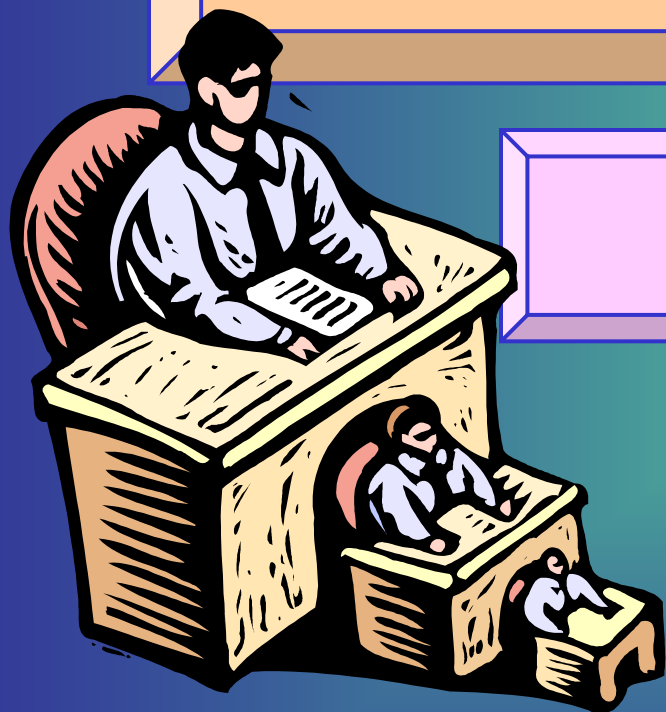


# רקורסיה מבעד למשקפי הפרדיגמה



רגוניס נוע

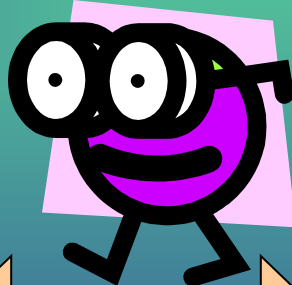
מכללת  
בית-ברל

מכון ויצמן  
למדע

הוראה  
בתיכון

# רקורסיה מבעד למשקפי הפרדיגמה

תיכנות  
פרוצדורלי



תיכנות  
לוגי

# רקורסיה היא הצהרתית בכל פרדיגמה

המבנה הבסיסי :

- מקרה פרטי (מקרה קצה)
- מקרה מורכב בשלב ה-  $N - 1$

המוגדר על ידי אותו מקרה  
בשלב ה-  $N - 1$  (מוקטן)

הוא הצהרתי

**תיכנות לוגי הוא  
תיכנות הצהרתי**

**רקורסיה היא  
הצהרתית  
בכל פרדיגמה**

- **לכן קל יותר להקנות את עקרונות הרקורסיה בפרדיגמה הצהרתית.**
- **המעבר לפרדיגמה הפרוצדורלית הוא בהיר וקל יותר.**

# סדר ההרצאה

תיכנות לוגי על קצה המזלג

רקורסיה טבעית בתיכנות לוגי

רקורסיה חשבונית בתיכנות לוגי

מעבר לרקורסיה חשבונית בתיכנות פרוצדורלי

רקורסיה ברשימות בתיכנות לוגי

רקורסיה ברשימות בתיכנות פרוצדורלי

# שפת תיכנות בלוגיקה

## מתאפיינת בדברים הבאים:

★ שפה הצהרתית - משתמשת בלוגיקה להצגת מידע במחשב. המידע מוצג כאוסף של הנחות – טענות בלוגיקה.

★ שפה דומה לשפה טבעית.

★ מנגנון ההיסק של השפה "יודע" להבחין בין טענות בלוגיקה שניתן להסיק אותן על פי חוקי הלוגיקה מתוך אוסף ההנחות לבין כאלה שלא ניתן להסיקן.

# דוגמא:

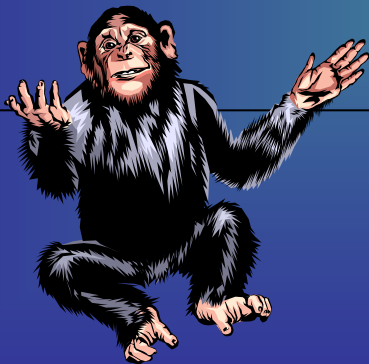
בעל\_חיים :- עוף.  
בעל\_חיים :- זוחל.  
בעל\_חיים :- יונק.  
יונק :- כלב.  
יונק :- קוף.  
כלב.  
קוף.  
נחש.

חוקים

עובדות

?- בעל\_חיים.

כ.



אוסף טענות  
בלוגיקה  
=  
בסיס ידע  
=  
תכנית פרולוג

שאלתה

תשובות  
מנגנון ההיסק

# דוגמא:

בעל\_חיים(X) :- עוף(X).

בעל\_חיים(X) :- זוחל(X).

בעל\_חיים(X) :- יונק(X).

יונק(X) :- כלב(X).

יונק(X) :- קוף(X).

כלב(בוני).

קוף(קינג\_קונג).

קוף(דודו).

?- בעל\_חיים(מי).

מי = בוני;

מי = קינג\_קונג;

מי = דודו;

לא

חוקים

עובדות



אוסף טענות

בלוגיקה

=

בסיס ידע

=

תכנית פרולוג

שאילתה

תשובות

מנגנון ההיסק



# מה בין שפת תיכנות פרוצדורלית לשפת תיכנות הצהרתית?

תיכנות הצהרתי

ראש חוק  
מתקיים אם  
מרכיבי אופ  
החוק  
מתקיימים

תכנית היא אוסף  
טענות שיכולות  
להשתתף בהיסק

תיכנות פרוצדורלי

אם ... חלוק ...  
קצו ...  
... אם ... אלי ...  
... עשה ...  
קרא -f ...

תכנית היא רצף  
של הוראות לביצוע

# תכנית פרוצדורלית מול תכנית בלוגיקה

## תכנית הצהרתית

### התכנית:

תיאור הצהרתי של הבעיה,  
אוסף טענות בלוגיקה –  
בסיס ידע.

### "הרצת" תכנית:

על ידי הצגת שאילתה.

### אופן פעולה:

על פי מנגנון היסק בלוגיקה.  
המנגנון בוחר את ההנחות  
שיכולות להשתתף בהיסק, כדי  
להוכיח את השאילתה.  
אין משמעות לסדר ההנחות.

## תכנית פרוצדורלית

### התכנית:

אלגוריתם המתאר איך לפתור  
את הבעיה.

### הרצת תכנית:

מתן הוראת ביצוע.

### אופן פעולה:

ביצוע סידרתי של ההוראות  
הכתובות בתכנית. כל משפטי  
התכנית מתבצעים על פי הסדר  
שמכתיב מראש האלגוריתם.

# דוגמא - משפחה

הורה (אברהם, יצחק).  
הורה (שרה, יצחק).  
הורה (יצחק, יעקב).  
הורה (יעקב, יוסף).

זכר (אברהם).  
נקבה (שרה).  
זכר (יצחק).  
זכר (יעקב).  
זכר (יוסף).

עובדות

אב (Y, X) :- הורה (Y, X), זכר (X).

אם (Y, X) :- הורה (Y, X), נקבה (X).

סבא (Y, X) :- אב (Z, X), הורה (Y, Z).

אחים (Y, X) :- הורה (X, Z), הורה (Y, Z),  $Y \neq X$ .

בני\_דודים (Y, X) :- הורה (X, 1X), הורה (Y, 1Y).

אחים (1Y, 1X).

חוקים



# דוגמא – אב קדמון רקורסיה טבעית !

% אב\_קדמון (Y, X) – אב קדמון של Y  
מדור כלשהו

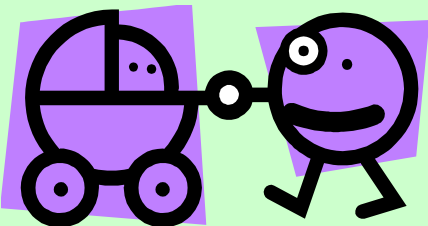
אב\_קדמון (Y, X) :-

הורה (Y, X).

אב\_קדמון (Y, X) :-

הורה (Z, X),

צאצא (Y, Z).



# עקרונות המתאר הרקורסיבי

## מתאר המוגדר בעזרת עצמו

המתאר מורכב משני חלקים:

- מקרה פרטי (מקרה קצה)
- מקרה מורכב בשלב ה-  $N$  –  $i$

המוגדר על ידי אותו מקרה

בשלב ה-  $N - 1$  (מוקטן)

✓ יכולים להיות מספר מקרים פרטיים

✓ יכולים להיות מספר מקרים מורכבים

✓ במקרה המורכב יכול להיות יותר משימוש

חוזר אחד בדגם "המוקטן"

# הדגמה גרפית של המתאר הרקורסיבי

אב קדמון

אב קדמון

בן של אב קדמון

בן של אב קדמון

אב קדמון

צאצא

צאצא

צאצא

דור 1

הורה

n-1 דורות

m דורות

$n=m+1$

n דורות



# דוגמא – שרשרת מזון

## רקורסיה טבעית !

עובדות

% אוכל  $(Y, X)$  –  $X$  אוכל מזון  $Y$   
אוכל (פרה, עשב). אוכל (אריה, פרה).

חוקים

% צרכן\_מזון  $(Y, X)$  –  $X$  צורך מזון  $Y$

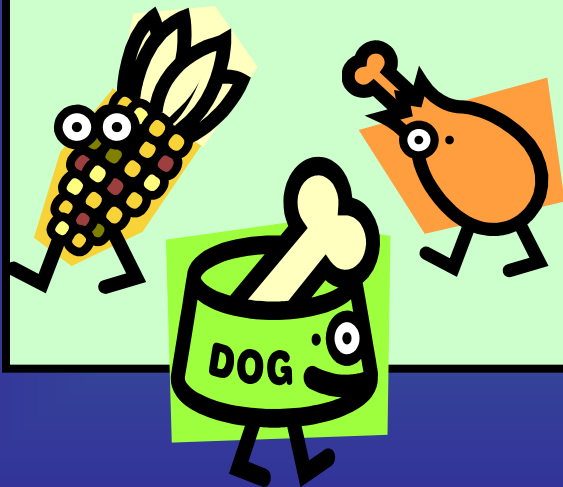
צרכן\_מזון  $(Y, X)$  :-

אוכל  $(Y, X)$ .

צרכן\_מזון  $(Y, X)$  :-

אוכל  $(Z, X)$ ,

צרכן\_מזון  $(Y, Z)$ .



# דוגמא – חפש את האפרוח

## רקורסיה טבעית !

% נמצאת\_בתוך ( \_ביצה\_קטנה, \_ביצה\_גדולה).  
נמצאת\_בתוך (כחולה, ירוקה).  
נמצאת\_בתוך (ירוקה, אדומה).  
נמצאת\_בתוך (אפרוח, כחולה).

עובדות

% עוטפת ( \_ביצה\_עוטפת, \_ביצה).  
עוטפת (Y, X) :- נמצאת\_בתוך (X, Y).  
עוטפת (Y, X) :- נמצאת\_בתוך (X, Z),  
עוטפת (Y, Z).

חוקים





# דוגמא – קווי נסיעה באוטובוס

## רקורסיה טבעית !

% קו\_נסיעה (מוצא, יעד) – קו ישיר  
קו\_נסיעה (מטולה, קירות\_שמונה).  
קו\_נסיעה (קרית\_שמונה, טבריה).

% ניתן\_להגיע (מוצא, יעד) - מסלול כלשהו

ניתן\_להגיע (מוצא, יעד) :-

קו\_נסיעה (מוצא, יעד).

ניתן\_להגיע (מוצא, יעד) :-

קו\_נסיעה (מוצא, ביניים),

ניתן\_להגיע (ביניים, יעד).



# דוגמא – מספר דורות

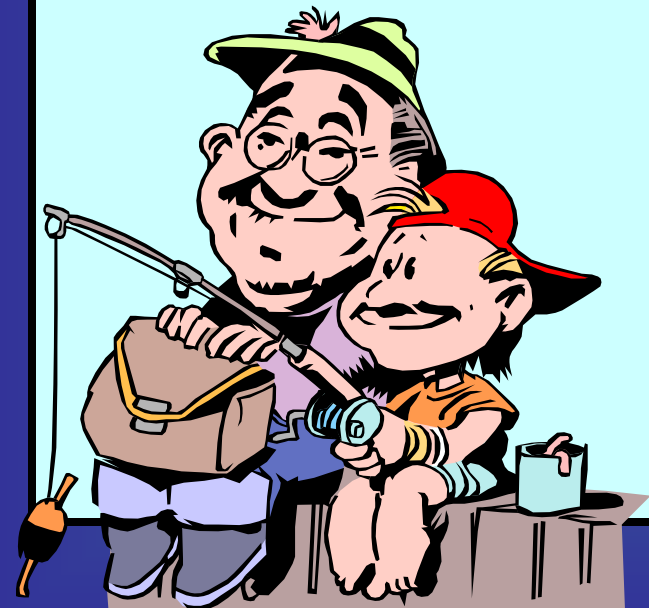
## רקורסיה חשבונית

% מספר\_דורות  $(N, Y, X)$  – אב קדמון של  $Y$   
 $N$  מספר הדורות ביניהם

מספר\_דורות  $(1, Y, X)$  :-  
הורה  $(Y, X)$ .

מספר\_דורות  $(N, Y, X)$  :-  
הורה  $(Z, X)$ ,

מספר\_דורות  $(M, Y, Z)$ ,  
 $N = M + 1$ .



# דוגמא – חפש את האפרוח

## רקורסיה חשבונית

% נמצאת בתוך ( \_ביצה\_ קטנה, \_ביצה\_ גדולה).  
נמצאת בתוך (כחולה, ירוקה).  
נמצאת בתוך (ירוקה, אדומה).  
נמצאת בתוך (אפרוח, כחולה).

% מספר ביצים ( \_ביצה\_ עוטפת, \_ביצה\_ , N).  
מספר ביצים ( X , Y , 2 ) :-  
נמצאת בתוך ( X , Y ).  
מספר ביצים ( N , Y , X ) :-  
נמצאת בתוך ( X , Z ),  
מספר ביצים ( M , Y , Z ),  
1 + M =: N



# דוגמא – קווי נסיעה באוטובוס

## רקורסיה חשבונית

% קו\_נסיעה (מוצא, יעד) – קו ישיר  
קו\_נסיעה (מטולה, קירית\_שמונה).  
קו\_נסיעה (קרית\_שמונה, טבריה).

% מספר\_קוים (מוצא, יעד, N) - מסלול במספר קוים  
מספר\_קוים (מוצא, יעד, 1) :-  
קו\_נסיעה (מוצא, יעד).  
מספר\_קוים (מוצא, יעד, N) :-  
קו\_נסיעה (מוצא, ביניים),  
מספר\_קוים (ביניים, יעד, M),  
1 + M =: N

# דוגמא – קווי נסיעה באוטובוס

## רקורסיה חשבונית - מחיר

% קו\_נסיעה ( $\_מוצא$ ,  $\_יעד$ ,  $\_מחיר$ ) – קו ישיר  
קו\_נסיעה (מטולה, קירית\_שמונה, 20).  
קו\_נסיעה (קרית\_שמונה, טבריה, 30).

% סכום\_נסיעה ( $\_מוצא$ ,  $\_יעד$ , S) - מסלול במספר קוים  
סכום\_נסיעה ( $\_מוצא$ ,  $\_יעד$ , S) :-  
קו\_נסיעה ( $\_מוצא$ ,  $\_יעד$ , S).  
סכום\_נסיעה ( $\_מוצא$ ,  $\_יעד$ , S) :-  
קו\_נסיעה ( $\_מוצא$ ,  $\_ביניים$ , 1S),  
סכום\_נסיעה ( $\_ביניים$ ,  $\_יעד$ , 2S),  
 $2S + 1S =: S$

# דוגמא – עצרת !

## רקורסיה חשבונית

% עצרת (F, N) – מקבל ערך N ומחזיר את N! ב-F.

$$0! = 1$$

עצרת (1, 0).

$$1! = 1$$

עצרת (1, 1).

$$N! = (N-1)! * N$$

עצרת (F, N) :-

$$, 1 < N$$

$$, N - 1 =: M$$

עצרת (F1, M),

$$.F1 * N =: F$$

# דוגמא – סידרת פיבונאצי

## רקורסיה חשבונית

% פיבונאצי (F, N) – מקבל ערך N ומחזיר את הערך של האיבר ה-N בסידרת פיבונאצי בתוך F.

0, 1, 1, 2, 3, 5, 8, 13, ..

בסידרת פיבונאצי

האיבר הראשון הוא – 0

האיבר השני הוא – 1

וכל איבר אחר הוא

סכום שני קודמיו

פיבונאצי (0, 1).

פיבונאצי (1, 2).

פיבונאצי (F, N) :-

$$N_1 = N - 1,$$

$$N_2 = N - 2,$$

פיבונאצי (F1, N1),

פיבונאצי (F2, N2),

$$F = F_1 + F_2.$$



# דוגמא – מספר טבעי רקורסיה חשבונית

% מספר\_טבעי (N) – 'כן' אם N טבעי 'לא' אחרת.

N טבעי אם

המספר הקטן ממנו ב-1  
הוא טבעי

מספר\_טבעי (1) .

מספר\_טבעי (N) :-  
,  $0 < N$

,  $N - 1 =: M$

מספר\_טבעי (M) .





# רקורסיה היא הצהרתית בכל

## פרדיגמה

$$0! = 1$$

$$1! = 1$$

$$N! = (N-1)! * N$$

N טבעי אם

המספר הקטן ממנו ב- 1

הוא טבעי

$$X^0 = 1$$

$$X^y = X^{y-1} * X$$

0, 1, 1, 2, 3, 5, 8, 13, ..

בסידרת פיבונאצי

האיבר הראשון הוא – 0

האיבר השני הוא – 1

וכל איבר אחר הוא

סכום שני קודמיו

$$a_1 = a$$

$$a_n = a_{n-1} + d$$

# דוגמא – עצרת !, רקורסיה חשבונית, מעבר לפסקל (פרוצדורה)

```
procedure Factorial_p( N:  
    integer; var F:integer);  
var  
    F1 : integer;  
begin  
    if N = 0 then  
        F := 1  
    else  
        if N = 1 then  
            F := 1  
        else  
            begin  
                Factorial_p( N-1, F1);  
                F := F1 * N;  
            end;  
        end;  
end;
```

# דוגמא – עצרת !, רקורסיה חשבונית, מעבר לפסקל (פונקציה)

```
function Factorial_f( N: integer):  
    integer;  
  
begin  
    if N = 0 then  
        Factorial_f := 1  
    else  
        if N = 1 then  
            Factorial_f := 1  
        else  
            Factorial_f :=  
                Factorial_f(N-1) * N;  
        end;  
    end;  
end;
```

# דוגמא – סידרת פיבונאצי, רקורסיה חשבונית, מעבר לפסקל (פרוצדורה)

```
procedure Fibonacci_p( N: integer;  
                        var F:integer);  
var  
    F1, F2 : integer;  
begin  
    if N = 1 then  
        F := 0  
    else  
        if N = 2 then  
            F := 1  
        else  
            begin  
                Fibonacci_p( N-1, F1);  
                Fibonacci_p( N-2, F2);  
                F := F1 + F2;  
            end;  
        end;  
end;
```

# דוגמא – סידרת פיבונאצי, רקורסיה חשבונית, מעבר לפסקל (פונקציה)

```
function Fibonacci_f( N: integer):  
    integer;  
begin  
    if N = 1 then  
        Fibonacci_f := 0  
    else  
        if N = 2 then  
            Fibonacci_f := 1  
        else  
            Fibonacci_f :=  
Fibonacci_f(N-1) + Fibonacci_f(N-2);  
end;
```

# דוגמא – מספר טבעי, רקורסיה חשבונית מעבר לפסקל (פונקציה)

```
function Natural_Num( N: integer):  
                                boolean;  
begin  
    if N > 0 then  
        if N = 1 then  
            Natural_Num := true  
        else  
            Natural_Num :=  
                Natural_Num(N-1)  
        else  
            Natural_Num := false;  
    end;  
end;
```

# דוגמא – נמצא ברשימה רקורסיה ברשימות

איבר נמצא ברשימה אם:

- הוא האיבר הראשון ברשימה או
- הוא נמצא בהמשך הרשימה

# דוגמא – נמצא ברשימה

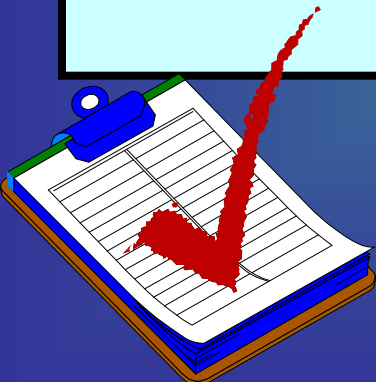
## רקורסיה ברשימות - פרולוג

% נמצא\_ברשימה (L ,X)

נמצא\_ברשימה (X , [ \_ | X ]).

נמצא\_ברשימה (X , [ Z | \_ ] ) :-

נמצא\_ברשימה (Z ,X).





# דוגמא – נמצא ברשימה רקורסיה

## מערך - בפסקל

```
function In_List(A:arr_type;N:integer;
                P: integer;
                X: integer): boolean;
begin
  if P <= N then
    begin
      if X = A[P] then
        In_List := true
      else
        In_List := In_List(A,P+1,X);
      end
    end
  else
    In_List := false;
  end;
end;
```

# דוגמא – נמצא ברשימה

## רקורסיה ברשימות - פסקל

```
function In_List( L: list_type;  
                 P: list_pos;  
                 X: list_info_type): boolean;  
var Y: list_info_type;  
begin  
  if not list_is_empty(L) then  
    begin  
      list_retrieve(L,P,Y);  
      if X = Y then  
        In_List := true  
      else  
        In_List := In_list(L,  
                           list_next(L,P),X);  
      end  
    end  
  else  
    In_List := false;  
end;
```

# דוגמא – קטן ברשימה

## הגדרה רקורסיבית

הקטן ברשימה הוא:

• האיבר היחיד ברשימה א

• האיבר הקטן ב

הראשון ברשימה ל

הקטן בין יתר אברי הרשימה

# דוגמא – קטן ברשימה

## רקורסיה ברשימות - פרולוג

הנחה: הרשימה לא ריקה

% קטן ברשימה (קטן, רשימה)

קטן ברשימה (X, [X]).

קטן ברשימה (X, [Z | X]) :-

קטן ברשימה (Z, Y),

$Y > X$ .

קטן ברשימה (Y, [Z | X]) :-

קטן ברשימה (Z, Y),

$X > Y$ .

# דוגמא – קטן ברשימה רקורסיה במערך - פסקל

```
function List_Small( A:arr_type;  
                    N,P:integer):integer;  
var Y:integer;  
begin  
  if P = N then  
    List_Small:= A[N]  
  else  
    begin  
      Y := List_Small(A,N,P+1);  
      if X < Y then  
        List_Small:= X  
      else  
        List_Small:= Y;  
    end;  
end;  
end;
```

# דוגמא – קטן ברשימה

## רקורסיה ברשימות - פסקל

```

function List_Small( L: list_type;
                    P: list_pos) :
                    integer;
var X,Y : list_info_type;
begin
  list_retrieve(L, P, X);
  if list_next(L, P) = list_end(L)
  then
    List_Small:= X
  else
    begin
      Y := List_Small
          (L, list_next(L, P));
      if X < Y then
        List_Small:= X
      else
        List_Small:= Y;
    end;
  end;
end;

```

תורה