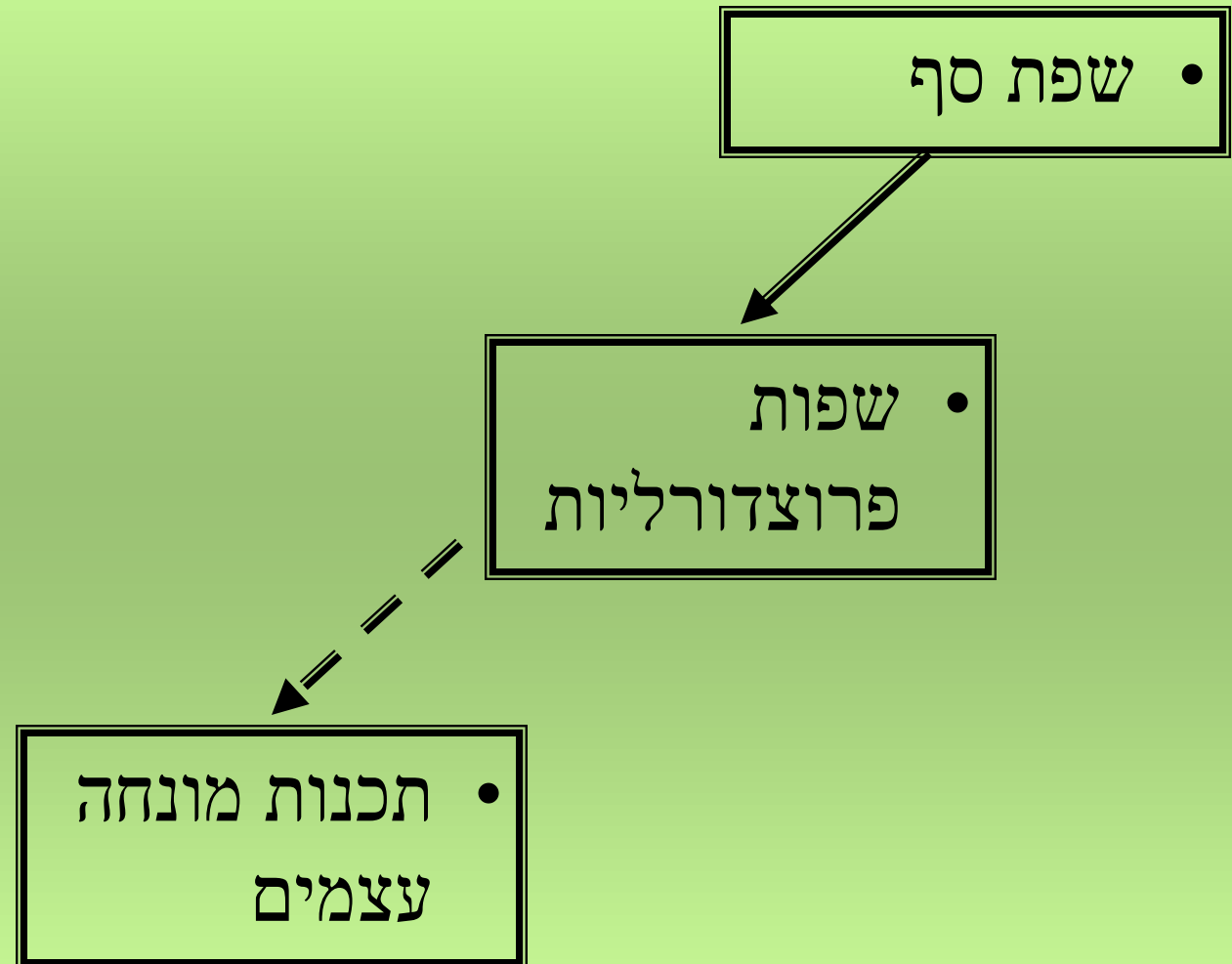


# מהו תכנות לוגי

צעדים ראשונים

# גישות מקובלות בעולם התוכנה



PASCAL :

Read(Num);

```
push offset insertnum  
call displayMsg  
xor bx,bx
```

AGAIN:

```
call ReadC  
cmp al,13  
je OUTSIDE  
cmp al,'9'  
jg WRONGINPUT  
cmp al,'0'  
jl WRONGINPUT
```

```
xor ah,ah  
push ax  
call charToInt  
push ax  
mov ax,bx  
mov cx,TEN  
mul cx  
mov bx,ax  
cmp dx,0  
jne TOBIG  
pop ax  
add bx,ax  
jmp AGAIN
```

WRONGINPUT:

```
push offset err1  
call displayMsg  
xor bx,bx  
jmp AGAIN
```

TOBIG:

# היסק לוגי :

- מסקנה הנובעת מצירוף חוקים ועובדות .

- לדוגמה :

— כשיורד עליך גשם אתה נרטב (חוק).

— יורד גשם מעל נקודה X (עובדה).

— אתה נמצא בנקודה X (עובדה).

— מסקנה : אתה נרטב !!!

- עוד דוגמאות ...

## עוד דוגמא :

- מי שמעתיק בממנים ונתפס נענש קשות (חוק).
- מי שמעתיק ממנים ולא נתפס, לא עובר את המבחן (חוק).

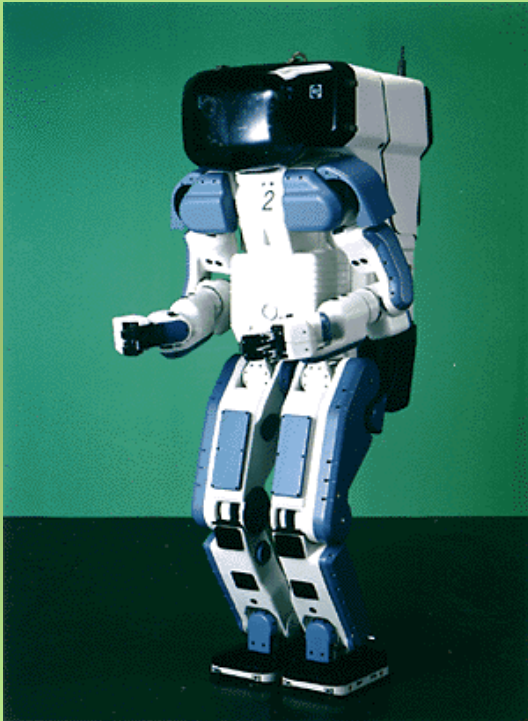
## • מסקנה :

העתקת בממנים , אז אחד מהשתיים או שנענשת קשות או שלא עברת את המבחן .

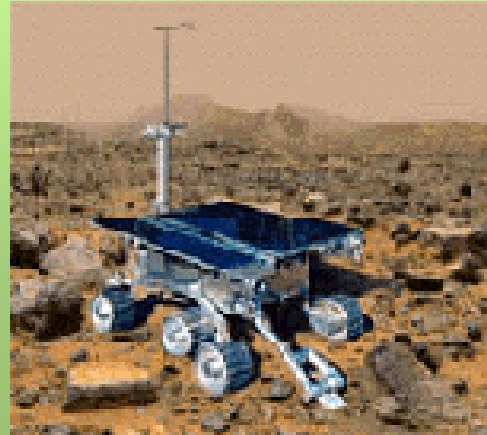
# למה ללמוד תכנות לוגי ?

- מדעי המחשב כמדע .
- החסרונות בתכנות פרוצדורלי .
- שימושים בתכנות לוגי בתעשייה .

# שימושים



Labor



Science



Search engines

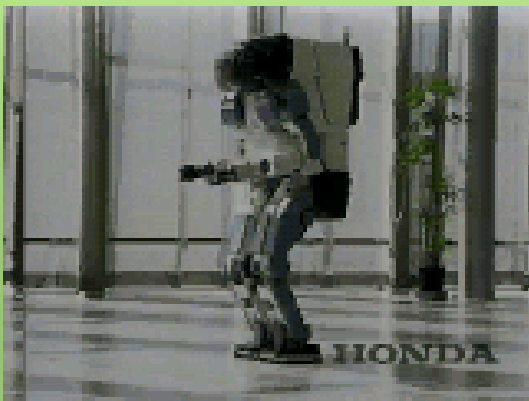


Medicine/  
Diagnosis



Appliances

# Honda Humanoid Robot



Walk



Turn

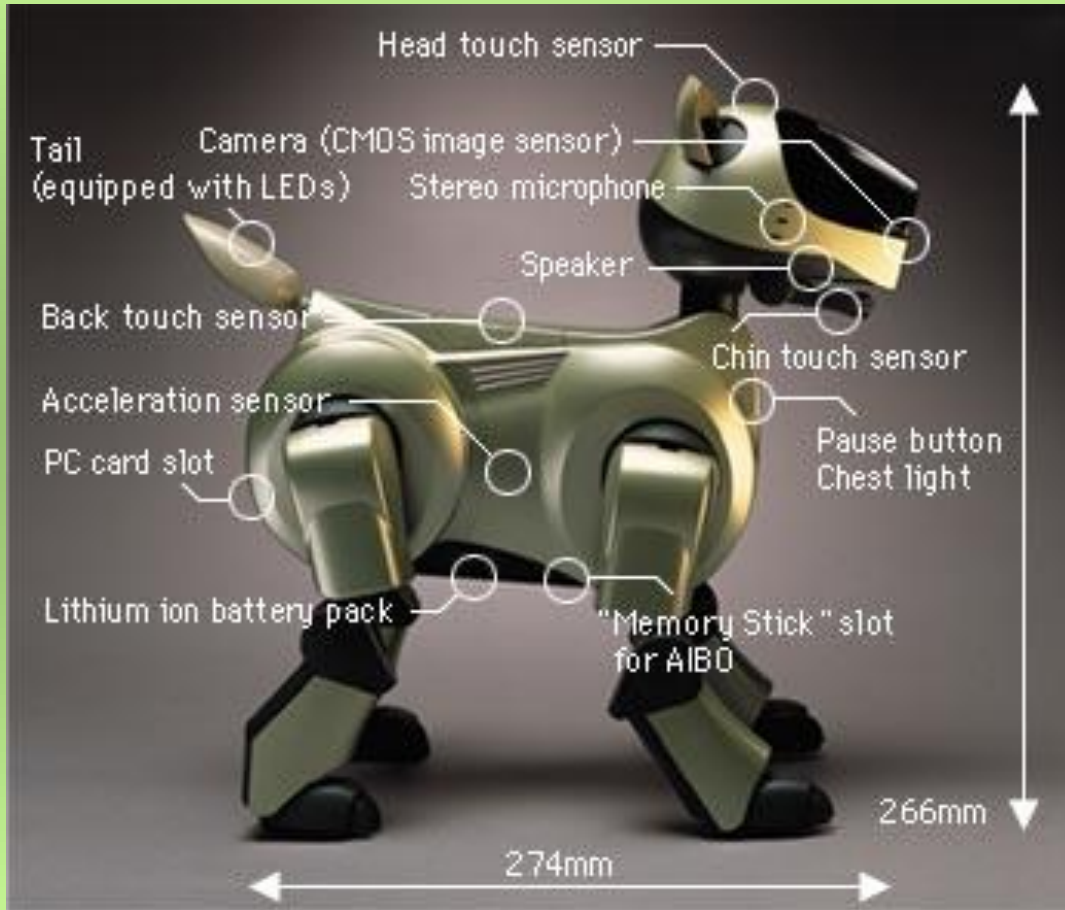


Stairs

<http://world.honda.com/robot/>



# AIBO



<http://www.aibo.com>

ש. עזר ראשון - הכרות עם תכנות לוגי

11/11/2011

# Question Answering



<http://aimovie.warnerbros.com>

<http://www.ai.mit.edu/projects/infolab/>

# דוגמא ראשונה :

## A biblical family database

---

father(terach,abraham).

father(terach,nachor).

father(terach,haran).

father(abraham,isaac).

father(haran,lot).

father(haran,milcah).

father(haran,yiscah).

mother(sarah,isaac).

male(terach).

male(abraham).

male(nachor).

male(haran).

male(isaac).

male(lot).

female(sarah).

female(milcah).

female(yiscah).

?- father(X,abraham).

Prolog will answer :

X = terach

?- father(X,Y).

Prolog will answer :

X=terach,Y=abraham ;

X=terach,Y=nachor ;

X=terach,Y=haran ;

X=abraham, Y=isaac ;

X=haran,Y=lot ;

X=haran,Y=milcah ;

X=haran,Y=yiscah ;

NO

woman(mia).  
woman(jody ).  
woman(yolanda ).  
playsGuitar(jody ).

?- woman(mia ).  
Prolog will answer  
yes

?- playsGuitar(jody).  
Prolog will again answer  
yes

?-playsGuitar(mia).  
We will get the answer  
  
no

?- tattooed(jody).  
no

?- woman(X).

Prolog will again answer

X = mia

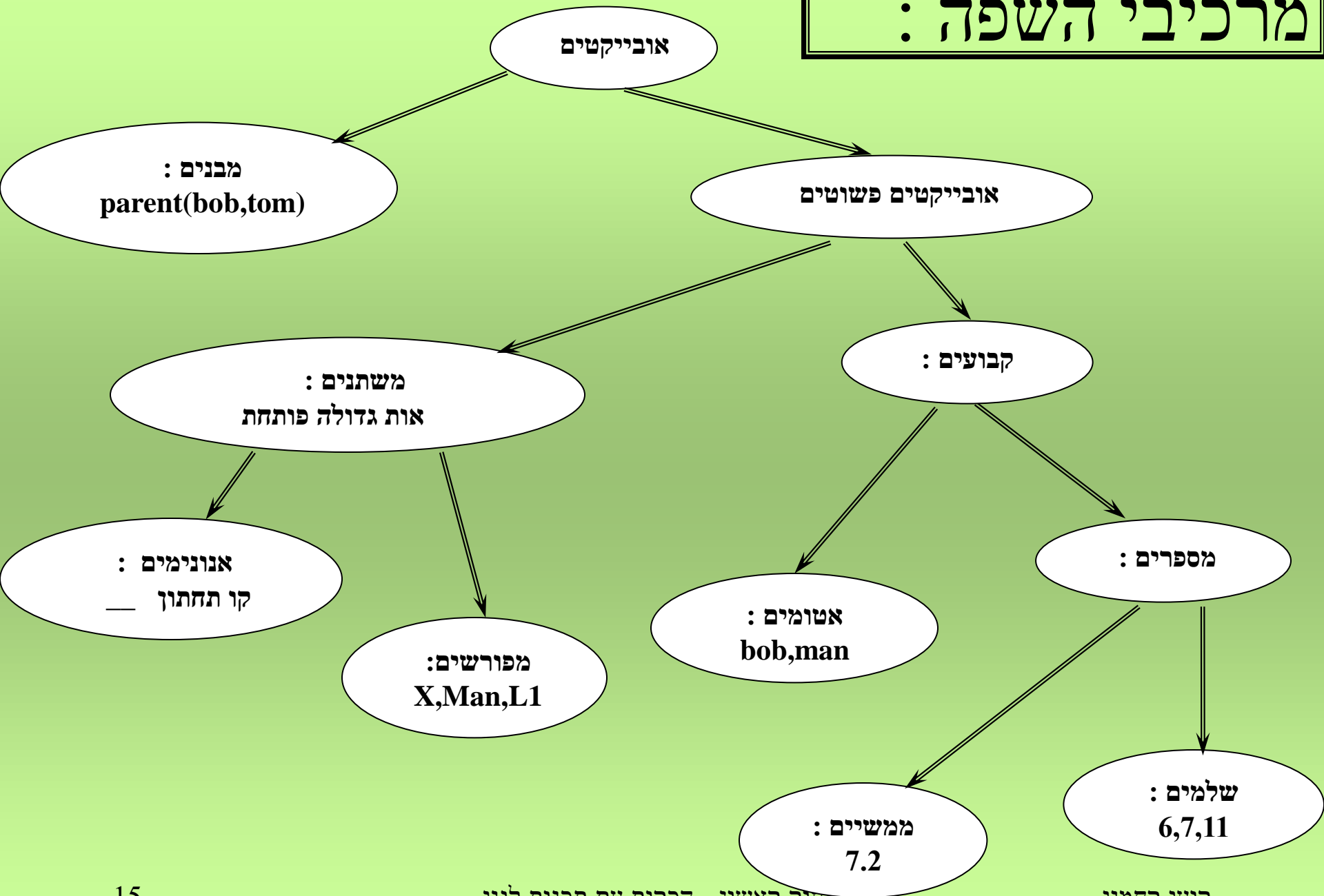
and if we try to get another result (by type spacebar) we get

X = jody

and one more time:

X = yolanda

# מרכיבי השפה :



- **אטומים** – מחרוזת שמתחילה באות קטנה יזוהו באופן יחידני ע"י ערך המחרוזת שלהם.
- **משתנים** - מתחילים באות גדולה או בקו תחתי .
- **משתנה אנונימי** – יתחיל בקו תחתי וישמש לסימון מקום בהצהרה שבו הערך לא משמעותי .

$\text{happy}(X):-\text{parent}(X,Y) . == \text{happy}(X):- \text{parent}(X,_) .$

-ערך המשתנה השני לא משנה בכלל לכן נסמן אותו בתור אנונימי .

-אם משתנה הוא אנונימי ערכו לא יופיע בתשובה.

-משתנים הם אובייקטים כללים ולא לכתובות בזיכרון.

- **מבנה** - אובייקט המכיל מספר רכיבים (משתנים קבועים או יחסים ומבנים) .



# חוקים :

```
parent(terach,bob) .  
parent(tom,bob) .  
parent(tom,liz) .  
parent(liz,roy) .  
parent(bob,ann) .  
parent(bob,pat) .  
parent(pat,jim) .
```

```
male(bob) .  
male(tom) .  
male(jim) .
```

```
female(terach) .  
female(liz) .  
female(ann) .  
female(pat) .
```

בהינתן בסיס העובדות הבא :

## חוקים :

אם נגדיר את החוק :

```
mother(X,Y) :-  
    female(X),parent(X,Y) .
```

כשנשאל :

```
?- mother(X,liz).  
no
```

```
?- mother(X,roy).  
X = liz
```

הגדר יחס המזהה אב במשפחה :

```
father(X,Y) :-  
    male(X) , parent(X,Y) .
```

הגדר יחס המזהה סב וסבתא במשפחה :

```
grandfather(X,Y) :-  
    male(X) , parent(X,Z) , parent(Z,Y) .  
grandmother(X,Y) :-  
    female(X) , parent(X,Z) , parent(Z,Y) .
```

הגדר יחס המזהה אח ואחות במשפחה :

```
brother(X,Y) :-
```

```
    male(X) , parent(Z,X) , parent(Z,Y) , X\=Y.
```

```
sister(X,Y) :-
```

```
    female(X) , parent(Z,X) , parent(Z,Y) , X\=Y.
```

חוקים המשתמשים בחוקים אחרים :

הגדר יחס המזהה דוד במשפחה :

```
uncle(X,Y) :- brother(X,Z) , parent(Z,Y) .
```

## הגדר יחס המזהה בין דוד במשפחה :

```
cousin(X,Y) :-  
    parent(P1,X) , parent(P2,Y) , parent(Z,P1) ,  
    parent(Z,P2) , P1 \= P2 .
```

או בדרך אחרת :

```
cousin(X,Y) :- parent(P1,X) , parent(P2,Y) ,  
    (brother(P1,P2)  
    ;  
    sister(P1,P2)) .
```

הצע דרך נוספת ...

## הגדר יחס המזהה צאצא במשפחה :

```
successor (X,Y) :-parent (Y,X) .  
successor (X,Y) :-parent (Y,Z) , successor (X,Z) .
```

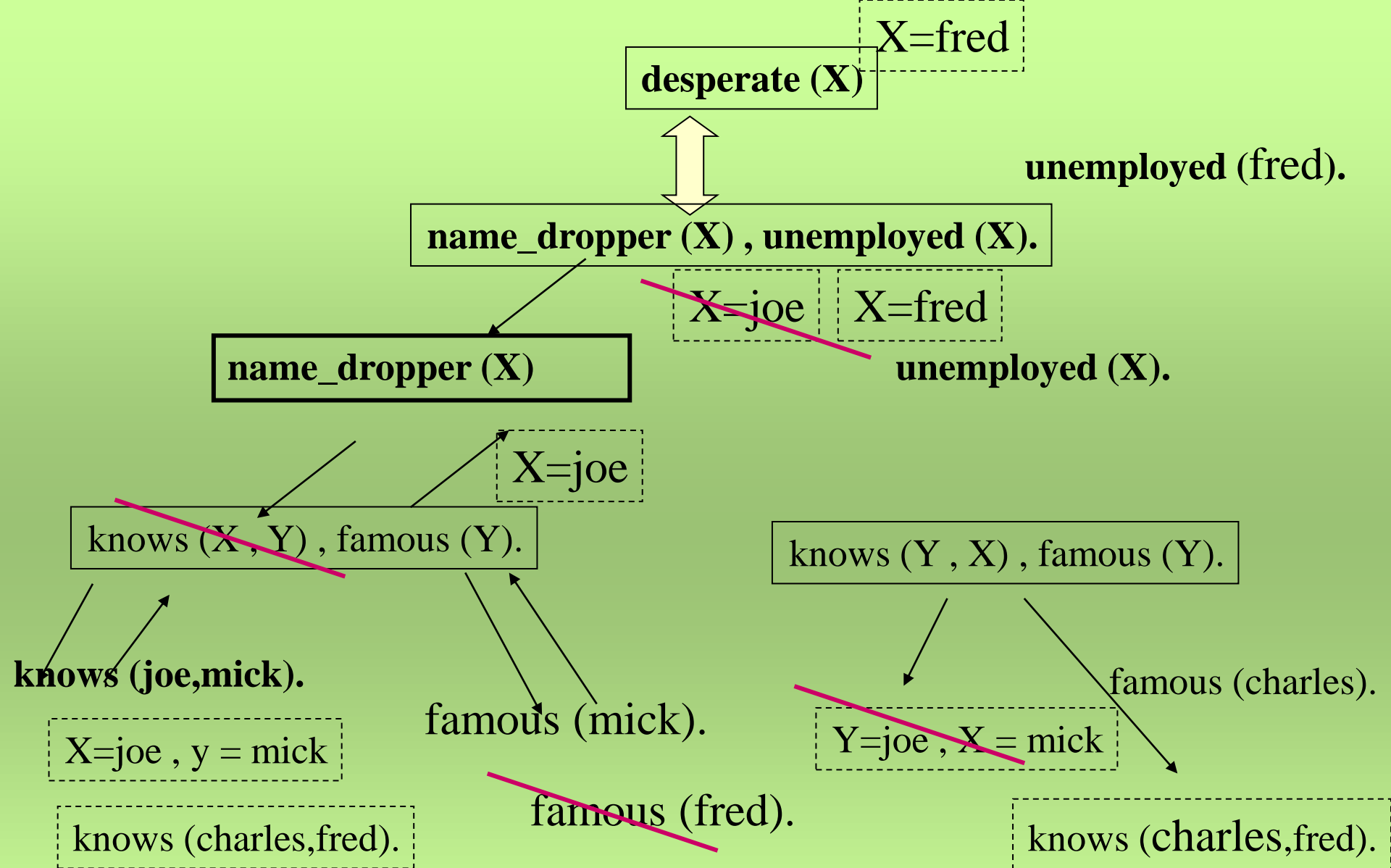
או בצורת כתיבה אחרת :

```
successor (X,Y) :-parent (Y,X)  
;  
parent (Y,Z) , successor (X,Z) .
```

```
desperate(X) :- name_dropper (X) , unemployed (X) .
name_dropper(X) :- knows (X , Y) , famous (Y) .
name_dropper(X) :- knows (Y , X) , famous (Y) .
knows (joe,mick) .
knows (charles,fred) .
famous (mick) .
famous (charles) .
unemployed (mick) .
unemployed (fred) .
```

מה תהיה התשובה לשאלתה:

```
?- desperate (Who) .
```





```
intst(1).  
intst(X):-  
    intst(X1),  
    X is X1+1.
```

# תרגול נוסף :

ניתן לתאר אנשים העומדים בתור זה אחרי זה בעזרת אוסף עובדות מהסוג:

```
stands_right_behind(Person1, Person2)
```

עובדה כזו היא נכונה אם Person1 עומד בתור מידית אחרי Person2.

הגדר פרדיקט

```
stands_behind( Person1, Person2)
```

שיהיה נכון אם Person1 עומד בתור אחרי Person2 (לאו דווקא מידית אחריו).

למשל, בהינתן העובדות:

```
stands_right_behind ( ran, tami).
```

```
stands_right_behind ( tami, gal).
```

```
stands_right_behind ( gal, ziv).
```

ניתן יהיה להסיק בעזרת הפרדיקט stands\_behind ש- ran עומד בתור אחרי gal או אחרי ziv.

## השתמש בפרדיקט

`stands_in_front_of` (שהגדרת בסעיף א') לצורך הגדרת פרדיקט `stands_behind`  
(Person1, Person2)

שיהיה נכון אם Person1 עומד בתור לפני Person2.

למשל:

```
?- stands_in_front_of ( ziv, tami).
```

```
yes
```

```
?- stands_in_front_of ( ran, ziv).
```

```
no
```

השתמש בפרדיקטים שהגדרת לעיל כדי להגדיר יחס המבטא אפשרות להעביר חפץ כלשהו מ-  
Person1 ל- Person2.

(ההעברה אפשרית אם ורק אם הם עומדים באותו תור או ש- (Person1= Person2).  
למשל:

?- can\_pass ( tami, ziv).

yes

?- can\_pass ( tami, yaron).

no

פתרון לדוגמא :

- עליכם לכתוב תוכנית העוזרת לגדי, לרינה וללירן לרכוש כלב עפ"י הנתונים הבאים.
- לאסי היא כלבה ענקית ורגועה בצבע בז'.
  - מקס הוא כלב בגודל בינוני בצבע אפור, בעל אופי אדיש.
  - ברנדי הוא כלב גדול ותוקפן שצבעו לבן.
  - ליידי היא כלבה קטנה ושחורה בעלת אופי תוקפני.
  - ידוע שהצבעים חום ובז' הם דומים.
  - גדי מעוניין בכלב זכר בגודל ענק, או לחילופין בכלב (או כלבה) גדול ותוקפן.
  - רינה מעונינת בכלב (או כלבה) קטן, או לחילופין בכלב (או כלבה) שאינו קטן ואינו תוקפן. בכל מקרה חשוב לרינה שצבע הכלב יהיה חום או צבע דומה לחום.
  - לירן מעוניין בכלב שאינו קטן.
  - יש לבחור לכל אחד כלב **שונה**.

תאר את כל המידע ע"י חוקים ועובדות בשפת פרולוג.

הגדר פרדיקט `choose3dogs(GadiDog, RinaDog, LiranDog)` שבוחר שלושה כלבים שונים לגדי, רינה ולירן.

```
dog(lasi,female,bez,xl,relax).
dog(max,male,gray,m,cool).
dog(brandi,male,white,l,killer).
dog(lady,female,black,s,killer).
```

```
sameColor(bez,brown).
```

```
choose3dogs(GadiDog,RinaDog,LiranDog):-
    ( dog(GadiDog,male,_,xl,_)
      ;
      dog(GadiDog,_,_,l,killer)),
    ((sameColor(RinaDogColor,_,_);sameColor(_,RinaDogColor)),
     (dog(RinaDog,_,RinaDogColor,s,_)
      ;
      dog(RinaDog,_,RinaDogColor,RinaDogSize,Tem),RinaDogSize\=s,Tem\=killer)),
    dog(LiranDog,_,_,LiranDogSize,_,_),LiranDogSize\=s,
    GadiDog\=RinaDog,LiranDog\=RinaDog,GadiDog\=LiranDog.
```

## שליפת נתונים :

- נתון הביטוי המורכב:
  - `movie(Name, Type, Serial_Num, Length, customer(Id_Num, name(Fname, Sname)), borrow_date(Day, Month, Year), Period)`
- המגדיר מידע לגבי סרט הניתן להשאלה בספריית וידאו. שם הסרט Name והוא מסוג Type (מתח, דרמה וכו'), מספרו הסידורי Serial\_Num ואורכו Length דקות.
- הסרט הושאל למנוי (ששמו הפרטי Sname, שם משפחתו Fname, ומספר זהותו Id\_Num), בתאריך מסוים, לתקופה של Period ימים.

# שליפת נתונים :

כתוב בעזרת הביטוי שלעיל חוקים בפרולוג בעזרתם ניתן לשלוף את המידע הבא:

1. שמות כל המנויים ששאלו סרט כלשהו בתאריך מסוים.
2. שמות כל הסרטים מסוג נתון שהושאלו אי-פעם.
3. האם סרט מסוים מושאל כרגע (יש לספק תאריך).
4. שמות כל המנויים ששאלו אי-פעם סרט.
5. שמות כל הסרטים הנמצאים כרגע אצל מנוי בעל מספר זהות מסוים.
6. באיזה תאריך יוחזר סרט מסוים המושאל כרגע.
7. שמות כל הסרטים מסוג נתון שיוחזרו בתאריך מסוים לספריה ואורך כל אחד מהם.
8. סוגי הסרטים שהושאלו בתאריך מסוים בספריה.

הנח כי סרט מוחזר בפועל ביום האחרון להשאלתו



# פיתרון

נגדיר פרדיקטים לטיפול בתאריכים:

(נניח אורך חודש קבוע, 30 ימים)

הוספת ימים לתאריך מסוים לקבלת תאריך חדש:

$\text{add}(\text{Days}, \text{date}(D, M, Y), \text{date}(D1, M1, Y1)):-$

$D1 \text{ is } (D + \text{Days}) \bmod 30,$

$M1 \text{ is } (M + ((D + \text{Days}) // 30)) \bmod 12,$

$Y1 \text{ is } Y + (M + ((D + \text{Days}) // 30)) // 12.$

בדיקה איזה תאריך מוקדם יותר:

$\text{earlier}(\text{date}(\_, \_, Y), \text{date}(\_, \_, Y1)):- Y < Y1.$

$\text{earlier}(\text{date}(\_, M, Y), \text{date}(\_, M1, Y)):- M < M1.$

$\text{earlier}(\text{date}(D, M, Y), \text{date}(D1, M, Y)):- D < D1.$

# פיתרון

1. names1(Name,date(D,M,Y)):-  
    movie(\_\_\_\_,customer(\_\_\_\_,Name),borrow\_date(D,M,Y),\_\_\_\_).
2. names\_of\_movies1(Movie,Type):-  
    movie(Movie,Type,\_\_\_\_,\_\_\_\_,\_\_\_\_,\_\_\_\_).
3. is\_borrowd(Movie,CurrentDate):-  
    movie(Movie,\_\_\_\_,\_\_\_\_,\_\_\_\_,borrow\_date(D,M,Y),P),  
    add(P,date(D,M,Y),ReturnDate),  
    earlier(CurrentDate, ReturnDate).
4. names2(Name,Movie):-  
    movie(Movie,\_\_\_\_,\_\_\_\_,customer(\_\_\_\_,Name),\_\_\_\_,\_\_\_\_).

# פיתרון

5. name\_of\_movies2(Movie,IdNum,CurrentDate):-

```
movie(Movie,_,_,_,customer(IdNum,_),borrow_date(D,M,Y),P),  
,  
    add(P,date(D,M,Y),ReturnDate),  
    earlier(CurrentDate, ReturnDate).
```

לא השתמשנו כאן בפרדיקט is\_borrowd כי יתכן שיש כמה סרטים עם אותו שם, וזה שמושאל כרגע הוא לא זה שהיה אצל האדם המבוקש.

6. return\_date(Movie,CurrentDate,ReturnDate):-

```
movie(Movie,_,_,_,_,borrow_date(D,M,Y),P),  
add(P,date(D,M,Y),ReturnDate),  
earlier(date(D,M,Y),CurrentDate).
```

# פיתרון

7. name\_of\_movies3(Movie,Type,ReturnDate,Length):-

```
movie(Movie,Type,_,Length,_,borrow_date(D,M,Y),P),  
    add(P,date(D,M,Y),ReturnDate).
```

8. types(Type,date(D,M,Y)):-

```
movie(_,Type,_,_,_,borrow_date(D,M,Y),_).
```

# סיכום בקטנה

- תכנות לוגי מקביל ודומה לתכנות פונקציונאלי
- במהלך הקורס ננסה להתרגל לדרך מחשבה קצת שונה בהתמודדות עם פתרון בעיות.
- הגישה שונה אבל מתגמלת ותשמש כלי חשוב בארגז הכלים שלכם בכל מקרה .
- סמסטר מוצלח !!!!