

חלק יבש

לפניכם מספר שאלות, חלקן על דברים שראינו בכתה וחלקן לא. כתבו את הפתרונות אליהן בקובץ **dry.pdf**.

שימו לב - שאלה 3 היא שאלת בונוס ללא ניקוד. השאלה תיבדק ותקבלו עליה משוב, אך היא לא מהווה מרכיב בציון התרגיל. השאלה נועדה רק על מנת שתוכלו לתרגל את החומר של פרק 7 שהתחלנו בהרצאה 11 (22.6.23) ונמשיך בהרצאה 12. מומלץ לפתור אותה רק בתום הרצאה מספר 12 (29.6.23).

שאלה 1

בהרצאה ראינו שעבור טיפוסים פולימורפיים מסויימים יש ערכים קונקרטיים השייכים לטיפוס. למשל, ראינו שהרשימה הריקה היא הערך היחיד מהטיפוס הפולימורפי $list(\tau)$. עבור כל אחד מהטיפוסים הבאים, תנו דוגמה אחת בשפת ML לערך מהטיפוס (אם קיים כזה). הניחו שניתן להשוות ערכים מסוג τ .

שימו לב - אין לתת כדוגמה פונקציות שעלולות לזרוק חריגות.

1. $\tau \rightarrow (\tau \times \tau)$
2. $(\tau \times \tau) \rightarrow \tau$
3. $\tau \rightarrow bool$
4. $bool \rightarrow \tau$
5. $list(\tau) \rightarrow bool$
6. $bool \rightarrow list(\tau)$
7. $\tau \rightarrow (bool \rightarrow \tau)$
8. $(\tau \times \tau) \rightarrow bool$
9. $\tau \rightarrow \tau \rightarrow bool$
10. $\tau \rightarrow \sigma \rightarrow bool$

שאלה 2

בשאלה זו נעסוק בחתכים (cuts) ב-Prolog.

1. מה ההבדל בין חתך ירוק לחתך אדום?

2. לכל אחד מהחתכים בפרדיקטים הבאים, הכריעו האם החתך אדום או ירוק.

עבור כל חתך ירוק, הסבירו את המוטיבציה להוסיף אותו לפרדיקט.

עבור חתך אדום, תנו דוגמה לשאילתה עבורה הפרדיקט יתן תוצאה שונה אם מסירים את החתך.

a. ענו עבור הפרדיקט `sign/2`. ניתן להניח שבשאילתה על הפרדיקט, הפרמטר

הראשון קשור והשני אינו קשור:

`sign(X, negative) :- X < 0, !.`

`sign(X, zero) :- X = 0, !.`

`sign(X, positive).`

b. ענו עבור הפרדיקט `member/2`:

`member(E, [E| Xs]) :- !.`

`member(E, [X| Xs]) :- member(E, Xs).`

c. ענו עבור הפרדיקט `member/2` אולם הפעם הניחו ששני הפרמטרים שמקבל הינם קשורים.

שאלה 3 (בנוס ללא ביקוד)

סִירְנָדָה היא הרחבה דמיונית של שפת C המאפשרת לציין באמצעות מילת המפתח `eager` או `lazy` לצד הארגומנט הפורמלי האם הוא מועבר לפונקציה באופן "להוט" (`eager`) או באופן "עצל" (`evaluation order`) או באופן "עצל" (`lazy evaluation order`), בהתאמה. בשאלה זו נציג בפניכם מספר מימושים אפשריים לפונקציה `repeat`, המספקת פונקציונליות דומה למנגנון `repeat-until` שראינו בשפת פסקל. נשתמש בפונקציה `repeat` כדי לממש פונקציית `gcd`:

```
int gcd(eager int m, eager int n) {  
    int m_;  
    return repeat ((m_=m,m=n,n=m_%n,m), n==0);  
}
```

בכל סעיף, הכריעו האם המימוש של `repeat` תקין או לא. במידה והמימוש לא תקין, ציינו מה עלול לקרות בחישוב של `gcd(m,n)` מבין האפשרויות הבאות:

- החזרת ערך שגוי.
- שגיאת זמן ריצה (חלוקה באפס).
- רקורסיה אינסופית.

1.

```
int repeat(lazy int exp, lazy bool until) {  
    return until ? exp : repeat(exp, until);  
}
```
2.

```
int repeat(lazy int exp, lazy bool until) {  
    int e = exp;  
    return until ? e : repeat(e, until);  
}
```
3.

```
int repeat(lazy int exp, lazy bool until) {  
    int e = exp;  
    return until ? e : repeat(exp, until);  
}
```

```
4. int repeat(lazy int exp, lazy bool until) {  
    int e = exp;  
    return until ? exp : repeat(exp, until);  
}
```

```
5. int repeat(lazy int exp, lazy bool until) {  
    int e = exp;  
    bool u = until;  
    return until ? exp : repeat(exp, u);  
}
```

חלק רטוב

- קבצים המסופקים לחלק זה תוכלו למצוא בגיטהאב של הקורס - [Homework6](#).
- פתרון לכל אחת מהשאלות יש לכתוב בקובץ ששמו [hw6_q1.pl](#) עבור שאלה [i](#).
- לכל שאלה מסופקת לכם דוגמת קלט ופלט רצוי. לצורך העניין את הפתרון שלכם לשאלה 1 תוכלו לבדוק בעזרת הפקודות הבאות:

```
swipl -q hw6_q1.pl < q1.in > q1.out
diff -wB <(sed '1,/===TEST START===/{N;d}' q1.out)
q1.expected
```

- יש לבדוק ולהריץ את פתרונותיכם לתרגיל זה בדוקר **twyair/safot-hw:5**
- [מדריך התקנה ושימוש בדוקר](#)

שאלה 1

בשאלה זו נכתוב פרדיקטים המבצעים חישובים על מספרים שלמים המיוצגים לפי מודל המספרים הטבעיים של פאנו.

הערות:

- לאורך כל השאלה אין להשתמש באופרטורים של חישוב מספרים שלמים הבנויים בשפה.
- בשאלה זו לא יבדקו שאליות המסתפקות עבור יותר מערך אחד.

מספר טבעי במערכת מוגדר באופן הבא:

- 0 מיוצג על ידי האטום **bot**.
- המספר $n+1$ מיוצג על ידי הפנקטור **s** כאשר הפרמטר שלו הוא הייצוג של n .

לדוגמא:

- 0: **bot**
- 1: **s(bot)**
- 2: **s(s(bot))**

וכו'.

1. הגדירו את הפרדיקט $\text{add}/3$ המתקיים כאשר הפרמטר השלישי הוא הסכום של שני הפרמטרים הראשונים.
ניתן להניח ששניים מהפרמטרים קשורים.
לדוגמא:

```
?- add(R,s(s(bot)),s(s(s(s(s(bot)))))).  
R = s(s(s(bot))).
```

2. הגדירו את הפרדיקט $\text{multiply}/3$ המתקיים כאשר הפרמטר השלישי הוא המכפלה של שני הפרמטרים הראשונים.
ניתן להניח ששניים מהפרמטרים קשורים.
לדוגמא:

```
?- multiply(s(s(bot)),s(bot),Res).  
Res = s(s(bot)).
```

3. הגדירו את הפרדיקט $\text{power}/3$ המתקיים כאשר הפרמטר השלישי מתקבל על ידי העלאה של הפרמטר הראשון בחזקת הפרמטר השני.
ניתן להניח ששני הפרמטרים הראשונים קשורים.
לדוגמא:

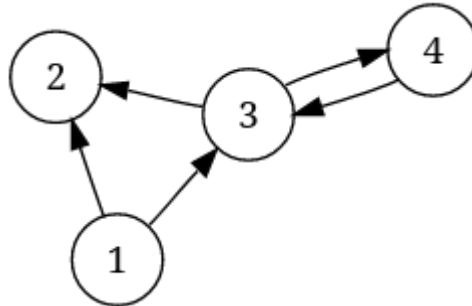
```
power(s(s(s(bot))),s(s(bot)),Res).  
Res = s(s(s(s(s(s(s(s(s(bot))))))))).
```

הערה - הפתרון לחישוב 0^0 אינו מוגדר, לכן לא ייבדק.

שאלה 2

לאחר שהשימוש בתכנות לוגי הביא אותו לפריצת דרך במחקר שלו, החליט המתרגל האחראי בקורס שפות תכנות להמשיך להיעזר בסטודנטים לטובת המחקר (מבוסס על סיפור לא אמיתי).

בשאלה זו נשתמש בייצוג של גרפים שראינו ב**[תרגיל בית 5](#)**.



הערות:

- בכל הפרדיקטים שתממשו בשאלה, ניתן להניח שהגרף המתקבל הינו קשור וחוקי והפרמטרים האחרים אינם קשורים.
 - לצורך הפתרון, תוכלו להשתמש בפרדיקטים **legal_graph**, **edge**, **path**, **circle** ו-**dag** שהגדרתם בתרגיל הקודם.
 - במידה ותרצו להשתמש בפרדיקטים אלו, ניתן לייבא את הפתרון שלכם מתרגיל בית 5 על ידי הוספת השורה הבאה בתחילת התכנית (כאשר הקובץ hw5_q2.pl נמצא באותה תיקיה):
- :- [hw5_q2].
- אין צורך להגיש מחדש את הקבצים של תרגיל הבית הקודם. הבדיקה האוטומטית תשתמש בפתרון רשמי של תרגיל בית 5.

1. [מיון טופולוגי](#) הוא סידור של כל צמתי הגרף, כך שלכל $j < i$ לא קיים מסלול מהצומת ה- j בסידור אל הצומת ה- i .
הגדירו את הפרדיקט `topological_sort/2` כאשר הפרמטר הראשון הינו גרף והשני הוא מיון טופולוגי של צמתי הגרף (המיוצג כרשימה של צמתים).
לדוגמה:

```
?- topological_sort(graph([[2, 3], [], [2, 4], [3]]), S).  
false.
```

```
?- topological_sort(graph([[2, 3], [], [2, 4], []]), S).  
S = [1, 3, 2, 4] ;  
S = [1, 3, 4, 2].
```

2. [רכיב קשיר היטב](#) בגרף הוא אוסף צמתים המקיים בו לכל זוג צמתים u ו- v , קיים מסלול מ- v ל- u ומסלול מ- u ל- v . בנוסף הרכיב הינו מקסימלי - הוספה של כל צומת אחר שאינו ברכיב תפגום בהיותו רכיב קשיר היטב.
הגדירו את הפרדיקט `scc/2` המתקיים כאשר הפרמטר הראשון הוא גרף והפרמטר השני הוא רשימה של כל הרכיבים הקשירים היטב בגרף (כל אחד מהרכיבים הוא רשימת הצמתים השייכים לו).
לדוגמה:

```
?- scc(graph([[2, 3], [], [2, 4], [3]]), S).  
S = [[1], [2], [3, 4]].
```

הערות:

- אין חשיבות לסדר של הרכיבים או לסדר של הצמתים בכל רכיב.
- עבור שאילתות עם הפרדיקט `scc`, תיבדק רק התוצאה הראשונה של הפרדיקט.

הנחיות

- את הפתרון לתרגיל הבית יש להגיש בקובץ zip המכיל את הקבצים הבאים בלבד:
hw6_q1.pl, hw6_q2.pl, dry.pdf.
- על החלק היבש להיות מוקלד. אין להגיש סריקה או צילום של התשובות לחלק זה.
- שם קובץ ההגשה יהיה EX6_ID1_ID2.zip כאשר ID1, ID2 הם מספרי ת.ז. של המגישים.
- מסופק לכם סקריפט לבדיקת עצמית של קובץ הגשה בשם **selfcheck.sh**. אנא בדקו את תקינות ההגשה שלכם בעזרת הסקריפט בסביבת ה-docker.
- בודקי התרגילים **מאוד** אוהבים Memes. שתפו את תחושותיכם במהלך פתירת התרגיל באמצעות Memes מתאימים ב-pdf של החלק היבש. Memes מצחיקים בצורה יוצאת דופן יזכו את היוצרים בבונוס.

בהצלחה!