

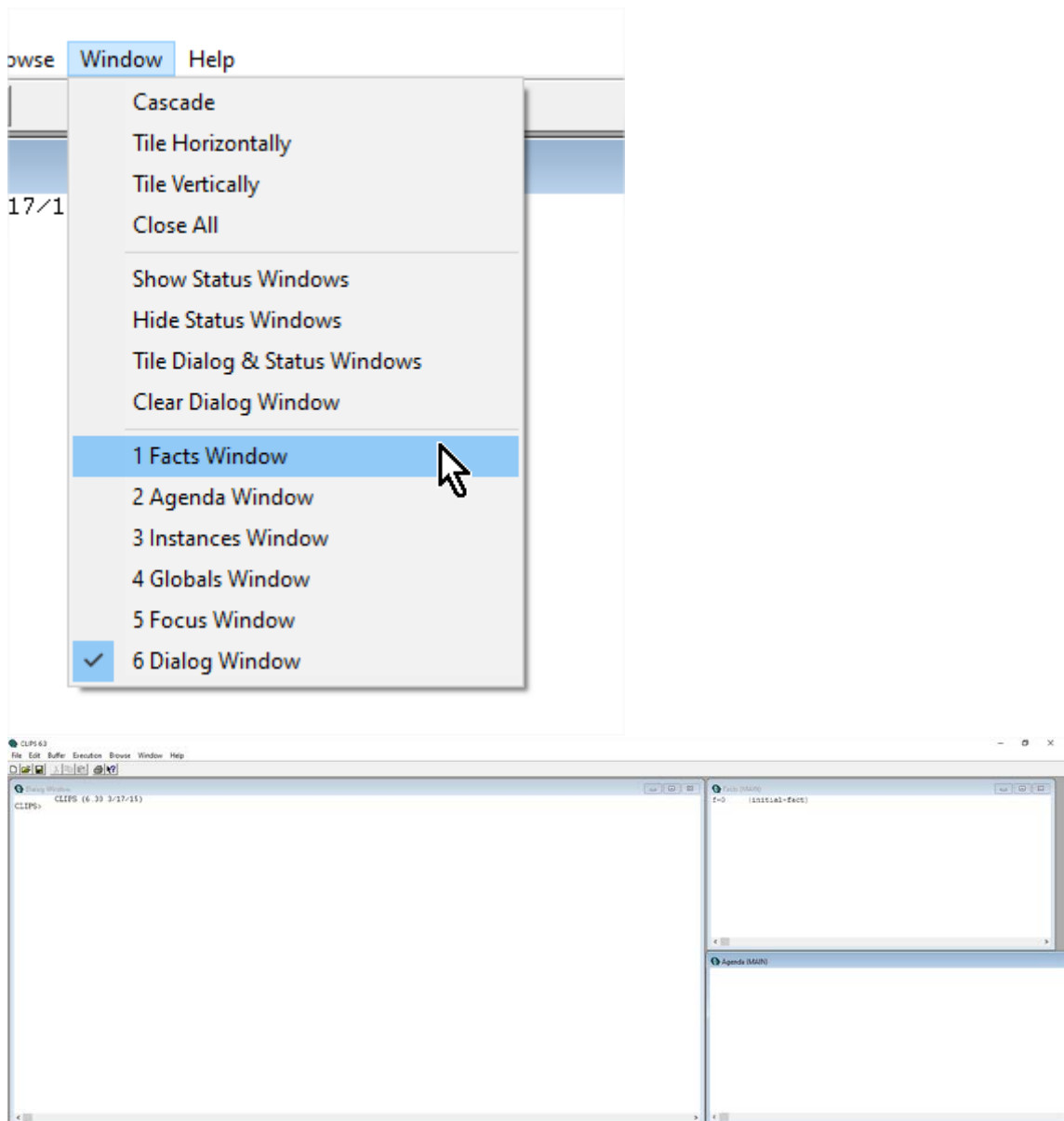
Práctica 4

Sistemas basados en reglas

El primer objetivo de esta práctica consiste en conocer las técnicas de representación de conocimiento basadas en reglas y, más concretamente, en aprender las características básicas del lenguaje CLIPS.

Apartado 1.

Toma de contacto con CLIPS. Arrancar el entorno del sistema CLIPS y abrir las ventanas Agenda y Hechos. Para ello, elija Facts Window y Agenda Window de la opción Window que aparece en el menú principal.



El entorno tiene un editor integrado aunque puedes usar cualquier editor de textos y luego cargar el archivo con load.

Los archivos clips tienen extensión .clp

Escribir las siguientes reglas:

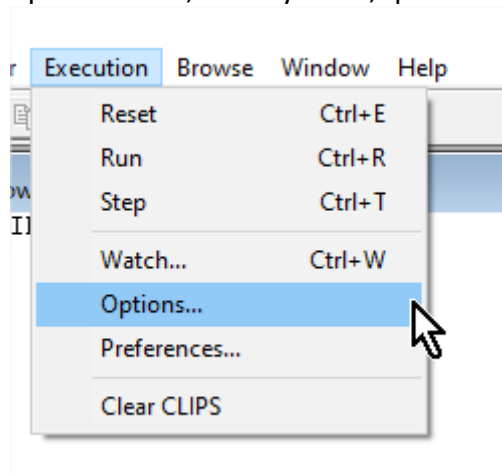
```
(defrule nombreJuan
  (nombre Juan)
=>
  (printout t "Tu nombre de pila es Juan" crlf)
)
(defrule Hola
  (nombre Juan)
  (apellido-1 Perez)
  (apellido-2 Lopez)
=>
  (printout t "Hola Juan Perez Lopez" crlf)
)
```

Y los siguientes hechos en la base de hechos inicial.

```
(assert (nombre Ana))
(assert (nombre Juan))
(assert (apellido-1 Diaz))
(assert (apellido-1 Perez) (apellido-2 Lopez))
(assert (apellido-1 Martinez))
```

La base de hechos inicial también se puede introducir utilizando el mandato (deffacts)

- ¿Se ha activado alguna regla?. ¿En qué orden se han activado?
- Para comenzar la ejecución hay que escribir (run) o usar la opción run del menú ¿Qué reglas se han ejecutado?. ¿En qué orden lo han hecho? ¿Por qué crees que se han ejecutado en ese orden?
- Reiniciar el sistema con (reset). ¿Qué hubiera pasado si reiniciamos con (clear) en lugar de con (reset)?
- Observación: En el menú Execution existen, entre otras, las opciones Run, Reset y Clear, que evitan tener que escribirlos.



Apartado 2.

Crea un archivo **familia.clp**, ejecútalo y observa el funcionamiento. Ejecutar algunos ejemplos de prueba y comprobar cómo se modifica la memoria de trabajo. Comenta si afecta el cambio de estrategia de resolución de conflictos en el resultado del sistema.

Añade las reglas que añaden todos los hechos (primos p1 p2) que indica que p1(h o m) es primo hermano de p2 (h o m).

```
(deffacts inicio
  (dd juan maria rosa m)
  (dd juan maria luis h)
  (dd jose laura pilar m)
  (dd luis pilar miguel h)
  (dd miguel isabel jaime h)
  (dd pepe rosa juanito h))
```

;; los hechos ordenados dd representan la información de descendientes directos
(dd padre madre hijo hombre/mujer)

```
(defrule padre
  (dd ?x ? ?y ?)
  =>
  (assert (padre ?x ?y)))
```

Apartado 3. (ejercicio 6 de la hoja de ejercicios del tema 3)

En el archivo dado **ej6.clp** se proporciona un sistema en Clips que da soporte a un servicio de búsqueda de pareja. El sistema dispondrá de datos iniciales de distintas personas que estarán establecidos como hechos. Por ejemplo, el nombre, el sexo, la edad y el tipo de pareja que busca (hombre o mujer). Además, se dispone de datos sobre su número de amigos en Facebook (si no tiene cuenta entonces valor 0) y sobre sus gustos (música, lectura, cine, teatro). El enunciado completo lo tienes en la hoja de ejercicios.

Ejecuta el sistema, indica cuál es el fallo y corrígelo, vuelve a hacer las pruebas necesarias y comenta los resultados.

Apartado 4.

Dado el archivo **cocina.clp** se trata de analizar y comprender el funcionamiento del Sistema Experto denominado STOVE, desarrollado por Thad Fiebich, de la Universidad Johannes Kepler (Linz, Austria). Se trata de un especialista en reparación de cocinas, tanto de gas como eléctricas, cuyo funcionamiento se basa en un sistema de preguntas alternativas (generalmente SI/NO), mediante el cual el programa tratará de determinar el problema de la cocina y su posible solución.

Realiza un análisis del sistema y de su funcionamiento y describe el árbol de decisión asociado.