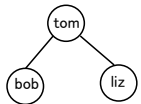


# 家族関係の記述

1

## (1) データベースの記述

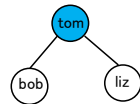
2



parent(tom,bob)  
parent(tom,liz)

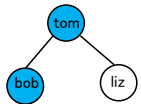
定数、述語は小文字から始める

3



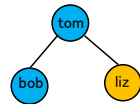
parent(tom,bob)  
parent(tom,liz)  
male(tom)

4



parent(tom,bob)  
parent(tom,liz)  
male(tom)  
male(bob)

5

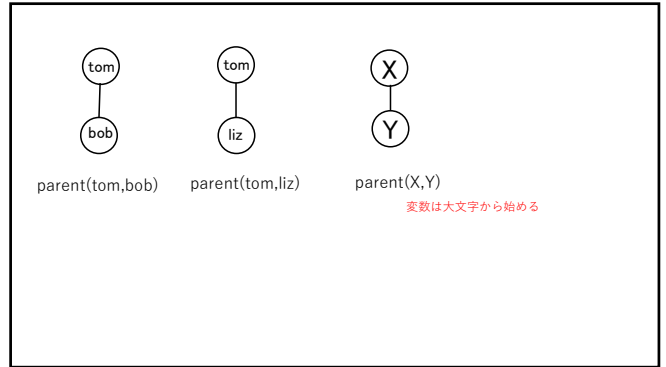


parent(tom,bob)  
parent(tom,liz)  
male(tom)  
male(bob)  
female(liz)

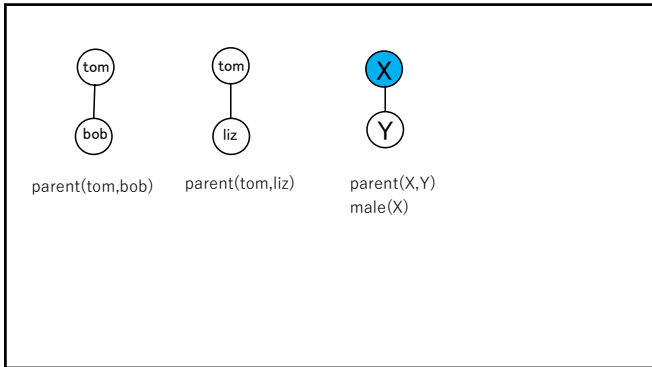
6

(2) 父親の定義

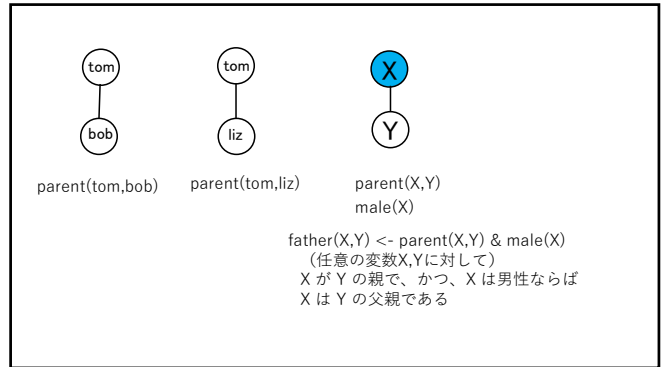
7



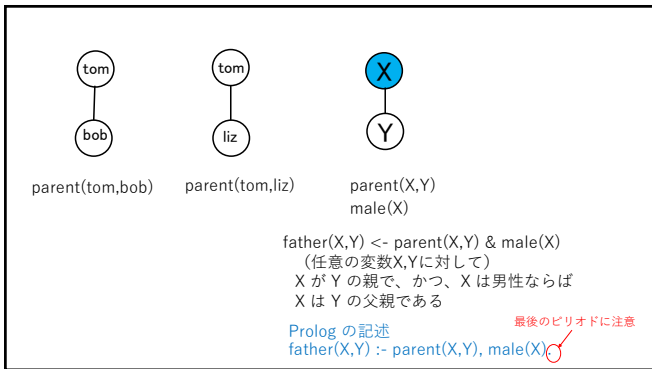
8



9



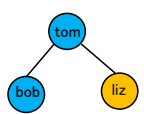
10



11

(3) 引数0の述語の定義

12



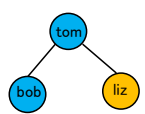
```

parent(tom,bob)
parent(tom,liz)
male(tom)
male(bob)
female(liz)
    
```

tb <- parent(tom,bob)  
 tom が bob の親であるならば tb である

Prolog の記述  
 tb :- parent(tom,bob).

13



```

parent(tom,bob)
parent(tom,liz)
male(tom)
male(bob)
female(liz)
    
```

tb <- parent(tom,bob)  
 tom が bob の親であるならば tb である

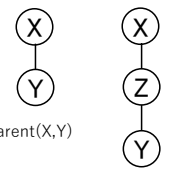
ok <- parent(tom,bob), parent(tom,liz)  
 tom が bob の親で、かつ、tom が liz の親であるならば ok である

Prolog の記述  
 tm :- parent(tom,bob).  
 ok :- parent(tom,liz), parent(tom,liz).

14

(4) 祖父の定義

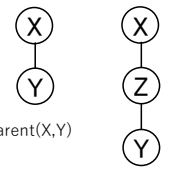
15



```

parent(X,Y)
parent(X,Z)
parent(Z,Y)
    
```

16



```

parent(X,Z)
parent(Z,Y)
grandfather(X,Y) <- father(X,Z) & parent(Z,Y)
    
```

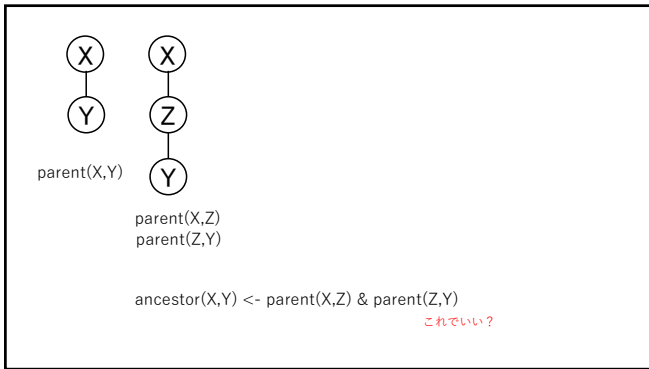
(任意の変数X,Yに対して)  
 X が Z の父親で、かつ、Z が Y の親ならば  
 X は Y の祖父である

Prolog の記述  
 grandfather(X,Y) :- father(X,Z), parent(Z,Y).

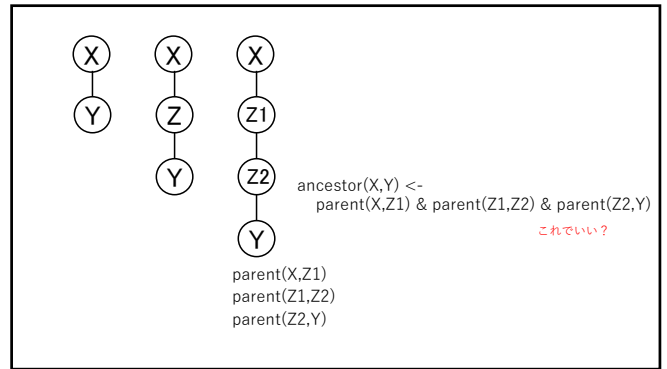
17

(5) 祖先の定義

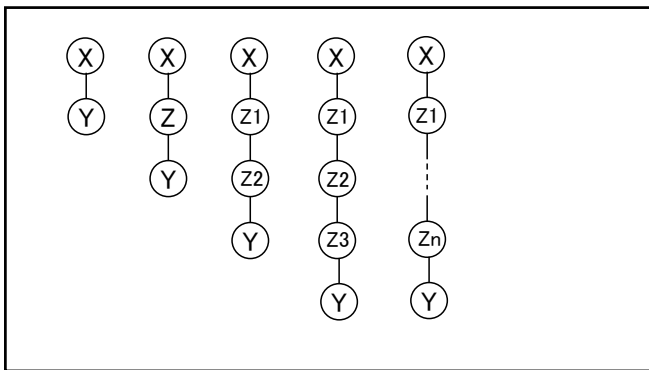
18



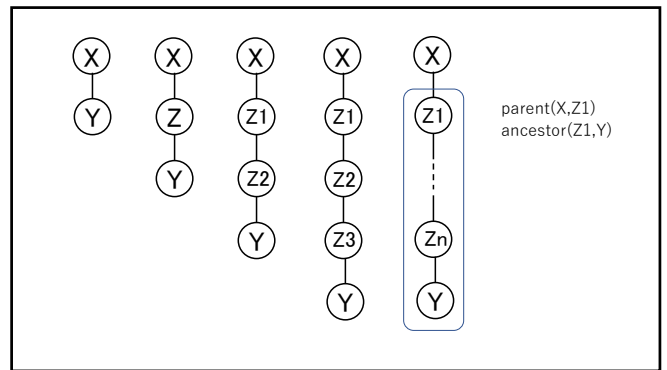
19



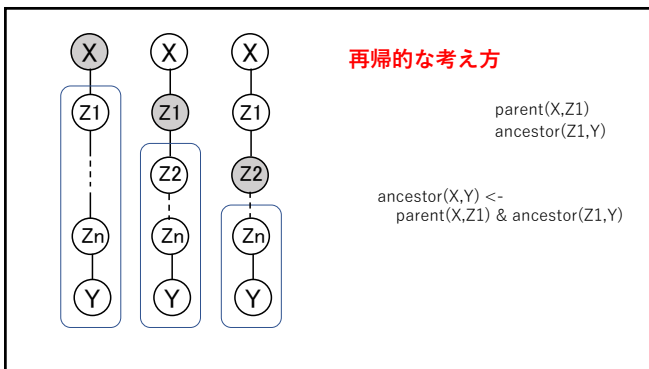
20



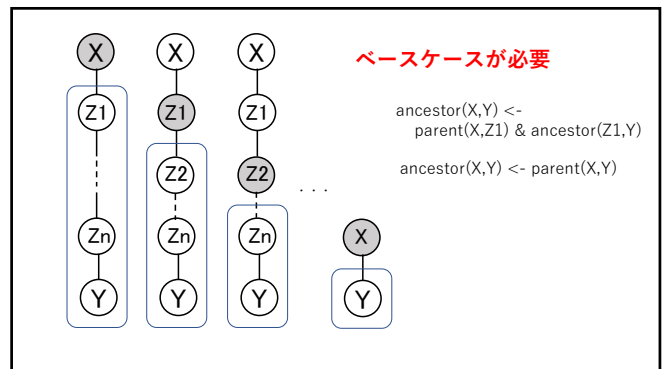
21



22



23



24

`ancestor(X,Y) <-`  
`parent(X,Z) & ancestor(Z,Y)`  
 (任意の変数X,Yに対して)  
 XがZの親で、かつ、ZがYの祖先ならば  
 XはYの祖先である  
  
`ancestor(X,Y) <- parent(X,Y)`  
 (任意の変数X,Yに対して)  
 XがYの親ならばXはYの祖先である  
  
 Prologの記述  
`ancestor(X,Y) :- parent(X,Z), ancestor(Z,Y).`  
`ancestor(X,Y) :- parent(X,Y).`  
 これでいい?

25

`ancestor(X,Y) <-`  
`parent(X,Z) & ancestor(Z,Y)`  
 (任意の変数X,Yに対して)  
 XがZの親で、かつ、ZがYの祖先ならば  
 XはYの祖先である  
  
`ancestor(X,Y) <- parent(X,Y)`  
 (任意の変数X,Yに対して)  
 XがYの親ならばXはYの祖先である  
  
 Prologの記述  
`ancestor(X,Y) :- parent(X,Z), ancestor(Z,Y).`  
`ancestor(X,Y) :- parent(X,Y).`  
 これでいい?  
 論理的にはOKだが動作としてはまずい

26

`ancestor(X,Y) <-`  
`parent(X,Z) & ancestor(Z,Y)`  
 (任意の変数X,Yに対して)  
 XがZの親で、かつ、ZがYの祖先ならば  
 XはYの祖先である  
  
`ancestor(X,Y) <- parent(X,Y)`  
 (任意の変数X,Yに対して)  
 XがYの親ならばXはYの祖先である  
  
 Prologの記述  
`ancestor(X,Y) :- parent(X,Y).`  
`ancestor(X,Y) :- parent(X,Z), ancestor(Z,Y).`  
 Prologは上から下、左から右に実行するので、無限ループを避ける必要がある

27