Sistemas Expertos deberes

Prolog

Prolog es un lenguaje conversacional, el sistema Prolog mantiene un diálogo continuo con el programador desde el inicio de la sesión hasta el final de la misma. Este diálogo toma generalmente la forma de un interrogatorio, a lo largo del cual el programador planteará preguntas al sistema Prolog. Por su parte, el sistema Prolog responderá cada una de las preguntas formuladas por el programador en la medida en que esto sea posible

Prolog le indica al programador que está esperando a que éste le formule una pregunta mostrando en pantalla el siguiente símbolo ?-

Base de conocimiento de Prolog

Prolog consulta una base de conocimiento. Al iniciar una sesión Prolog, esta base de conocimiento almacena un conocimiento básico que incluye, entre otras cosas, conceptos y definiciones de la aritmética de los números naturales

Predicados sobre directorios y ficheros

Los programas Prolog se almacenarán en ficheros de texto (con extensión '.pl'). Prolog adquiere nuevos conocimientos consultando (es decir, leyendo) estos programas. Para facilitar al programador el acceso a los programas almacenados en los ficheros, Prolog define un conjunto de predicados especiales que permiten navegar por el sistema de ficheros y visualizar los directorios.

Creación, ejecución, modificación y consulta de programas Prolog

El primer paso para escribir un programa Prolog consiste en crear el fichero que lo almacenará. Esto se hace mediante el predicado crea, que recibe cuatro parámetros: el nombre del fichero a crear, el nombre del autor, el curso y el código de la máquina en que se está trabajando

Ejecutar un programa Prolog consiste realmente en formular una pregunta a la que Prolog intentará responder haciendo uso del programa (base de conocimientos). Por ejemplo, para “ejecutar” el programa ‘patos.pl’ podemos plantear a Prolog la siguiente cuestión ?- esPato(lucas).

editar el fichero 'patos.pl'. Puesto que el fichero ya existe, no es necesario que empleemos el predicado crea. En esta ocasión editaremos el fichero mediante el predicado edit, que recibe como parámetro el nombre del fichero a editar

Un punto importante a destacar es que Prolog no recuerda lo que aprendió en las sesiones anteriores. Cada vez que se inicia una nueva sesión, Prolog parte sólo del conocimiento básico, independientemente del conocimiento que hubiera adquirido en las sesiones anteriores. Así, si damos por terminada (mediante el predicado halt) la sesión en que enseñamos a Prolog acerca de los patos

DENDRAL

Es un sistema experto que permite resolver la cuestión planteada anteriormente a través de un proceso de búsqueda de generación y prueba jerárquica que se divide en tres partes funcionales: plan, generación y prueba. Su base de conocimientos se desglosa en dos conjuntos de reglas correspondientes a cada una de las fases de desarrollo del sistema

Área de trabajo

Consistía en calcular todos los compuestos que podían dan lugar al número másico de la molécula inicial, teniendo en cuenta el número másico de cada uno de los átomos en los que se dividía el compuesto y las restricciones de valencia. Estas restricciones permitieron podar el árbol de posibles soluciones rápidamente, reduciendo el coste computacional de la búsqueda exhaustiva que se estaba realizando

Intentaba modelar el procedimiento inferencial del experto químico para encontrar la estructura molecular de la combinación que se consideraba solución: representar dicha estructura en forma de grafo.

En que lenguaje esta basado

Estaba escrito en el lenguaje de programación LISP , que se consideraba el lenguaje de la IA debido a su flexibilidad. Dendral

Clips

CLIPS es una herramienta que provee un entorno de desarrollo para la producción y ejecución de sistemas expertos. Fue creado a partir de 1984, en el Lyndon B. Johnson Space Center de la NASA. Los fondos cesaron a principios de los años 1990, y hubo un mandato de la NASA para comprar software comercial.

Como otros lenguajes para sistemas expertos, CLIPS estructura el conocimiento en hechos y reglas. Los hechos son información sobre el entorno que se usa para razonar. Mientras que las reglas son los elementos que permiten que el sistema evolucione, normalmente modificando hechos.

En que lenguaje esta basado

CLIPS incorpora un completo lenguaje orientado a objetos (COOL) para la elaboración de sistemas expertos. Aunque está escrito en C, su interfaz más próxima se parece a LISP

CADUCEUS

CADUCEUS fue un sistema experto médico programado para realizar diagnósticos en medicina interna. Su nombre deriva de Caduceo, un vocablo de origen griego (κηρύκειο) relacionado con la mitología.

Área de trabajo

Si bien CADUCEUS trabajó utilizando un motor de inferencia similar al de MYCIN, realizó una serie de cambios para hacer frente a la complejidad adicional de las enfermedades internas; puede haber varias enfermedades simultáneas y los datos generalmente son defectuosos y escasos

En que lenguaje esta basado

Esta herramienta está basada en el lenguaje de programación

Ejercicio 2

Suponga que una agencia de viaje desea que diseñemos un S.E. para ayudar a los clientes a elegir un destino vacacional. Analizar lo siguiente

Es un problema adecuado para la implementación de un SE

Si, ya que en base a recomendaciones de otros usuarios o viajes frecuentes de otros clientes a cierto destino se podria diseñar un SE que en base a su presupuesto, fecha, motivo de viaje y tiempo de vuelo, se pueda elegir un destino acorde a los gustos del cliente.

Plantee una idea de cómo realizaría el proceso de adquisición de conocimientos

Se empezaría recopilando datos como fecha de vuelo, presupuesto, lugar turístico referente a ocasiones especiales (luna de miel, camping, exploración, etc) y lugares favoritos.

Que estructura lógica cree que podría usar para implementar dicho sistema

En base a la problemática de covid es factible realizar un SE para diagnosticar su contagio

Si, ya que al conocer la forma de trasmisión y los síntomas que se presentan podrían ayudar a diagnosticar la enfermedad en el paciente.

```
import math
```

```
import re


def vcedula(texto):
   if len(cedula) != 10:
      raise Exception("Error numero de cedula incompleto")
   else:
      multiplicador = [2, 1, 2, 1, 2, 1, 2, 1, 2]
      ced_array = list(map(lambda k: int(k), list(cedula)))[0:9]
      ultimo_digito = int(cedula[9])
      resultado = []
      arr = map(lambda x, j: (x, j), ced_array, multiplicador)
      for (i, j) in arr:
         if i * j < 10:
            resultado.append(i * j)
         else:
            resultado.append((i * j)-9)
      if ultimo_digito == int(math.ceil(float(sum(resultado)) / 10) * 10) - sum(resultado):
         return True
      else:
         return False


print("Elija una opcion")
print("1.Crear nuevo usuario")
print("2.borrar usuario")
print("3.actualizar usuario")
print("4.Buscar usuario")
print("5.Crear nuevo numero")
print("6.borrar numero")
print("7.actualizar numero")
print("8.Borrar numero")
opcion=input()
```

```
if opcion=="1":
    print("Ingrese el nombre")
    nombre=input()
    print("Ingrese el apellido")
    apellido=input()
    print("Ingrese la cedula")
    cedula=input()
    aux=vcedula(cedula)

    if aux==True:
        usuario=[nombre,apellido,cedula,direccion,fecha]
        archivo = open("baseDatos.txt", "a")
        archivo.write(str(usuario)+"\n")
        archivo.close()
    else:
        print("Cedula incorrecta")

    print("Ingrese la fecha de nacimiento")
    fecha=input()
    print("Ingrese la direccion")
    direccion=input()
    archivo.write(str(usuario)+"\n")
    archivo.close()


if opcion=="2":
    print("Ingrese la cedula a eliminar")
    cedula=input()
    archivo = open("baseDatos.txt", "r")
    lineas = archivo.readlines()
```

```
    for lin in lineas:
        if cedula in lin:
            auxcedula=lin
    archivo.close()

    archivo = open("baseDatos.txt", "w")
    for line in lineas:
        print("valor del aux "+auxcedula)
        print("valor de line "+line)
        if line!=str(auxcedula):
            archivo.write(line)
    archivo.close()



if opcion=="3":
    print("Ingrese la cedula del usuario a actualizar")
    cedula=input()
    archivo = open("baseDatos.txt", "r")
    lineas = archivo.readlines()
    for lin in lineas:
        if cedula in lin:
            auxcedula=lin
    archivo.close()

    archivo = open("baseDatos.txt", "w")
    for line in lineas:
        print("valor del aux "+auxcedula)
        print("valor de line "+line)
        if line!=str(auxcedula):
            archivo.write(line)
    archivo.close()
```

```
    print("Ingrese el nombre")
    nombre=input()
    print("Ingrese el apellido")
    apellido=input()
    print("Ingrese la fecha de nacimiento")
    fecha=input()
    print("Ingrese la direccion")
    direccion=input()
    print("Ingrese la cedula")
    cedula=input()
    aux=vcedula(cedula)

    if aux==True:
        usuarioAct=[nombre,apellido,cedula,direccion,fecha]
        archivo = open("baseDatos.txt", "a")
        archivo.write(str(usuarioAct)+"\n")
        archivo.close()
    else:
        print("Cedula incorrecta")



if opcion=="4":
    print("Ingrese la cedula a buscar")
    cedula=input()
    archivo = open("baseDatos.txt", "r")
    lineas = archivo.readlines()
    for lin in lineas:
        if cedula in lin:
            auxcedula=lin
            print(auxcedula)
```

```
    archivo.close()



if opcion=="5":
    print("Ingrese el numero de telefono")
    telefono=input()
    print("Ingrese el tipo de telefono")
    tipTelefono=input()
    print("Ingrese el codigo de area")
    area=input()
    print("Ingrese la operadora")
    operadora=input()

    agenda=[telefono,tipTelefono,area,operadora]
    archivo1 = open("baseDatosTelf.txt", "a")
    archivo1.write(str(agenda)+"\n")
    archivo1.close()



if opcion=="6":
    print("Ingrese el telefono a eliminar")
    telefono=input()
    archivo = open("baseDatosTelf.txt", "r")
    lineas = archivo.readlines()
    for lin in lineas:
        if telefono in lin:
            auxtelf=lin
    archivo.close()

    archivo = open("baseDatosTelf.txt", "w")
```

```
    for line in lineas:
        print("valor del aux "+auxtelf)
        print("valor de line "+line)
        if line!=str(auxtelf):
            archivo.write(line)
    archivo.close()



if opcion=="7":
    print("Ingrese el numero de telefono a actualizar")
    cedula=input()
    archivo = open("baseDatosTelf.txt", "r")
    lineas = archivo.readlines()
    for lin in lineas:
        if cedula in lin:
            auxtelf=lin
    archivo.close()

    archivo = open("baseDatosTelf.txt", "w")
    for line in lineas:
        print("valor del aux "+auxtelf)
        print("valor de line "+line)
        if line!=str(auxtelf):
            archivo.write(line)
    archivo.close()

    print("Ingrese el numero de telefono")
    telefono=input()
    print("Ingrese el tipo de telefono")
    tipTelefono=input()
    print("Ingrese el codigo de area")
```

```
    area=input()
    print("Ingrese la operadora")
    operadora=input()



    telfAct=[telefono,tipTelefono,area,operadora]
    archivo = open("baseDatosTelf.txt", "a")
    archivo.write(str(telfAct)+"\n")
    archivo.close()



if opcion=="8":
    print("Ingrese el numero a buscar")
    telefono=input()
    archivo = open("baseDatosTelf.txt", "r")
    lineas = archivo.readlines()
    for lin in lineas:
        if telefono in lin:
            auxtelf=lin
            print(auxtelf)
    archivo.close()
```