

Conception et implémentation

IFT-17586 Intelligence artificielle I - A2002

Séance 4

- ◆ Les opérateurs
- ◆ Les listes comme structures de données

© Capus, Potvin et Tourigny, 2003

1

Les opérateurs



$2 + 2$

$a \rightarrow b$

$a \wedge b$

© Capus, Potvin et Tourigny, 2003

2

Les opérateurs

◆ Opérateurs arithmétiques

- Fonctions : **+**, **-**, *****, **/**, **sqrt**, **abs**, etc.
- Seuls les opérateurs arithmétiques forcent l'évaluation des expressions

?- 3+2 **:=** 5. ?- 3+2 **==** 5.

- Opérateur d'évaluation « is » et opérateur d'unification « = »

?- A **is** 2+3. ?- A **=** 2+3.

Les opérateurs

unification		identité		égalité	
=	\=	==	\==	:=	:=
?- 1+1=2. no	?- 1+1\=2. yes	?- 1+1==2. no	?- 1+1\==2. yes	?- 1+1:=2. yes	?- 1+1:=2. no
?- 1 = 1. yes	?- 1 \= 1. no	?- 1 == 1. yes	?- 1 \== 1. no	?- 1 := 1. yes	?- 1 := 1. no
?- 1 = 2. no	?- 1 \= 2. yes	?- 1 == 2. no	?- 1 \== 2. yes	?- 1 := 2. no	?- 1 := 2. yes
?- a = a. yes	?- a \= a. no	?- a == a. yes	?- a \== a. no	?- a := a. Type Error	?- a := a. Type Error
?- a = b. no	?- a \= b. yes	?- a == b. no	?- a \== b. yes	?- a := b. Type Error	?- a := b. Type Error
?- A = A. A = _	?- A \= A. no	?- A == A. A = _	?- A \== A. no	?- A := A. Instantiation Error	?- A := A. Instantiation Error
?- A = B. A = B = _	?- A \= B. no	?- A == B. no	?- A \== B. A = _, B = _	?- A := B. Instantiation Error	?- A := B. Instantiation Error

Les opérateurs

◆ Opérateurs dans WIN-PROLOG : op/3

- Précédence de 0 (forte) à 1200 (faible)
 - Ex: la précédence du « non » est plus forte que celle du « et »
- Types
 - Préfixe (fx, fy) , postfixe (xf, yf) , infixe (xfx, xfy)
 - Exemple préfixe : non a.
 - Exemple postfixe : a est_vrai.
 - Associativité : à droite, à gauche, non-associatif
 - Exemple d'associativité à droite :
 - a et b et c $\Leftrightarrow$ a et (b et c).
 - non non a $\Leftrightarrow$ non (non a).

Les opérateurs

◆ Exemple avec WIN-PROLOG

% sans opérateurs définis

```
non( A ):- A, !, fail.  
non( _ ).
```

```
et( A, B ):- A, B.
```

```
a.  
b.
```

```
| ?- et(a,b). | ?- a et b.
```

```
yes Syntax Error
```

% avec opérateurs définis

```
:- op(200,xfy,et), op(100,fy,non).  
non A :- A, !, fail.  
non _ .
```

```
A et B :- A, B.
```

```
a.  
b.
```

```
| ?- et(a,b). | ?- a et b.
```

```
yes yes
```

Les listes comme structures de données

◆ Exemples de listes :

[a, b, 3, c, pomme, 'a b c']

[]

◆ Accès aux éléments : 2 façons

« , » : élément par élément

?- [A, B, C] = [30, 20, 50].

« | » : premier(s) élément et reste de la liste

?- [X | Y] = [30, 20, 50].

Les listes comme structures de données

◆ Le prédicat membre/2 :

– membre(A,B) est vrai si A est membre de la liste B

%condition d'arrêt

membre(X,[X|_]).

%appel récursif

membre(X,[_|L]):- membre(X,L).

Les listes comme structures de données

◆ Les prédicats ensemblistes

- Construire des listes à partir des arguments des faits connus
- 3 prédicats : bagof/3, setof/3, findall/3
- Particularités :
 - bagof/3 et setof/ tiennent compte de la valeur de tous les arguments
 - findall/3 ne s'occupent pas des arguments non demandés pour la liste
 - setof/3 construit des listes triées

Les listes comme structures de données

◆ Les prédicats ensemblistes

- findall/3 : exemple

```
auto( julie, honda, rouge ).
auto( jean, honda, rouge ).
auto( marc, chrysler, rouge ).
auto( nathalie, honda, noire ).

| ?- findall( P, auto(P, M, C), R ).
P=_, M=_, C=_, R=[julie, jean, marc, nathalie]

| ?- findall( P, auto(P, _, _), R ).
P = _ , R = [julie,jean,marc,nathalie]
```

Les listes comme structures de données

◆ Les prédicats ensemblistes

– bagof/3 : exemple

auto(julie, honda, rouge).
auto(jean, honda, rouge).
auto(marc, chrysler, rouge).
auto(nathalie, honda, noire).

?- bagof(P, auto(P, M, C), R).

P=_, M=honda, C=noire, R=[nathalie] ;

P=_, M= chrysler, C=rouge, R=[marc] ;

P=_, M=honda, C=rouge, R=[julie, jean]

no

© Capus, Potvin et Tourigny, 2003

11

Les listes comme structures de données

◆ Les prédicats ensemblistes

– bagof/3 : exemple

auto(julie, honda, rouge).
auto(jean, honda, rouge).
auto(marc, chrysler, rouge).
auto(nathalie, honda, noire).

?- bagof(P, C^auto(P, M, C), R).

P=_, C=_, M= chrysler, R=[marc] ;

P=_, C=_, M=honda, R=[julie, jean, nathalie] ;

no

^{^2}
quantification
existentielle

⇒ Listes des personnes en fonction de la marque de leur auto

© Capus, Potvin et Tourigny, 2003

12

Les listes comme structures de données

◆ Les prédicats ensemblistes

– bagof/3 : exemple

auto(julie, honda, rouge).

auto(jean, honda, rouge).

auto(marc, chrysler, rouge).

auto(nathalie, honda, noire).

?- bagof(P, M^auto(P, M, C), R).

P = _ , M = _ , C = noire , R = [nathalie] ;

P = _ , M = _ , C = rouge , R = [julie,jean,marc] ;

no

⇒ Listes des personnes en fonction de la couleur de leur auto

© Capus, Potvin et Tourigny, 2003

13

Les listes comme structures de données

◆ Les prédicats ensemblistes

– bagof/3 : exemple

auto(julie, honda, rouge).

auto(jean, honda, rouge).

auto(marc, chrysler, rouge).

auto(nathalie, honda, noire).

?- bagof(P, C^M^auto(P, M, C), R).

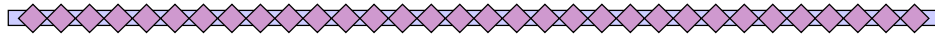
P = _ , C = _ , M = _ , R = [julie,jean,marc,nathalie]

⇒ Listes non triée des personnes qui ont une auto

© Capus, Potvin et Tourigny, 2003

14

Les listes comme structures de données



◆ Les prédicats ensemblistes

– bagof/3 : exemple

auto(julie, honda, rouge).

auto(jean, honda, rouge).

auto(marc, chrysler, rouge).

auto(nathalie, honda, noire).

?- setof(P, auto(P, M, C), R).

P = _ , C = _ , M = _ , R = [jean,julie,marc,nathalie]

⇒ Listes triée des personnes qui ont une auto