

Intelligence Artificielle

Introduction à l'environnement logiciel Clips



CLIPS C Langage Integrated Production System



Version 6.3

PRESENTATION DE CLIPS

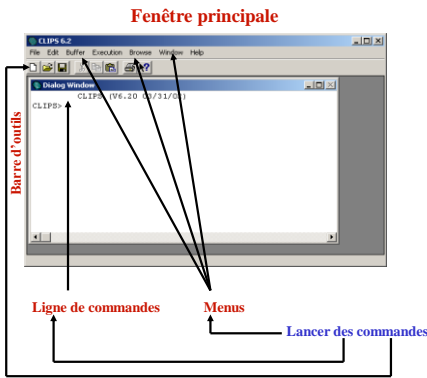
- Outil de représentation de connaissances
- CLIPS intègre un moteur d'ordre 1
- Logiciel du domaine public réalisé par la NASA
- CLIPS est non monotone (faits rétractables)
- CLIPS permet le chaînage avant (mais on peut simuler le chaînage arrière en utilisant des méta-règles)
- CLIPS est en régime irrévocable (pas de backtrack, mais on peut également simuler le backtrack)

PRESENTATION DE CLIPS

Caractéristiques

- Différents formalismes de représentation des connaissances:
 - Règles de production
 - Objets
 - Procédures
- Un mécanisme d'inférence intégré
- Un langage
 - Puissant
 - Portable
 - Ouvert
 - Extensible

ENVIRONNEMENT CLIPS



Codage de systèmes à base de règles de production avec l'environnement logiciel CLIPS

Les informations manipulées dans Clips

Informations symboliques

- Les symboles (symbol)
- Les chaînes de caractères (string)

Informations numériques

- Les nombres (number)
 - Les entiers (integer)
 - Les réels (float)

Codage de systèmes à base de règles de production avec l'environnement logiciel CLIPS

Les informations manipulées dans Clips

```

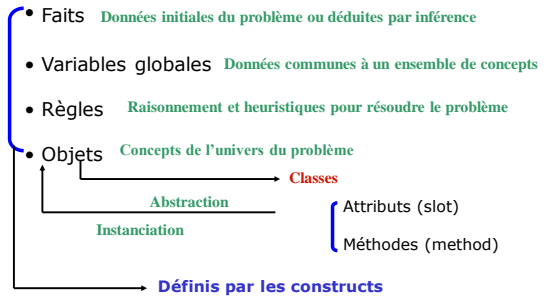
: : : : :
: : : DES SYMBOLES
: : : : :
CLIPS> EPS1
EPS1
CLIPS> informatique
informatique
: : : : :
: : : DES STRINGS
: : : : :
CLIPS> "EPS1"
"EPS1"
CLIPS> "a et b"
"a et b"
: : : : :
: : : DES NOMBRES
: : : : :
CLIPS> 747
747
CLIPS> 17.02
17.02
    
```

Constantes

Les symboles, les strings et les nombres renvoient leur propre valeur

Codage de systèmes à base de règles de production avec l'environnement logiciel CLIPS

Formes de connaissances



LA Codage de systèmes à base de règles de production avec l'environnement logiciel CLIPS

Les faits

• Les faits (facts)

- un objet **une instance d'une classe**
 - une structure : nom + suite de paire (attribut valeur) **un template**
 - une liste **(atome-1 atome-2 ... atome-n)**
 - symbole
 - symbole, nombre, string
- Relation entre atome-2 atome-3 ... atome-n
- Enregistrement (Pascal, C)
Relation (SGBDR)

Codage de systèmes à base de règles de production avec l'environnement logiciel CLIPS

Les faits

Exemples de faits de type liste

(Grand-pere Philippe Bernard) "Philippe est le grand père de Bernard"
 (Est-parti Jacques "à" Paris) "Jacques est parti à Paris"
 (Est-de altitude 10000 "mètres") "L'altitude est de 10000 mètres"

Affirmation des faits → Fonction assert

Codage de systèmes à base de règles de production avec l'environnement logiciel CLIPS

Les faits

La fonction assert

Action

Affirme un ou plusieurs fait(s)

Syntaxes

- 1- (**assert** <un-fait>)
- 2- (**assert** <une-liste-de-faits>)

<un-fait> : fait à affirmer

<une-liste-de faits> : la liste des faits à affirmer

Codage de systèmes à base de règles de production avec l'environnement logiciel CLIPS

Les faits

La fonction assert

Exemples Syntaxe 1

```
!
::: AFFIRMATIONS DES FAITS
:::
DLIPS> (assert(Grand-pere Philippe Bernard))
<Fact-0>
DLIPS> (assert(Est-de altitude 10000 "mètres"))
<Fact-1>
DLIPS> (assert(Est-parti-a Jacques Paris))
<Fact-2>
```

rajout du fait d'indice 0 à la liste des faits

affichage du fait d'indice 0

Format d'affichage : <Fact-xxx>

indice du fait

Codage de systèmes à base de règles de production avec l'environnement logiciel CLIPS

Les faits

La fonction assert

Exemples Syntaxe 2

```
;;*****  
;; AFRMATIONS DES FAITS  
;;*****
```

```
CLIPS> (assert  
(Grand-pere Philippe Bernard)  
(Est-de altitude 10000 "mètres")  
(Est-parti-a Jacques Paris)  
)  
<Fact-2>
```

Exemples Syntaxe 3

```
CLIPS> (+ 3 4)  
Renvoie 7
```

Codage de systèmes à base de règles de production avec l'environnement logiciel CLIPS

Les faits

Définition et affirmation d'une collection de faits initiaux

- Définir la collection des faits initiaux avec le constructeur **deffacts**
- Rajouter cette collection de faits initiaux à la base de faits avec la commande **reset**

Codage de systèmes à base de règles de production avec l'environnement logiciel CLIPS

Les faits

Le construct deffacts

Action

Définit une liste de faits initiaux

```
(deffacts <nom-deffacts> |<commentaires>|  
  <fait-1>  
  <fait-2>  
  ....  
  <fait-n>  
)
```

<nom-deffacts> : identificateur de la liste de faits
|<commentaires>| : commentaires facultatifs commençant par un ;
<fait-1> ... <fait-n> : les faits initiaux à définir

Codage de systèmes à base de règles de production avec l'environnement logiciel CLIPS

Les faits

La commande reset

Action

Rajoute à la base de faits une liste de faits initiaux défini par deffacts

Syntaxe

(reset)

Codage de systèmes à base de règles de production avec l'environnement logiciel CLIPS

Les faits

Définition par defaults et affirmation par reset

Exemple

```
CLIPS> (defacts base-de-faits
(Grand-pere Philippe Bernard)
(Est-de altitude 10000 "mètres")
(Est-parti-a Jacques Paris)
)
CLIPS> (reset)
CLIPS> (facts)
f-0 (initial-fact)
f-1 (Grand-pere Philippe Bernard)
f-2 (Est-de altitude 10000 "mètres")
f-3 (Est-parti-a Jacques Paris)
For a total of 4 facts.
CLIPS> |
```

Annotations:
 - "définition" points to the first line of the code block.
 - "affirmation" points to the "(reset)" command.
 - "affichage" points to the "(facts)" command.
 - "fait initial rajouté automatiquement" points to the "initial-fact" line.
 - "taille de la liste des faits" points to the "For a total of 4 facts." line.

Codage de systèmes à base de règles de production avec l'environnement logiciel CLIPS

Les faits

La commande facts

Action

Affiche la liste des faits (fact list)

Syntaxe

(facts |<deb>| |<fin>| |<max>|)

Exemples

```
CLIPS> (facts)
f-0 (Grand-pere Philippe Bernard)
f-1 (Est-de altitude 10000 "mètres")
f-2 (Est-parti-a Jacques Paris)
For a total of 3 facts.
CLIPS> |
```

Annotations:
 - "<deb> : indice du premier fait" points to the first parameter in the syntax.
 - "<fin> : indice du dernier fait" points to the second parameter.
 - "<max> : maximum d'affichage" points to the third parameter.
 - "indice du fait" points to the "f-0" label.
 - "Taille de la liste des faits" points to the "For a total of 3 facts." line.
 - "Identificateurs : <f-xxx>" points to the "f-0" label.

Codage de systèmes à base de règles de production avec l'environnement logiciel CLIPS

Les faits

La fonction retract

Action

Retire un fait de la liste des faits

Syntaxe

(retract <un-indice>) <un-indice> : indice du fait à retirer

Exemple

```
CLIPS> (retract 2)
CLIPS> (facts)
f-0 (Grand-pere Philippe Bernard)
f-1 (Est-de altitude 10000 "mètres")
For a total of 2 facts.
```

Annotation: "Retire le fait d'indice 2" points to the "(retract 2)" command.

Codage de systèmes à base de règles de production avec l'environnement logiciel CLIPS

Les faits

La commande clear

Action

Retire tous les faits de la base de faits et toutes les règles de la base de règles

Syntaxe

(clear)

Exemple

```
CLIPS> (facts)
f-0 (Grand-pere Philippe Bernard)
f-1 (Est-de altitude 10000 "mètres")
For a total of 2 facts.
CLIPS> (clear)
CLIPS> (facts)
CLIPS> |
```

Codage de systèmes à base de règles de production avec l'environnement logiciel CLIPS

Les faits

Les templates

- Des **structures de données** similaires à celles des enregistrements dans C et Pascal et des relations dans les bases de données
- Une structure est composé de :
 - un nom
 - des champs (**slots**)
- Une structure permet de définir pour un slot :
 - une valeur par défaut (**default**),
 - une intervalle de valeurs possibles (**range**)
 - une liste de valeurs possibles (**allowed values**)

Codage de systèmes à base de règles de production avec l'environnement logiciel CLIPS

Les faits

Le constructeur deftemplate

Action

Définit une structure

Syntaxe

```
(deftemplate <nom-template> |<commentaires>|  
  <slot champ-1> | <multislot champ-1>  
  <slot champ-2>  
  ....  
  <champ-n>  
)
```

<nom-template> : identificateur de la structure
|<commentaires>| : commentaires facultatifs commençant par un ;
<champ-1> ... <champ-n> : les champs de la structure

Codage de systèmes à base de règles de production avec l'environnement logiciel CLIPS

Les faits

Le constructeur deftemplate

Définition de la structure : exemple

```
(deftemplate Vol ; une relation d'une base de données  
  (slot Numvol (type STRING) ; utilisée par une agence  
  (default "Vol Tun728")) ; de voyage marseillaise  
  (slot Prov (type STRING) ; spécialisée dans les  
  (default "Marseille")) ; voyages à destination du Maghreb  
  (slot Dest (type STRING)  
  (allowed-values "Tunis" "Alger" "Casablanca")  
  (default "Tunis"))  
  (slot Durvol (type FLOAT)  
  (range 0.00 15.00) (default 1.25))  
)
```

Codage de systèmes à base de règles de production avec l'environnement logiciel CLIPS

Les faits

Le constructeur deftemplate

Affirmation de faits de type structure : exemple

```
(assert(Vol))  
(assert  
  (Vol  
  (Numvol "Vol Alg879") (Dest "Alger") (Durvol  
  2.25)  
  )  
  )  
(assert  
  (Vol (Numvol "Vol Cas235") (Dest "Casablanca")  
  (Durvol 2.45)  
  )  
  )
```

Codage de systèmes à base de règles de production avec l'environnement logiciel CLIPS

Les faits

Le constructeur deftemplate

Structure d'une base de faits de type template

```
CLIPS> (facts)
f-0 (Vol (Numvol "Vol Tun728") (Prov "Marseille") (Dest "Tunis") (Durvol 1.25))
f-1 (Vol (Numvol "Vol Alg879") (Prov "Marseille") (Dest "Alger") (Durvol 2.25))
f-2 (Vol (Numvol "Vol Cas295") (Prov "Marseille") (Dest "Casablanca") (Durvol 2.45))
For a total of 3 Facts.
```

Codage de systèmes à base de règles de production avec l'environnement logiciel CLIPS

Les règles

Le constructeur defrule

Action

Définit une règle

Syntaxe

```
(defrule <nom-règle> |<commentaires>|
  <condition-1>
  <condition-2>
  ....
  <condition-n>
=>
  <action-1>
  <action-2>
  ....
  Action-n
)
```

<nom-règle> : nom de la règle

<condition-1> ... <condition-n> :
partie gauche de la règle

<action-1> ... <action-n> :
partie droite de la règle

Codage de systèmes à base de règles de production avec l'environnement logiciel CLIPS

Les faits

Le constructeur deftemplate

Les templates et les règles

```
(assert
(Vol
  (Numvol "Vol Tun635")
)
)

(defrule vols-pour-tunis
(Vol (Numvol ?Numero) (Dest "Tunis"))

=>
(printout t "Numéro de vol: " ?Numero crlf)
)
```

Codage de systèmes à base de règles de production avec l'environnement logiciel CLIPS

Les faits

Le constructeur deftemplate

Les templates et les règles

```
CLIPS> (run)
Numéro de vol: Vol Tun635
Numéro de vol: Vol Tun728
CLIPS>
```

Codage de systèmes à base de règles de production avec l'environnement logiciel CLIPS

Les règles

Le construct defrule

Exemple

Soit la règle : « si le patient a des crises de glycémie quand il est à jeun et si le patient est âgé de 0 à 10 ans alors il s'agit d'un diabète de type gestationnel »

Représentation en Clips avec defrule : syntaxe 1

```
(defrule type-de-diabete "quel type de diabete"
  (glycemie periode a-jeun)
  (age interval 0-10-ans)
=>
  (assert(diabete type gestationnel))
)
```

Codage de systèmes à base de règles de production avec l'environnement logiciel CLIPS

Les règles

La commande rules

Action

Affiche la liste des règles

Syntaxe

(rules)

La commande ppdefrule

Action

Affiche le contenu d'une règle

Syntaxe

(ppdefrule <nom-règle>)

<nom-règle> : nom de la règle à afficher

Codage de systèmes à base de règles de production avec l'environnement logiciel CLIPS

Les règles

Exemple complet

```
CLIPS>
(defrule type-de-diabete ; quel type de diabete
  (glycemie periode a-jeun)
  (age interval 0-10-ans)
=>
  (assert(diabete type gestationnel))
)
CLIPS> (rules)
type-de-diabete
For a total of 1 defrule.
CLIPS> (ppdefrule type-de-diabete)
(defrule MAIN::type-de-diabete
  (glycemie periode a-jeun)
  (age interval 0-10-ans)
=>
  (assert (diabete type gestationnel)))
CLIPS>
```

Définition d'une règle

Affichage de la liste des règles

Affichage du contenu d'une règle

Codage de systèmes à base de règles de production avec l'environnement logiciel CLIPS

Les faits

La fonction undefrule

Action

Retire une règle de la base de règles

Syntaxe

(undefrule <un-indice>)

Exemple

(undefrule type-de-diabete)

Codage de systèmes à base de règles de production avec l'environnement logiciel CLIPS

Les fonctions d'E/S

La fonction printout

Action

Affiche une expression à l'écran

Syntaxe

```
(printout t <expression>
)
```

<expression> : expression à afficher

Codage de systèmes à base de règles de production avec l'environnement logiciel CLIPS

Les fonctions d'E/S

La fonction modify

Action

Modifie un fait

Syntaxe

```
(modify <fact-index> <slot-modifier>+)
      <slot-modifier> (<slot-name> <slot-value>)
```

Exemple

```
(modify 0 (age 24) )
```

Codage de systèmes à base de règles de production avec l'environnement logiciel CLIPS

Les fonctions d'E/S

La fonction duplicate

Action

Dupliquer un fait

Exemple

```
(duplicate 2 (nom "Sabeur" ) )
```

Codage de systèmes à base de règles de production avec l'environnement logiciel CLIPS

Les fonctions d'E/S

Les variables:

Stocker des information. Elle doit être précédé par ?

Exemple 1

```
(defrule Changer-Employe
?fl <- (tache (employe "Sami"))
=>
(modify ?fl (employe "Sabeur")))
```

Codage de systèmes à base de règles de production avec l'environnement logiciel CLIPS

Les fonctions d'E/S

Les variables:

Stocker des information. Elle doit être précédé par ?

Exemple 1

```
(defrule substitute-instructor
"Replace the instructor Jim with
Joe" ?Jim's-Course <- (course
(instructor "Jim")) => (modify
?Jim's-Course (instructor "Joe")))
```

Exemple 2

```
(defrule r1
?y<-(birthday ?nom)
?fact<- (person(nom ?n)(age ?a))
=>
(modify ?fact(age(+ ?a 1)))
(retract fact))
```

Codage de systèmes à base de règles de production avec l'environnement logiciel CLIPS

Les fonctions d'E/S

La fonction test

Action

Permet d'effectuer un test de condition

Exemple

```
(test (> age 20))

(test (> ?poid 100))
```