

Examen Interciclo

Nombre: Edison Huinaizaca

Problema

1. Se desea modelizar el conocimiento de un experto para el diagnóstico de enfermedades de las plantas, basadas en las siguientes reglas. • Las plantas requieren diferentes tipos de nutrientes para desarrollarse adecuadamente. • Tres de los nutrientes de mayor importancia son el nitrógeno, el fósforo y el potasio. • Una deficiencia en alguno de estos nutrientes puede producir varios síntomas.

Escribir como reglas en CLIPS las siguientes reglas heurísticas que indican qué deficiencia se está produciendo:

- Si la planta crece muy poco entonces puede tener una deficiencia de nitrógeno.
- Si la planta tiene un color amarillo pálido entonces puede tener una deficiencia de nitrógeno.
- Si las hojas tienen un color pardo rojizo entonces la planta puede tener una deficiencia de nitrógeno.
- Si la raíz de la planta tiene poco crecimiento entonces puede tener una deficiencia de fósforo.
- Una planta con tallo fusiforme puede tener una deficiencia de fósforo.
- Una planta con color púrpura puede tener una deficiencia de fósforo.
- Un retraso en la madurez de una planta puede deberse a una deficiencia de fósforo.
- Si los bordes de las hojas aparecen chamuscados, la planta puede tener una deficiencia de potasio.
- Una planta con los tallos debilitados puede tener una deficiencia de potasio.
- Una planta con semillas o frutas marchitas puede tener una deficiencia de potasio.

La entrada del programa debe ser una descripción de los síntomas de la planta. La salida debe indicar de qué nutriente o nutrientes se ha producido una deficiencia imprimiendo en pantalla. Hacerlo de forma que no aparezcan en pantalla múltiples salidas indicando una misma deficiencia.

Desarrollo

Para el desarrollo de este problema se usó el programa CLIPS el cual es una Herramienta de desarrollo y manejo de sistemas expertos la cual nos ayudara en el desarrollo de este problema a continuación se presentara el código escrito en CLIPS.

```
In [ ]: (defrule tipo-problemas_plantas1
(initial-fact)
=>
(printout t "la planta crece poco si/no?" crlf)
(assert (tipo-problema (read)))
)

(defrule preguntas
(or(tipo-problema no)(tipo-problema si))
=>
(printout t "la planta tiene color amarillo(si/no)?" crlf)
(assert (problemas-planta(read)))
)
(defrule preguntas1
(or(tipo-problema no)(tipo-problema si))
=>
(printout t "tiene color pardo rojizo(si/no)?" crlf)
(assert (problemas-planta1(read)))
)

(defrule preguntas2
(or(tipo-problema no)(tipo-problema si))
=>
(printout t "la raíz de la planta tiene poco crecimiento (si/no)?" crlf)
(assert (problemas-planta2(read)))
)

(defrule preguntas3
(or(tipo-problema no)(tipo-problema si))
=>
(printout t "la planta tiene un tallo fusiforme (si/no)?" crlf)
(assert (problemas-planta3(read)))
)

(defrule preguntas4
(or(tipo-problema no)(tipo-problema si))
=>
(printout t "la planta tiene color púrpura (si/no)?" crlf)
(assert (problemas-planta4(read)))
)

(defrule preguntas5
(or(tipo-problema no)(tipo-problema si))
=>
(printout t "la planta tiene un retraso en la madurez(si/no)?" crlf)
(assert (problemas-planta5(read)))
)

(defrule preguntas6
(or(tipo-problema no)(tipo-problema si))
=>
(printout t "la planta tiene los bordes de las hojas chamuscados(si/no)?" crlf)
(assert (problemas-planta6(read)))
)
(defrule preguntas7
```

```

(or(tipo-problema no)(tipo-problema si))
=>
(printout t "la planta tiene los tallos debilitados (si/no)?" crlf)
(assert (problemas-planta7(read)))
)
(defrule preguntas8
(or(tipo-problema no)(tipo-problema si))
=>
(printout t "la planta tiene semillas o frutas marchitas (si/no)?" crlf)
(assert (problemas-planta8(read)))
)

(defrule resultado1
(or(tipo-problema si)
(problemas-planta si)
(problemas-planta1 si))
=>(printout t "Tiene deficiencia de nitrógeno" crlf)
)

(defrule resultado2
(or
(problemas-planta2 si)
(problemas-planta3 si)
(problemas-planta4 si)
(problemas-planta5 si))
=>(printout t "Tiene deficiencia de fósforo" crlf)
)

(defrule resultado3
(or
(problemas-planta6 si)
(problemas-planta7 si)
(problemas-planta8 si))
=>(printout t "Tiene deficiencia potasio" crlf)
)

```

Explicación del código:

- la primera sección consta con lo siguiente:
- (defrule preguntas2
- (or(tipo-problema no)(tipo-problema si))
- =>
- (printout t "la raíz de la planta tiene poco crecimiento (si/no)?" crlf)
- (assert (problemas-planta2(read))))

esta parte es la parte del tes lacual ara la pregunta y guardara el resultado en la variable problemas-planta2(read) luego esta la sección 2 la cual es:

- (defrule resultado3
 - (or
 - (problemas-planta6 si)
 - (problemas-planta7 si)

- (problemas-planta8 si))
- =>(printout t "Tiene deficiencia potasio" crlf)) esta seccion dara respuesta a los enusitados
segun ballan respondiendo

Conclusiones

Se puede decir que estos lenguajes de programación fue un lenguaje muy util para resolver este tipo de problemas puesto a que dependiendo el problema este lenguaje puede ser fasil o difisil de iplementarlo

In []: