

Temas Selectos de IA

PROLOG

M.C Said Zamora

Lenguajes lógicos

- Utilizan constructores lógicos para establecer relaciones.
- Átomos.

PROLOG

- Hechos
- Reglas
- Base de conocimiento.
- Búsquedas.

Base de conocimiento

- mujer(ana).
- mujer(juany).
- mujer(yolanda).
- tocaguitarra(juany).
- fiesta.

Búsquedas

- ?- mujer(ana).
- ?- mujer(genoveva).
- ?- party.

Creación de reglas.

- feliz(yolanda).
- escuchamusica(ana).
- escuchamusica(yolanda):- feliz(yolanda).

Búsquedas

- ?- mujer(X).

Base de conocimiento

- gusta(vicente,ana).
- gusta(marsellus,ana).
- gusta(calabaza,conejodechocolate).
- gusta(conejodechocolate,calabaza).

- celos(X,Y):- gusta(X,Z), gusta(Y,Z).

Archivos .pl

- ?- [kb2].
- ?- ['c:/Documents and Settings/.../Prolog/kb2.pl'].
- ?- listing.

Átomos

- String de caracteres.
- Secuencia arbitraria de caracteres entre comillas simples
- ' & ^ % & # @ \$ & * '
- String de caracteres especiales
- @ = == == > ; :-

Números

- Enteros
- Flotantes

Variables

- X , Y , Variable , _variables , X_526 , Lista , Lista24

Actividad

- VINCENT
- Footmassage
- variable23
- Variable2000
- big_kahuna_burger
- 'big kahuna burger'
- big kahuna burger
- 'Jules'
- _Jules
- '_Jules'

Unificación

- Dos términos se unifican si son el mismo átomo
- $?- =(\text{mia}, \text{mia}).$
- $?- =(\text{mia}, \text{vincent}).$

Términos complejos

- Se construyen a partir de constantes, números y variables.
- Son llamados estructuras.
- Es una función seguido de sus argumentos.

Términos complejos

- `canta(juany)`
- `busca(X,padre(padre(padre(ana))))`

Aridad

- Número de argumentos que posee un termino complejo.
- mujer(ana)

Predicado

- gusta(vincent,mia)
- gusta(vincent,marsellus,mia)
- gusta/2
- gusta/3

Unificación de términos complejos

- Pertenecen al mismo funtor.
- Tienen la misma aridad.
- Sus argumentos pueden ser unificados.
- Las variables deben ser compatibles.

- ?- 2 = 2.

- yes

- ?- mia = vincent.

- No

- ?- 'mia' = mia.

- yes

- ?- '2' = 2.

- no

- ?- mia = X.

- X = mia

- yes

- $\text{?- } k(s(g), Y) = k(X, t(k)).$

- $X = s(g)$

- $Y = t(k)$

Recursividad

- `digiere(X,Y) :- hacomido(X,Y).`
- `digiere(X,Y) :-`
 - `hacomido(X,Z),`
 - `digiere(Z,Y).`
- `hacomido(mosquito,sangre(cacerolo)).`
- `hacomido(rana,mosquito).`
- `hacomido(caiman,rana).`

Significado Declarativo y Procedimental

- Lo que dice, lo que significa como declaraciones lógicas.
- Si X digiere a Y, Prolog puede utilizar esta regla para preguntar si X se ha comido a Y.

- ?- digiere(caiman,mosquito).
- ?- hacomido(caiman,mosquito).
- Al unificar caimán con X y mosquito con Y
- ?- hacomido(caiman,Z),
- digiere(Z,mosquito).

- ?- hacomido(caiman,Z).
- ?- digiere(Z,mosquito).
- ?- hacomido(caiman,rana).
- ?- digiere(rana,mosquito).
- ?- hacomido(rana,mosquito).

Árbol genealógico

- `hijo(anne,bridget).`
- `hijo(bridget,caroline).`
- `hijo(caroline,donna).`
- `hijo(donna,emily).`
- `descendiente(X,Y) :- hijo(X,Y).`
- `descendiente(X,Y) :- hijo(X,Z),`
 `descendiente(Z,Y).`