



برنامه نویسی پرولوگ فصل اول

سید ناصر رضوی

razavi@Comp.iust.ac.ir

۱۳۸۳



مقدمه

- مکانیزم های پایه ای در پرولوگ
 - فراکردهای پرولوگ (Clauses)
 - حقایق (Facts)
 - قوانین (Rules)
 - رویه ها (procedures)
- مکانیزم عقبگرد در پرولوگ (backtracking)
- تفاوت معنای توصیفی و رویه ای یک برنامه



تعریف روابط بوسیله حقایق

- پرولوگ یک زبان برنامه نویسی برای محاسبات نمادین و غیر عددی می باشد.
- پرولوگ برای حل مسائلی که شامل اشیاء و روابط میان آنها می باشند بسیار مناسب می باشد.
- مثال : روابط خانوادگی

– این حقیقت که Tom پدر Bob است را در پرولوگ به صورت زیر نمایش می دهیم:

parent(tom, bob).

در اینجا parent نام رابطه و Tom و Bob آرگومانهای رابطه می باشند.
توجه: اسامی خاص مانند tom باید با یک حرف کوچک الفبا نوشته شوند.

تعریف روابط بوسیله حقایق

- این برنامه از شش فراکرد تشکیل شده است.
- هر فراکرد یک نمونه از رابطه parent می باشد.

parent(tom, bob).

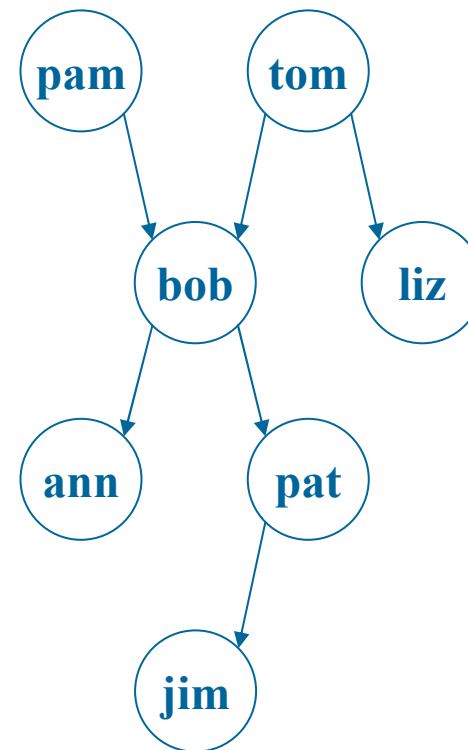
parent(pam, bob).

parent(tom, liz).

parent(bob, ann).

parent(bob, pat).

parent(pat, jim).



تعریف روابط بوسیله حقایق

آیا Bob پدر Pat می باشد؟

?- parent(bob, pat).
yes

این به صورت یک حقیقت در برنامه ذکر شده است.

?- parent(liz, pat).
no

در برنامه چنین رابطه ای بیان نشده است.

?- parent(tom, ben).
no

برنامه نام Ben را نشنیده است.

تعریف روابط بوسیله حقایق

چه کسی والد Liz می باشد؟

?- parent(X, liz).

پرولوگ در پاسخ به این پرسش سعی می کند برای X مقداری بیابد که عبارت بالا درست باشد.

X = tom

فرزندان Bob چه کسانی می باشند؟

?- parent(bob, X).

X = ann;

X = pat;

no

مکانیزمی برای درخواست پاسخهای بیشتر

تعریف روابط بوسیله حقایق

چه کسانی والد چه کسانی می باشند؟

?- parent(X, Y).

X = pam

Y = bob;

X = tom

Y = bob;

X = tom

Y = liz;

...

تعریف روابط بوسیله حقایق

جد Jim چه کسی است؟

(۱) چه کسی والد Jim است؟ (فرض کنیم Y باشد)

(۲) چه کسی والد Y است؟ (فرض کنیم X باشد)

?- parent(Y, jim), parent(X, Y).

X = bob

Y = pat

اگر ترتیب دو درخواست را در پرسش بالا عوض کنیم معنای منطقی ثابت باقی می ماند و بنابراین پرسش زیر همان پاسخ بالا را تولید می کند.

?- parent(X, Y), parent(Y, jim).

تعریف روابط بوسیله حقایق

نوه های Tom چه کسانی هستند؟

?- parent(tom, X), parent(X, Y).

X = bob

Y = ann;

X = bob

Y = pat

تعریف روابط بوسیله حقایق

آیا Ann و Pat والد مشترکی دارند؟

(۱) چه کسی والد Ann می باشد؟ (فرض می کنیم X باشد)

(۲) آیا این X والد Pat نیز هست؟

?- parent(X , ann), parent(X , pat).

$X = \text{bob}$



تعریف روابط بوسیله حقایق

- در پرولوگ تعریف روابط به سادگی توسط یک n -تایی از اشیاء که رابطه را ارضاء می کنند، انجام می شود.
- کاربر به سادگی می تواند از پرولوگ در باره روابط تعریف شده در برنامه پرس و جو کند.
- یک برنامه پرولوگ از **فراکردها** تشکیل می شود.
- پرسش ها شامل یک یا چند **هدف** می باشند، مانند:
 $\text{parent}(X, \text{ann}), \text{parent}(X, \text{pat})$
- اگر پاسخ پرولوگ به یک پرسش مثبت باشد در این صورت می گوئیم پرسش **قابل ارضاء** و **غیر قابل ارضاء** می باشد.
- اگر پرسشی دارای پاسخهای متعددی باشد، پرولوگ هر تعداد پاسخی را که کاربر بخواهد پیدا می کند.



تعریف روابط بوسیله قوانین

برنامه روابط خانوادگی را با افزودن اطلاعاتی مثلا درباره جنسیت افراد می توان تعمیم داد:

female(pam).

male(tom).

male(bob).

female(liz).

female(pat).

female(ann).

male(jim).



روابط یکتایی و دوتایی.

تعریف روابط بوسیله قوانین

- تعریف رابطه offspring (عکس رابطه parent)

(۱) تمام روابط offspring را مانند روابط parent اضافه کنیم:
مثلا

offspring(liz, tom).

(۲) از یک **قانون** کلی استفاده کنیم:

offspring(Y, X) :- parent(X, Y).

یعنی بازاء هر X و Y ، اگر X والد Y باشد، آنگاه Y فرزند X می باشد.

تعریف روابط بوسیله قوانین

- تفاوت میان حقایق و قوانین:
 - یک حقیقت چیزی است که همواره و بدون هیچ قید و شرطی درست می باشد.
 - از طرف دیگر یک قانون چیزهایی را بیان می کند که برای درستی آنها باید برخی از شرایط ارضاء شوند.
- بنابراین قوانین از دو قسمت تشکیل می شوند:
 - بخش شرط (condition part) و
 - بخش نتیجه (conclusion part).

$\underbrace{\text{offspring}(Y, X)}_{\text{head}} :- \underbrace{\text{parent}(X, Y)}_{\text{body}}.$

تعریف روابط بوسیله قوانین

آیا Liz فرزند Tom است؟

?- offspring(liz, tom).

نحوه پاسخ دهی پرولوگ:

چون قانون اسلاید قبل بازا هر X و Y قابل اعمال است، پس بر اشیاء خاص Liz و Tom نیز قابل اعمال می باشد. برای اعمال این قانون بر این اشیاء خاص، باید Y را با Liz و X را با Tom در قانون فوق جایگزین کنیم (نمونه دهی)

$X = tom$, $Y = liz$

پس از جایگذاری، یک نمونه خاص از قانون را به شکل زیر بدست می آوریم:

offspring(liz, tom) :- parent(tom, liz)

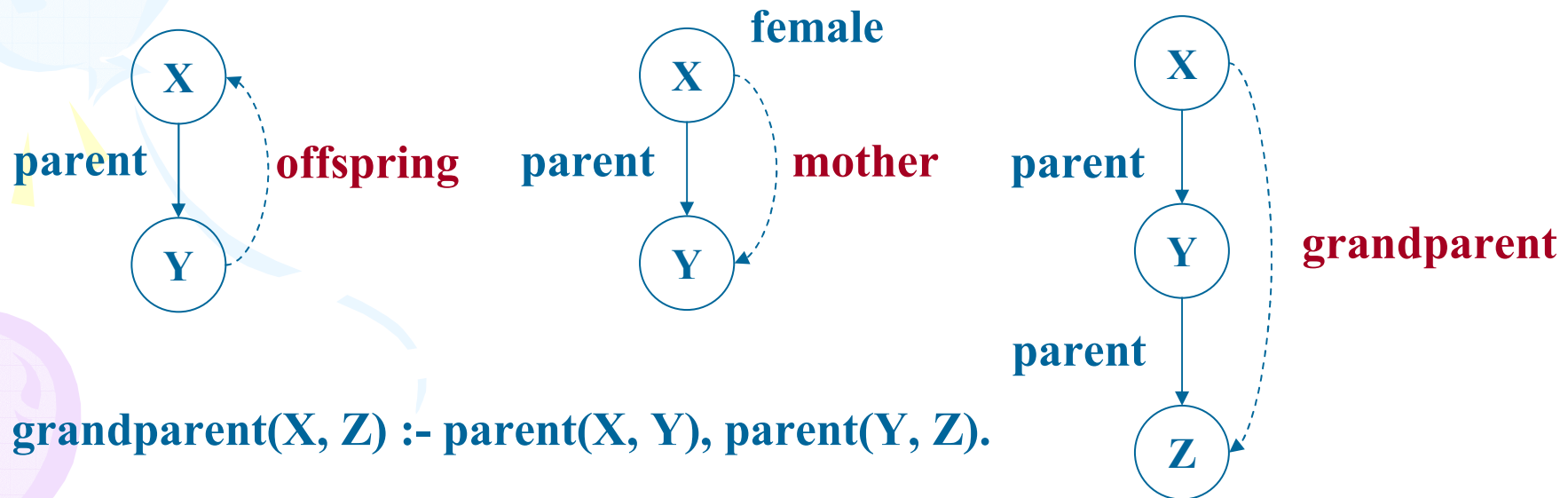
اکنون پرولوگ سعی می کند که دریابد آیا بخش شرط قانون بالا درست است یا خیر. چون بخش شرط درست است (با توجه به حقایق موجود در برنامه)، بنابراین پاسخ پرولوگ yes خواهد بود.

تعریف روابط بوسیله قوانین

تعریف رابطه مادری:

$\text{mother}(X, Y) :- \text{parent}(X, Y), \text{female}(X).$

توجه: در قانون فوق علامت ویرگول به معنای ترکیب عطفی (And) شرایط می باشد.



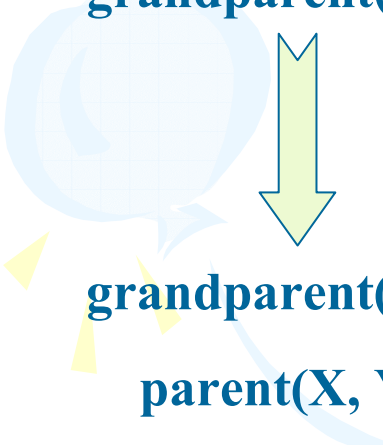
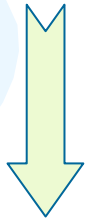
$\text{grandparent}(X, Z) :- \text{parent}(X, Y), \text{parent}(Y, Z).$




تعریف روابط بوسیله قوانین

نحوه قالب بندی برنامه ها

grandparent(X, Z) :- parent(X, Y), parent(Y, Z).



**grandparent(X, Z) :-
parent(X, Y),
parent(Y, Z).**



نحوه مناسب نوشتن قوانین:

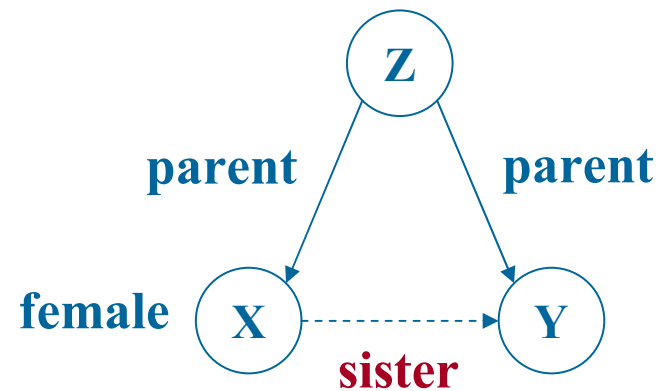
تعریف روابط بوسیله قوانین

تعریف رابطه خواهری (sister)

sister(X, Y) :-
parent(Z, X),
parent(Z, Y),
female(X).

?- sister(ann, pat).
yes

?- sister(X, pat).
X = ann;
X = pat





تعریف روابط بوسیله قوانین

تصحیح رابطه خواهری

sister(X, Y) :-

parent(Z, X),

parent(Z, Y),

female(X),

different(X, Y).





تعریف روابط بوسیله قوانین

- نکات مهم:

- تعمیم برنامه های پرولوگ با افزودن فراکردهای جدید.

- انواع فراکردها:

- حقایق: بیانگر چیزهایی که همواره و بدون شرط درست می باشند.
- قوانین: بیانگر چیزهایی که درستی آنها بستگی به برخی شرایط دارد.
- پرسش ها: وسیله ای که کاربر با آن می تواند از برنامه پرسد چه چیزهایی درست هستند.

- فراکردها از دو قسمت تشکیل می شوند (سرآیند و بدنه)

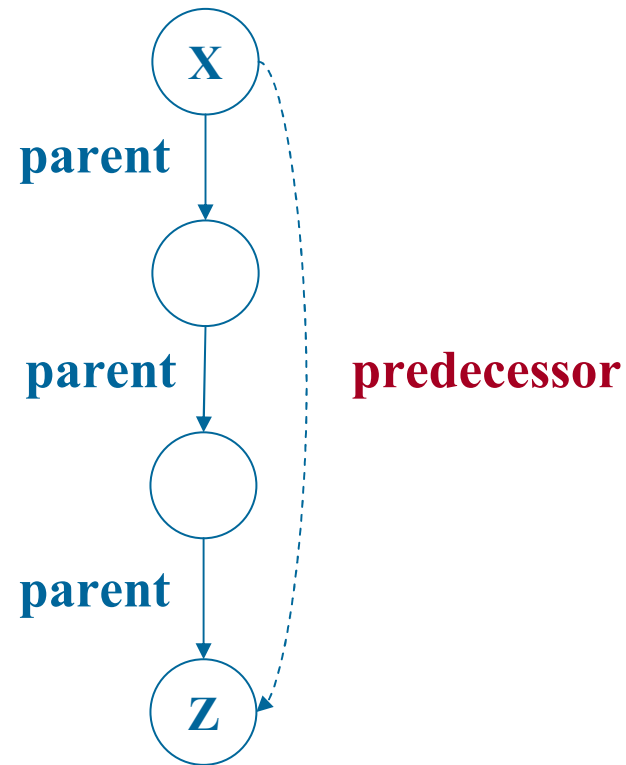
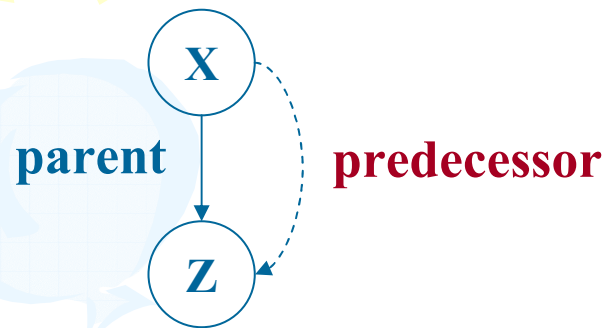
- حقایق: قسمت بدنه آنها تهی
- پرسش ها: فقط دارای بدنه
- قوانین: دارای هر دو قسمت

قوانین بازگشتی

تعریف رابطه اجدادی (predecessor)

جد مستقیم (والد)

جد غیر مستقیم



قوانین بازگشتی

تعریف رابطه predecessor:

predecessor(X, Z) :-

parent(X, Z).

predecessor(X, Z) :-

parent(X, Y),

parent(Y, Z).

predecessor(X, Z) :-

parent(X, Y1),

parent(Y1, Y2),

parent(Y2, Z).

...

قوانین بازگشتی

تعریف predecessor به صورت بازگشتی:

```
predecessor(X, Z) :-      %pr1  
    parent(X, Z).
```

```
predecessor(X, Z) :-      %pr2  
    parent(X, Y),  
    predecessor(Y, Z).
```

```
?- predecessor(pam, X).  
X = bob;  
X = ann;  
X = pat;  
X = jim
```



قوانین بازگشتی

تعریف رویه (procedure):

مجموعه تمام فراکردهای مربوط به یک رابطه، مانند رابطه predecessor در مثال روابط خانوادگی که توسط دو فراکرد تعریف شده است.

انواع توضیحات (comment) در پرولوگ:

`/* this is a comment */`

`% this is also a comment`





نحوه پاسخ گویی به پرسشها در پرولوگ

در پرولوگ هر پرسش شامل یک یا چند هدف می باشد.

پرولوگ برای پاسخ گویی به پرسشها سعی می کند این اهداف را ارضاء کند.
ارضاء هدف یعنی اینکه نشان دهیم هدف درست است (با این فرض که روابط موجود در برنامه درست می باشند).


ارضاء هدف یعنی اینکه نشان دهیم هدف مورد نظر به طور منطقی از حقایق و قوانین موجود در برنامه پیروی می کند.

اگر پرسشی شامل متغیر باشد، پرولوگ باید دریابد که کدام اشیاء خاص (به جای متغیرها) اهداف را برآورده می کنند. اگر چنین اشیائی یافت شوند، پرولوگ نمونه دهی های متغیرها را به کاربر نشان می دهد و گرنه پرولوگ پاسخ منفی (no) برمی گرداند.



نحوه پاسخ گویی به پرسشها در پرولوگ

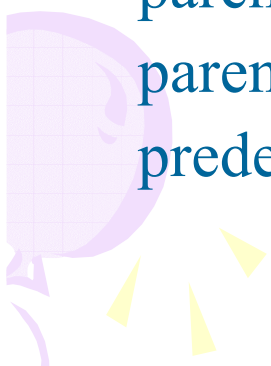
مثال:



fallible(X) :- man(X).
man(socrates).

?- fallible(socrates).
yes

یک مثال پیچیده تر:



?- predecessor(tom, pat).
parent(bob, pat) \Rightarrow predecessor(bob, pat)
parent(tom, bob) *and* predecessor(bob, pat) \Rightarrow
predecessor(tom, pat)

نحوه پاسخ گویی به پرسشها در پرولوگ

predecessor(tom, pat)

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

X = tom, Z = pat

هدف جدید:

parent(tom, pat)

چون هیچ فراکردی در برنامه وجود ندارد که بخش سرآیند آن با این هدف
تطابق (match) داشته باشد، بنابراین این هدف مردود می شود.

اکنون پرولوگ به سمت هدف اول **عقبگرد** می کند تا روش دیگری برای
ارضای آن پیدا کند. بنابراین قانون دوم predecessor را امتحان می کند.



نحوه پاسخ گویی به پرسشها در پرولوگ

predecessor(X, Z) :-

parent(X, Y),

predecessor(Y, Z).

مانند قبل متغیرهای X و Z با tom و pat نمونه دهی می شوند:

X = tom, Z = pat

اکنون هدف اصلی یعنی predecessor(tom, pat) با دو هدف جدید زیر جایگزین می شود:

parent(tom, Y),

predecessor(Y, pat)



اکنون پرولوگ سعی می کند این دو هدف را به ترتیب نوشتن آنها ارضاء کند.



نحوه پاسخ گویی به پرسشها در پرولوگ

هدف اول یعنی $\text{parent}(\text{tom}, Y)$ به راحتی با یک حقیقت برنامه یعنی $\text{parent}(\text{tom}, \text{bob})$ تطابق می یابد، بنابراین هدف اول بانمونه دهی $Y=\text{bob}$ ارضاء می شود و هدف دوم به هدف زیر تبدیل می شود:

$\text{predecessor}(\text{bob}, \text{pat})$

دوباره برای ارضاء این هدف قانون اول predecessor امتحان می شود، یعنی قانون زیر:

$\text{predecessor}(X', Z') :-$

$\text{parent}(X', Z').$

از تطابق هدف و بخش سرآیند این قانون نمونه دهی زیر بدست می آید:

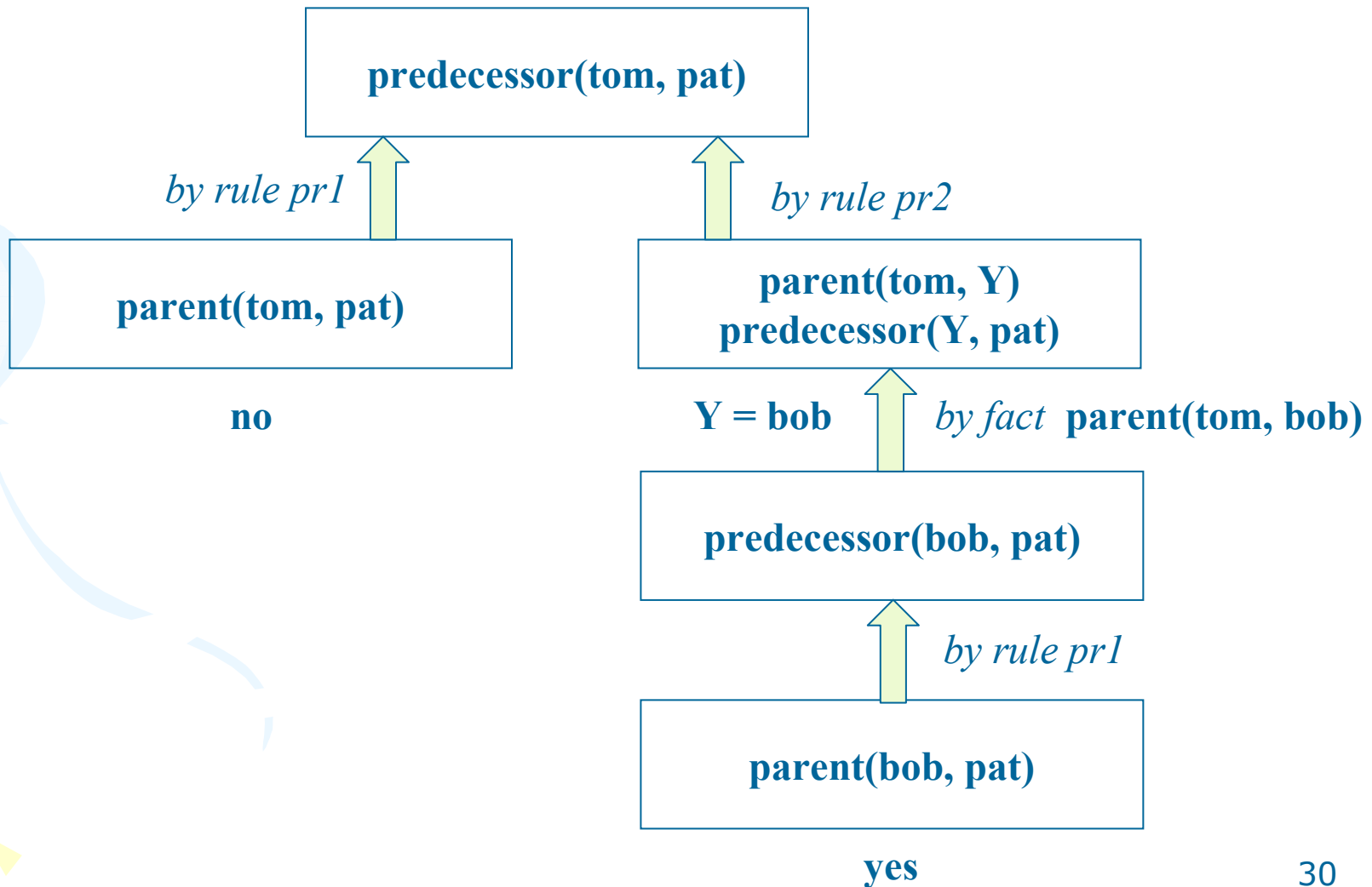
$X' = \text{bob}, Z' = \text{pat}$

و هدف فعلی با هدف جدید زیر جایگزین می شود:

$\text{parent}(\text{bob}, \text{pat})$

این هدف بلافاصله قابل ارضاء می باشد. و بنابراین هدف اصلی نیز ارضاء می شود.

نحوه پاسخ گویی به پرسشها در پرولوگ






معانی توصیفی و رویه ای برنامه ها

معنای توصیفی: تعیین می کند خروجی برنامه چیست (what)

معنای رویه ای: تعیین می کند خروجی برنامه چگونه بدست می آید (how)



از آنجا که نتایج برنامه توسط معنای توصیفی آن تعیین می شوند، از نظر اصولی معنای توصیفی برای نوشتن برنامه کافی می باشد.

فهم جنبه های توصیفی ساده تر است و نوشتن برنامه را ساده می کند.