

Objetivo:

- Consolidar los conocimientos adquiridos en clase de los sistemas expertos basados en reglas.

Enunciado:

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.

In [2]:

```
!pip3 install clipspy
from clips import Environment, Symbol

environment = Environment()
```

```
Requirement already satisfied: clipspy in /usr/local/lib/python3.7/dist-packages (0.3.3)
Requirement already satisfied: cffi>=1.0.0 in /usr/local/lib/python3.7/dist-packages (from clipspy) (1.14.4)
Requirement already satisfied: pycparser in /usr/local/lib/python3.7/dist-packages (from cffi>=1.0.0->clipspy) (2.20)
```

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.

REGLAS

cargamos el documento de clips en donde tenemos establecidas las reglas

In [3]:

```
environment.load('examen.clp')
```

```
In [4]: environment.reset()  
environment.run()
```

Out[4]: 0

```
In [5]: for rule in environment.rules():  
        print(rule)  
  
(defrule MAIN::pregunta1  
  (respuesta1 si)  
  =>  
  (assert (planta-crece-poco)))  
  
(defrule MAIN::pregunta2  
  (respuesta2 si)  
  =>  
  (assert (planta-amarillo-palido)))  
  
(defrule MAIN::pregunta3  
  (respuesta3 si)  
  =>  
  (assert (hojas-pardo-rojizo)))  
  
(defrule MAIN::pregunta4  
  (respuesta4 si)  
  =>  
  (assert (raiz-crece-poco)))  
  
(defrule MAIN::pregunta5  
  (respuesta5 si)  
  =>  
  (assert (planta-tallo-fusiforme)))  
  
(defrule MAIN::pregunta6  
  (respuesta6 si)  
  =>  
  (assert (planta-color-purpura)))  
  
(defrule MAIN::pregunta7  
  (respuesta7 si)  
  =>  
  (assert (planta-retraso-madurez)))  
  
(defrule MAIN::pregunta8  
  (respuesta8 si)  
  =>  
  (assert (bordes-hojas-chamuscados)))  
  
(defrule MAIN::pregunta9  
  (respuesta9 si)  
  =>  
  (assert (planta-tallos-debilitados)))  
  
(defrule MAIN::pregunta10  
  (respuesta10 si)  
  =>  
  (assert (frutas-semillas-marchitas)))  
  
(defrule MAIN::nitrogeno-1  
  (or (planta-crece-poco)  
      (planta-amarillo-palido)  
      (hojas-pardo-rojizo))  
  =>  
  (assert (deficiencia-nitrogeno))  
  (printout t "La planta tiene una deficiencia de nitrógeno." crlf))
```

```
(defrule MAIN::fosforo-1
  (or (raiz-crece-poco)
       (planta-tallo-fusiforme)
       (planta-color-purpura)
       (planta-retraso-madurez))
  =>
  (assert (deficiencia-fosforo))
  (printout t "La planta tiene una deficiencia de fosforo." crlf))

(defrule MAIN::potasio-1
  (or (bordes-hojas-chamuscados)
       (planta-tallos-debilitados)
       (frutas-semillas-marchitas))
  =>
  (assert (deficiencia-potasio))
  (printout t "La planta tiene una deficiencia de Potasio." crlf))
```

In [151]

```
for fact in environment.facts():
    print(fact)
```

(initial-fact)

- 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.

In [152]

```
respuesta1= input('La planta crece muy poco? (si/no) \n')
environment.assert_string('(respuesta1 '+respuesta1+')')

respuesta2= input('La planta tiene un color amarillo pálido. (si/no) \n')
environment.assert_string('(respuesta2 '+respuesta2+')')

respuesta3= input('Las hojas tienen un color pardo rojizo? (si/no) \n')
environment.assert_string('(respuesta3 '+respuesta3+')')

respuesta4= input('La raíz de la planta tiene poco crecimiento? (si/no) \n')
environment.assert_string('(respuesta4 '+respuesta4+')')

respuesta5= input('Una planta con tallo fusiforme? (si/no) \n')
environment.assert_string('(respuesta5 '+respuesta5+')')

respuesta6= input('Una planta con color púrpura? (si/no) \n')
environment.assert_string('(respuesta6 '+respuesta6+')')

respuesta7= input('Un retraso en la madurez de una planta? (si/no) \n')
environment.assert_string('(respuesta7 '+respuesta7+')')

respuesta8= input('Los bordes de las hojas aparecen chamuscados? (si/no) \n')
environment.assert_string('(respuesta8 '+respuesta8+')')

respuesta9= input('Una planta con los tallos debilitados? (si/no) \n')
environment.assert_string('(respuesta9 '+respuesta9+')')
```

```

respuesta10= input('Una planta con semillas o frutas marchitas? (si/no) \n')
environment.assert_string('(respuesta10 '+respuesta10+')')

```

```

La planta crece muy poco? (si/no)
si
La planta tiene un color amarillo pálido. (si/no)
no
Las hojas tienen un color pardo rojizo? (si/no)
si
La raíz de la planta tiene poco crecimiento? (si/no)
no
Una planta con tallo fusiforme? (si/no)
si
Una planta con color púrpura? (si/no)
no
Un retraso en la madurez de una planta? (si/no)
si
Los bordes de las hojas aparecen chamuscados? (si/no)
no
Una planta con los tallos debilitados? (si/no)
si
Una planta con semillas o frutas marchitas? (si/no)
no

```

Out[152... ImpliedFact: f-10 (respuesta10 no)

```

In [153...
nums=[respuesta1,respuesta2,respuesta3,respuesta4,respuesta5,respuesta6,resp
n=1
environment.run()
for fact in environment.facts():
    if fact.template.name == 'deficiencia-nitrogeno':
        print('Resultado: '+'La planta tiene deficiencia de nitrogeno')
    elif fact.template.name == 'deficiencia-fosforo':
        print('Resultado: '+'La planta tiene deficiencia de fosforo')
    elif fact.template.name == 'deficiencia-potasio':
        print('Resultado: '+'La planta tiene deficiencia de potasio')

for i in nums:
    if i == 'no':
        n=n+1
    if (n-1) == len(nums):
        print('no se registra solucion')

```

```

Resultado: La planta tiene deficiencia de potasio
Resultado: La planta tiene deficiencia de fosforo
Resultado: La planta tiene deficiencia de nitrogeno

```

```

In [154...
for fact in environment.facts():
    print(fact)

```

```

(initial-fact)
(respuesta1 si)
(respuesta2 no)
(respuesta3 si)
(respuesta4 no)
(respuesta5 si)
(respuesta6 no)
(respuesta7 si)
(respuesta8 no)
(respuesta9 si)

```

f-10 (respuesta10 no)
f-11 (planta-tallos-debilitados)
f-12 (deficiencia-potasio)
f-13 (planta-retraso-madurez)
f-14 (deficiencia-fosforo)
f-15 (planta-tallo-fusiforme)
f-16 (hojas-pardo-rojizo)
f-17 (deficiencia-nitrogeno)
f-18 (planta-crece-poco)

In []: