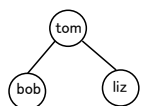


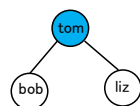
家族関係の記述

(1) データベースの記述

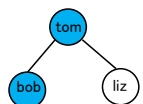


parent(tom,bob)
parent(tom,liz)

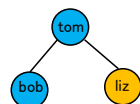
定数、述語は小文字から始める



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



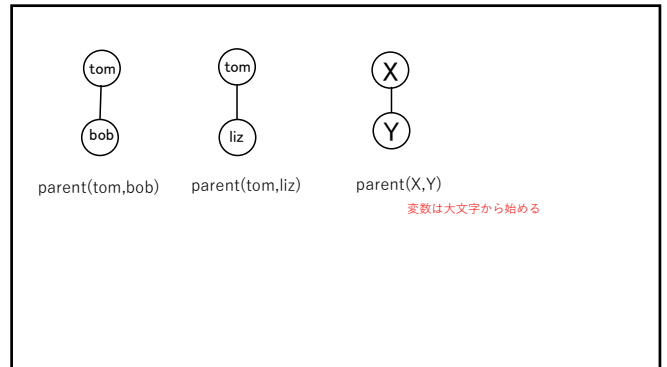
parent(tom,bob)
parent(tom,liz)
male(tom)
male(bob)



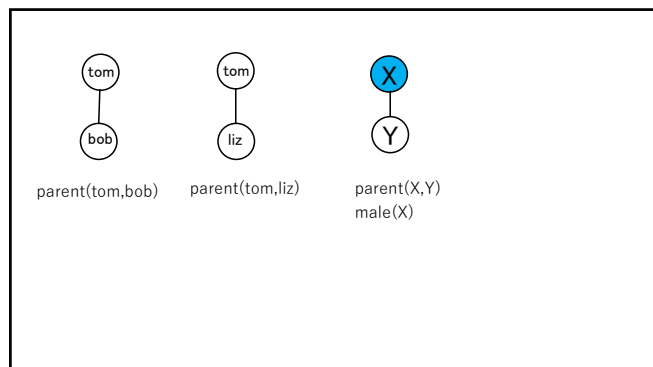
parent(tom,bob)
parent(tom,liz)
male(tom)
male(bob)
female(liz)

(2) 父親の定義

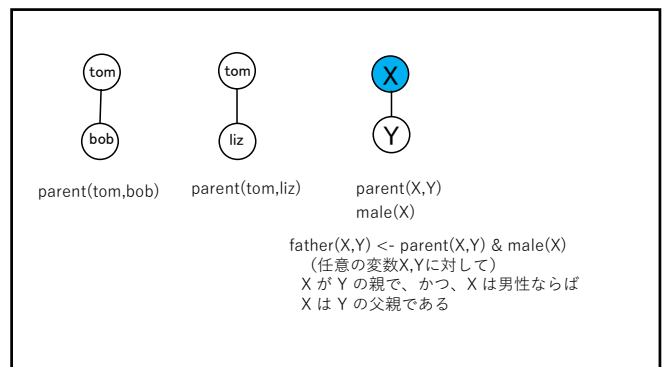
7



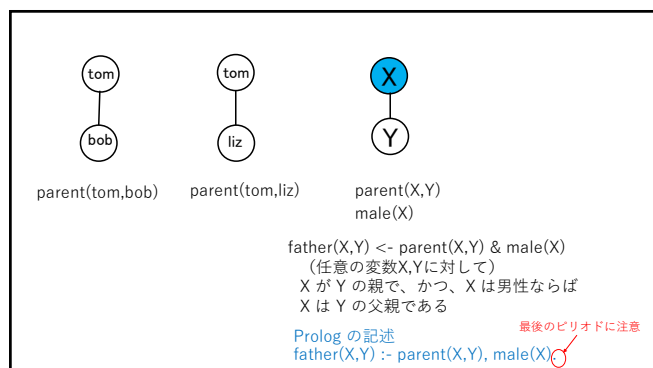
8



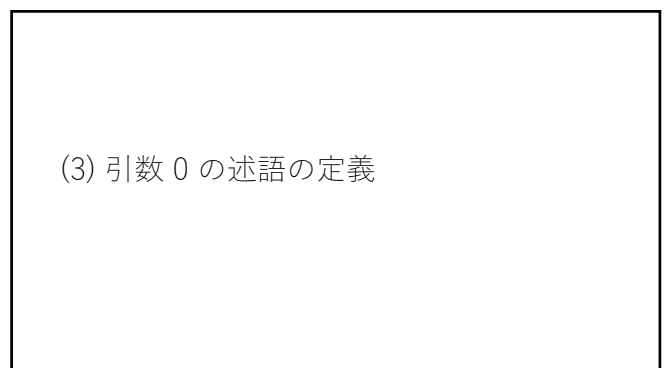
9



10

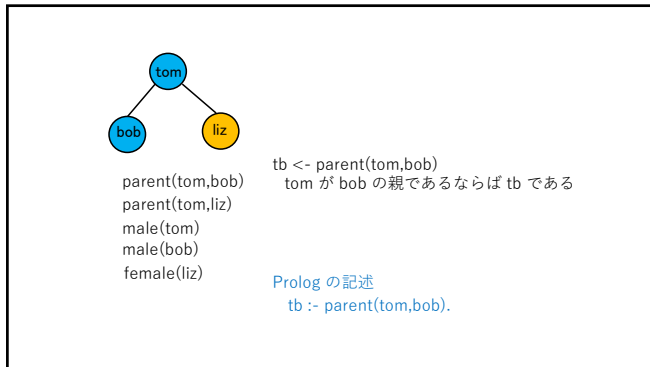


11

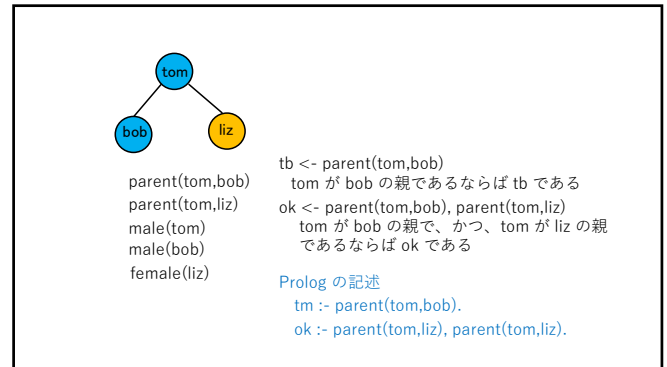


12

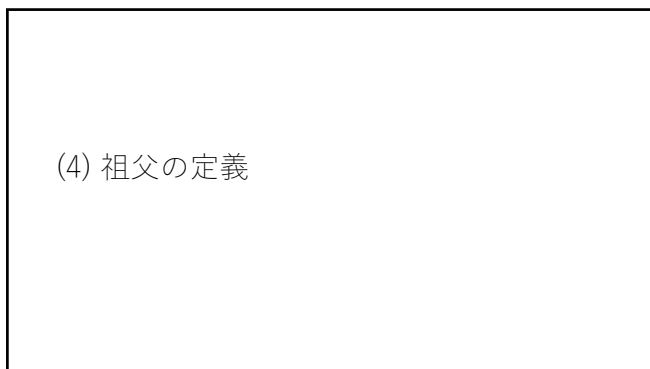
(3) 引数 0 の述語の定義



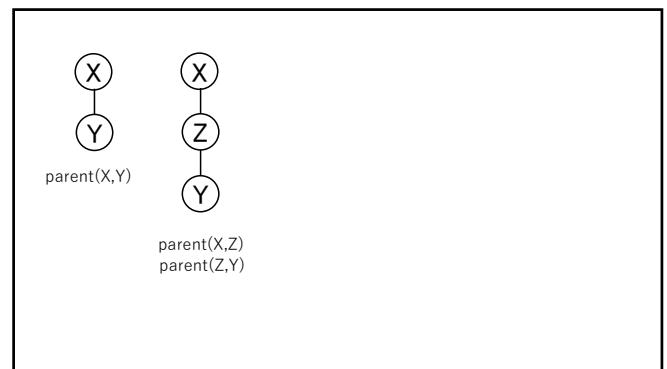
13



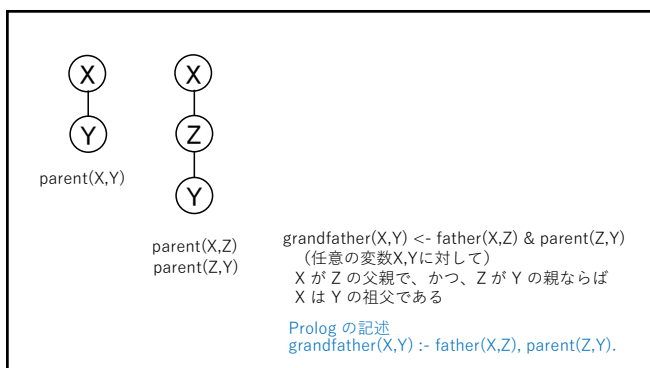
14



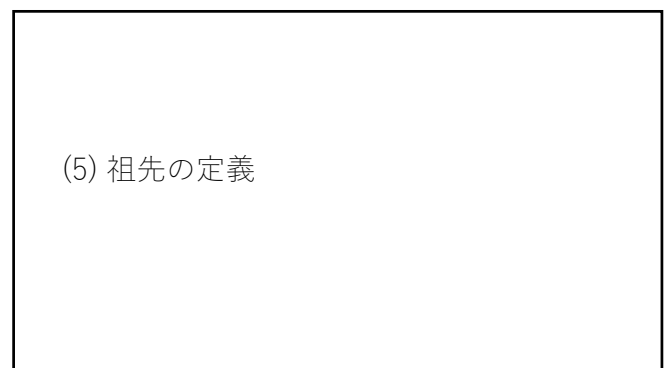
15



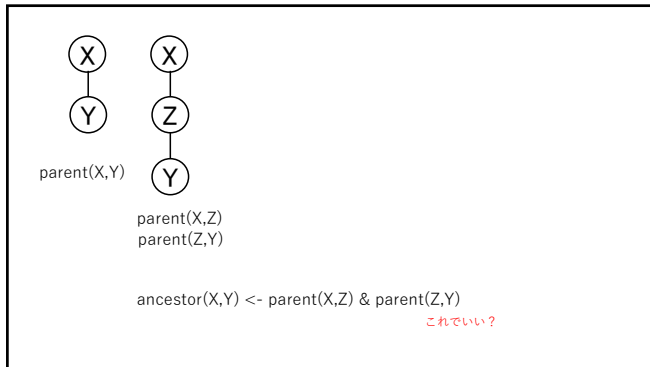
16



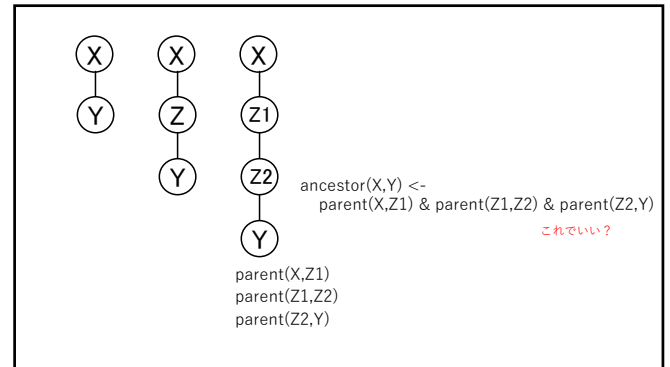
17



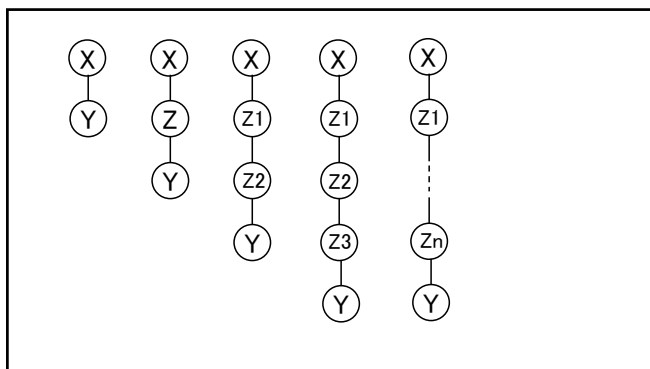
18



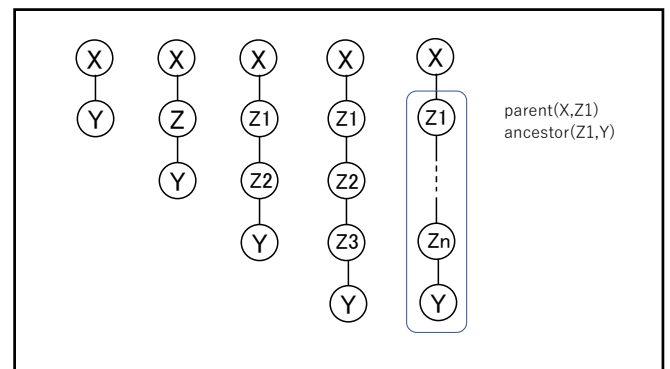
19



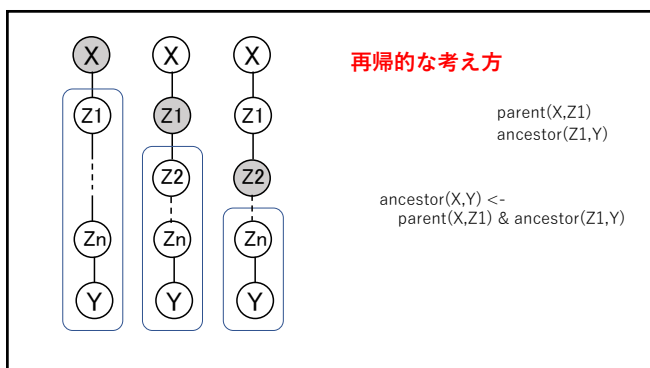
20



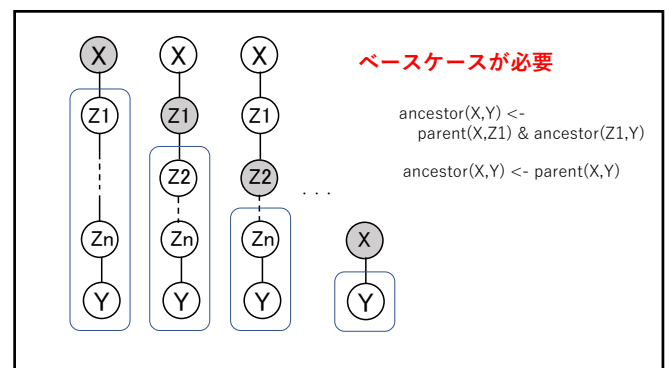
21



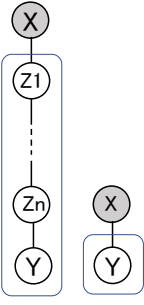
22



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
  
```

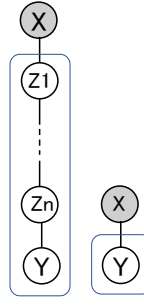
Prolog の記述

```

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
  
```

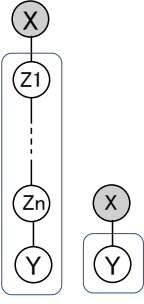
Prolog の記述

```

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