فصل دوم

سید ناصر رضوی

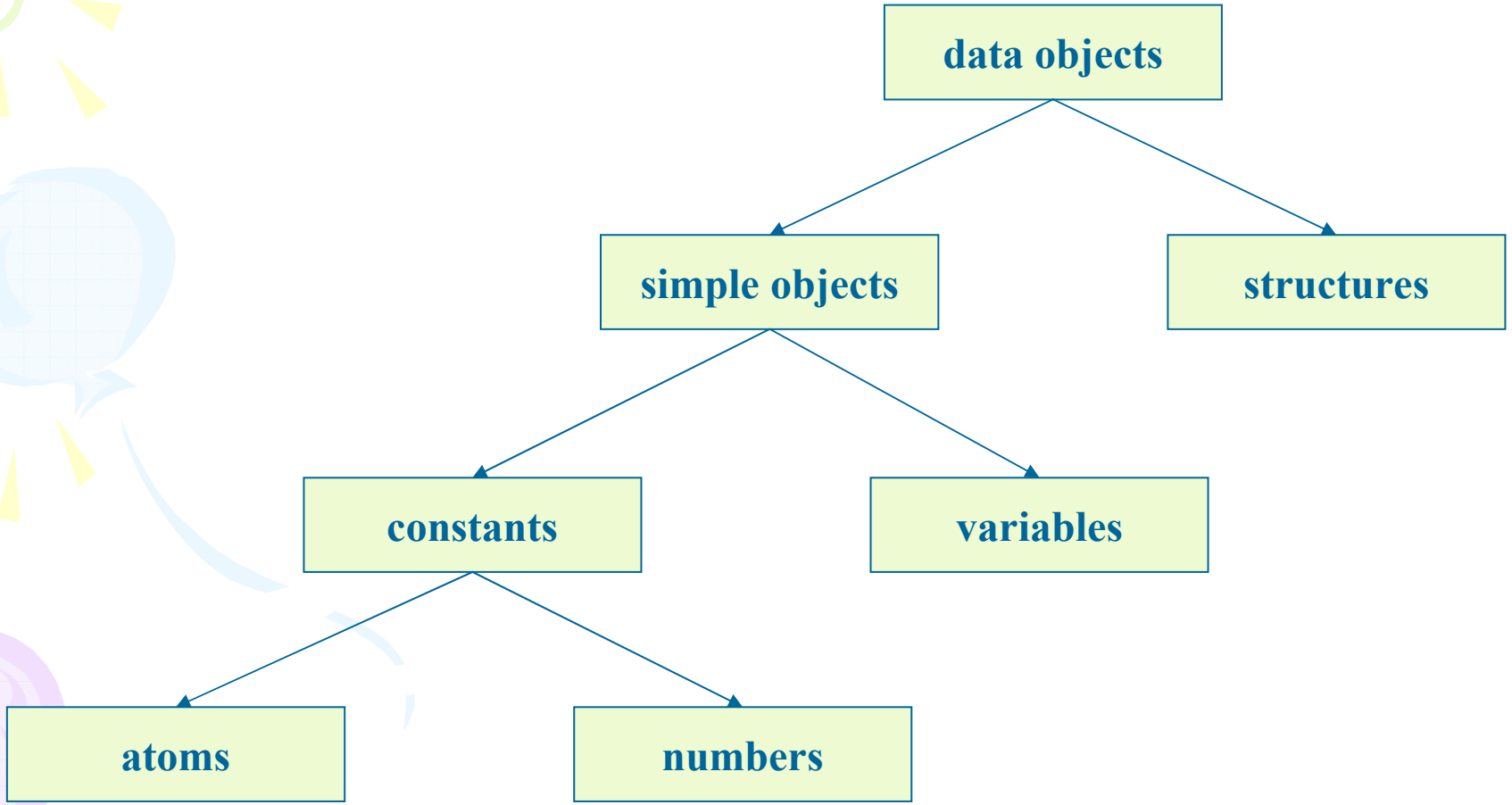
razavi@Comp.iust.ac.ir

۱۳۸۳

مقدمه

- اشیاء داده ای ساده
  - اتم ها (ثابت های غیر عددی)
  - اعداد
  - متغیرها
- اشیاء ساخت یافته
- تطابق
- معنای توصیفی یک برنامه
- معنای رویه ای یک برنامه
- ارتباط معنای توصیفی و رویه ای یک برنامه
- تغییر معنای رویه ای توسط تغییر دادن ترتیب فراکردها و اهداف

اشیاء داده ای



اشیاء داده ای

- در پرولوگ نوع یک شیء توسط ساختار گرامری آن مشخص می شود.
  - متغیرها با یک حرف الفبایی بزرگ شروع می شوند.
  - اتم ها با یک حرف الفبایی کوچک شروع می شوند.
- در پرولوگ اتم ها و متغیر ها رشته هایی از کاراکترهای زیر می باشند:
  - حروف بزرگ A، B، ... ، Z
  - حروف کوچک a، b، ... ، z
  - ارقام 0، 1، ... ، 9
  - کاراکتر های خاص مانند ~ _ & . : = < > / * - +

اشیاء داده ای

- اتم ها به سه شکل می توانند ساخته شوند:

– رشته هایی از حروف، ارقام و کاراکتر زیرخط " _ "، که با یک حرف کوچک شروع می شوند مانند :

  - anna، nil، x25 ، x2AB ، x_ ، miss_Jones

– رشته هایی از کاراکترهای خاص مانند:

  - ::= ، .:. ، ... ، = = = = > ، <--->

– رشته هایی از کاراکترها که در یک کوتیشن تک قرار دارند:

  - 'Tom' ، 'South_America' ، 'Miss Jones'

اشیاء داده ای

- اعداد در پرولوگ:
  - اعداد صحیح : 1, 1313, 0, -97
  - اعداد حقیقی: 3.14, -0.0035, 100.2
  - اعداد حقیقی در پرولوگ کمتراستفاده می شوند.
- متغیرها: رشته هایی از حروف، ارقام و کاراکتر زیرخط می باشند که با یک حرف بزرگ و یا کاراکتر زیر خط شروع می شوند، مانند:
  - X, Result, ShoppingList, _x23, _23

متغیرها

- اگر متغیری در یک عبارت فقط یک بار ظاهر شود، می توان به جای آن از متغیر بی نام (anonymous)استفاده نمود:

```
haschild(X) :- parent(X, Y).
haschild(X) :- parent(X, _).
```

```
sombody_has_child :- parent(X, Y).
sombody_has_child :- parent(_, _).
```

اگر متغیر بی نام در یک پرسش قرار بگیرد، مقدار آن به خروجی فرستاده نمی شود:

```
?- parent(X,_).
```

توجه: حوزه لغوی یک متغیر یک فراکرد است.

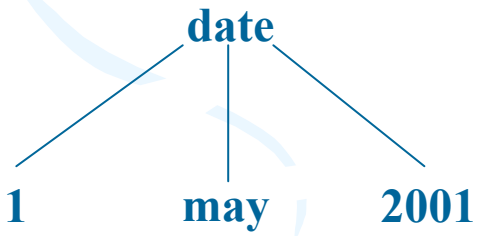
ساختارها

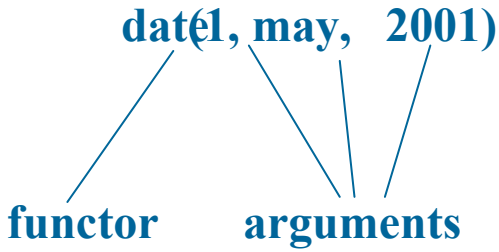
- اشیاء ساخت یافته (ساختارها):اشیایی که شامل چندین مولفه می باشند. خود مولفه ها می توانند ساختار باشند.

date(1, may, 2001).

date(Day, may, 2001).

عبارت ها(terms): تمامی اشیاء داده ای در پرولوگ عبارت محسوب می شوند، مانند may و date(1, may, 2001).





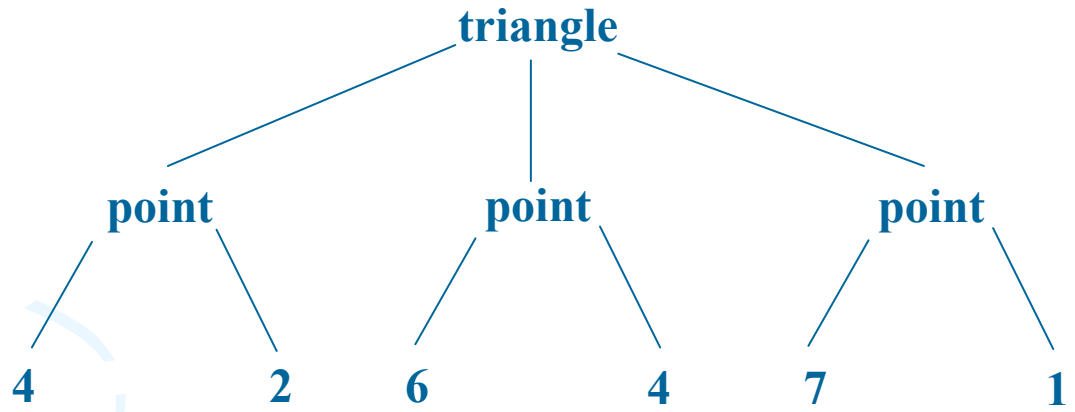
ساختارها

P1 = point(1,1)

P2 = point(2,3)

S = seg(P1, P2) = seg(point(1,1) , point(2,3))

T = triangle(point(4,2) , point(6,4) , point(7,1))



ساختارها

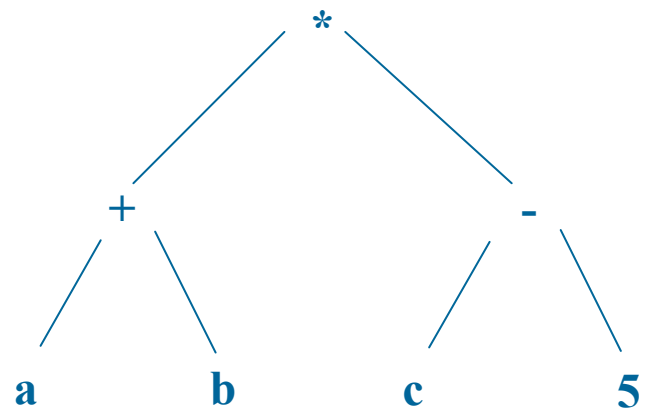
- هر functor بوسیله دو مشخصه تعریف می شود:
  (۱) نام
  (۲) چندی(arity) ، تعداد آرگومانها

بنابراین می توان در یک برنامه تعاریف زیر را داشت:

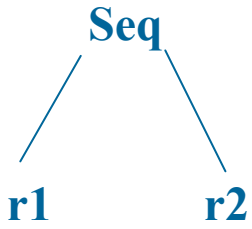
point(X1, Y1), point(X, Y, Z)

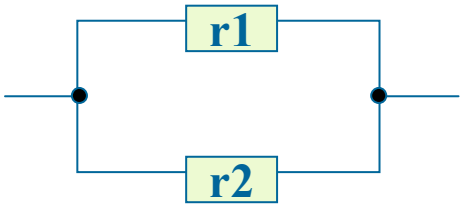
مثال:

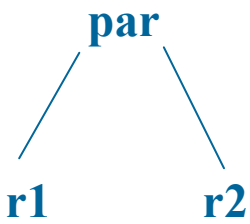
(a + b) * (c – 5)
*(+(a, b), -(c, 5))

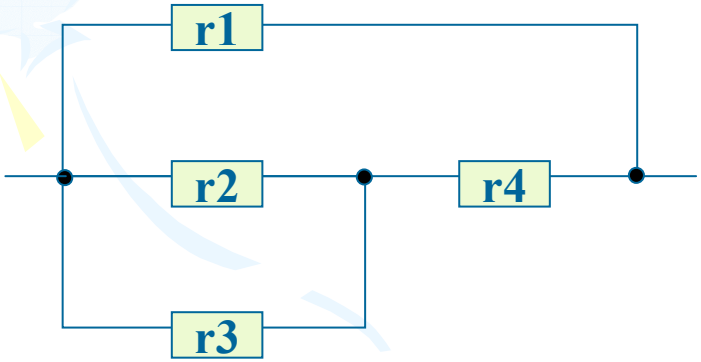


ساختارها

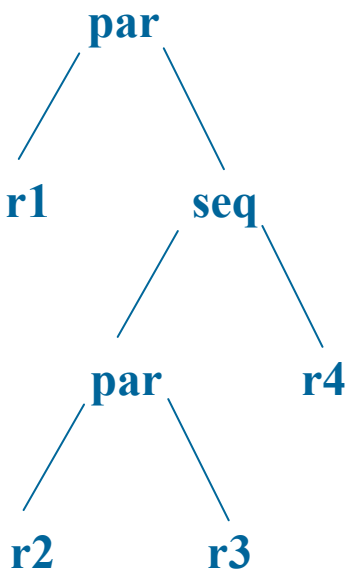








par(r1, seq(par(r2, r3), r4))



تطابق

- عمل تطابق مهمترین عملی است که برروی عبارت ها در پرولوگ انجام می گیرد.
- گوییم دو عبارت داده شده، باهم تطابق دارند اگر:

(۱) یکسان باشند، یا

(۲) متغیرها در هر دو عبارت بتوانند به گونه ای با اشیاء جایگزین شوند که بعد از جایگزینی‌عبارت ها با هم یکسان شوند. مثلا:

date(D, M, 2001) ,
date(D1, may, Y1)

دو عبارت بالا با هم تطابق دارند اگر:

۱- D با D1 جایگزین شود.

۲- M با may جایگزین شود.

۳- Y1 با 2001 جایگزین شود.

D =D1

M = may

Y1 = 2001

تطابق

- عبارت های زیر قابل تطابق نمی باشند:

```
date(D, M, 2001),
date(D1, M1, 1444).

date(X, Y, Z),
point(X, Y, Z)
```

تطابق

- **عمل تطابق** فرآیندی است که دو عبارت را به عنوان ورودی دریافت می کند و امکان تطابق یافتن آنها را بررسی می کند
  - اگر دو عبارت قابل تطابق نباشند، این فرآیند **مردود** می شود.
  - وگرنه فرآیند تطابق **موفقیت آمیز** می باشد و متغیرها در هر دو عبارت ورودی با مقادیری جایگزین می کند که هر دوعبارت به یک عبارت یکسان تبدیل شوند.

```
?- date(D, M, 2001) =   date(D1, may, Y1).
D = H1
M = may
D1 = H1
Y1 = 2001
```

تطابق

- **توجه:** عمل تطابق در پرولوگ همیشه منجر به عمومی ترین نمونه دهی ها می شود.

```
?- date(D, M, 2001) = date(D1, may, Y1),
   date(D, M, 2001) = date(15, M, Y).
D = 15
M = may
D1 =15
Y1 =2001
Y = 2001
```

تطابق

- الگوریتم تطابق دو عبارت S, T:

(۱) اگر S, T هردو ثابت باشند، درصورتی قابل تطابق هستند که هر دو شیء یکسانی باشند.

(۲) اگر S یک متغیر و T هر چه باشد، آنگاه قابل انطباق هستند و S با T نمونه دهی می شود و برعکس.

(۳) اگر S و T ساختار باشند، تنها در صورتی تطابق می یابند که:

الف- S و T دارای functor اصلی یکسانی باشند، و

ب- تمام مولفه های متناظرشان تطابق یابند.

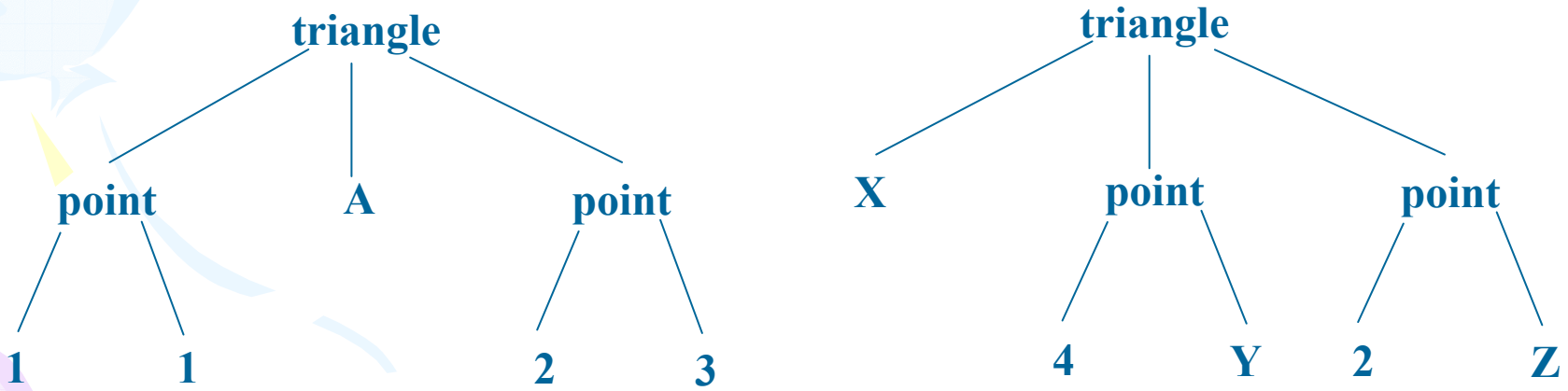
تطابق

?- triangle(point(1, 1), A, point(2, 3)) = triangle(X, point(4, Y) , point(2, Z)).

A = point(4, Y)

X = point(1, 1)

Z = 3



تطابق

- مثال : تعریف خاصیت عمودی و افقی برای پاره خط ها

vertical(seg(point(X, Y), point(X, Y1))).
horizontal(seg(point(X, Y), point(X1, Y))).

```
?- vertical(seg(point(1, 1), point(1, 2) ) ).
  yes
?- vertical(seg(point(1, 1), point(2, Y) ) ).
  no
?- horizontal(seg(point(1, 1), point(2, Y) ) ).
  Y = 1
?- vertical(seg(point(2, 3), P).
  P = point(2, _5B)
?- vertical(S), horizontal(S).
  S = seg(point(X, Y), point(X, Y) )
```

معنای توصیفی برنامه ها

- **P :- Q, R.**

معنای توصیفی: P درست است اگر Q و R درست باشند.

معنای رویه ای:

برای ارضاء P ابتدا Q و سپس R را ارضاء کن.

برای حل مسأله P ابتدا زیر مسأله Q و سپس زیر مسأله R را حل کن.

بنابراین تفاوت در اهمیت ترتیب بخش شرایط می باشد.

معنای توصیفی برنامه ها

- **P :- Q; R.**

معنای توصیفی: P درست است اگر Q درست باشد **یا** R درست باشد.
معادل با:

P :- Q.
P :- R.

اولویت کاما از سمی کالن بیشتر است:

P :- Q, R; S, T, U.

معادل است با:

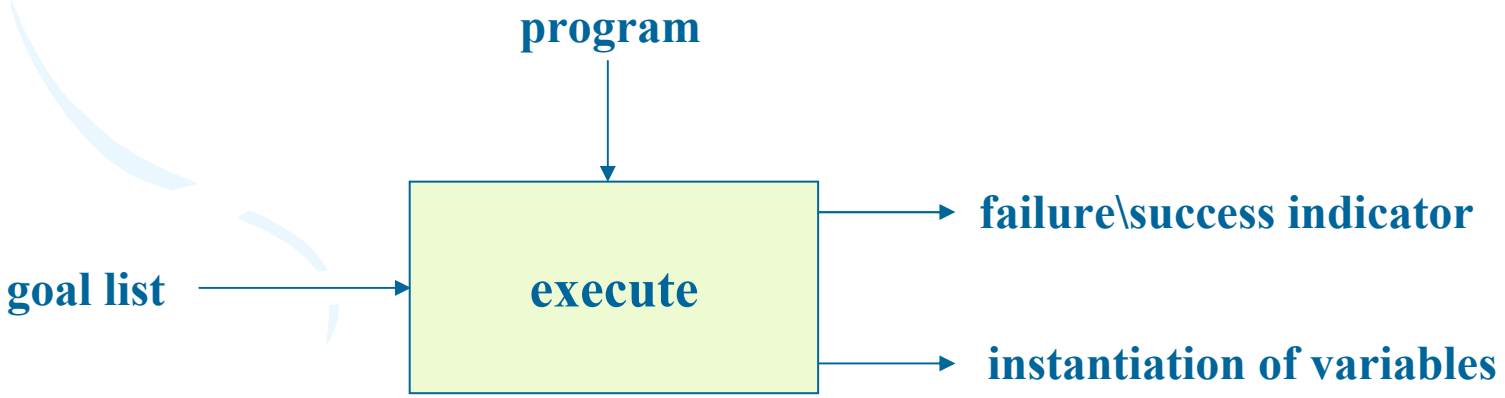
P :- (Q, R); (S, T, U).

یا:

P :- Q, R.
P :- S, T, U.

معنای رویه ای برنامه ها

- معنای رویه ای **چگونگی** پاسخ گویی پرولوگ به پرسشها را مشخص می کند.
- بنابراین معنای رویه ای پرولوگ، یک رویه برای اجرای لیستی از اهداف (پرسش) در رابطه با یک برنامه می باشد.



مسأله میمون و موز

- **مسأله:** یک میمون در کنار در ورودی اتاقی قرار دارد. در وسط اتاق یک موز از سقف آویزان است.میمون گرسنه است و می خواهد خودش را به موز برساند، اما دستش به آن نمی رسد. در کنار پنجره یک جعبه وجود دارد که میمون می تواند از آن استفاده کند. میمون می تواند اعمال زیر را انجام دهد:
  - کف اتاق راه برود.
  - از جعبه بالا برود.
  - جعبه را به این طرف و آن طرف جابجا کند.
  - موز رابگیرد به شرطی که روی جعبه باشد و جعبه نیز در وسط اتاق و زیر موز باشد.
- سوال این است که آیا میمون می تواند به موزش برسد یا خیر و اگر می تواند کدام دنباله از اعمال را باید انجام دهد؟

مسأله میمون و موز

- **فرموله سازی مسأله:** در این مسأله می توان دنیا را در هر لحظه توسط حالتی بیان کرد که درآن موقعیت اشیاء مشخص شده باشند. حالت می تواند با زمان (انجام عمل) تغییر کند.

state(P1, P2, P3, H)

موقعیت افقی میمون — موقعیت عمودی میمون — موقعیت جعبه — داشتن/نداشتن موز

- **حالت اولیه:**

state(atdoor, onfloor, atwindow, hasnot)

- **فرموله سازی هدف:**

state(_, _, _, has)

مسأله میمون و موز

- فرموله سازی عملیات :

move(State1, Move, State2)

(۱) عمل گرفتن موز:

move(state(middle, onbox, middle, hasnot),
grasp,
state(middle, onbox, middle, has)).

مسأله میمون و موز

- فرموله سازی عملیات :

move(State1, Move, State2)

(۲) عمل راه رفتن کف اتاق:

```
move(state(Pos1, onfloor, Box, Has),
     walk(Pos1, Pos2),
     state(Pos2, onfloor, Box, Has) ).
```

مسأله میمون و موز

- فرموله سازی عملیات :

move(State1, Move, State2)

(۲) عمل جابه جا کردن جعبه:

```
move(state(Pos1, onfloor, Pos1, Has),
     push(Pos1, Pos2),
     state(Pos2, onfloor, Pos2, Has) ).
```

مسأله میمون و موز

- فرموله سازی عملیات :

move(State1, Move, State2)

(۲) عمل بالا رفتن از جعبه:

```
move(state(Pos, onfloor, Pos, Has),
     climb,
     state(Pos, onbox, Pos, Has) ).
```

مسأله میمون و موز

- پرسش اصلی که برنامه باید پاسخ دهد این است که آیا دنباله ای از عملیات وجود دارند که میمون را از حالت اولیه مسأله به حالت هدف (نهایی) برسانند.

canget(State)

که می تواند به صورت زیر بیان شود:

```
canget(state(_, _, _, has) ).
canget(State1) :-
    move(State1 , Move, State2),
    canget(State2).
```

مسأله میمون و موز(برنامه کامل)

```
move(state(middle, onbox, middle, hasnot),
        grasp,
        state(middle, onbox, middle, has) ).
move(state(Pos, onfloor, Pos, Has),
        climb,
        state(Pos, onbox, Pos, Has) ).
move(state(Pos1, onfloor, Pos1, Has),
        push(Pos1, Pos2),
        state(Pos2, onfloor, Pos2, Has) ).
move(state(Pos1, onfloor, Box, Has),
        walk(Pos1, Pos2),
        state(Pos2, onfloor, Box, Has) ).

canget(state(_, _, _, has) ).
canget(State1) :-
      move(State1, Move, State2),
      canget(State2).
```

مسأله میمون و موز

- حل مسأله:

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

دنباله عملیات:

walk(atdoor, atwindow),
push(atwindow, middle),
climb,
grasp

ترتیب فراکردها و اهداف

- خطر حلقه بی پایان

P :- P.

این فراکرد از لحاظ معنای توصیفی کاملا درست است (P درست است اگر P درست باشد).
اما از لحاظ رویه ای مشکل ایجاد می کند (حلقه بی پایان)

در برنامه میمون و موز عملیات را به ترتیب زیر در برنامه نوشتیم:
گرفتن موز، بالارفتن، جابه جا کردن جعبه و راه رفتن
این ترتیب اولویت اعمال مختلف را برای میمون مشخص می کنند، یعنی مثلا میمون بین راه رفتن و گرفتن موز دومی را ترجیح می دهد. این اولویت ها به میمون در حل مسأله کمک می کنند. حال اگر ترتیب عملیات را عوض کنیم و عمل راه رفتن را بالاتر از بقیه اعمال در برنامه قراردهیم میمون تا ابد در اتاق قدم خواهد زد و هیچگاه به موز نمی رسد (اگر زنده بماند!) .

ترتیب فراکردها و اهداف

- تمرین۱

رویه predecessor را به هر چهار شکل ممکن بنویسید و اجرا کنید. (با عوض کردن ترتیب دو فراکرد و نیز دو هدف موجود در فراکرد دوم). کدامیک را ترجیح می دهید؟ آیا می توانید برای برخورد با چنین مواردی یک قانون کلی ارائه دهید؟

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

تحقیق

- درمورد ارتباط پرولوگ با زبانهای منطقی (از لحاظ دستوری و معنایی) و نیز نحوه اثبات در پرولوگ تحقیق کنید.