



INSTITUTO POLITECNICO NACIONAL



ESCUELA SUPERIOR DE CÓMPUTO

FUNDAMENTOS DE INTELIGENCIA ARTIFICIAL

Prof. Hernández Cruz Macario

PRÁCTICA 06

4BM1

Corona Reyes Mauricio Dassel

Martinez Méndez Diego

Pacheco Sánchez Rodrigo

Ingeniería en Inteligencia Artificial

Resumen:

Nuestra práctica consiste en desarrollar un sistema experto con información de distintas especies de animales, con el fin de consultar e imprimir dicha información empleando reglas establecidas en un archivo prolog. Nuestro programa fue desarrollado empleando las herramientas de la biblioteca de Python, Tkinter para desarrollar la interfaz de usuario y que la experiencia con nuestro programa fuera lo más cómoda y sencilla posible.

Introducción:

En el presente trabajo desarrollamos un programa empleando el lenguaje de programación Python y Prolog para la elaboración de un sistema experto combinando ambas tecnologías.

Prolog es un lenguaje de programación basado en el paradigma de programación lógica, asociado en muchas ocasiones con la inteligencia artificial y el lenguaje computacional, el nombre de este lenguaje de programación proviene de combinar las palabras "PROgramación" y "LÓGica", prolog.

Se trata de un lenguaje de programación declarativo, funciona con base en hechos y reglas para dar sentido y "lógica" a su razonamiento en la ejecución de programas a partir de relaciones lógicas.

Los sistemas expertos según la definición del profesor Edward Feigenbaum, se tratan de programas inteligentes para ordenadores que emplean conocimiento humano y procedimientos inferenciales en la resolución de problemas.

Los sistemas expertos están compuestos por hechos y heurísticas, donde los hechos se tratan de cuerpos de información, mientras que los heurísticos son reglas de razonamiento plausible en las que se basa el sistema para obtener una solución al problema planteado.

Para nuestro programa, nosotros hemos ingresado tanto los hechos como las heurísticas dentro de nuestro programa prolog, para facilitar el razonamiento de la consulta creamos reglas de inferencia basadas en los hechos, finalmente nuestro código en Python obtendrá el tipo de especie que el usuario ingrese y obtendrá su información a partir del programa prolog.

Desarrollo:

Comenzando con la descripción de nuestro programa, elegimos el lenguaje de programación python para desarrollar nuestra práctica y utilizamos la biblioteca Tkinter para poderlo implementar, puesto que ofrece diversas herramientas que facilitaron la elaboración de nuestro programa y su interfaz gráfica.

A continuación, presentamos las reglas de inferencia propuestas en nuestro archivo prolog así como la información ingresada en el mismo archivo que servirá para que nuestro sistema experto busque a través del nombre científico del animal y pueda mostrar la información más adelante en nuestro programa Python.

```
frame(elephantidae, subclase_de(proboscidea), propiedades([son(elefantes)])).
frame(equidae, subclase_de(perissodactyla), propiedades([son(compatibles)])).
frame(rhinocerotidae, subclase_de(perissodactyla), propiedades([tien(cuerno)])).
frame(ursidae, subclase_de(carnivora), propiedades([son(osos)])).
frame(noctilionoidea, subclase_de(chiroptera), propiedades([es(pescador)])).
frame(animal, subclase_de(objeto), propiedades([puede(sentir), puede(respirar)])).
frame(mamifero, subclase_de(animal), propiedades([puede(mamar), respira(aire)])).
frame(artiodactilo, subclase_de(mamifero), propiedades([tiene(pesugnas_pares), comen(plantas)])).
frame(carnivora, subclase_de(mamifero), propiedades([comen(carne)])).
frame(primates, subclase_de(mamifero), propiedades([tiene(cerebro_desarrollado)])).
frame(proboscidea, subclase_de(mamifero), propiedades([es(grande)])).
frame(perissodactyla, subclase_de(mamifero), propiedades([tiene(pesugnas_impares)])).
frame(chiroptera, subclase_de(mamifero), propiedades([tien(alas), es(roedor)])).
frame(camelidos, subclase_de(artiodactilo), propiedades([familia_de(camellos)])).
frame(canidae, subclase_de(carnivora), propiedades([puede(comer_vegetales)])).
frame(suidae, subclase_de(artiodactilo), propiedades([son(inteligentes)])).
frame(hominidae, subclase_de(primates), propiedades([son(grandes_simios)])).
frame(felidae, subclase_de(carnivora), propiedades([son(felinos)])).

frame(elefante_asiatico, subclase_de(elephantidae), propiedades([nombre_cientifico(elephas_maximus),
imagen('elefante.png')])).
frame(caballo, subclase_de(equidae), propiedades([nombre_cientifico(equus_caballus), ruido(relincha),
imagen('caballo.png')])).
frame(rhinoceronte_blanco, subclase_de(rhinocerotidae), propiedades([nombre_cientifico(ceratotherium_simum),
imagen('rino.png')])).
frame(oso_pardo, subclase_de(ursidae), propiedades([nombre_cientifico(ursus_arctos), pelaje(marron),
imagen('pardo.png')])).
frame(oso_polar, subclase_de(ursidae), propiedades([nombre_cientifico(ursus_maritimus), pelaje(blanco),
imagen('polar.png')])).
frame(panda, subclase_de(ursidae), propiedades([nombre_cientifico(ailuropoda_melanoleuca),
pelaje(blanco_y_negro), imagen('panda.png')])).
frame(murcielago, subclase_de(noctilionoidea), propiedades([nombre_cientifico(noctilio_albiventris),
imagen('dracula.png')])).
frame(vicugna, subclase_de(camelidos), propiedades([nombre_cientifico(vicugna_vicugna),
imagen('vicugna_vicugna.png')])).
frame(guanaco, subclase_de(camelidos), propiedades([nombre_cientifico(lama_guanicoe),
imagen('guanaco.png')])).
frame(llama, subclase_de(camelidos), propiedades([nombre_cientifico(lama_pacos), imagen('llama.png')])).

frame(perro, subclase_de(canidae), propiedades([nombre_cientifico(canis_lupus_familiaris),
imagen('guagua.png')])).
frame(lobo_rojo, subclase_de(canidae), propiedades([nombre_cientifico(canis_rufus),
imagen('lobo_rojo.png')])).
frame(coyote, subclase_de(canidae), propiedades([nombre_cientifico(canis_latrans), imagen('coyote.png')])).

frame(babirusa, subclase_de(suidae), propiedades([nombre_cientifico(babirusa_babirusa),
imagen('babirusa.png')])).
frame(jabali, subclase_de(suidae), propiedades([nombre_cientifico(puma_o_sus_scrofa),
imagen('jabali.png')])).
frame(chimpanzee, subclase_de(hominidae), propiedades([nombre_cientifico(pan_troglodytes), vive_en(selvas),
imagen('chimpanzee.png')])).
frame(gorila, subclase_de(hominidae), propiedades([nombre_cientifico(gorilla_gorilla),
vive_en(bosques_costeros), imagen('gorila.png')])).
frame(orangutan, subclase_de(hominidae), propiedades([nombre_cientifico(pongo_pygmaeus), vive_en(selvas),
imagen('orangutan.png')])).
frame(puma, subclase_de(felidae), propiedades([nombre_cientifico(puma_concolor), es(pequeña),
emite(maullidos), imagen('puma.png')])).
frame(leopardo, subclase_de(felidae), propiedades([nombre_cientifico(panthera_pardus), es(rapido),
imagen('leopardo.png')])).
frame(gato_montes, subclase_de(felidae), propiedades([nombre_cientifico(leopardus_geoffroyi), es(chiquito),
imagen('montes.png')])).

que_es(X):-((instancia(X),es(Clase,X));
(subclase(X,subclase_de(Clase)),Clase=objeto,write('Es '),writeln(Clase),fail.
es(Clase,obj):- frame(obj,instancia_de(Clase),_).
es(Clase,obj):- frame(obj,instancia_de(Clase),_),subclase(Clase,obj).
subclase(C1,C2):- frame(C1,subclase_de(C2),_).
subclase(C1,C2):- frame(C1,subclase_de(C3),_),subclase(C3,C2).
subclase(X):-frame(X,subclase_de(_),_).
instancia(X):-frame(X,instancia_de(_),_).
propiedades(objeto):-!.
propiedades(X):-frame(X,subclase_de(Y),propiedades(Z)),imprime(Z),propiedades(Y).
propiedades(X):-frame(X,instancia_de(Y),propiedades(Z)),imprime(Z),propiedades(Y).
props(X):-((instancia(X),propiedades(X));subclase(X),propiedades(X)).
imprime([]):-!.
imprime([H|T]):-write(H),imprime(T).
about(X):-que_es(X);props(X).
```

Iniciando con la descripción de nuestro código en Python, comenzamos estableciendo todos los Labels o elementos de nuestra interfaz tales como el texto, nombre del equipo, los animales disponibles en nuestro programa, etc.

Para poder distribuirlos adecuadamente utilizamos el método `.grid()`, propio de la biblioteca Tkinter.

```
1 from tkinter import *
2 import os
3
4 root = Tk()
5 root.title("Práctica 06 - FIA")
6 root.iconbitmap("images/pk.ico")
7 root.maxsize(700,770)
8
9 especieslista="\npuma\nleopardo\ngato_montes\nelefante_asiatico\ncaballo\nrinoceronte_blanco\noso_pardo\noso_polar\npanda\nmurcielago\nvicuña\nguanaco\nllama\nperu
10
11 Label(root, text="PRACTICA 06 - FIA 48H1 \nEQUIPO:", bg="white", font=("Helvetica Bold", 25)).grid(pady=5, row=0, column=0, columnspan=7)
12 Label(root, text="CORONA REYES MAURICIO DASSEL\nMARTINEZ MENDEZ DIEGO\nPACHECO SANCHEZ RODRIGO\n", bg="white", font=("Helvetica Bold", 15)).grid( pady=5, row=1, co
13 Label(root, text="Ingresa una especie: ", bg="white", font=("Helvetica Bold", 15)).grid( pady=5, row=2, column=0)
14
15
16 Label(root, text="Especies disponibles: ", bg="white", font=("Arial bold", 15)).grid(pady=5, row=2, column=3)
17 listaesp = Label(root, justify=LEFT, text=especieslista)
18 listaesp.config(bg="white")
19 listaesp.grid(pady=5, row=1, rowspan = 7, column= 3)
20
21 especieInput = Entry(root, width=30)
22 especieInput.config(highlightbackground="gray", highlightcolor="black", highlightthickness=2)
23 especieInput.grid(pady=20, row=3, column=0)
24
25 textResult = Text(
26     width=50,
27     height=5,
28     font=("Arial", 12)
29 )
30 textResult.grid(pady=5, row=5, column=0)
31
32 image_label = Label(root)
33 image_label.grid(pady=40, row=6, column=0)
34 image_path = "images/pk.png"
35 image = PhotoImage(file=image_path)
36 image_label.image = image
37 image_label.configure(image=image, height=200, width=200, bg="white")
38
39 textoaux = "Escriba un animal del listado!"
```

A continuación, tenemos a la función principal que realiza la consulta de nuestro programa prolog a partir del texto ingresado por el usuario.

Para mostrar la imagen del animal en cuestión, recorreremos el texto proveniente de nuestro “output.txt” hasta encontrar el atributo de “imagen” donde leeremos la ruta de la imagen y la insertaremos en el label correspondiente.

Ejecución del programa:

PRACTICA 06 - FIA 4BM1

EQUIPO:


**CORONA REYES MAURICIO DASSEL
MARTINEZ MENDEZ DIEGO
PACHECO SANCHEZ RODRIGO**

Ingresar una especie:

Buscar

Especies disponibles:

- puma
- leopardo
- gato_montes
- ratoneta_arctica
- caballo
- rinoceronte_blanco
- oso_pardo
- oso_polar
- pantera
- morsa(ciego)
- viciogna
- guinea
- fennec
- perro
- tobo_ngo
- coyote
- babirusa
- javalí
- chimpancé
- gorila
- orangutan



Podemos ver que para facilitar la búsqueda contamos en la parte derecha con un listado de los animales disponibles a consultar.

Práctica 06 - FIA

—

□

×

PRACTICA 06 - FIA 4BM1


EQUIPO:

CORONA REYES MAURICIO DASSEL
MARTINEZ MENDEZ DIEGO
PACHECO SANCHEZ RODRIGO

Ingresa una especie:

Buscar

Es primates
Es mamifero
Es animal
nombre cientifico gorilla gorilla
vive en bosques costeros
..



Especies disponibles:

puma
leopardo
gato_montes
elefante_asiatico
caballo
rinoceronte_blanco
oso_pardo
oso_polar
panda
murcielago
vicugna
guanaco
llama
perro
lobo_rojo
coyote
babirusa
jabalí
chimpancé
gorila
orangutan

Al escribir el nombre de algún animal en el listado y dar click en el botón buscar, podemos ver como la información de dicho animal, así como una imagen ilustrativa al mismo, es desplegada en nuestro programa.

Incluso, al intentar buscar un animal que no se encuentre en el listado, aparece un mensaje de error que indica lo siguiente:

Práctica 06 - FIA

PRACTICA 06 - FIA 4BM1

EQUIPO:

CORONA REYES MAURICIO DASSEL
MARTINEZ MENDEZ DIEGO
PACHECO SANCHEZ RODRIGO

Ingresa una especie:

unicornio

Buscar

Especies disponibles:

puma
leopardo
gato_montes
elefante_asiatico
caballo
rinoceronte_blanco
oso_pardo
oso_polar
panda
murcielago
vicugna
guanaco
llama
perro
lobo_rojo
coyote
babirusa
jabali
chimpancé
gorila
orangutan

Escriba un animal del listado!



Conclusiones:

Como pudimos aprender, los sistemas expertos se tratan de programas bastante útiles para la resolución de problemas, capaces de razonar por medio de relaciones lógicas y fundamentados en el conocimiento que les proveamos. Al investigar un poco más aprendimos que los sistemas expertos recientemente diseñados buscan crear tomas de decisiones más eficaces y reducir el tiempo de decisión, actualmente aún no se considera que superen a expertos humanos en la toma de decisiones, pero se espera que estos sistemas puedan aportar técnicas de inducción y razonamiento complementarias a las propuestas por expertos humanos y así ayudar a una toma de decisiones más eficaz colaborando tanto la máquina como el humano.

Referencias

- Harmon, N. P., & King, D. R. (1985). *Expert Systems: Artificial Intelligence in Business*. John Wiley & Sons.
- Clocksin, W. F., & Mellish, C. S. (1990). *Programmieren in Prolog* (H. Sterner, Trad.; 1990a ed.). Springer.