Untitled

2025-05-07

¿Qué es CLIPS?

CLIPS es la abreviación de: C Language integrated Production System.

Es una variación muy especializada de LISP por el estilo.

Herramienta para el desarrollo de sistemas expertos, creada en el Johnson Space Center (NASA) en 1986.

¿Para qué sirve CLIPS

Permite resolver problemas que normalmente resolverian "expertos humanos" gracias a su base de conocimiento sobre el dominio que se desea emplea.

Soporta la inclusión de programación lógica, programacipon imperativa y hasta orientada a objetos.

Su motor de inferencia (hacia adelante, hacia atras o hibrido,que paso aqui?) permite el diseño de programas dirigidos por datos.

Permite realizar el procesamiento de forma interactiva mediante la ventana de comandos, o por lotes. Admite la depuración.

El sistema de producción que incluye:

- Mantenimiendo de la verdad con el encadenamiento hacia adelante
- Adición dinámica de reglas y hechos
- Doferemtes estrategias de resolución de conflictos

Componentes básicos:

- Base de hechos: Datos introducidos e inferidos
- Base de conocimientos: Reglas, funciones, ...
- Mecanismo de inferencia: Controla la ejecución

CLIPS proporciona tres elementos básicos de programación:

- Tipos primitivos de datos.
- Funciones para la manipulación de los datos.
- Constructores.

Elementos básicos: Órdenes

Una vez abierta la interfaz, veamos algunas órdenes o comandos básicos de la operación.

Todo comando se escbribe siempre entre paréntesis (RECUERDA QUE ESTO ES LIPS). Los comandos se ejecutan habitalmente en top-level y no devuelven un valor directamente, al contrario de las funciones que SÍ retornan un valor:

- (exit) Cierra la interfaz CLIPS
- (run) Lanza la ejecución del programa CLIPS actualmente cargado. Se le puede indicar el número máximo de reglas a lanzar.
- (clear) Elimina todos los hechos y reglas almacenados en memoria, equivalente a cerrar y rearrancar CLIPS.
- (reset) Elimina sólo los hechos, no las reglas, anulando la agenda y añadiendo los elementos definidos por defecto o iniciales.
- (watch) Permite realizar depuración del programa.

Representar conocimiento en CLIPS

- Reglas: Se destinan principalmente al conocimiento heurístico basado en la experiencia.
- Programación orientada a objetos: destinada principalmente a los conocimientos procedimentales. Se admiten las cinco características generalmente aceptadas de la programación orientada a objetos: clases, manejadores de mensajes, abstracción, encapsulación, herencia y polimorfismo. Las reglas pueden coincidencia de patrones en objetos y hechos.

Definiendo una lista

Puedes usar el comando *assert* para declarar lo que acabas introducir (duck) como un argumento

```
# Así declar un hecho %
(assert (duck))
#<Fact-1>
```

Esto indica que CLIPS guardo a *duck* como un hecho en su lista de hechos y que le da un identificador 1.

Se puede consultar la base de hechos con

```
(facts)
#f-1 (duck)
```

Nota que internamente, el fact es identificado por el termino *f-1* por CLIPS. Cada hecho insertado en la lista de hechos tiene asignado un identificador unico que empieza con la letra "f" y seguido por su indice de hecho. Dicha lista de hechos incrementara conforme se agreguen hechos y les pondra sus correspondientes identificadores.

Un hecho que ya existe no lo modifica, esto debido a que CLIPS no acepta hechos repetidos, salvo que por alguna razón quieras torserle la mano y duplicar un hecho con **set-fact-duplication**.

Se pueden remover los hechos declarados con *(clear)*:

Remueve los hechos.

Campos

Un campo puede ser entendido como un espacio que va a contener un valor o varios valores.

Estos campos pueden tener o no un nombre asignado.

El orden dentro de los campos sin nombrar es significativo. Por ejemplo, si un hecho se define como:

```
(Brian duck)
```

Y es interpretado por una regla como que el cazador Brian diasparo al pato, entonces el hecho

```
(duck Brian)
```

Deberia significar que el cazador duck le disparo a Brian.

En contraste, el orden de los campos nombrados es no significativo, como vas a ver despues con el deftemplate.

En realidad, es una buena practica iniciar le hecho con la relacaión que describe a los campos. Un mejor hecho pudiera ser:

```
(hunter-game duck Brian)
```

Para implicar que el primer campo es el cazador y el segundo campo es el juego.

Ahora, aqui son necesarios unas definiciones.

Una **lista** es un grupo de objetos que NO implican orden. Diciendo eso, una lista es ordenada significando que la posición en la lista es significativa.

Un multicampo es una secuenda de campos, cada uno de ellos puede tener un valor. El ejemplo del *(duck Brian)* y *(Brian duck)* son hechos de multicampos. Sí un campo NO tiene valor, el simbolo especial **nil**, el cual significa "nada" podria ser usado para un campo vacio como un contenedor, por ejemplo

```
(duck nil)
```

Significario que el pato asesino no lograria conseguir ningun trofeo hoy.

NOTA que nil es necesario para indicar un contenedor, aun si este NO tiene valor. Por ejemplo, piensa en un campo analogo a un buzon de correso. Ahi hay una gran diferencia entre un buzon vacio, y un no buzon del todo. Sin el comando *nil*, el hecho se convierte en un hecho de campo unico (duck). Si una regla depende de dos campos, la cosa no funcionaria con un solo campo como vas a ver más adelante.

Existen diferentes tipos de campos disponibles:

- float
- integer
- symbol
- string
- external-address
- fact-address
- instance-name
- instance-address

Elementos básicos: Tipos de datos

- Reales (float): 1.5, -0.7, 3.5e-10
- Enteros (integer): 1, -1, +3, 65
- Simbolos (symbols): Cualquier secuencia de caracteres que no siga el formato de numéros.
- Cadenas (strings): Deben estar entre comillas
- Direcciones externas (external-address): Estructura de datos externa devuelta por una función escrita en C o Ada
- Direcciones de hechos (fact-address): Hechos referenciados por su posición o por un nombre
- Nombres de instancias (instance-name)
- Direcciones de instancias (instance-address address)

¿Que es son CAMPOS?

¿Que es son Agendas?

¿Que es son Hechos?

¿Que es son Reglas?

¿Que es son Plantillas?