

ה א ו נ י ב ר ס י ט ה    ה פ ת ו ח ה

20596

**שפת פרולוג**  
**והיבטים לבניה מלאכותית**  
חוברת הקורס – קיץ 2021ג

כתבה : אילנה בס

יולי 2021 – סמסטר קיץ – תשפ"א

**פנימי – לא להפצה.**

© כל הזכויות שמורות לאוניברסיטה הפתוחה.

## תוכן העניינים

א	אל הסטודנט
ג	1. התרגול בקורס
ג	2. מהדורים
ד	3. לוח זמנים ופעילויות
ה	4. תיאור המטלות
ה	4.1 מבנה המטלות
ה	4.2 חומר הלימוד הדרוש לפתרון המטלות
ו	4.3 ניקוד המטלות
ז	5. התנאים לקבלת נקודות זכות
1	ממ"ן 11
7	ממ"ן 12
11	ממ"ן 13
15	ממ"ן 14
19	ממ"ן 15
23	ממ"ן 16
25	ממ"ן 17



## אל הסטודנט,

אנו מקדמים את פניך בברכה עם הצטרפותך אל הלומדים בקורס "שפת פרולוג והיבטים לבינה מלאכותית".

בחוברת זו תמצא את לוח הזמנים של הקורס, תנאים לקבלת נקודות זכות ואת המטלות.

הקורס בסמסטר קיץ נמשך 9 שבועות בלבד ולכן יידרש ממך מאמץ ניכר לעמוד בעומס ובלוח הזמנים. חשוב להקפיד על לימוד החומר והגשת המטלות בקצב שקבענו, כדי להבטיח סיום מוצלח של הקורס. **בגלל משך הסמסטר הקצר, אין אפשרות לדחות את הגשת המטלות.**

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"ם בכתובת:

<http://www.openu.ac.il/shoham>

מידע על שירותי ספרייה ומקורות מידע שהאוניברסיטה מעמידה לרשותכם, תמצאו באתר הספרייה באינטרנט <http://www.openu.ac.il/Library>.

ניתן לפנות 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. התרגול בקורס**

מרכזי הלימוד, ובהם המחשבים, יעמדו לרשותך לצורך כתיבת התכניות והרצתן. לבירור המועדים ושעות התרגול של קבוצתך עיין ב"לוח מפגשים ומנחים".

## **2. מהדריס**

ניתן להוריד מהאינטרנט מהדריס לשפת פרולוג.

המהדריס המומלצים הם:

**1. Amzi Prolog** הניתן להורדה מהאתר <http://www.amzi.com/>

**2. Swi Prolog** הניתן להורדה מהאתר <http://www.swi-prolog.org/download/stable>

דף הבית של Swi - Prolog : <http://www.swi-prolog.org>

**3. LPA WIN-PROLOG**

ניתן להוריד גירסה לתקופה מוגבלת של 28 יום מהקישור:

[http://www.lpa.co.uk/dow\\_tri.htm](http://www.lpa.co.uk/dow_tri.htm)

דף הבית של LPA : <http://www.lpa.co.uk>

### 3. לוח זמנים ופעילויות (20596/ 2021)

שבוע הלימוד	תאריכי שבוע הלימוד	יחידת הלימוד המומלצת	מפגשי ההנחיה*	תאריך אחרון למשלוח הממ"ן (למנחה)
1	9.7.2021-4.7.2021	1,2		
2	16.7.2021-11.7.2021	3		ממ"ן 11 16.7.2021
3	23.7.2021-18.7.2021 (א צום טי באב)	4,5		
4	30.7.2021-25.7.2021	5,6		ממ"ן 12 30.7.2021
5	6.8.2021-1.8.2021	8,9		ממ"ן 13 6.8.2021
6	13.8.2021-8.8.2021	11,12		
7	20.8.2021-15.8.2021	12,14		ממ"ן 14 20.8.2021
8	27.8.2021-22.8.2021	14,15		ממ"ן 15 27.8.2021
9	3.9.2021-29.8.2021	24		ממ"ן 16 3.9.2021

מועדי בחינות הגמר יפורסמו בנפרד

**תאריך אחרון למשלוח ממ"ן 17 (פרויקט הגמר) – 5.11.2021**

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



## 4. תיאור המטלות

קרא היטב עמודים אלו לפני שתתחיל לענות על השאלות

בקורס זה 7 מטלות. המטלה האחרונה מהווה פרויקט גמר ואותה יש להגיש לאחר תום הסמסטר.

יש להגיש לפחות 3 מטלות מבין מטלות 11-16. הגשת פרויקט הגמר (ממ"ן 17) היא חובה. אם שאלה מסוימת בממ"ן אינה ברורה לך, אל תהסס להתקשר אל המנחה (בשעות הייעוץ הטלפוני שלו) לצורך קבלת הסבר. להלן תמצא הסבר על אופן הפתרון הנדרש וכיצד לשלוח את המטלה למנחה.

### 4.1 מבנה המטלות וצורת הגשתן

במטלות שבקורס ישנן שאלות משני סוגים:

**שאלות רגילות:** שאלות "יבשות" שאינן דורשות הרצת תכניות במחשב. הן נועדו לבדוק את הבנתך בחומר הלימוד.

את הפתרונות לשאלות כאלה עליך לרשום על דף נייר בכתב יד ברור ובצורה מסודרת.

**שאלות הרצה:** בשאלות אלה עליך לכתוב תכניות ולהריץ אותן במחשב.

בשאלות הרצה עליך לשלוח למנחה:

א. את התכנית המודפסת, לאחר שבדקת שהיא מבצעת את הנדרש ממנה ללא טעויות.

ב. הדפסה של מספר דוגמאות ריצה של התכנית. תכנית להרצה שתישלח ללא דוגמאות ריצה (דוגמאות קלט והפלט שהופק עבורך) לא תיבדק!

ג. יש לצרף את קובץ המקור של התכנית.

#### שים לב!

אין להגיש תכניות להרצה בכתב-יד! כל התכניות חייבות להיות מורצות ומודפסות במחשב.

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

#### תיעוד:

בכל תכנית, בין אם נדרשת להרצה ובין אם לאו, הוסף תיעוד בגוף התכנית המסביר מהו תפקידו של כל משתנה, מה מבצע כל פרדיקט וכל הסבר נוסף החשוב להבנת מהלך פעולתה של התכנית. יש לתת שמות משמעותיים למשתנים וליחסים (פרדיקטים) המופיעים בתכניות. יש להקפיד על קריאות ובהירות תוך שימוש בהיסח (אינדטציה) מסודרת ואחידה.

### 4.2 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 1-2
ממ"ן 12	פרקים 3-4
ממ"ן 13	פרקים 5-6
ממ"ן 14	פרקים 8-9
ממ"ן 15	פרקים 11-12
ממ"ן 16	פרקים 14 ו-24
ממ"ן 17 - פרוייקט	פרקים 1-24

### 4.3 ניקוד המטלות

המשקל הכולל של ממ"נים 11-16 הוא 20 נקודות. עליך לצבור לפחות 11 נקודות. משקלו של ממ"ן 17 (פרויקט גמר) הוא 20 נקודות.

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

הכנת המטלות 11-16 חייבת להיעשות ע"י כל סטודנט בנפרד.  
מטלות שלא יבוצעו באופן עצמאי – ייפסלו !!!

להלן פירוט הניקוד לכל מטלה:

ממ"ן	ניקוד
11	3
12	4
13	3
14	4
15	3
16	3
17	20

### 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 (כשהממוצע 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.

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

א. הגשת 3 מטלות לפחות מבין מטלות 11-16 במשקל 11 נקודות לפחות.

ב. ציון של 60 נקודות לפחות בבחינת הגמר.

ג. ציון 60 לפחות בפרויקט הגמר.

ד. ציון סופי בקורס של 60 נקודות לפחות.

# מטלת מנחה (ממ"ן) 11

הקורס: 20596 - שפת פרולוג והיבטים לבינה מלאכותית

חומר הלימוד למטלה: פרקים 1-2

משקל המטלה: 3 נקודות

מספר השאלות: 3

מועד אחרון להגשה: 16.7.2021

סמסטר: ג2021

(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"

הערות:

- על מנת לבדוק ששני עצמים  $X$  ו- $Y$  שונים (אין ביניהם matching) ניתן לרשום  $X \neq Y$ .
- שים לב לחומר הלימוד המצויין בראש כל מטלה. אין להשתמש בפרדיקטים או במרכיבי שפה שעדיין לא נלמדו אלא רק במה שנלמד עד הפרקים המצויינים בראש המטלה (כולל). למשל, בממ"ן זה אין להשתמש באריתמטיקה (הנלמדת בפרק 3); בממ"נים 11 ו-12 אין להשתמש בפרדיקט  $\text{not}$  (הנלמד בפרק 5).

**שים לב!** בשאלת הרצה, עליך להקפיד על ההוראות להגשת תרגילי הרצה שפורטו בעמוד יז. בייחוד, יש להקפיד על צירוף תדפיסי קלט / פלט וכן דיסקט עם קוד המקור.

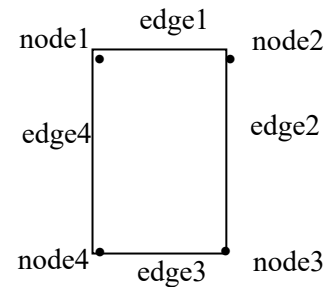
השאלות מופיעות בעמודים הבאים

### שאלה 1 - לא להרצה (27 נקודות)

להלן נתון אוסף עובדות המגדירות צלעות של שני מלבנים, אשר אינם נוגעים זה בזה. כל מלבן מוגדר על-ידי 4 הצלעות המחברות בין קודקודיו. (לא נתונות קואורדינטות של קודקודי המלבנים). (אחד המלבנים משורטט להלן לצורך המחשה)

```
directed_edge( edge1, node1, node2).
directed_edge( edge2, node3, node2).
directed_edge( edge3, node4, node3).
directed_edge( edge4, node1, node4).

directed_edge( edge5, node5, node6).
directed_edge( edge6, node6, node7).
directed_edge( edge7, node8, node7).
directed_edge( edge8, node8, node5).
```



הגדר את הפרדיקטים הבאים :

א. (4 נק')  $\text{edge}(X, \text{NodeI}, \text{NodeJ})$  מצליח אם קיימת צלע (X) המחברת את הנקודות NodeI ו-NodeJ. לדוגמא,  $\text{edge}(\text{edge1}, \text{node2}, \text{node1})$  צריך להצליח.

ב. (7 נק')  $\text{vertical}(\text{EdgeI}, \text{EdgeJ})$  מצליח אם שתי הצלעות EdgeI ו-EdgeJ ניצבות זו לזו.

ג. (9 נק')  $\text{parallel}(\text{EdgeI}, \text{EdgeJ})$  מצליח אם הצלעות EdgeI ו-EdgeJ מקבילות זו לזו (ושייכות כמובן לאותו מלבן).

ד. (7 נק')  $\text{rectangular}(\text{EdgeI}, \text{EdgeJ}, \text{EdgeK}, \text{EdgeL})$  מצליח אם 4 הצלעות הנתונות כארגומנטים מגדירות מלבן.

### הערה:

הפרדיקטים צריכים לעבוד על כל אוסף עובדות המגדיר מלבנים, ולא דווקא על האוסף שניתן בשאלה זו.

המשך המטלה בעמודים הבאים

## שאלה 2 - לא להרצה (33 נקודות: א'-6 נק'; ב'-4 נק'; ג'-15 נק'; ד'-8 נק')

ניתן לתאר את המשפחות הגרות ישירות זו מעל זו באותו בנין רב קומות בעזרת אוסף עובדות מהסוג `live_direct_above (Family1, Family2)`. עובדה כזו היא נכונה אם `Family1` גרה ישירות מעל `Family2`.

א. הגדר פרדיקט `live_above (Family1, Family2)` שיצליח אם `Family1` גרה מעל `Family2` בבניין (לאו דוקא ישירות מעליה).  
לדוגמא, בהינתן העובדות שלהלן:

`live_direct_above (frid, stein).`  
`live_direct_above (asher, frid).`  
`live_direct_above (goren, asher).`  
`live_direct_above (yarden, fogel).`  
`live_direct_above (cohen, yarden).`  
`live_direct_above (zamir, cohen).`  
`live_direct_above (levi, adar).`  
`live_direct_above (adar, segal).`  
`live_direct_above (segal, avraham).`

ניתן יהיה להסיק בעזרת הפרדיקט `live_above` שמשפחת `cohen` גרה מעל משפחת `fogel` וכן שמשפחת `goren` גרה מעל משפחת `stein`.  
לא ניתן יהיה להסיק שמשפחת `levi` גרה מעל משפחת `stein`, למשל.

ב. הגדר פרדיקט `live_below (Family1, Family2)` שיהיה נכון אם `Family1` גרה מתחת ל-`Family2` בבניין. לצורך הגדרת פרדיקט זה השתמש בפרדיקט `live_above` שהגדרת בסעיף א'.  
דוגמאות:

?- `live_below (stein, goren).`  
yes  
?- `live_below (cohen, fogel).`  
no

ג. בסעיף זה נניח שישנם מספר בנייני מגורים.

נניח כי נתונות גם עובדות מן הסוג `live_in_first_floor (Family)` שיהיו נכונות עבור משפחות הגרות בקומה ראשונה.  
בהתייחס לדוגמה שבסעיף א' נוסיף את העובדות הבאות:

`live_in_first_floor (stein).`  
`live_in_first_floor (fogel).`  
`live_in_first_floor (avraham).`

(i) הגדר פרדיקט live\_in\_same\_floor (Family1, Family2) שיצליח אם המשפחה

Family1 גרה באותה קומה של המשפחה Family2. (לאו דווקא באותו בניין).

למשל:

?- live\_in\_same\_floor (cohen, adar).

yes

(ii) הגדר פרדיקט above (Family1, Family2) שיצליח אם המשפחה Family1

גרה בקומה גבוהה יותר מהקומה שבה גרה Family2 (לאו דווקא באותו בניין).

למשל:

?- above (goren, yarden).

yes

ד. הצג בעזרת שרטוט (בדומה לזה שבאיור 1.11 בספר) את המעקב אחר האופן שבו מנסה פרולוג

לענות על השאלות:

?- live\_above (goren, stein).

?- live\_below (fogel, cohen).

**המשך המטלה בעמוד הבא**

### שאלה 3 - הרצה (40 נקודות)

ניתן להגדיר את המספרים הטבעיים בדרך לוגית, ע"י שימוש במבנה נתונים רקורסיבי, כדלהלן:

1. המספר הטבעי הנמוך ביותר, אפס, יוגדר כאטום 0.

2. אם  $X$  הוא מספר טבעי, אזי המספר הטבעי הבא  $(X+1)$  יוגדר כ- $s(X)$ .

אם כך, המספר 1 הוא  $s(0)$ , המספר 2 הוא  $s(s(0))$ , המספר 3 הוא  $s(s(s(0)))$  וכן הלאה.

א. כתוב פרדיקט  $\text{plus}(X,Y,Z)$  אשר שלושת הארגומנטים שלו הם מספרים טבעיים בייצוג הנ"ל,

כאשר  $Z$  הוא הסכום של  $X$  ושל  $Y$ .

**דוגמה:**

?-  $\text{plus}(s(s(0)), s(s(s(0))), Z)$ .

$Z = s(s(s(s(s(0)))))$

?-  $\text{plus}(s(s(0)), Y, s(s(s(s(s(0)))))$ .

$Y = s(s(s(0)))$

?-  $\text{plus}(X, s(s(s(0))), s(s(s(s(s(0)))))$ .

$X = s(s(0))$

ב. כתוב פרדיקט  $\text{times}(X,Y,Z)$  אשר שלושת הארגומנטים שלו הם מספרים טבעיים בייצוג

הנ"ל, כאשר  $Z$  הוא מכפלתם של  $X$  ו- $Y$ .

**דוגמה:**

?-  $\text{times}(s(s(0)), s(s(s(0))), Z)$ .

$Z = s(s(s(s(s(s(s(0)))))$

?-  $\text{times}(s(s(0)), Y, s(s(s(s(s(s(0)))))$ .

$Y = s(s(s(0)))$

?-  $\text{times}(X, s(s(s(0))), s(s(s(s(s(s(0)))))$ .

$X = s(s(0))$

**הערה:** כאן מותר לפרולוג להיכנס ללולאה אינסופית אם מבקשים יותר מפתרון אחד.

ג. כתוב פרדיקט  $\text{not\_greater}(X,Y)$  אשר שני הארגומנטים שלו הם מספרים טבעיים בייצוג

הנ"ל, כאשר  $X$  אינו גדול מ- $Y$ .

**דוגמה:**

?-  $\text{not\_greater}(s(s(0)), s(s(s(0))))$ .

yes

?-  $\text{not\_greater}(X, s(s(s(0))))$ .

$X = 0$  ;

$X = s(0)$  ;

$X = s(s(0))$  ;

$X = s(s(s(0)))$

הסבר היטב, גם באופן הצהרתי (לוגי) וגם באופן פרוצדורלי, את תשובת פרולוג לשאלתא :

?-  $\text{not\_greater}(s(s(0)), Y)$ .





# מטלת מנחה (ממ"ן) 12

הקורס: 20596 - שפת פרולוג והיבטים לבינה מלאכותית

חומר הלימוד למטלה: פרקים 3-4

משקל המטלה: 4 נקודות

מספר השאלות: 3

מועד אחרון להגשה: 30.7.2021

סמסטר: ג2021

(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"

אם לא פתרת את ממ"ן 11, קרא תחילה את ההערות המופיעות בתחילתו.

שאלה 1 - הרצה (26 נקודות)

כתוב פרדיקט  $\text{translate}(X, \text{Num})$  אשר הארגומנט הראשון שלו הוא מספר בייצוג הלוגי (כפי שהוצג בממ"ן 11 שאלה 3) והארגומנט השני שלו הוא מספר בייצוג הרגיל בעל ערך זהה.

דוגמאות:

?-  $\text{translate}(\text{s}(\text{s}(\text{s}(\text{s}(\text{s}(0)))))$ , Num).

Num = 6

?-  $\text{translate}(X, 4)$ .

$X = \text{s}(\text{s}(\text{s}(0)))$

?-  $\text{translate}(X, 2)$ ,  $\text{translate}(Y, 3)$ ,  $\text{times}(X, Y, Z)$ ,  $\text{translate}(Z, \text{Num})$ .

$X = \text{s}(0)$ ,

$Y = \text{s}(\text{s}(0))$ ,

$Z = \text{s}(\text{s}(\text{s}(\text{s}(\text{s}(0)))))$ ,

Num = 6

הערה: בשאלה זו מותר לפרולוג להיכנס ללולאה אינסופית אם מבקשים יותר מפתרון אחד.

## שאלה 2 - הרצה (40 נקודות)

פעמים רבות מתעורר הצורך לערבב רשימות מספרים נתונה באופן שיראה כאקראי. בשאלה זו עליך לכתוב תכנית המערבבת רשימת אטומים נתונה בשיטה המזכירה ערבוב חפיסת קלפים כפי שיוסבר להלן.

נגדיר מיזוג 1-1 (perfect\_shuffle) של שתי רשימות כך :

בהינתן שתי רשימות :  $[X_1, X_2, \dots, X_n]$   $[Y_1, Y_2, \dots, Y_m]$  (n, m טבעיים)

הרשימה שתתקבל לאחר ביצוע מיזוג 1-1 היא אחת מהרשימות הבאות (עפ"י היחס בין n ל-m) :

$$\begin{cases} [X_1, Y_1, X_2, Y_2, \dots, X_n, Y_n] & n = m \\ [X_1, Y_1, \dots, X_n, Y_n, Y_{n+1}, \dots, Y_m] & m > n \\ [X_1, Y_1, \dots, X_m, Y_m, X_{m+1}, \dots, X_n] & n > m \end{cases}$$

כדי לערבב רשימה נבצע מיזוג 1-1 לשני חצאיה מספר פעמים.

8 נק') א. כתוב והרץ פרדיקט range( Low, High, List) המקבל כקלט שני מספרים טבעיים Low ו-High ומחזיר רשימה המכילה את כל האיברים שבין Low ו-High (כולל אותם).

דוגמה : range( 2, 8, List).

List = [2,3,4,5,6,7,8]

16 נק') ב. כתוב והרץ פרדיקט המקבל כקלט רשימה של מספרים וחוצה אותה לשתי רשימות שהפרש אורכיהן הוא לכל היותר 1.

דוגמה :

divide ( [1,2,3,4,5,6], List1, List2).

List1=[1,2,3]

List2=[4,5,6]

16 נק') ג. כתוב והרץ תכנית shuffle(List, Times, NewList) המקבלת כקלט רשימת מספרים List ומערבבת אותה על-ידי ביצוע Times פעמים מיזוג 1-1 לחציה הימני

עם חציה השמאלי. הרשימה המתקבלת כתוצאה מהערבוב היא NewList.

דוגמה :

shuffle( [1,2,3,4,5,6,7,8,9,10], 4, List).

List=[1,5,9,4,8,3,7,2,6,10]

כדוגמת הרצה, הדפס את תוצאת העירבוב של רשימת כל המספרים שבין 1 ל-27, 4 פעמים.

### שאלה 3 – הרצה (34 נקודות)

נתונה החידה הבאה :

ברחוב מסוים ישנה שורה של חמישה בתים,  
האנגלי חי בבית האדום,  
לספרדי יש כלב,  
האיש בבית הירוק שותה קפה,  
האוקראיני שותה תה,  
הבית הלבן צמוד לבית הירוק,  
האיש שברשותו חתול מעשן מרלבורו,  
האיש בבית הצהוב מעשן טיים,  
האיש בבית האמצעי שותה חלב,  
הנורבגי גר בבית הראשון,  
הבעלים של השועל גר בצמוד לאיש שמעשן מונטנה,  
מעשן הטיים גר צמוד לאיש שברשותו סוס,  
האיש שמעשן לקי-סטרייק שותה מיץ תפוזים,  
היפני מעשן פרלמנט,  
הנורבגי גר צמוד לבית הכחול.

מהי ארץ המוצא של הבעלים של הזברה, ומהי ארץ המוצא של האיש שאוהב לשתות מים ?

1. כל אדם מגדל בעל חיים אחד בלבד : זברה או בעל-חיים אחר.
2. כל אדם שותה משקה אחד בלבד : מים או משקה אחר.

כתוב והרץ **תכנית** בשפת פרולוג המוצאת את פתרון החידה.  
יש לייצג באופן מתאים את נתוני החידה כך שהתכנית תוכל למצוא את פתרונה.  
שים לב שהנתון הבסיסי הוא בית ולכל בית יש חמישה מאפיינים שונים.



# מטלת מנחה (ממ"ן) 13

הקורס: 20596 - שפת פרולוג והיבטים לבינה מלאכותית

חומר הלימוד למטלה: פרקים 5-6

משקל המטלה: 3 נקודות

מספר השאלות: 4

מועד אחרון להגשה: 6.8.2021

סמסטר: ג2021

(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 - הרצה (20 נקודות)

א. כתוב והרץ תכנית `merge(SortedList1, SortedList2, MergedList)` המקבלת כקלט שתי רשימות של מספרים, ממויינות (בסדר לא יורד) וממזגת אותן באופן יעיל לרשימה ממויינת אחת.

התכנית תשתמש ב- `Cuts` ירוקים.

דוגמה:

`merge([1,5,7], [3,5], MergedList)`?

`MergedList=[1,3,5,5,7]`

ב. כתוב והרץ גירסה חדשה ל- `merge` המשתמשת ב- `Cuts` אדומים.

הערה: עליך לציין לגבי כל `cut` האם הוא אדום או ירוק.

שאלה 2 - הרצה (30 נקודות)

נתון ריבוע בגודל  $3 \times 3$ :



כתוב והרץ תכנית לניהול המשחק איקס-מיקס-דריקס, בין שני שחקנים אנושיים.

כל שחקן ממלא בתורו ריבוע פנוי בסימן  $X$  או  $O$  (אין חשיבות לסימן "המתחיל", אך יש למלא את הסימנים לסירוגין).

המנצח הוא השחקן שמצליח ליצור שלשה רצופה מסימנו. דהיינו שורה, עמודה או אלכסון מאותו סימן שבו השתמש במשחק.

על התוכנית לספק ממשק נוח לניהול המשחק, שיכלול את הצגת לוח המשחק. כמו כן, התוכנית צריכה להתמודד עם קלטים לא חוקיים.

### שאלה 3 - הרצה (30 נקודות)

כתוב פרדיקט  $\text{bestof}(\text{Val}, \text{GT}, \text{Goal})$  הפועל בדומה ל- $\text{findall}$  אך במקום לשמור את כל הפתרונות הוא שומר רק את הפתרון הטוב ביותר, כפי שנקבע עפ"י יחס הסדר GT. כאשר Val הוא הערך ה"טוב" ביותר עבורו המטרה Goal מתקיימת. על מנת להגדיר איזה ערך הוא הטוב ביותר משתמשים ביחס הסדר הבינארי GT.

**הערה:** יש להקפיד על יעילות ניצול הזיכרון, כל שלעולם לא יישמרו בזיכרון פתרונות שכבר נסרקו (פרט לפתרון הטוב ביותר עד כה). בפרט, **אסור להשתמש בפרדיקטים**  $\text{setof}$ ,  $\text{findall}$  ו- $\text{bagof}$ .

#### דוגמאות:

בהינתן העובדות הבאות:

```
point(1,2).
point(1,10).
point(2,3).
point(5,2).
point(4,4).
```

בשאלתה הבאה אנו מחפשים את קואורדינטת ה-Y (הארגומנט השני של point) **הגדולה ביותר**:

```
?- bestof(Y,>,point(_,Y)).
```

$Y = 10$

בשאלתה הבאה אנו מחפשים את קואורדינטת ה-Y (הארגומנט השני של point) **הקטנה ביותר**:

```
?- bestof(Y,<,point(_,Y)).
```

$Y = 2$

בשאלתה הבאה אנו מחפשים את קואורדינטת ה-X (הארגומנט הראשון של point) **הגדולה ביותר**:

```
?- bestof(X,>,point(X,_)).
```

$X = 5$

בשאלתה הבאה אנו מחפשים את קואורדינטת ה-X (הארגומנט הראשון של point) **הקטנה ביותר**:

```
?- bestof(X,<,point(X,_)).
```

$X = 1$

בכל הדוגמאות האלו, עבור יחס הסדר השתמשנו ב"גדול" או ב"קטן" המובנים בשפת פרולוג. כמובן שניתן להשתמש גם ביחסים המוגדרים ע"י המשתמש.

#### שאלה 4 - הרצה (20 נקודות)

כתוב פרדיקט `match_words(Atoms, Target, Seq)` המקבל רשימת אטומים (Atoms) ומילת מטרה (Target). הפרדיקט מחשב את רשימת האטומים (Seq), המכילה אטומים מתוך Atoms, אשר שירשורם יחדיו (לפי הסדר בו הם נמצאים בתוך Seq), יוצר את מילת המטרה.  
**הערה:** אינך נדרש למימוש יעיל בשאלה זו. כמו כן, מותר שפתרון יחזור על עצמו מספר פעמים.

#### דוגמאות:

```
?- match_words([cd,de,da,a,def,bc,abc], abcde, Seq).
```

```
Seq = [a,bc,de] ;
```

```
Seq = [abc,de]
```

```
?- match_words([lpa, lp, pro, 'a p',' ', win, rol, og, log], 'lpa prolog', Seq).
```

```
Seq = [lpa,' ',pro,log] ;
```

```
Seq = [lp,'a p',rol,og]
```





# מטלת מנחה (ממ"ן) 14

הקורס: 20596 - שפת פרולוג והיבטים לבניה מלאכותית

חומר הלימוד למטלה: פרקים 8-9

משקל המטלה: 4 נקודות

מספר השאלות: 3

מועד אחרון להגשה: 20.8.2021

סמסטר: 2021

(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 - הרצה (30 נקודות)

נגדיר:

(I) T הוא עץ זיגזגי עולה אם:

א. T הוא עץ ריק, או -

ב.  $T = t(\text{Left}, \text{Root}, \text{Right})$  כך ש-:

(i) Left ו-Right הינם עצים זיגזגיים יורדים (ראה הגדרה להלן).

(ii) ערכי הצמתים ב-Left קטנים ממש מ-Root.

(iii) ערכי הצמתים ב-Right גדולים ממש מ-Root.

(II) T הוא עץ זיגזגי יורד אם:

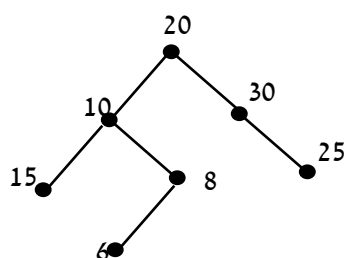
א. T הוא עץ ריק, או -

ב.  $T = t(\text{Left}, \text{Root}, \text{Right})$  כך ש-:

(i) Left ו-Right הינם עצים זיגזגיים עולים.

(ii) ערכי הצמתים ב-Left גדולים ממש מ-Root.

(iii) ערכי הצמתים ב-Right קטנים ממש מ-Root.



דוגמה:

- העץ שערך שורשו הוא 20 הינו עץ זיגזגי עולה.
- העץ שערך שורשו הוא 10 הינו עץ זיגזגי יורד.

כתוב תכנית המקבלת כקלט עץ בינרי ומצליחה אם"ם העץ הינו עץ זיגזגי עולה.

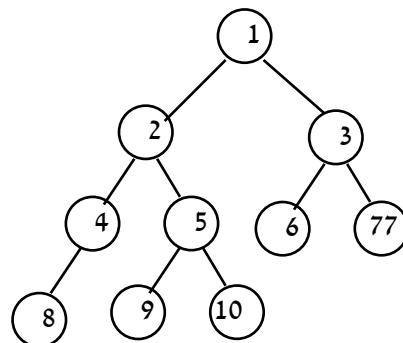
## שאלה 2 – הרצה (30 נקודות)

נתון עץ בינרי המכיל מספר שלם בכל אחד מצמתיו.  
כתוב 3 פרדיקטים לסריקת העץ בשלושה אופנים :  
בסדר תחילי (preorder), בסדר תוכי (inorder) ובסדר סופי (postorder). כל אחד מהפרדיקטים יצור את רשימת עלי העץ המתקבלת מאופן הסריקה המתאים.  
**יש לכתוב תוכנית יעילה המשתמשת ברשימות הפרש.**

### הגדרות:

סריקה על-פי **סדר תחילי** : תחילה מבקרים בשורש, לאחר מכן נסרק תת-העץ השמאלי בסדר תחילי ולבסוף נסרק תת-העץ הימני בסדר תחילי.  
סריקה על-פי **סדר תוכי** : תחילה נסרק תת-העץ השמאלי בסדר תוכי, לאחר מכן מבקרים בשורש ולבסוף נסרק תת-העץ הימני בסדר תוכי.  
סריקה על-פי **סדר סופי** : תחילה נסרק תת-העץ השמאלי בסדר סופי, לאחר מכן נסרק תת-העץ הימני בסדר סופי ולבסוף נסרק השורש.

יש להריץ את התכנית עבור העץ שלהלן :



רשימות העלים שיתקבלו מהרצת התכנית על העץ הנתון :

[1,2,4,8,5,9,10,3,6,7]	תחילי :
[8,4,2,9,5,10,1,6,3,7]	תוכי :
[8,4,9,10,5,2,6,7,3,1]	סופי :

**המשך המטלה בעמוד הבא**

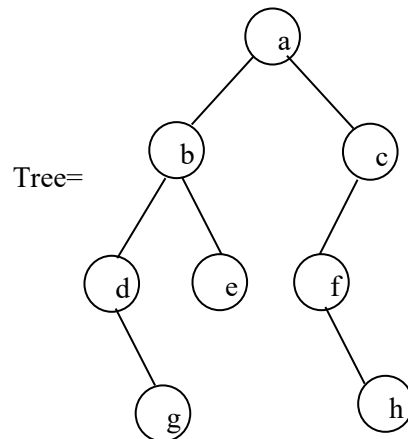
### שאלה 3 - הרצה (40 נקודות)

כתוב פרדיקט  $\text{inorder\_place}(\text{Tree}, X, \text{Place})$  המקבל כקלט עץ בינרי  $\text{Tree}$  ואיבר כלשהו  $X$  ומחזיר ב- $\text{Place}$  את מיקומו של האיבר  $X$  בסריקה תוכנית של העץ  $\text{Tree}$ . אם  $X$  לא מופיע בעץ יש להחזיר את הערך -1.

אין להפוך את העץ לרשימה.

ניקוד מלא ינתן לביצוע משימה זו בסריקה אחת של העץ (כלומר ביקור יחיד בכל צומת).

**דוגמה:** עבור העץ:



התשובות לשאלות הבאות הן:

?-  $\text{inorder\_place}(\text{Tree}, c, \text{Place})$ .

$\text{Place} = 8$

?-  $\text{inorder\_place}(\text{Tree}, d, \text{Place})$ .

$\text{Place} = 1$

?-  $\text{inorder\_place}(\text{Tree}, r, \text{Place})$ .

$\text{Place} = -1$



# מטלת מנחה (ממ"ן) 15

הקורס: 20596 - שפת פרולוג והיבטים לבינה מלאכותית

חומר הלימוד למטלה: פרקים 11-12

משקל המטלה: 3 נקודות

מספר השאלות: 3

מועד אחרון להגשה: 27.8.2021

סמסטר: 2021

(אב)

## קיימות שתי חלופות להגשת מטלות:

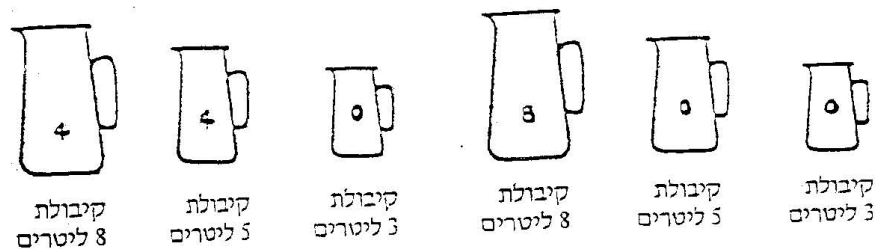
- שליחת מטלות באמצעות מערכת המטלות המקוונת באתר הבית של הקורס
  - שליחת מטלות באמצעות הדואר או הגשה ישירה למנחה במפגשי ההנחיה
- הסבר מפורט ב"נוהל הגשת מטלות מנחה"

## שאלה 1 - הרצה (40 נקודות)

נתונים שלושה כדים היכולים להכיל 8, 5 ו-3 ליטרים מים בהתאמה. בהתחלה הכד הגדול מלא (מכיל 8 ליטר מים) ושני האחרים ריקים. המטרה היא למצוא דרך למזוג את המים כך שלבסוף הכד הגדול (בעל קיבולת 8 ליטרים) יכיל 4 ליטרים והכד הבינוני (בעל קיבולת 5 ליטרים) יכיל 4 ליטרים.

מצב מטרה

מצב התחלתי



על הכדים אין כל סימון ולכן יש למלא לגמרי כדים ריקים או לרוקן לגמרי תכולתו של כד אחד לתוך כד אחר בעל קיבולת גדולה יותר.

א. מהו מרחב המצבים? (אין צורך לפרט את כל המצבים, מספיק לתת תיאור פרמטרי מדויק

של המצבים, ולהדגים ע"י כמה מצבים.)

רמז: היעזר באופרטור '/' כדי לציין את נפח המים בכד מתוך קיבולת הכד.

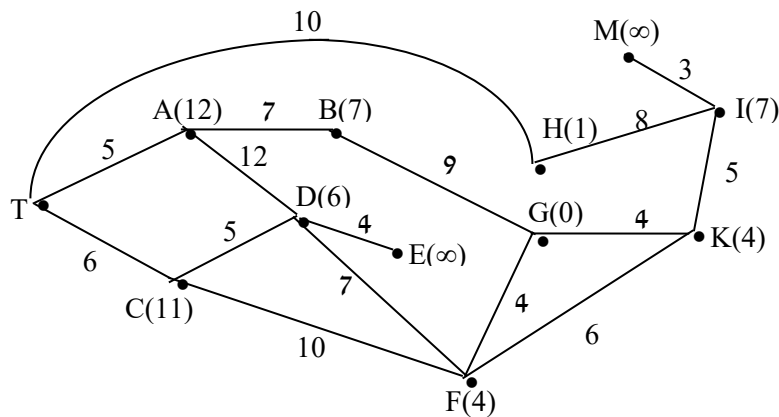
ב. מהן פעולות המזיגה האפשריות (אופרטורים) המאפשרות מעבר ממצב אחד למשנהו במרחב המצבים?

**שים לב:** אין הכוונה לחישובי כמויות מדויקים.  
השאלה מתייחסת למעשה לכל שלושה כדים (כדוגמת הללו) כך שקיבולת הכד הגדול אינה גדולה מסכום הקיבולות של שני הכדים האחרים.

ג. כתוב והרץ תכנית למציאת סדרת המעברים (המזיגות) הפותרת בעיה זו לפי שיטת החיפוש לעומק. **עליך למצוא פתרון במינימום צעדים.** התכנית צריכה להיות כללית ולהתאים לכל שלושה כדים כאמור המקיימים את האילוץ שבסעיף ב'.  
**רמז:** את חישובי הכמויות עצמם לגבי כל אחד מהמעברים ניתן לבצע בעזרת פרוצדורה נפרדת שהקלט שלה הוא כמויות המים בכדים במצב הנוכחי (לפני המזיגה) והיא מחשבת (ומחזירה כפלט) את כמויות המים בכדים במצב הבא (לאחר המזיגה).

## שאלה 2 (30 נקודות)

נתונה מפת הרחובות שלהלן :



מטרת החיפוש היא למצוא את הדרך הקצרה ביותר מ-T ל-G.  
אורכי הרחובות מצויינים על גבי הקטעים המייצגים אותם במפה. (למשל אורך הרחוב מ-T ל-A הוא 5). בסוגריים מצויין המרחק האווירי מכל צומת ל-G.

- א. (15 נק') תאר את סדר סריקת הצמתים במפה לפי אלגוריתם החיפוש Best First. הנח כי ערכי פונקציית ההערכה ההיוריסטית הם המרחקים האוויריים.
- ב. (10 נק') גישה היוריסטית נוספת לפתרון בעיות היא האלגוריתם המכונה Hill-Climb שהוא הכללה של חיפוש לעומק. בחיפוש לעומק לא נקבע כל כלל לפיו בוחרים את הבן הבא, ולכן סדר הבחירה בין אחים נקבע שרירותית ("משמאל לימין"). לעומת זאת, ב-Hill-Climb בוחרים מבין הבנים של הצומת הנוכחי X (בהם עדיין לא ביקרנו) את הצומת Y שהוא הקרוב ביותר למטרה בהסתמך על הידע ההיוריסטי ועל אורך הקשת המובילה לצומת Y. למעשה ב-Hill-Climb בוחרים בכל פעם בן של הצומת הנוכחי שנראה כמבטיח ביותר אך לא מתייחסים כלל לבנים של צמתים אחרים, כלומר בוחרים בצומת הטוב ביותר לוקלית. כדי להימנע ממעגלים בזמן החיפוש יש לדאוג לא לחזור באותו מסלול על צמתים שבהם כבר היינו בהתאם למוסבר בספר הלימוד ב-Fig 11.7.

תאר את סדר סריקת הצמתים במפה לפי אלגוריתם Hill-Climb.

- ג. (5 נק') איזה מבין שני האלגוריתמים שבהם השתמשת (Best first ו-Hill Climb) מצא דרך קצרה יותר מ-T לצומת המטרה G ? הסבר.



### שאלה 3- הרצה (30 נקודות)

בתרשים 2.14 בספר מתואר פרדיקט הפותר את בעיית הקוף והבננה. ובסעיף 2.6.1 צוינה רגישות הפתרון לסידור אופטימלי של פסוקיות הפרדיקט move.

ניתן לבטל רגישות זאת תוך שימוש בהעמקה איטרטיבית (iterative-deepening, עמ' 248). עליך לתקן את הפרדיקט כך שהוא יעבוד כראוי עבור כל סדר של הפסוקיות. למשל עבור הסדר המתקבל כאשר מקדימים את הפסוקית האחרונה של היחס move (זו שמבצעת את הפעולה walk), והופכים אותה לפסוקית ראשונה.

כמו כן על הפרדיקט להראות את הדרך שעשה הקוף על מנת להשיג את הבננה, עפ"י הדוגמא שלהלן.

שים לב, כיוון שמשתמשים בהעמקה איטרטיבית, דרך זו היא תמיד הקצרה ביותר.

#### דוגמה:

?- canget(state(atdoor, onfloor, atwindow, hasnot), Path).

```
Path = [state(atdoor,onfloor,atwindow,hasnot),
        walk(atdoor,atwindow) - state(atwindow,onfloor,atwindow,hasnot),
        push(atwindow,middle) - state(middle,onfloor,middle,hasnot),
        climb - state(middle,onbox,middle,hasnot),
        grasp - state(middle,onbox,middle,has)]
```

#### הערה:

אל תנסה למנוע מעגלים ע"י בדיקה שמצב נתון לא מופיע כבר במסלול הנוכחי. טכניקה כזאת לא מתאימה כאן, כי מצבים חדשים נוצרים ע"י השמת ערכים למשתנים שהיו חופשיים במצב הקודם.

# מטלת מנחה (ממ"ן) 16

הקורס: 20596 - שפת פרולוג והיבטים לבינה מלאכותית

חומר הלימוד למטלה: פרקים 14 ו-24

משקל המטלה: 3 נקודות

מספר השאלות: 1

מועד אחרון להגשה: 3.9.2021

סמסטר: 2021

(א.ב.)

## 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 - להרצה (100 נקודות)

אלגוריתם אלפא-ביתא מתייחס אל מרחב המצבים כאל עץ, בו לכל צומת ניתן להגיע במסלול יחיד מהשורש. אולם, בפועל מרחב המצבים הוא גרף ולא עץ, דהיינו ניתן להגיע למצב מסויים במסלולי משחק שונים. משמעות עובדה זו היא שחלק ניכר מפעולת הסריקה של עץ המשחק מיותר לחלוטין, מכיוון שאנו מחשבים את ערך האלפא-ביתא של כל מצב פעמים רבות, עבור כל צומת בעץ המייצג אותו. כל תכניות המשחק המודרניות מנצלות תופעה זו, והגישה המועדפת היא לא לוותר על אלגוריתם האלפא-ביתא בעל המוניטין הרב, אלא לשנותו מעט כדי שלא יבצע עבודה מיותרת. הפתרון הידוע הוא שמירת ערכו של כל צומת שאנו סורקים בטבלת גיבוב (hash-table), כך שאם נזדקק להערכה נוספת של צומת זה, לא נצטרך לחשב את ערכו מחדש, אלא נשלוף אותו מהטבלה. תוצאות גישה זו מרשימות מאוד וחוסכות כ-80% מזמן החיפוש.

בתרגיל זה ניישם שיטה זו בשפת פרולוג עבור משחק  $x-o$  (כפי שהוגדר בממ"ן 13) שמרחב מצביו קטן דיו כדי להיות מוכל, בשלמותו, בזכרון המחשב.

עליכם להרחיב את התוכנית משאלה 2 בממ"ן 13, כך שתנהל משחק  $x-o$  בין שחקן אנושי לבין המחשב. המחשב ישחק תוך כדי הישענות על הרחבת אלגוריתם האלפא-ביתא השומרת צמתים שנסרקו. יש לסרוק את עץ המשחק עד להגעה למצב סופי, אין צורך בפונקציה היוריסטית.

## הדרכה:

ממש את חיפוש אלפא-ביתא כפי שהוא מופיע בספר בעמוד 589, עבור המשחק שהוגדר לעיל. באלגוריתם המובא בספר, מספר פרדיקטים לא מומשו, כי הם תלויים ביישום הספציפי, ועליך לממשם.

המשך השאלה בעמוד הבא

הפרדיקטים הללו הם :

`moves(Pos, PosList)` - פרדיקט זה מחזיר את רשימת כל המצבים אליהם ניתן להגיע ע"י צעד אחד ממצב נתון. חשוב לשים לב לכך שפרדיקט זה נכשל אם הגענו למצב סופי (סיום המשחק בניצחון, הפסד או תיקו), אשר לא קיים לו צעד המשך.

`min_to_move(Pos)`, `max_to_move(Pos)` - Pos הוא מצב, והצד שתורו לנוע מוגדר בתוך `Pos`. פרדיקטים אלו מצליחים אם השחקן שתורו לנוע הוא Max/Min.

`staticval(Pos)` - מחזיר הערכה מספרית לטיב מצב. במקרה שלנו מעריכים מצבים סופיים ולכן, כאמור, אין צורך בהערכה היוריסטית, אלא יש לתת ערך 1 כש-Max מנצח, ערך 1- כש-Min מנצח וערך 0 במקרה של תיקו.

עליך לשפר את אלגוריתם אלפא-ביתא, על-ידי זכירת ערכו של כל צומת שנסרק כעובדה בבסיס הנתונים. בכל פעם שאלפא-ביתא יתבקש לחשב ערך של מצב, הוא יבדוק האם ערך זה שמור בזיכרון. אם הערך אכן נמצא בזיכרון, ניתן לשולפו מיידית מהזיכרון ואין צורך בהמשך החיפוש. אם הוא ייתקל במצב חדש עליו לחשב את ערכו בשיטה הרגילה, ולהכניס לזיכרון על-ידי `assert` עובדה שתכיל את שם המצב והערך שחושב עבורו.

התכנית שתבנה תתבסס באופן ניכר על מנגנון ה- `asserts` וחשוב מאד להשתמש בו באופן זהיר. יש לבצע `retract` או `retractall` לכל פרדיקט עם סיום השימוש בו, כדי לנקות את הזכרון. `retractall` הוא פרדיקט המוחק מבסיס הנתונים את כל הפסוקיות השייכות לפרדיקט מסויים. לדוגמה אם רוצים למחוק כל הפסוקיות של הפרדיקט `known( Pos, Val)` יש לבצע :

`retractall( known( _,_ )).`

שים לב כי מהדרים מסוימים של פרולוג לא מאפשרים להשתמש ביחס שעדיין לא הוגדר אך בעתיד יבוצע לו `assert` (הם לא מחזירים `no` כפי שניתן לצפות, אלא מפסיקים את פעולת התכנית). פתרון אפשרי הוא לבצע באתחול התכנית `assert` ליחס המבוקש עם ערכי דמה לא משמעותיים, כדי שניתן יהיה להשתמש בו.

#### הערה:

תוכל להיעזר בקוד המקור של אלגוריתם אלפא-ביתא שהובא בספר, אותו תוכל להשיג באתר האינטרנט המלווה את ספר הקורס : [www.booksites.net/bratko](http://www.booksites.net/bratko).

במידה ותיתקל בבעיות זיכרון, תוכל להגדיל את הזיכרון של פרולוג. הדרך לבצע זאת ב-LPA Win-Prolog היא ע"י ארגומנטים משורת הפקודה, למשל להעלות את מערכת הפרולוג ע"י הפקודה :

"C:\Program Files\WIN-PROLOG\PRO386W.EXE" /p4096 /b1024 /l2048 /r1024 /s1024 /h4096 /i1024

פרטים נוספים תמצא בעזרה של ה-LPA תחת הנושא Command-Line Switches.

# מטלת מנחה (ממ"ן) 17

**הקורס:** 20596 - שפת פרולוג והיבטים לבינה מלאכותית

**חומר הלימוד למטלה:** פרוייקט גמר

**משקל המטלה:** 20 נקודות

**מספר השאלות:** 1

**מועד אחרון להגשה:** 5.11.2021

**סמסטר:** 2021

(אב)

רצוי מאוד לבצע את הפרוייקט בזוגות אך ניתן לבצעו גם באופן עצמאי.

**שימו לב:**

הפרוייקט כמוהו כבחנית בית ולכן על כל זוג סטודנטים (או סטודנט יחיד) להכינו **באופן עצמאי**. עליכם לפתח בעצמכם את כל חלקי הפרוייקט. לא ניתן להשתמש במימוש מוכן מכל מקור שהוא.

**רשימת נושאים מוצעים לפרוייקט הגמר:**

1. מערכת מומחה:

- \* לאיבחון רפואי (Medical Diagnostic)
- \* לייעוץ בנושא השקעות
- \* לאיתור תקלות (במחשב, למשל)
- \* סימולטור לפסיכולוג/פסיכיאטר
- \* להגדרת צמחים או בעלי-חיים
- \* להגדרת משפחת מלוכה
- \* לפתרון בעיית תזמון והקצאת משאבים (Scheduling)

2. תכנית המשחקת משחק כלשהו המיועד לשני שחקנים שאחד מהם הוא המשתמש והאחר הוא המחשב.

## א.דרישות כלליות:

### 1) תיעוד

התיעוד צריך לכלול:

i. הערה ראשית ובה:

Programmer	-	שם המגיש/ים
File Name	-	שם הקובץ
Description	-	תיאור כללי של התכנית
Input	-	קלט (אם יש)
Output	-	פלט (אם יש)
Synopsis	-	איך להריץ את התכנית (הוראות הפעלה למשתמש)

- ii. מסמך תיעוד חיצוני המפרט את השיטות והאלגוריתמים בהם משתמשים בפרויקט. יש לציין כל קשר או שוני של האלגוריתם הנבחר לאלגוריתמים דומים שבספר. יש להגדיר באופן ברור את מבנה (או מבני) הנתונים המרכזיים. במקרה בו משתמשים בפונקציה היוריסטית יש להגדירה בדיוק רב ובאופן ברור (בעברית, לא בפרולוג).
- iii. הסבר קצר לכל פרדיקט מהותי (אין צורך לתעד פרדיקטים טריוויאליים כמו conc, member או כל פרדיקט מובן מאליו).
- iv. שימוש בשמות משמעותיים (לפרדיקטים ולמשתנים) שייקלו את קריאת התכנית והבנתה.
- v. הסבר מפורט לנקודות מסובכות / לא ברורות (אם יש כאלו).
- vi. הסבר על תוספות ושיפורים (אם ישנם).
- vii. הסבר על שגיאות בהן מטפלת התכנית.

2) **יעילות.** יש לתת דגש על כתיבת אלגוריתמים יעילים.

3) **מבניות התכנית** (מן הפרדיקט הראשי ומטה).

4) **ממשק נוח למשתמש:** התכנית צריכה לתת למשתמש הסבר ברור וקריא (בזמן הריצה). יש להימנע, לדוגמא, מהדפסת מסך הסבר ארוך מהמסך עצמו וגלגולו מבלי שהמשתמש יספיק לקרוא את כל ההוראות.

צריך להניח שהמשתמש הוא "משתמש טיפש" והתכנית צריכה להסביר את עצמה. כמו כן צריך לאפשר למשתמש לבצע פעולות אלמנטריות כמו יציאה באמצע או התחלה מחדש. הכוונה אינה לבנות ממשק מלא ומקיף אלא ממשק מצומצם שעובד טוב.

5) **טיפול בשגיאות:** התכנית לא יכולה להניח שהקלט מהמשתמש הוא תקין וצריכה לטפל בדרך כלשהי בקלט שגוי.

6) **ביצועים:** תגובת המחשב בזמן סביר! (יש לציין באיזה מחשב ובאיזה קומפיילר של פרולוג מומשה התכנית). אם מדובר במשחק חייבת להיות אפשרות להריצו עבור לוחות בגודל סביר (למשל דמקה בלוח 8 על 8).

## ב. שיפורים ותוספות

להלן שיפורים ותוספות אפשריים:

- 1) טיפול אלגנטי יותר בשגיאות (למשל זיהוי קלט "כמעט נכון" של המשתמש, כמו למשל הרחבת הרעיון של means בעמוד 373 בספר).
- 2) כלליות (לוח בגודל  $n$  במשחק או חוקים כלליים במערכת מומחה).
- 3) שימוש בבסיס נתונים מורכב ומעניין במערכת מומחה.
- 4) רעיון מקורי לפרויקט עצמו.

## ג. דרישות ממערכת מומחה

- 1) לאפשר למשתמש כמה אופציות פעולה או כמה תשובות אפשריות (זאת אומרת לא לכפות עליו לענות תמיד ברצף על שאלות סגורות).
- 2) לממש כל אחת מהאופציות הבאות:
  - קליטת עובדות (או כללים) מהמשתמש תוך כדי ריצה.
  - מימוש how או why.
  - הכנסת הסתברויות דיסקרטיות (זאת אומרת לא טווח בלתי מוגבל אלא מספר דרגות מוגדרות מראש כמו "בטוח", "קרוב לודאי", "אולי", "בטוח שלא") במקום רק "כן" או "לא".
- 3) לממש אופציה נוספת שאינה מופיעה בספר, לפי בחירתכם, ולתעד אותה בפירוט.

## ד. דרישות ממשחק

- 1) הכוונה למשחק של מחשב מול שחקן ולא לפתרון בעיה (כמו בעיית המלכות בפרק 4).
- 2) נדרש מימוש של שיטת חיפוש (רצוי אלפא-ביתא).
- 3) שימוש בפונקציה היוריסטית מתאימה.
- 4) אם ניתן, רצוי להגדיר משחק כללי עם לוח בגודל  $n \times n$ .
- 5) ניתן כמובן להמציא משחק מקורי או שילוב של משהו מוכר עם משהו אחר.
- 6) יש לאפשר מס' דרגות קושי במשחק.
- 7) ממשק גרפי.

## ה. ניקוד הפרויקט

- 65% - נכונות התכנית וקיום הדרישות (הכלליות בסעיפים א'3- א'5 והספציפיות בסעיף ג' או ד')
- 20% - יעילות וביצועים.
- 15% - תיעוד (כמפורט בסעיף א'1).
- ייתכן בונוס מסויים על שיפורים ותוספות אך בכל מקרה הציון המקסימלי של הפרויקט לא יעלה על 100.

## ו. הרחבת דוגמאות למערכת מומחה

### (1) מערכת להגדרת ציפורים:

- יכולה לאפשר למשתמש: (א) לקבל מידע על ציפור מסוימת.  
(ב) להגדיר (ע"י הצגת שאלות למשתמש) ציפור מסוימת שראה או לפחות את המשפחה אליה היא שייכת.  
(ג) לדלות רשימה של כל הציפורים מאותה משפחה או בעלות אותה תכונה.  
(ד) להוסיף עובדות חדשות (ציפורים נוספות) או חוקים חדשים (איך לאפיין את הציפורים).

### (2) מערכת לעזרה בבישול

תחלק למשל את המאכלים לסוגים שונים: אפיה, טיגון, סלטים וכו'.  
בהינתן מאכל רצוי ע"י המשתמש, תדריך אותו מה לקנות (אחרי שתברר מה יש בבית) ואיך להכין אותו.  
אין הכוונה לבסיס נתונים (בלבד) של מתכונים וחומרים. הכוונה למערכת חכמה יותר שתדע לבד שלכל מאכל מטוגן, למשל, יש צורך בשמן. אולי גם תציין אלטרנטיבות (כלליות) כשחסרים חומרים (מרגרינה במקום שמן או פירורי לחם במקום קמח וכד').  
בנוסף, אם המאכל לא יצא טעים מסיבה כלשהי (למשל תפל מדי) על המערכת לדעת לשפר את המתכון לקראת הכנתו בפעם הבאה (במקרה זה להוסיף תבלין כלשהו ו/או מלח).

## ז. הנחיות לגבי הגשת הפרוייקט

- (1) ב-5.11.2021 תתבצע הצגה ובדיקה אינטראקטיבית של פרוייקט הגמר בבניין האוניברסיטה הפתוחה (יש לקבוע עם המנחה טלפונית את השעה המדויקת).
- (2) על כל סטודנט (או זוג) להביא לבדיקת הפרוייקט:
  - (i) קובץ עם קוד הפרוייקט והקומפיילר בו נעשה שימוש.
  - (ii) תדפיס של התכנית.
  - (iii) יש להכין מראש דוגמת קלט אשר תציג את הפרוייקט בצורה מתומצתת וממצה.
  - (iv) טופס מלווה למטלה.
- (3) יש לדאוג לכך שכל החומר הקשור בפרוייקט יוגש בצורה מסודרת.
- (4) ציון הפרוייקט לא ינתן בתום הצגתו, אלא לאחר בדיקה נוספת ומקיפה ע"י המנחה.

**בהצלחה!**