

Examen Sistemas Expertos

Enunciado

1. Se desea modelizar el conocimiento de un experto para el diagnostico 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.
- 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

Procedemos a cargar nuestro sistema basado en reglas, para ello utilizamos el metodo load y le enviamos el archivo .clp

```
In [1]: from clips import Environment, Symbol
environment = Environment()
environment.load('plantas.clp')
```

Procedemos a validar que se cargo el archivo de reglas con la siguiente instruccion

```
In [2]: environment.reset()
environment.run()

Out[2]: 1
```

Para revisar la base de reglas ejecutamos la siguiente instruccion:

```
In [3]: for rule in environment.rules():
        print(rule)

(defrule MAIN::tipo-problemas_plantas1
  (initial-fact)
  =>
  (printout t "La planta crece poco si/no?" crlf)
  (assert (tipo-problema (read))))

(defrule MAIN::preguntas
  (or (tipo-problema no)
      (tipo-problema si))
  =>
  (printout t "La planta tiene color amarillo palido (si/no)?" crlf)
  (assert (problemas-planta (read))))

(defrule MAIN::preguntas1
  (or (tipo-problema no)
      (tipo-problema si))
  =>
  (printout t "Las hojas tiene color pardo rojiso(si/no)?" crlf)
  (assert (problemas-planta2 (read))))

(defrule MAIN::preguntas2
  (or (tipo-problema no)
      (tipo-problema si))
  =>
  (printout t "La raiz de la planta tiene poco crecimiento (si/no)?" crlf)
  (assert (problemas-planta3 (read))))

(defrule MAIN::preguntas3
  (or (tipo-problema no)
      (tipo-problema si))
  =>
  (printout t "Una planta tiene tallo fusiforme (si/no)?" crlf)
  (assert (problemas-planta4 (read))))

(defrule MAIN::preguntas4
  (or (tipo-problema no)
      (tipo-problema si))
  =>
  (printout t "Una planta tiene color purpura (si/no)?" crlf)
  (assert (problemas-planta5 (read))))

(defrule MAIN::preguntas5
  (or (tipo-problema no)
      (tipo-problema si))
  =>
  (printout t "La planta tiene un retraso en la madurez (si/no)?" crlf)
  (assert (problemas-planta6 (read))))

(defrule MAIN::preguntas6
  (or (tipo-problema no)
      (tipo-problema si))
  =>
  (printout t "Los bordes de las hojas aparecen chamuscados (si/no)?" crlf)
  (assert (problemas-planta7 (read))))

(defrule MAIN::preguntas7
  (or (tipo-problema no)
      (tipo-problema si))
  =>
  (printout t "La planta tiene tallos debilitados (si/no)?" crlf)
  (assert (problemas-planta8 (read))))

(defrule MAIN::preguntas8
  (or (tipo-problema no)
      (tipo-problema si))
  =>
  (printout t "La planta tiene semillas o frutas marchitas (si/no)?" crlf)
  (assert (problemas-planta9 (read))))

(defrule MAIN::resultado1
  (or (tipo-problema si)
      (problemas-planta si)
      (problemas-planta2 si))
  =>
  (assert (deficiencia-nitrogeno))
  (printout t "Tiene deficiencia nitrogeno" crlf))

(defrule MAIN::resultado2
  (or (problemas-planta3 si)
      (problemas-planta4 si)
      (problemas-planta5 si)
      (problemas-planta6 si))
  =>
  (assert (deficiencia-fosforo))
  (printout t "Tiene deficiencia fosforo" crlf))

(defrule MAIN::resultado3
  (or (problemas-planta7 si)
      (problemas-planta8 si)
      (problemas-planta9 si))
  =>
  (assert (deficiencia-potasio))
  (printout t "Tiene deficiencia potasio" crlf))
```

Mostramos hechos registrados dentro de clips y la inferencia que obtuvo

```
In [4]: for fact in environment.facts():
        print(fact)
```

```
(initial-fact)
(tipo-problema EOF)
```

```
In [5]: pregunta1 = input("La planta crece poco si/no? \n")
pregunta2 = input("La planta tiene color amarillo palido (si/no)? \n")
pregunta3 = input("Las hojas tiene color pardo rojiso(si/no)? \n")
pregunta4 = input("La raiz de la planta tiene poco crecimiento (si/no)? \n")
pregunta5 = input("Una planta tiene tallo fusiforme (si/no)? \n")
pregunta6 = input("Una planta tiene color purpura (si/no)? \n")
pregunta7 = input("La planta tiene un retraso en la madurez (si/no)? \n")
pregunta8 = input("Los bordes de las hojas aparecen chamuscados (si/no)? \n")
pregunta9 = input("La planta tiene tallos debilitados (si/no)? \n")
pregunta10 = input("La planta tiene semillas o frutas marchitas (si/no)? \n")
```

```
fact_string_pre1 = '(tipo-problema ' + pregunta1 + ')'
fact_string_respuesta1 = environment.assert_string(fact_string_pre1)
fact_string_pre2 = '(problemas-planta ' + pregunta2 + ')'
fact_string_respuesta2 = environment.assert_string(fact_string_pre2)
fact_string_pre3 = '(problemas-planta2 ' + pregunta3 + ')'
fact_string_respuesta3 = environment.assert_string(fact_string_pre3)
fact_string_pre4 = '(problemas-planta3 ' + pregunta4 + ')'
fact_string_respuesta4= environment.assert_string(fact_string_pre4)
fact_string_pre5 = '(problemas-planta4 ' + pregunta5 + ')'
fact_string_respuesta5 = environment.assert_string(fact_string_pre5)
fact_string_pre6 = '(problemas-planta5 ' + pregunta6 + ')'
fact_string_respuesta6 = environment.assert_string(fact_string_pre6)
fact_string_pre7 = '(problemas-planta6 ' + pregunta7 + ')'
fact_string_respuesta7 = environment.assert_string(fact_string_pre7)
fact_string_pre8 = '(problemas-planta7 ' + pregunta8 + ')'
fact_string_respuesta8 = environment.assert_string(fact_string_pre8)
fact_string_pre9 = '(problemas-planta8 ' + pregunta9 + ')'
fact_string_respuesta9 = environment.assert_string(fact_string_pre9)
fact_string_pre10 = '(problemas-planta9 ' + pregunta10 + ')'
fact_string_respuesta10 = environment.assert_string(fact_string_pre10)
```

```
environment.run()
for fact in environment.facts():
    if fact.template.name == 'deficiencia-nitrogeno':
        print('Resultado: '+'Tiene deficiencia nitrogeno')
    elif fact.template.name == 'deficiencia-fosforo':
        print('Resultado: '+'Tiene deficiencia fosforo')
    elif fact.template.name == 'deficiencia-potasio':
        print('Resultado: '+'Tiene deficiencia potasio')
```

```
La planta crece poco si/no?
si
La planta tiene color amarillo palido (si/no)?
no
Las hojas tiene color pardo rojiso(si/no)?
no
La raiz de la planta tiene poco crecimiento (si/no)?
si
Una planta tiene tallo fusiforme (si/no)?
no
Una planta tiene color purpura (si/no)?
no
La planta tiene un retraso en la madurez (si/no)?
si
Los bordes de las hojas aparecen chamuscados (si/no)?
no
La planta tiene tallos debilitados (si/no)?
si
La planta tiene semillas o frutas marchitas (si/no)?
no
Resultado: Tiene deficiencia potasio
Resultado: Tiene deficiencia fosforo
Resultado: Tiene deficiencia nitrogeno
```

```
In [6]: for fact in environment.facts():
        print(fact)
```

```
(initial-fact)
(tipo-problema EOF)
(tipo-problema si)
(problemas-planta no)
(problemas-planta2 no)
(problemas-planta3 si)
(problemas-planta4 no)
(problemas-planta5 no)
(problemas-planta6 si)
(problemas-planta7 no)
f-10 (problemas-planta8 si)
f-11 (problemas-planta9 no)
f-12 (deficiencia-potasio)
f-13 (deficiencia-fosforo)
f-14 (problemas-planta EOF)
f-15 (problemas-planta2 EOF)
f-16 (problemas-planta3 EOF)
f-17 (problemas-planta4 EOF)
f-18 (problemas-planta5 EOF)
f-19 (problemas-planta6 EOF)
f-20 (problemas-planta7 EOF)
f-21 (problemas-planta8 EOF)
f-22 (problemas-planta9 EOF)
f-23 (deficiencia-nitrogeno)
```

```
In [ ]:
```