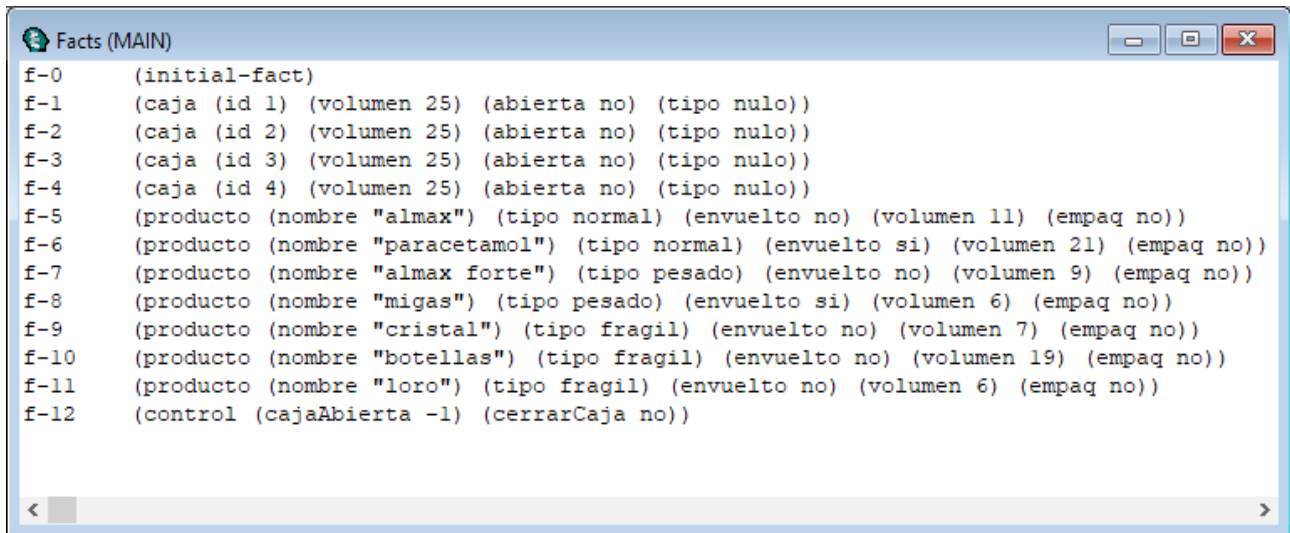


Práctica 5

Empaquetador de productos - Juan Chozas Sumera & Miguel Menéndez Galindo

Ejemplo de ejecución

Después de ejecutar (load ..) y (reset), esta es la primera tanda de hechos de la que partimos:



```
Facts (MAIN)
f-0      (initial-fact)
f-1      (caja (id 1) (volumen 25) (abierta no) (tipo nulo))
f-2      (caja (id 2) (volumen 25) (abierta no) (tipo nulo))
f-3      (caja (id 3) (volumen 25) (abierta no) (tipo nulo))
f-4      (caja (id 4) (volumen 25) (abierta no) (tipo nulo))
f-5      (producto (nombre "almax") (tipo normal) (envuelto no) (volumen 11) (empaqa no))
f-6      (producto (nombre "paracetamol") (tipo normal) (envuelto si) (volumen 21) (empaqa no))
f-7      (producto (nombre "almax forte") (tipo pesado) (envuelto no) (volumen 9) (empaqa no))
f-8      (producto (nombre "migas") (tipo pesado) (envuelto si) (volumen 6) (empaqa no))
f-9      (producto (nombre "cristal") (tipo fragil) (envuelto no) (volumen 7) (empaqa no))
f-10     (producto (nombre "botellas") (tipo fragil) (envuelto no) (volumen 19) (empaqa no))
f-11     (producto (nombre "loro") (tipo fragil) (envuelto no) (volumen 6) (empaqa no))
f-12     (control (cajaAbierta -1) (cerrarCaja no))
```

CLIPS> (run)

Caja 4 abierta. Tipo pesado

Producto migas empaquetado

Producto loro envuelto

Producto botellas envuelto

Producto cristal envuelto

Producto almax forte envuelto

Producto almax forte empaquetado

No se pueden empaquetar mas productos de tipo pesado

Caja 4 cerrada

Caja 3 abierta. Tipo fragil

Producto cristal empaquetado

Producto loro empaquetado

No se pueden empaquetar mas productos de tipo fragil

Caja 3 cerrada

Caja 2 abierta. Tipo fragil

Producto botellas empaquetado

No se pueden empaquetar mas productos de tipo fragil

Caja 2 cerrada

Caja 1 abierta. Tipo normal

Producto paracetamol empaquetado

No se pueden empaquetar mas productos de tipo normal

Caja 1 cerrada

Producto almax envuelto

A continuación se puede ver como quedan los hechos al final de la ejecución.

```
Facts (MAIN)
f-0      (initial-fact)
f-15     (producto (nombre "migas") (tipo pesado) (envuelto si) (volumen 6) (empaqa 4))
f-21     (producto (nombre "almax forte") (tipo pesado) (envuelto si) (volumen 9) (empaqa 4))
f-25     (caja (id 4) (volumen 10) (abierta no) (tipo pesado))
f-28     (producto (nombre "cristal") (tipo fragil) (envuelto si) (volumen 7) (empaqa 3))
f-30     (producto (nombre "loro") (tipo fragil) (envuelto si) (volumen 6) (empaqa 3))
f-34     (caja (id 3) (volumen 12) (abierta no) (tipo fragil))
f-37     (producto (nombre "botellas") (tipo fragil) (envuelto si) (volumen 19) (empaqa 2))
f-41     (caja (id 2) (volumen 6) (abierta no) (tipo fragil))
f-44     (producto (nombre "paracetamol") (tipo normal) (envuelto si) (volumen 21) (empaqa 1))
f-47     (control (cajaAbierta -1) (cerrarCaja no))
f-48     (caja (id 1) (volumen 4) (abierta no) (tipo normal))
f-49     (producto (nombre "almax") (tipo normal) (envuelto si) (volumen 11) (empaqa no))
```

Comentarios

Fue difícil encontrar la manera de abrir una sola caja a la vez, con lo cual añadimos un hecho "controlador" que mantiene dos slots para saber cuando hay abrir y cerrar cajas:

- El primer slot, *cajaAbierta* indica que caja está abierta. Si no hay ninguna caja abierta, el valor de *cajaAbierta* es -1. En el caso opuesto, tomaría el valor del slot *id* de la caja que esté abierta. Lo importante de este slot es que nos proporciona el conocimiento de que no existen cajas abiertas cuando su valor es -1.
- El segundo slot, *cerrarCaja* se puede interpretar como un valor booleano que nos avisa de que hay que cerrar una caja cuando su valor es *si*.

Asertando un único hecho controlador, podemos asegurar que en cualquier momento, como mucho habrá una sola caja abierta. Eso si, para modificar el segundo slot del controlador necesitamos algo más.

La regla *noQuedanProductosPorEmpaquetar* se encarga de "activar" la señal de *cerrarCaja* cuando no se puedan empaquetar mas productos en la caja que estuviese abierta en ese momento. Esta regla utiliza *find-all-facts* para contar los hechos de productos no empaquetados del tipo de la caja que esté abierta. *find-all-facts* una lista de todos los hechos que encajen con los patrones incluidos, y *length\$* nos permite contar cuantos hechos son. Puede activar *cerrarCaja* si los productos no empaquetados del tipo de la caja abierta es 0, o si los productos no empaquetados del tipo de la caja abierta que caben en la caja es 0.

Gracias al controlador y a la regla que activa *cerrarCaja*, la regla *cierraCaja* se implementa con facilidad. *abrirCaja* también se simplifica al saber que solo hay que abrir una caja cuando *cajaAbierta* = -1.