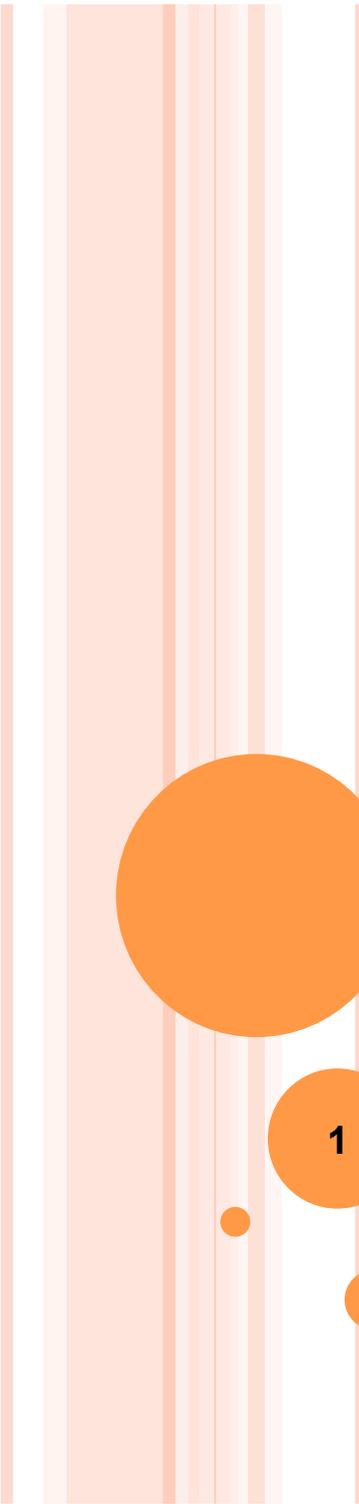


# INTRODUCTION AU PROJET

## UN « QUI EST-CE ? » À BASE DE RÈGLES



1

# CE QUE VOUS ALLEZ RÉALISER

> (quiestce)

Choisissez dans la liste suivante : (tigre guepard zebre)

Est ce que donne-lait? (oui-non) non

Est ce que avec-poils? (oui-non) oui

Est ce que raies-noires? (oui-non) oui

Est ce que predateur? (oui-non) non

Est ce que avec-sabots? (oui-non) oui

zebre

>

# COMMENT LE SYSTÈME RAISONNE-T-IL ?

## Utilisation d'une base de règles :

- SI mammifere ET raies-noires ET carnivore ET couleur-fauve ALORS tigre
- SI donne-lait ALORS mammifere
- SI avec-poils ALORS mammifere
- SI mammifere ET avec-sabots ALORS ongule
- SI mammifere ET ruminant ALORS ongule
- SI predateur ALORS mange-viande
- SI mange-viande ALORS carnivore
- SI mammifere ET carnivore ET taches-noires ET couleur-fauve ALORS guepard
- SI ongule ET raies-noires ALORS zebre

# QU'EST-CE QU'UNE BASE DE RÈGLES ?

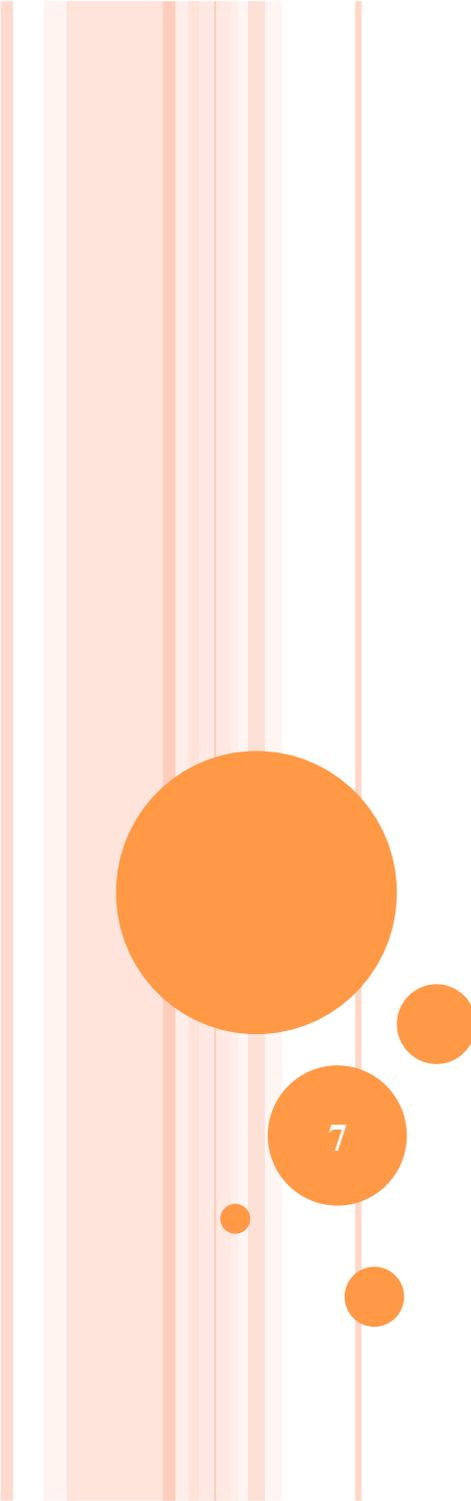
- Chaque règle a une ou plusieurs prémisses, et une seule conclusion
- On distingue dans cette base de règles :
  - des faits terminaux, qui ne figurent jamais dans la partie prémisse des règles
  - des faits observables, qui ne figurent jamais dans la partie conclusion des règles
  - des faits intermédiaires, qui figurent à la fois comme prémisse et comme conclusion
- Remarque : dans un « Qui est-ce ? » classique, il n'y pas de faits intermédiaires

## QUELLE INTERACTION AVEC L'UTILISATEUR ?

- On lui propose la liste des animaux qu'il peut choisir
- On lui pose des questions uniquement sur les faits observables
- On ne doit pas lui poser deux fois la même question

# PRINCIPE DE FONCTIONNEMENT DU SYSTÈME

- Pour chaque animal, on cherche si l'une des règles qui concluent dessus est déclenchable
  - c'est-à-dire si toutes les prémisses de la règle sont vraies
- Pour chaque prémisse
  - Si c'est un fait observable
    - soit on connaît sa valeur (car on l'a déjà demandée)
    - sinon on demande cette valeur à l'utilisateur
  - Sinon on cherche à déclencher une des règles qui permettent de démontrer ce fait
- On est donc dans le cadre d'un processus récursif (quelle surprise...)



# FONCTIONS UTILES

7

# INTERACTIONS AVEC L'UTILISATEUR

- Afficher à l'écran : fonction prédéfinie `display`

Ex :

```
> (display "Bonjour")
Bonjour
> (display 'Bonjour)
bonjour
> (define a 'bonjour)
> (display a)
bonjour
```

- Passer à la ligne : fonction prédéfinie `newline`

Ex : `(newline)`

# INTERACTIONS AVEC L'UTILISATEUR

- Lire au clavier : fonction prédéfinie `read`

Ex :

```
▶ (read)
toto
toto
```

- Utilisation du résultat de la fonction :

```
(let ((reponse (read)))
  (if (eq? reponse 'o)
      'toto
      'tata))
```

# SORTIR DU FONCTIONNEL

- Utilisation de la séquence : fonction prédéfinie `begin`

```
(define exemple
  (lambda ()
    (begin
      (display "Bonjour, souhaitez-vous continuer (o/n)")
      (newline)
      (let ((reponse (read)))
        (if (eq? reponse 'o)
            (exemple)
            (display "Au revoir !"))))))
```

# UTILISATION DE VARIABLES GLOBALES

- Une variable globale est une variable qui est définie en dehors des fonctions
  - (define toto 5)
- Par exemple dans ce projet on pourra gérer avec une variable globale la liste des faits observables qu'on a déjà demandés à l'utilisateur
- Cela évite de passer cette liste en paramètre de toutes les fonctions du programme
- L'initialisation de la variable est faite dans le fichier du programme
- La fonction set! permet de modifier sa valeur
  - (set! toto (+ toto 1))