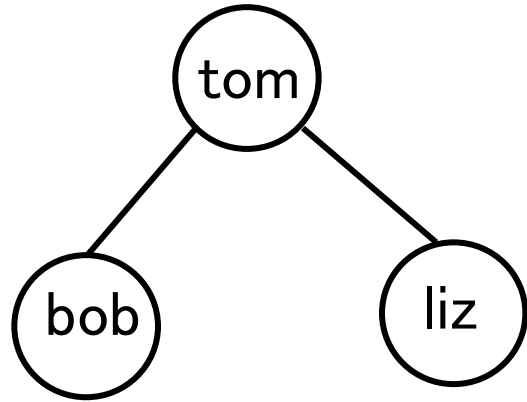


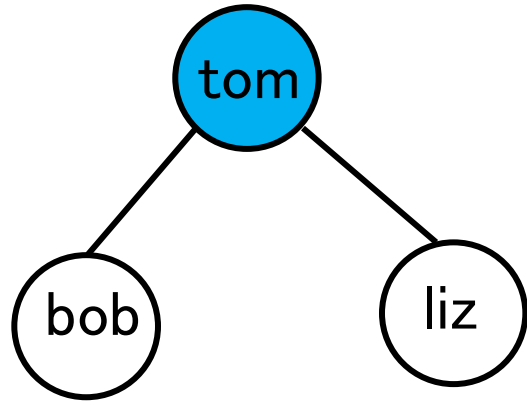
# 家族関係の記述

# (1) データベースの記述



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

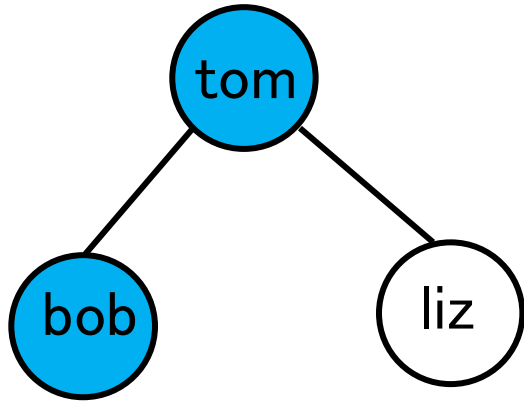
定数、述語は小文字から始める



parent(tom,bob)

parent(tom,liz)

male(tom)

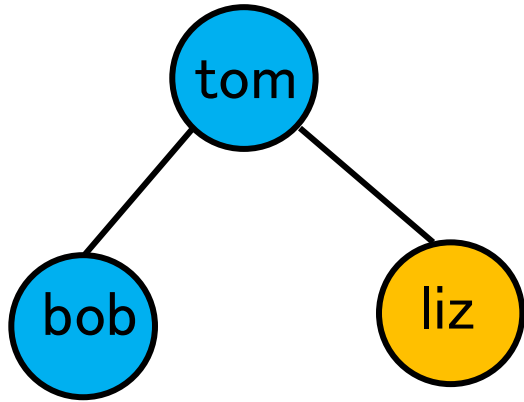


parent(tom,bob)

parent(tom,liz)

male(tom)

male(bob)



parent(tom,bob)

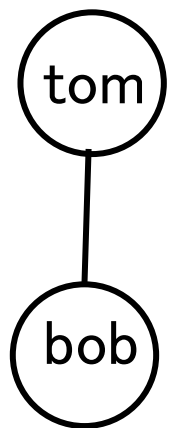
parent(tom,liz)

male(tom)

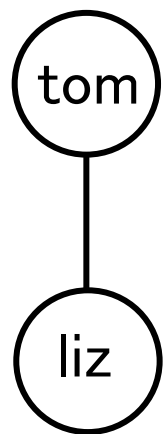
male(bob)

female(liz)

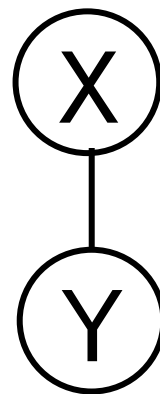
## (2) 父親の定義



parent(tom,bob)



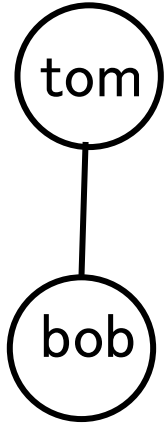
parent(tom,liz)



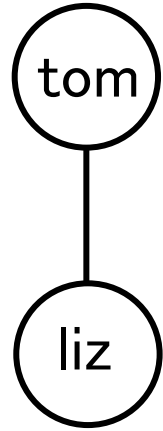
parent(X,Y)

変数は大文字から始める

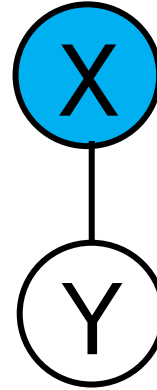




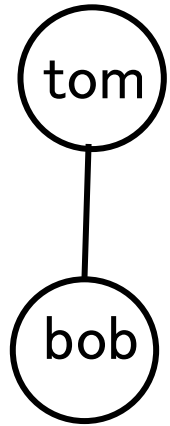
parent(tom,bob)



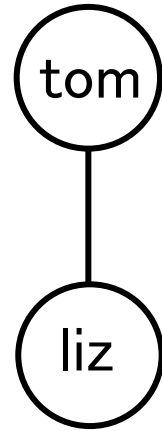
parent(tom,liz)



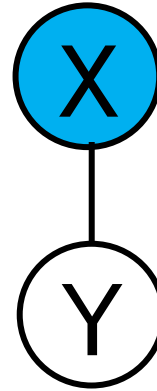
parent(X,Y)  
male(X)



parent(tom,bob)



parent(tom,liz)

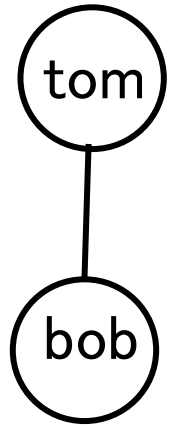


parent(X,Y)  
male(X)

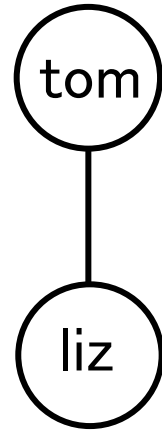
father(X,Y) <- parent(X,Y) & male(X)

(任意の変数X,Yに対して)

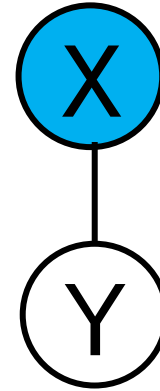
XがYの親で、かつ、Xは男性ならば  
XはYの父親である



parent(tom,bob)



parent(tom,liz)



parent(X,Y)  
male(X)

father(X,Y) <- parent(X,Y) & male(X)

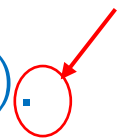
(任意の変数X,Yに対して)

XがYの親で、かつ、Xは男性ならば  
XはYの父親である

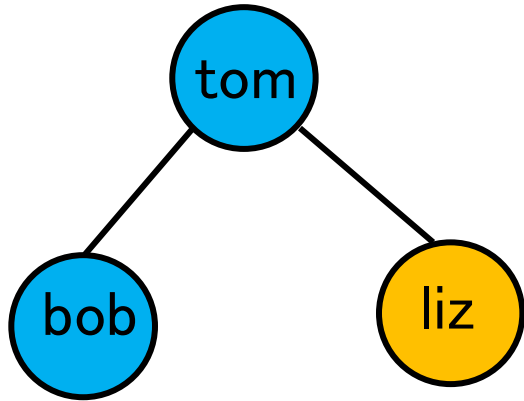
Prolog の記述

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

最後のピリオドに注意



### (3) 引数 0 の述語の定義



parent(tom,bob)

parent(tom,liz)

male(tom)

male(bob)

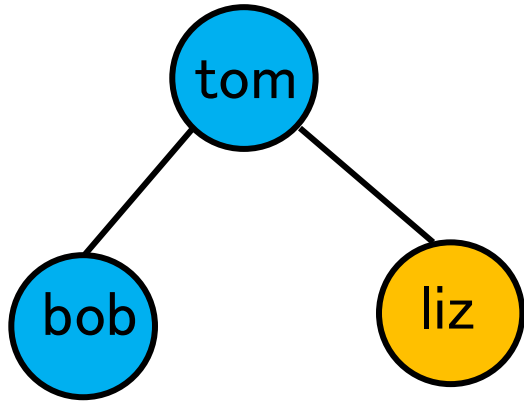
female(liz)

```
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).
```



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)
```

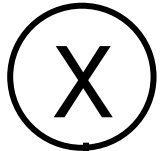
tom が bob の親で、かつ、tom が liz の親であるならば ok である

Prolog の記述

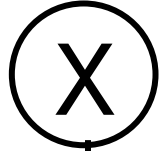
```
tm :- parent(tom,bob).
```

```
ok :- parent(tom,liz), parent(tom,liz).
```

## (4) 祖父の定義



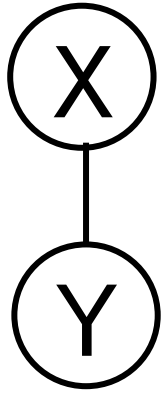
parent(X,Y)



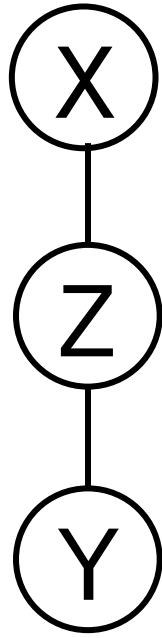
parent(X,Z)

parent(Z,Y)





parent(X,Y)



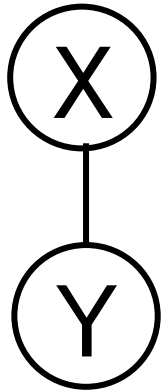
parent(X,Z)  
parent(Z,Y)

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

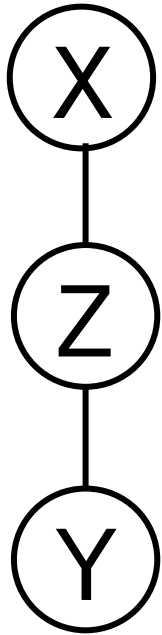
Prolog の記述

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

## (5) 祖先の定義



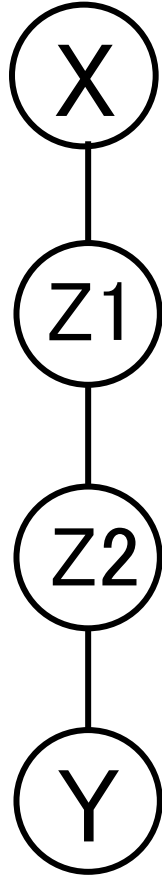
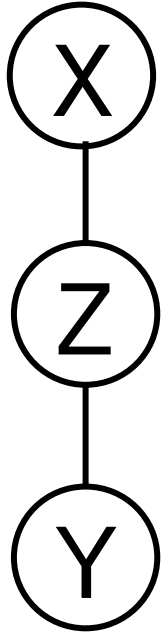
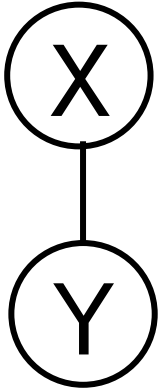
parent(X,Y)



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

ancestor(X,Y) <- parent(X,Z) & parent(Z,Y)

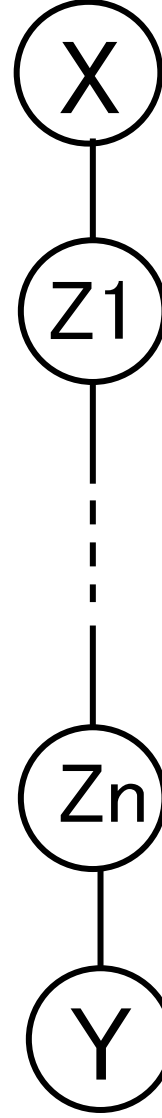
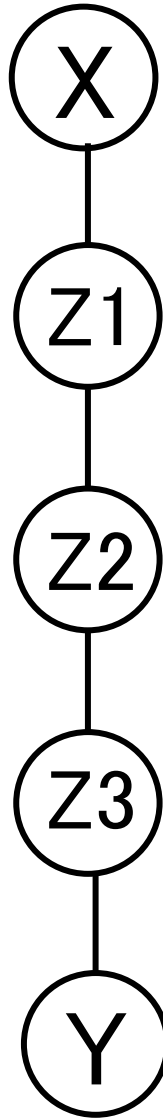
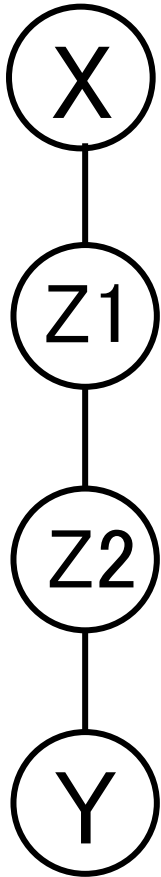
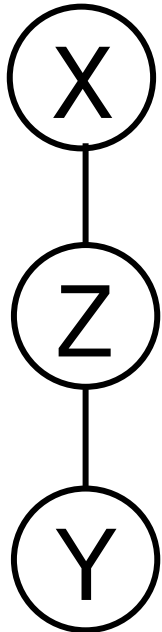
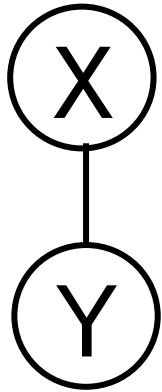
これでいい？

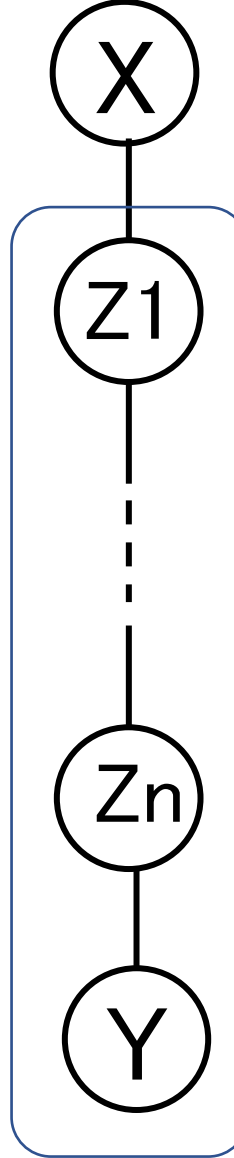
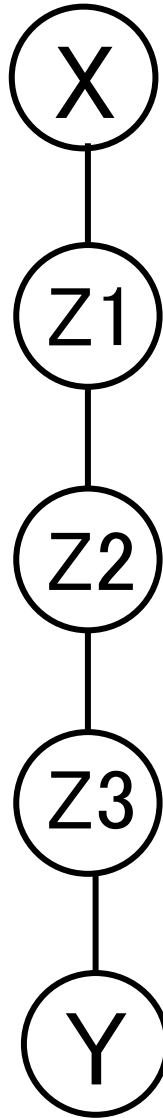
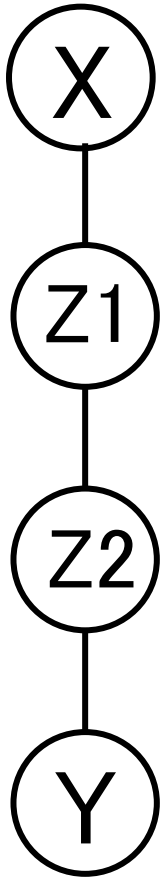
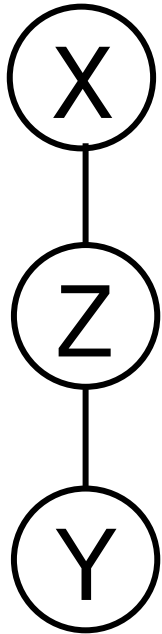
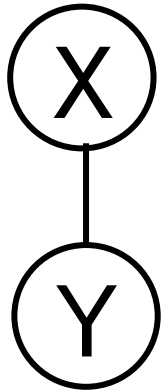


ancestor(X,Y) <-  
parent(X,Z1) & parent(Z1,Z2) & parent(Z2,Y)

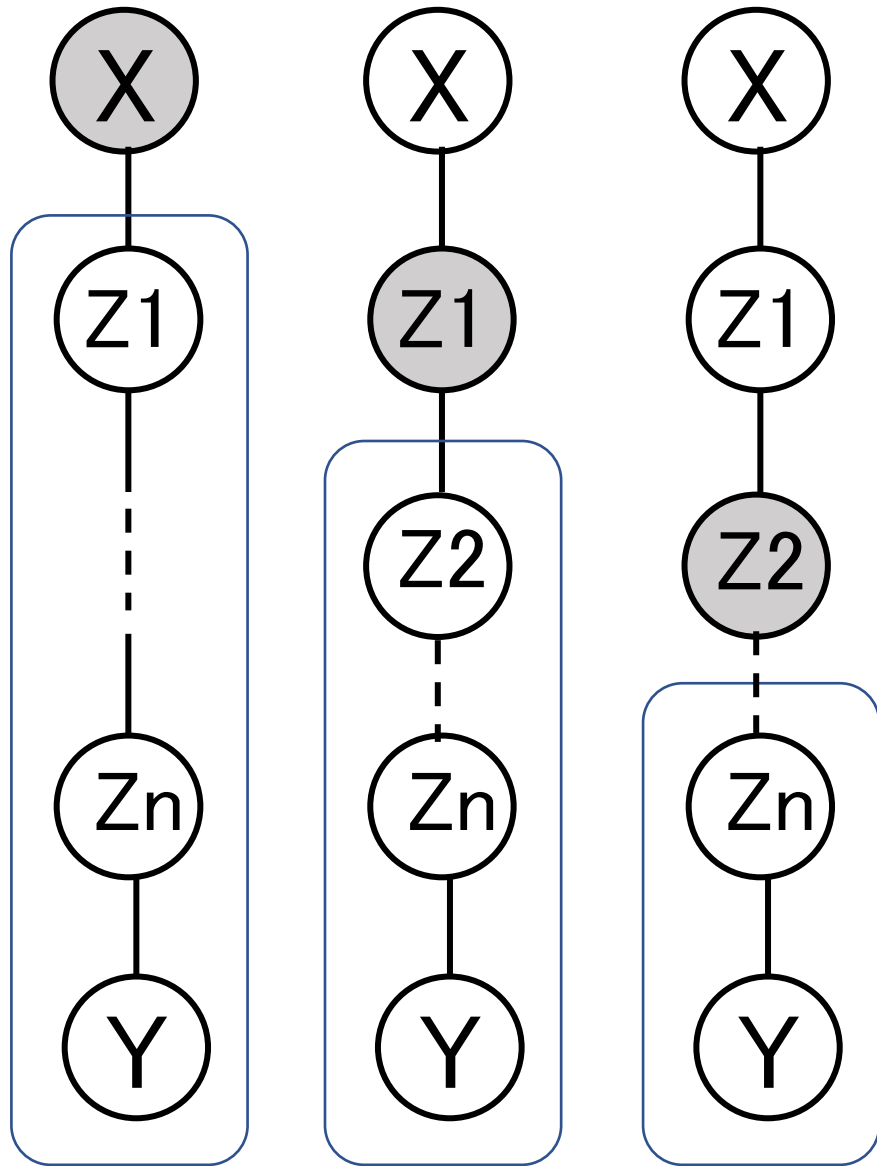
これでいい？

parent(X,Z1)  
parent(Z1,Z2)  
parent(Z2,Y)





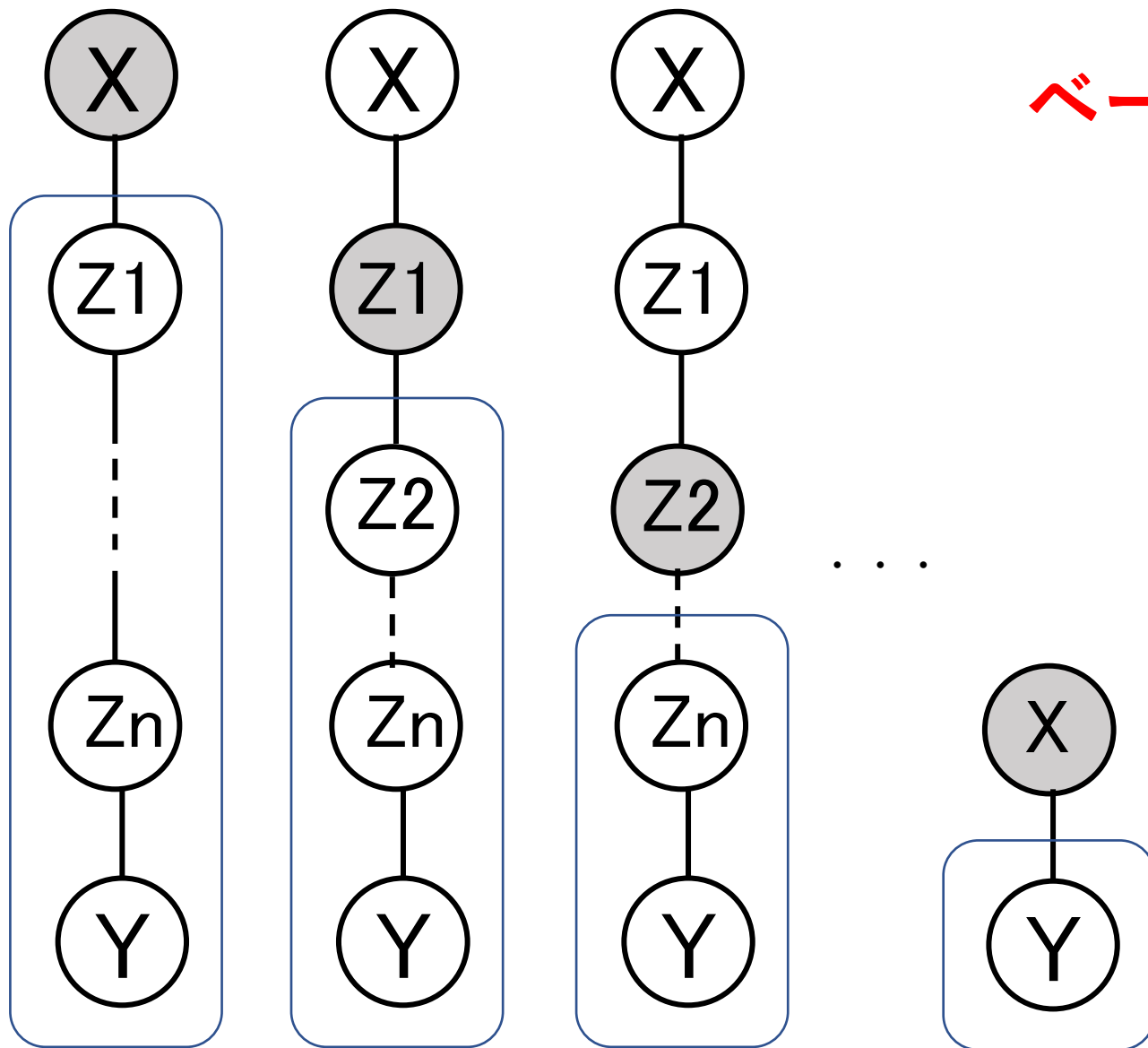
parent(X,Z1)  
ancestor(Z1,Y)



## 再帰的な考え方

parent(X,Z1)  
ancestor(Z1,Y)

ancestor(X,Y) <-  
parent(X,Z1) & ancestor(Z1,Y)

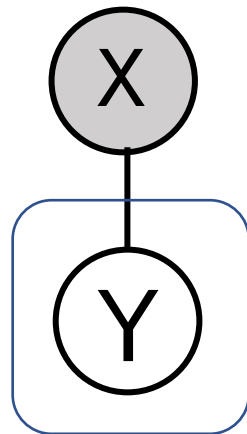
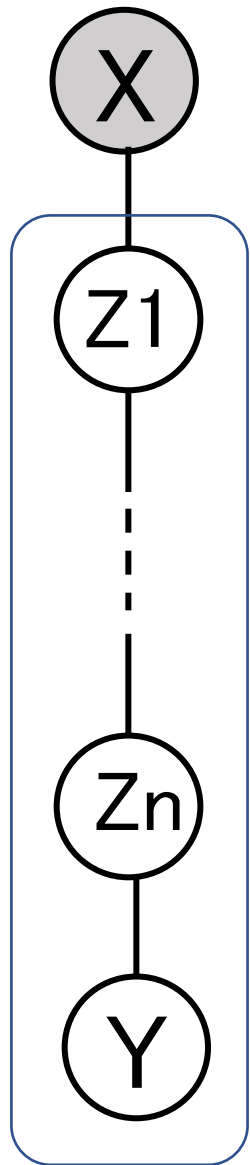


## ベースケースが必要

$\text{ancestor}(X, Y) \leftarrow$   
 $\text{parent}(X, Z1) \ \& \ \text{ancestor}(Z1, Y)$

$\text{ancestor}(X, Y) \leftarrow \text{parent}(X, Y)$





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)

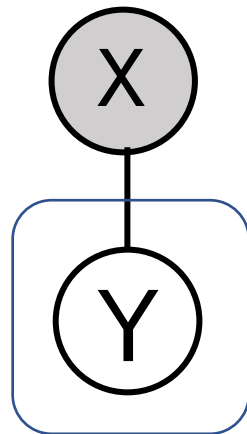
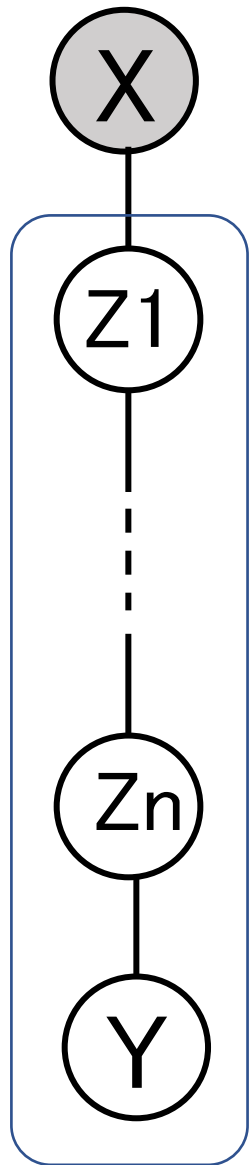
(任意の変数X,Yに対して)  
 XがYの親ならばXはYの祖先である

Prolog の記述

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

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

これでいい？



$\text{ancestor}(X,Y) \leftarrow$   
 $\text{parent}(X,Z) \ \& \ \text{ancestor}(Z,Y)$

(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

$\text{ancestor}(X,Y) \leftarrow \text{parent}(X,Y)$

(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

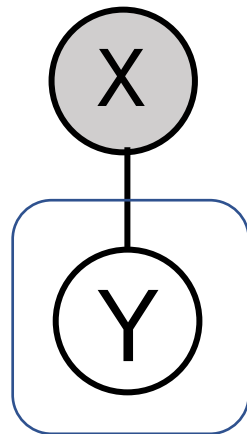
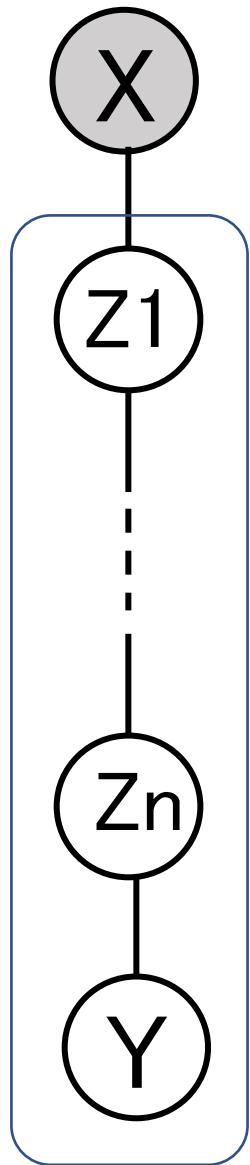
Prolog の記述

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

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

これでいい？

論理的にはOKだが動作としてはまずい



$\text{ancestor}(X,Y) \leftarrow$   
 $\text{parent}(X,Z) \ \& \ \text{ancestor}(Z,Y)$

(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

$\text{ancestor}(X,Y) \leftarrow \text{parent}(X,Y)$

(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

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

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

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