפסוק המקורי).

האם רזולוצית קלט שלמה להפרכה? הוכיחו.

שלמה להפרכה הכוונה: אם לוקחים טענה הנובעת מבסיס הידע ומוסיפים את שלילתה לבסיס הידע, אזי ניתן להסיק פסוק ריק (סתירה) בעזרת כללי ההיסק.

רמז: התבוננו בשלב האחרון של גזירת הפסוקית הריקה מן הפסוק המקורי.

### שאלה 4 (28 נק': 7 נק' לסעיף א'; 7 נק' לסעיף ב'; 14 נק' לסעיף ג')

גיל, רונן והדר שייכים למועדון הספורטאים. כל חבר במועדון זה הוא גולש סקי או מטפס הרים או שניהם. אין מטפס הרים שאוהב גשם וכל הגולשים אוהבים שלג. הדר אינה אוהבת את כל מה שגיל אוהב ואוהבת את כל מה שגיל אינו אוהב. גיל אוהב גשם ושלג.

הוכיחו באמצעות רזולוציה כי "הדר היא מטפסת הרים אך אינה גולשת סקי". לצורך כך:

א. תרגמו את המשפטים שלעיל לפסוקים בלוגיקה מסדר ראשון.

ב. המירו לצורת CNF.

ג. השתמשו בכלל הרזולוציה ובהאחדה (היכן שצריך) כדי להוכיח את המטרה:  
"הדר היא מטפסת הרים אך אינה גולשת סקי".

# מטלת מנחה (ממ"ן) 15 - להרצה

הקורס: 20551 – מבוא לבינה מלאכותית

חומר הלימוד למטלה: פרקים 1-10

משקל המטלה: 5

מספר השאלות: 14

מועד אחרון להגשה: 14.1.2017

סמסטר: 2017א

(אב)

קיימות שתי חלופות להגשת מטלות:

- שליחת מטלות באמצעות מערכת המטלות המקוונת באתר הבית של הקורס
  - שליחת מטלות באמצעות הדואר או הגשה ישירה למנחה במפגשי ההנחיה
- הסבר מפורט ב"נוהל הגשת מטלות מנחה"

## Graphplan



## הקדמה

במטלה זו, עליכם לממש חלקים מהאלגוריתם Graphplan ולתכנן יוריסטיקות מתוך גרף התכנון.

בחלק הראשון תשלימו את המימוש של Graphplan ותבדקו את האלגוריתמים על בעיות מהתחום של ה- "dock worker robot" (הוזכר במדריך הלמידה).  
בחלק השני תשתמשו בגירסה מוחלשת של גרף התכנון כדי ליצור יוריסטיקות ל-A\*.  
בחלק האחרון, תיצרו קבצים לבעיית [מגדלי האנוי](#).

הקוד עבור מטלה זו מורכב ממספר קבצי Python, חלקם יהיה עליכם לקרוא ולהבין כדי לבצע את המטלה ומהאחרים תוכלו להתעלם.  
את כל הקוד והקבצים הנלווים ניתן להוריד [כקובץ zip](#).  
מטלה זו נלקחה מהקורס המקביל באוניברסיטה העברית.

### הקבצים שיש לקרוא:

[graphPlan.py](#), אלגוריתמי ה-graphPlan – מודול האחראי על יצירת Graphplan, הרחבתו במידת הצורך ויצירת תכנית plan.

[planGraphLevel.py](#) ייצוג רמה אחת של הגרף (action layer & proposition layer)

[planningProblem.py](#) ייצוג בעיית התכנון כבעיית חיפוש

[hanoy.py](#) יצירת קובץ הבעיה

### קבצים שאין צורך להתעמק בקוד שלהם:

action.py

proposition.py

actionLayer.py

propositionLayer.py

util.py

### עליכם להגיש

יש למלא חלקים חסרים בקוד של [graphPlan.py](#), [planGraphLevel.py](#),

[planningProblem.py](#), ו-[hanoy.py](#) במהלך ביצוע המטלה.

עליכם לשלוח למנחה קבצים אלו (בלבד) וקובץ עם שמכם והתייעוד. הנכם מתבקשים לא לשנות שמות של פונקציות או מחלקות הנתונות בקוד.

לפתרון המטלה קראו החל משאלה 1 (ולמעשה גם 5 שורות לפניה) ועד הסוף בקישור:

<http://www.cs.huji.ac.il/~ai/graphplan/graphplan.html> ופתרו את כל השאלות.

# מטלת מנחה (ממ"ן) 16

הקורס: 20551 – מבוא לבינה מלאכותית

חומר הלימוד למטלה: פרק 10, פרקים 13-14

משקל המטלה: 2

מספר השאלות: 4

מועד אחרון להגשה: 28.1.2017

סמסטר: א2017

(אב)

## קיימות שתי חלופות להגשת מטלות:

- שליחת מטלות באמצעות מערכת המטלות המקוונת באתר הבית של הקורס
  - שליחת מטלות באמצעות הדואר או הגשה ישירה למנחה במפגשי ההנחיה
- הסבר מפורט ב"נוהל הגשת מטלות מנחה"

שאלה 1 ( 25 נק' : סעיף א' - 13 נק'; סעיף ב' - 12 נק')

נתייחס לפעולות המוגדרות ב-PDDL לבעיית ה-air cargo באיור 10.1 בספר הלימוד בעמוד 369. נגדיר מצב התחלתי ומצב מטרה עבור הבעיה כלהלן:

מצב התחלתי:

$At(P1, TLV) \wedge At(P2, LHR) \wedge At(C1, AMS) \wedge In(C2, P1) \wedge Plane(P1) \wedge Plane(P2) \wedge Cargo(C1) \wedge Cargo(C2) \wedge Airport(TLV) \wedge Airport(LHR) \wedge Airport(AMS)$

מצב מטרה:

$At(P1, LHR) \wedge At(P2, TLV) \wedge At(C1, LHR) \wedge In(C2, P1)$

א. שרטטו עץ חיפוש עד עומק 1 תוך שימוש בחיפוש קדימה (Forward state-space search).

פתחו את צומת השורש והראו את כל הפעולות שניתנות לביצוע במצב ההתחלתי, וכן את המצבים העוקבים המתקבלים מביצוע פעולות אלה. כדי למנוע סרבול יתר ניתן להשמיט בתיאור המצבים את הליטרלים המשתמשים בפרדיקטים: Plane, Airport, Cargo. שימו לב כי יש לציין את כל הפעולות הניתנות לביצוע.

ב. שרטטו שוב עץ חיפוש עד עומק 1 אך השתמשו בחיפוש לאחור (Backward state-space search).

פתחו את צומת השורש והראו את כל הפעולות שהינן רלבנטיות למצב המטרה הנתון והראו את המצבים הקודמים (predecessors) עבור פעולות אלה.

ניתן להשמיט כבסעיף הקודם את הליטרלים המשתמשים בפרדיקטים

Plane, Airport, Cargo.

בנוסף, ניתן להשתמש במשתנים כפרמטרים עבור פעולות.

## שאלה 2 (25 נק')

פתרו את שאלה 10.5 שבספר הלימוד.

## שאלה 3 (25 נק': 6 נק' לסעיף א'; 5 נק' לסעיף ב'; 14 נק' לסעיף ג')

ידוע כי מחלת השחפת (T) עלולה לגרום לשיעול ממושך (R) או לכאבים בחזה (H), וגם למימצאים חיוביים) בצילום רנטגן (X). סרטן הריאה (C) גם משפיע על הסיכוי שיתגלו ממצאים בצילום הרנטגן. הוכח כי סרטן הריאה עלול להיגרם כתוצאה מעישון (S) או מזיהום אוויר (P). עישון עלול לגרום לברונכיטיס כרונית (B) שהסימפטומים שלה כוללים סרטן ריאה, שיעול ממושך וקוצר נשימה (D).

א. שרטטו רשת בייסיאנית המתארת את יחסי הסיבה-תוצאה שלעיל. שמות הצמתים יהיו האותיות שצויינו בסוגריים.

ב. כתבו את הביטוי עבור ההתפלגות המשותפת על-פי מבנה הרשת הבייסיאנית ששרטטתם ויחסי האי-תלות בין משתנים הנובעים ממנה.

ג. כתבו איזו (אילו) מהטענות הבאות נכונה, איזו (אילו) אינה נכונה ואיזו(אילו) מהטענות לא נובעת ממבנה הרשת. צרפו הסבר קצר לכל טענה.

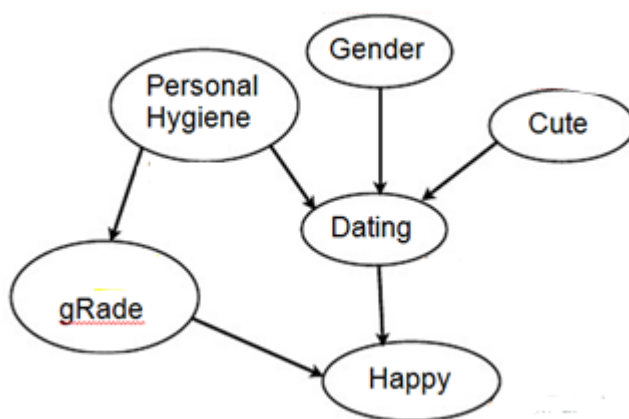
1. עישון וכאבים בחזה הם משתנים בלתי תלויים.
2. עישון וקוצר נשימה הם משתנים בלתי תלויים בהינתן שלפציינט יש ברונכיטיס.
3. ברונכיטיס וכאבים בחזה הם משתנים בלתי תלויים בהינתן שלפציינט יש שיעול ממושך וממצאים חיוביים בצילום הרנטגן.
4. תוצאת צילום הרנטגן ושיעול ממושך הם משתנים בלתי תלויים בהינתן שלפציינט יש סרטן.
5. עישון ומחלת השחפת אינם בלתי תלויים הם משתנים בלתי תלויים בהינתן שאובחן שיש לפציינט סרטן.
6. עישון וזיהום אוויר הם משתנים בלתי תלויים אם לא אובחן כל סימפטום/מחלה/מימצא.
7. אם לפציינט מימצאים חיוביים בצילום, אזי האבחון של סרן ריאה ושל מחלת השחפת הינם בלתי תלויים.

## המשך המטלה בעמודים הבאים

**שאלה 4 (25 נק': 1 נק' ל-1א'; 4 נק' ל-1ב'; 5 נק' לסעיף 2; 15 נק' לסעיף 3)**

להלן נתונה רשת בייסיאנית המראה שביעות רצון (רמת האושר) של סטודנטים בקורס "מבוא לבינה מלאכותית".

המשתנים  $Cute(C)$ ,  $Dating(D)$ ,  $Happy(H)$ , הם משתנים בוליאניים שערכיהם  $\{true, false\}$ .  
 המשתנה  $Gender(G)$  הוא משתנה בוליאני שערכיו הם  $\{male, female\}$ .  
 המשתנה  $Personal\ Hygiene(PH)$  הוא משתנה בוליאני שערכיו הם  $\{good, poor\}$ .  
 תכום הערכים של המשתנה  $gRade(R)$  הוא  $\{A/B, C, F\}$ .



1. השתמשו בהסתברויות המותנות הנובעות מהרשת הנתונה לעיל כדי לכתוב את הביטוי הנדרש עבור התפלגות ההסתברות בכל סעיף:

א.  $P(G, C, D, P, R, H)$

ב.  $P(H|R, D, C)$

פשטו ככל שניתן את הביטוי שהתקבל.

2. מה תוכלו להסיק מסעיף 1-ב' לגבי התלות המותנית שבין  $Happy$  ו- $Cute$ , בהינתן  $Dating$  ו- $gRade$ ?

3. להלן נתונות טבלאות ה-CPT עבור הרשת בייסיאנית שלעיל. חשבו בעזרתן את ההסתברויות הבאות:

א. מהי ההסתברות שסטודנט שמח בהינתן שהוא ממין זכר, יוצא לדייט, ההיגיינה

האישית שלו ירודה והציון שלו בקורס מבוא לבינה מלאכותית הוא A?

$$P(H=true | D=true, G=male, P=poor, R=A/B)$$

ב. מהי ההסתברות שקיים סטודנט כמתואר בסעיף א'?

$$P(H=true, D=true, G=male, P=poor, R=A/B)$$

ג. מהי ההסתברות שסטודנט נחמד בהינתן שהוא ממין זכר, שמח, יוצא לדייט וההיגיינה

האישית שלו טובה  $P(C | D=true, G=male, P=good, H=true)$ ?

P(PH)
0.7

P(G)
0.8

P(C)
0.6

C	G	PH	P(Dating)
<i>t</i>	<i>male</i>	<i>good</i>	0.6
<i>t</i>	<i>male</i>	<i>poor</i>	0.3
<i>t</i>	<i>female</i>	<i>good</i>	0.9
<i>t</i>	<i>female</i>	<i>poor</i>	0.8
<i>f</i>	<i>male</i>	<i>good</i>	0.3
<i>f</i>	<i>male</i>	<i>poor</i>	0.1
<i>f</i>	<i>female</i>	<i>good</i>	0.7
<i>f</i>	<i>female</i>	<i>poor</i>	0.6

PH	P(R)=A/B	P(R)=C	P(R)=F
<i>good</i>	0.7	0.25	0.05
<i>poor</i>	0.6	0.3	0.1

D	R	P(H)
<i>t</i>	A/B	0.8
<i>t</i>	C	0.7
<i>t</i>	F	0.4
<i>f</i>	A/B	0.7
<i>f</i>	C	0.6
<i>f</i>	F	0.3