

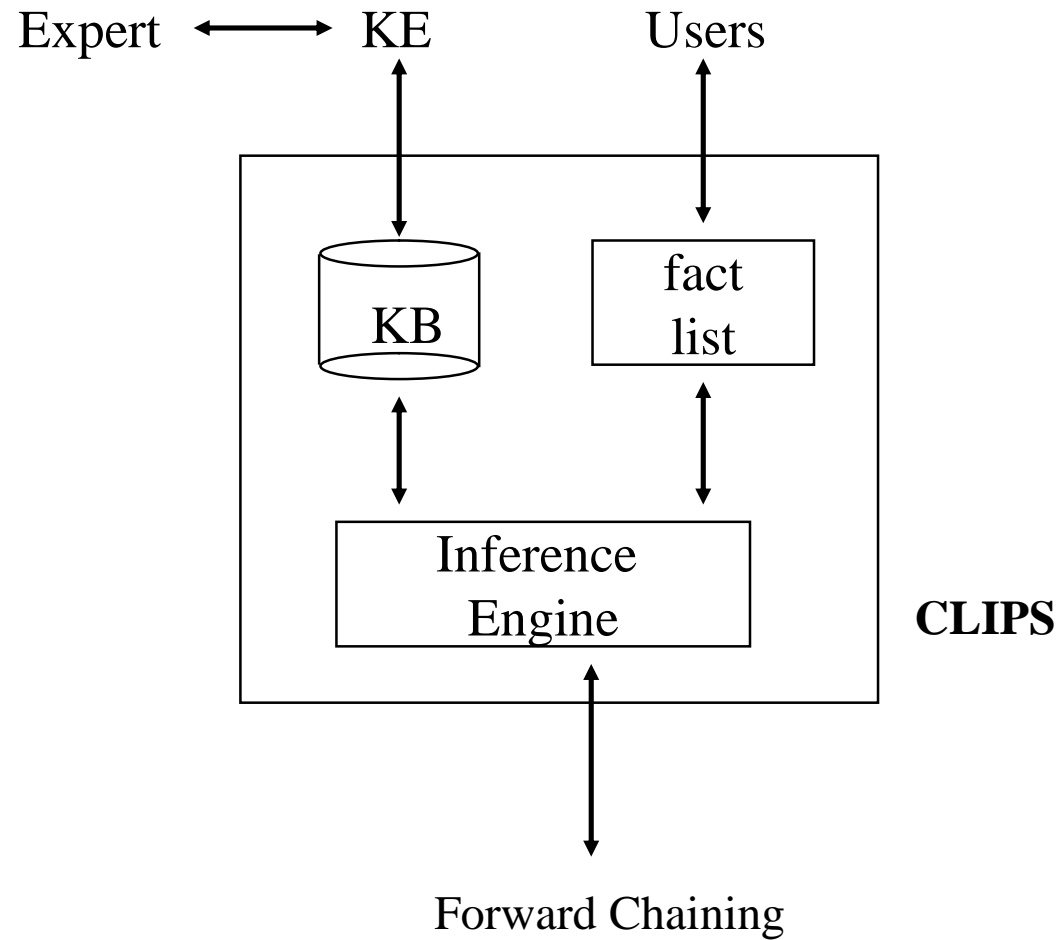
**CLIPS-01**

**Introduction to CLIPS**

By Gwo-Jen Hwang

- **CLIPS**

(C Language Integrated Production System)

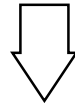


- **Notation**

- **1. fields**

- » **word type :**


- A printable ASCII character followed by zero or more characters.**



- A word may not begin with**

- < | & \$ ? + - ( ) ;**

- and may not contain**

- < | & ( ) ;**        **delimiters**

- e.g. valid words :**

- emergency**


- fire**

- emergency-fire**

- A-B-C**

- A-387**

- ! ? # \$ ^ \***

- fire ≠ FIRE ≠ Fire**        **Case-Sensitive**

» **String Type :**

**“This is a book”**

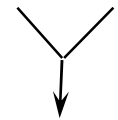
**“!?\$^”**

**“<-;() +-”**

**“fire” ≠ “fire ” ≠ “ fire” ≠ “ fire ”**

**““fire”” ⇒ “” + fire + “”**

**“\“fire\”” ⇒ ““fire””**



**special character: the following character is used as the contents of the string**

**“\\fire\\” ⇒ “\fire\”**

» **Number :** 1

**1.5**

**.7**

**+3**

**-1**

**65**

**3.5e10**

## 2. Facts

*A fact : A chunk of information*

(single-field)

(two fields)

(speed 38 mph)

(cost 78 dollars 23 cents)

(name "John Doe")

(Tuesday)

(day Tuesday)

(day Wednesday)

(emergency)

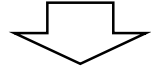
(fire)

(emergency fire)

(emergency flood)

### 3. Fact Templates

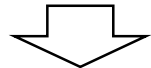
(emergency <type>)



(emergency fire)

(emergency flood)

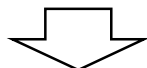
(action activate <device>)



(action activate sprinkler-system)

(action activate fire-alarm)

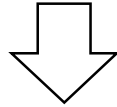
(action notify <group>)



(action notify fire-department)

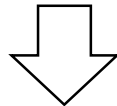
(action notify police-station)

(computer - components << a list of components>>)



(computer-components CPU disk-drive terminal)

(components <item> <<components-list>>)



(components computer CPU disk-drive terminal)

(components car tires engine body gas-tank)

## Basic Operations of CLIPS :

### 1. Start CLIPS :

```
C:\CLIPS> CLIPS ←┐  
CLIPS> ■
```

### 2. Leave CLIPS :

```
CLIPS> (exit) ←┐  
C:\> ■
```

### 3. Define facts :

(1) CLIPS> (assert (emergency fire)) ←┐

CLIPS> (assert (John 15 male)) ←┐

(2) Use READ or READLINE in rules :

(READ) ←┐

(READLINE) ←┐

(3) Use Deffacts name function :

(Deffacts status “Some facts about the emergency”

(emergency fire) }  
(fire class A) } two facts



## 4. Operations on facts :

1> show facts

```
CLIPS> (facts) ←|  
          f-0 (Tom 7015)  
          f-1 (John 8023)  
          f-2 (Mary 1235)  
          f-3 (Peter 1256)
```

2> delete facts

```
CLIPS> (retract 1) ←|  
CLIPS> (facts) ←|  
          f-0 (Tom 7015)  
          f-2 (Mary 1235)  
          f-3 (Peter 1256)
```

3> remove the definitions of facts

```
CLIPS> (deffacts status) ←|
```

## 5. Read facts or rules from files :

```
CLIPS> (LOAD filename) ←|      facts + rules  
CLIPS> (LOAD-facts filename) ←| facts only  
CLIPS> (SAVE filename) ←|  
CLIPS> (SAVE-facts filename) ←|
```

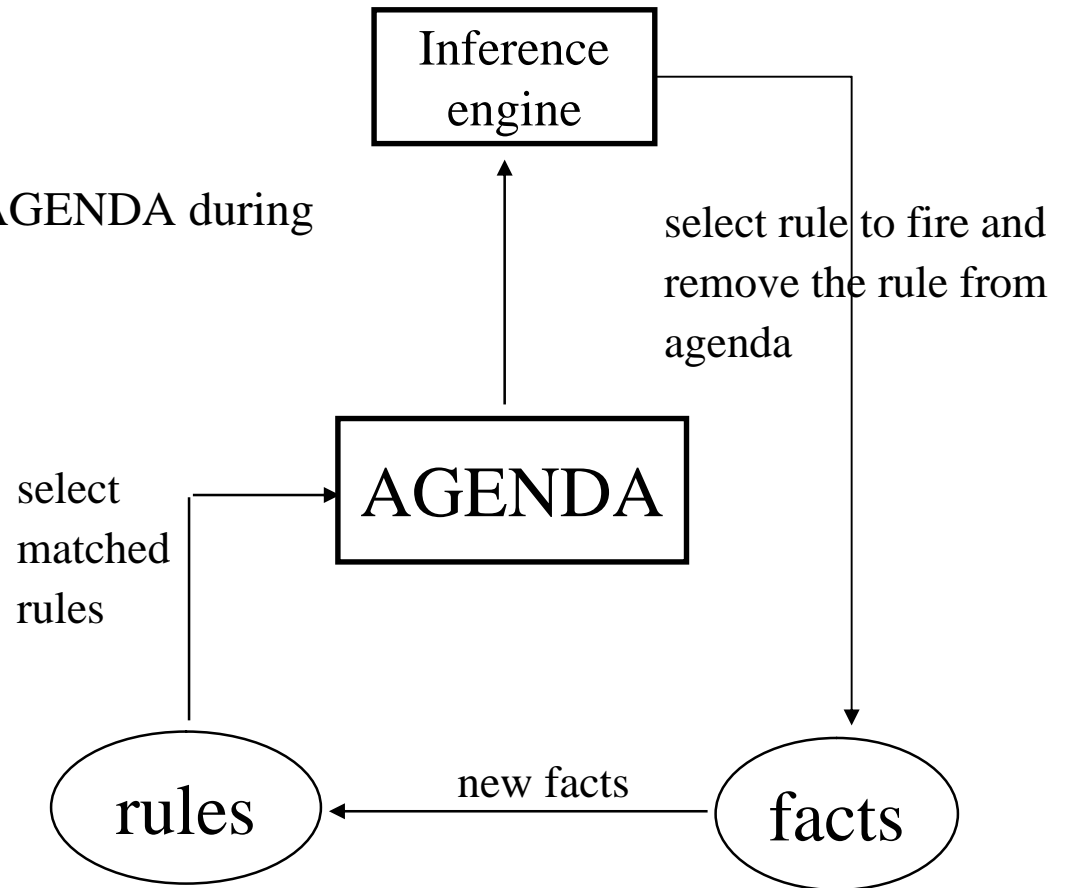
## 6. Reset command

CLIPS> (reset) ←

1> Clear temporary facts.

2> Insert facts defined by deffacts.

The matched rules will be put into AGENDA during the insertions of facts.



## 7. Execution

CLIPS> (Run) ←

## 8. Define rules

```
; This is a rule ... → comments ←
(defrule class-A-fire-emergency ; declare
IF [ (emergency fire) → rule name
      (fire-class A)
      =>
      (printout t "Activate sprinkler system" crlf)
      (assert (sprinkler system work))
      (system "sprinkler.exe") )
THEN
      → driver
```

## 9. Operations for rules:

CLIPS> (agenda) ←┘ list the rules in AGENDA

CLIPS> (rules)←┘ list the rules being defined

CLIPS> (ppdefrule rule-name)←┘ display contents of a rule

CLIPS> (excise rule-name)←┘ remove a rule

## 10. System call

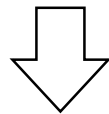
CLIPS> (system "DIR") ←┘

CLIPS> (system "command") ←┘return to DOS

CLIPS> (system "PE2") ←┘

## 11. The use of special symbol

CLIPS> (system "C:\\DOS\\CHKDSK.EXE") ←┘



1st '\': The following is some special symbol.

2nd '\': The special symbol used as the content of the list.