

# Arbre de recherche

- ▶ On parle d'arbre de recherche d'une question
  - \* Racine de l'arbre : question
  - \* Nœuds : points de choix (formule à démontrer)
  - \* Passage d'un nœud vers son fils en considérant l'une des règles et en effectuant une unification et une étape (pas) de démonstration

# Arbre de recherche

- \* Nœuds de gauche à droite dans l'ordre de déclaration des règles
- \* **Nœuds d'échec** : aucune règle ne permet de démontrer la première formule du nœud
- \* **Nœuds de succès** : ne contient plus aucune formule, tout a été démontré et les éléments de solution sont trouvés en remontant vers la racine de l'arbre

# Stratégie de Prolog

- ▶ Pour résoudre une question, Prolog construit l'arbre de recherche de la question
- ▶ Parcours en profondeur d'abord
  - \* nœud de succès : c'est une solution, Prolog l'affiche et cherche d'autres solutions
  - \* nœud d'échec : remontée dans l'arbre jusqu'à un point de choix possédant des branches non explorées

# Stratégie de Prolog

- ▶ On parle de **backtracking**. Si un tel nœud de choix n'existe pas, la démonstration est terminée, il n'y a pas d'autres solutions.
- ▶ Possibilité de branche infinie et donc de recherche sans terminaison...
- ▶ Attention à :
  - \* ordre des littéraux dans la queue de clause
  - \* ordre des clauses

# Un premier exemple

$f(a).$

$f(b).$

$g(a).$

$g(b).$

$h(b).$

$k(X):- f(X), g(X), h(X).$

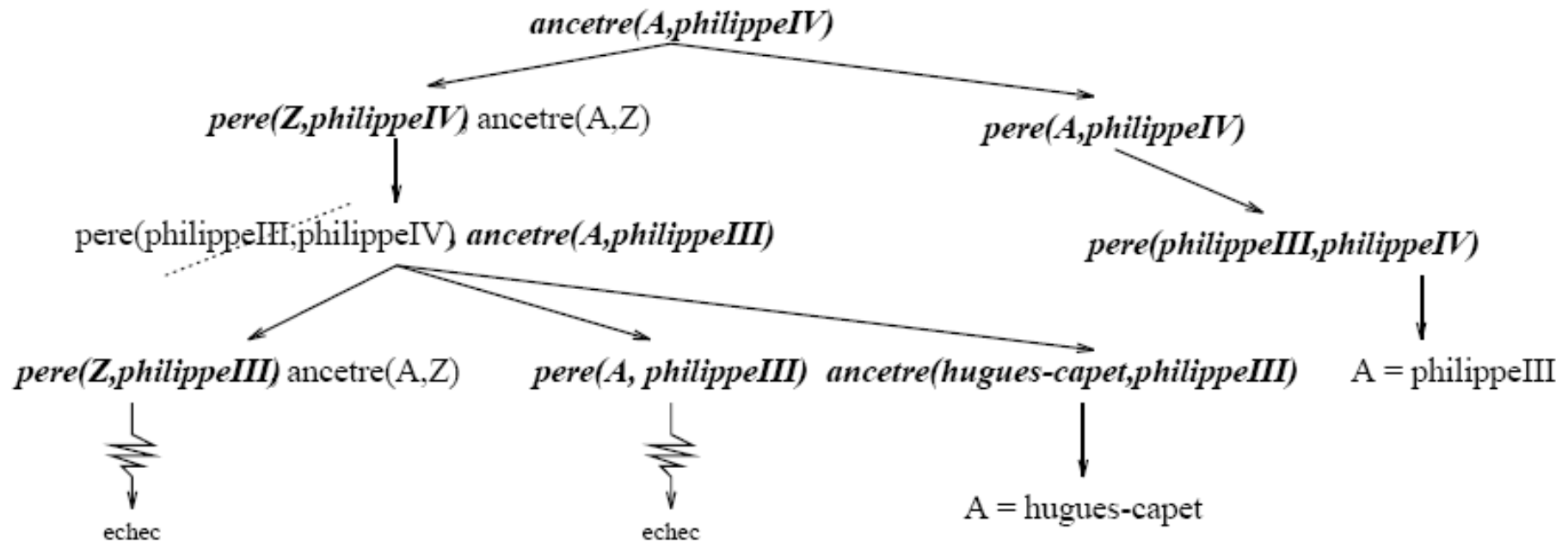
$?- k(Y).$

# Exemple: arbre de recherche

```
ancetre(X,Y) :- pere(Z,Y), ancetre(X,Z).  
ancetre(X,Y) :- pere(X,Y).  
ancetre(hugues_capet,philippeIII).  
pere(philippeIII,philippeIV).
```

```
?- ancetre(A,philippeIV).
```

# Exemple: arbre de recherche



# Un autre exemple (en 3 versions)

```
pere(charles,jean).  
noble(henri).  
noble(louis).  
noble(charles).  
noble(X):- pere(Y,X), noble(Y).
```

```
pere(charles,jean).  
noble(henri).  
noble(louis).  
noble(charles).  
noble(X):- noble(Y), pere(Y,X).
```

```
pere(charles,jean).  
noble(X):- pere(Y,X), noble(Y).  
noble(henri).  
noble(louis).  
noble(charles).
```

?- noble(jean).



# Un dernier exemple

```
aime(vincent,mimi).  
aime(marcel,mimi).  
jaloux(X,Y) :- aime(X,Z),aime(Y,Z).
```

```
?- jaloux(X,Y).
```

*Combien de solutions? ... 4*