Preguntas sobre Integración de CLIPS en C y Python

1. ¿Cómo se instala CLIPSPy?

Para instalar CLISPy necesitamos correr el siguiente comando:

>>$ pip install clipsy

2. ¿Se pueden evaluar en Python (mediante CLIPSPy) expresiones de CLIPS? ¿Cómo?

Si. Para ello sería necesario hacer lo siguiente:

```
import clips as cp
env = cp.Environment()
exp = "(create$ silla mesa)"
env.eval(exap)
```

3. Mediante CLIPSPy se pueden utilizar dentro de CLISP funciones de python. ¿Cómo se hace?

Para ello generaríamos nuestro entorno de CLIPS a través de Environment, suponiendo que ya tenemos definida una función fun en Python. Entonces, con esta clase utilizamos el método defina_function que nos define la función fun en CLIPS, que nos simula una función como si la hubieramos hecho con def function. Utilizando por último el método eval se podrá utilizar.

4. ¿Qué métodos de CLIPSPy asertan y retractan un hecho en CLISP?

Métodos que retractan hechos:

- retract(): ejecuta un retract sobre el hecho en el que se ejecuta

Métodos que asertan hechos:

- assert_string(string): te aserta el hecho de string
- assertit(): te aserta el hecho en el que se ejecuta.

5. ¿Cómo se ejecutaría en Python (mediante CLIPSPy) un sistema basado en reglas definido mediante un fichero .clp?

Para ello deberíamos crear un objeto de tipo Environment y con el método load(ruta), cargar el archivo .clp en el entorno del Environment que acabamos de crear.

6. Describe brevemente como convertirías un sistema basado en reglas definido mediante un fichero .clp en un fichero ejecutable

- Primero, cargar en CLIPS todas las construcciones que van a constituir el módulo en tiempo de ejecución.
- Ponenmos el flag RUN_TIME a 1. Luego, compilamos los archivos de c que se generan. El flag RUN_TIME se encuentra en el archivo setup.h
- Modificamos el main.c para embeber. Acto seguido, recompilamos todo el código fuente de CLIPS.
- Enlazamos los módulos regulares de CLIPS junntos, unto a cualquier módulo de funciones definidas por el usuario y todos los módulos de construcciones de C.

7. Describe brevemente cómo incluirías en CLISP una función definida en C

- Copio los archivos de CLIPS en el direcctorio correspondiente del usuario.
- Definiría la función que queremos añadir y un nuevo main.
- Modifico EnvUserFunctions (userfunction.c).
- Defino las construcciones usando la nueva función añadida, en un archivo llamado construcciones.clp. S
- Compilo todos los archivos de CLIPS, excepto el main.c, junto con los de usuario.
- Enlazo los archivos y ejecuto.

8. Describe brevemente cómo incluirías un sistema basado en reglas definido mediante un fichero .clp dentro de tu programa escrito en C.

- En el main incluiríamos clips.h
- Crearía un entorno de ejecucion de CLIPS.
- Ejecutando EnvLoad(<entorno>,<.clp>) ya podríamos ejecutarlo con las funciones EnvReset(<entorno>) y EnvRun(<entorno>,-1)

9. ¿Que funciones se utilizan para asertar o retractar un hecho en un sistema basado en reglas embebido en un programa de C?

Métodos que retractan hechos:

- int EnvRetract(<puntero al entorno>,<puntero al hecho a retractar>): retractamos el hecho pasado por parámetro.

Métodos que asertan hechos:

- void *EnvAssert(<puntero al entorno>,<puntero al hecho creado>): aserta a un entorno un hecho creado con la función CreateFact.

- void *EnvAssertString(<puntero al entorno>,<string del hecho>): aserta un hecho definido en forma de String.

10. ¿Se pueden ejecutar varios sistemas basados en reglas distintos dentro de un mismo programa de C?

Sí. Como podemos crear los entornos que queramos con CreateEnvironment(), en cada instancia (es decir, en cada entorno) podemos definir sistemas basdos en reglas distintos, consiguiendo ejecutar varios sistemas basados en reglas distintos dentro de un mismo programa de C.