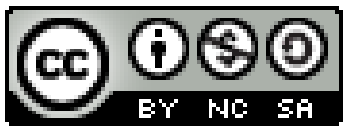
ΑΡΙΣΤΟΤΕΛΕΙΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΟΝΙΚΗΣ

ΑΝΟΙΚΤΑ
ΑΚΑΔΗΜΑΙΚΑ
ΜΑΘΗΜΑΤΑ

Συστήματα Γνώσης

Πρακτικό Κομμάτι Μαθήματος
Συναρτήσεις στο CLIPS

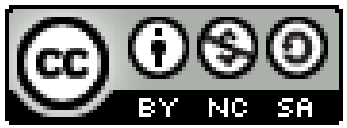
Νίκος Βασιλειάδης, Αναπλ. Καθηγητής
Τμήμα Πληροφορικής



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ
ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «Ανοικτά Ακαδημαϊκά Μαθήματα στο Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.

Συναρτήσεις στο CLIPS

Συναρτήσεις στο CLIPS

- Οι συναρτήσεις στο CLIPS έχουν παρόμοια μορφή με εκείνη των συναρτήσεων στην LISP
- Μορφή Κλήσης

```
(<όνομα συνάρτησης> όρισμα1 … όρισμαN)
```

- Τιμή: επιστρέφει στο σημείο στο οποίο εμφανίζεται η συνάρτηση

```
CLIPS> (assert (The number (+ 2 3)))
<Fact-0>
CLIPS> (facts)
f-0     (The number 5)
For a total of 1 fact.
```

Build-in Συναρτήσεις
Ορισμένες από το σύστημα

- Βασικές Αριθμητικές Συναρτήσεις

```
(+ <ορίσματα>)          (- <ορίσματα>)
(* <ορίσματα>)          (/ <ορίσματα>)
```

- Συναρτήσεις σύγκρισης αριθμών

```
(=  <ορίσματα>)          (<  <ορίσματα>)
(>=  <ορίσματα>)  (>  <ορίσματα>)
(<=  <ορίσματα>)  (<>  <ορίσματα>)
```

- Οι ακόλουθες συναρτήσεις επιστρέφουν **TRUE**

```
(>= 5 5 4 2 2 1)
(<=  2 3 3 4 6)
(<> 3 5)
```

Λογικές Συναρτήσεις

- Δυνατότητα να εκφραστούν πολύπλοκες συνθήκες κανόνων

```
(and  <ορίσματα>)
(or  <ορίσματα>)
(not <όρισμα>)
(eq <ορίσματα>)
(neq  <ορίσματα>)
```

- Οι ακόλουθες συναρτήσεις επιστρέφουν **TRUE**

```
(and (>= 5 5) (> 3 2 1) (= 10 10))
(and (not (= 10 7)) (= 9 9))
```

Λογικές Συναρτήσεις

```
(defrule days
   (month Jan mon)
   (or (hour 12) (hour 13))
  =>
   (assert (day is Monday noon time))
)

(defrule days
   (month Jan mon)
   (or (and (hour 12) (lunch break))
       (and (hour 17) (departure time))
   )
  =>
   (assert (office closed at this
  hour))
)
```

Συναρτήσεις Ελέγχου Τύπου

- Όταν πρέπει να εξακριβωθεί ο τύπος της τιμής μιας μεταβλητής

```
(numberp <όρισμα>)
(integerp <όρισμα>)
(floatp <όρισμα>)
(stringp <όρισμα>)
(symbolp <όρισμα>)
```

- Οι ακόλουθες συναρτήσεις επιστρέφουν **TRUE**

```
(numberp 4)    (symbolp day)
```

Παραδείγματα υπολογισμού συναρτήσεων

```
(+ 6 5 9)
20
(- 4 6 8)
-10
(* 2 3 (+ 1 2))
18
(< 2 6 8)
TRUE
(eq 8 8)
TRUE
```

- (eq 5 5.0)
- FALSE
- (= 4 4.0)
- TRUE
- (and (eq 2 3)(= 1 1))
- FALSE
- (and (eq 2 2) (< 2 3))
- TRUE

Συναρτήσεις χειρισμού πολλαπλών τιμών

(create$ <ορίσματα>)

- Επιστρέφει μια τιμή (λίστα) που μπορεί να ανατεθεί σε μεταβλητή πολλαπλών τιμών, την οποία δημιουργεί από τα ορίσματα.

```
CLIPS> (create$ the day is (grey as it
  was))
(the day is grey as it was)
```

(explode$ <string>)

- Επιστρέφει μια πολλαπλή τιμή (λίστα) την οποία δημιουργεί από αλφαριθμητικό.

```
CLIPS> (explode$ "the night was blue")
(the night was blue)
```

Συναρτήσεις χειρισμού πολλαπλών τιμών

(implode$ <multivalue>)

- Επιστρέφει το αντίστοιχο αλφαριθμητικό από μια πολλαπλή τιμή.
  - Είναι η αντίθετη εντολή της **explode**

```
CLIPS> (implode$ (create$ the night was
  blue))
"the night was blue"
```

- Είναι όμοια με:

```
(implode$ (explode$ "the night was blue"))
```

Συναρτήσεις χειρισμού πολλαπλών τιμών

(nth$ N <multivalue>)

- Επιστρέφει το N-οστό πεδίο μιας πολλαπλής τιμής

```
CLIPS> (nth$ 2 (create$ 3 4 5))
4
```

(member$ <symbol> <multivalue>)

- Επιστρέφει τη θέση του πεδίου **<symbol>** μέσα στην τιμή **<multivalue>** εφόσον αυτό υπάρχει.
  - Εάν δεν υπάρχει επιστρέφει **FALSE**.

```
CLIPS>(member$ c (create$ a b c))
3
```

Συναρτήσεις χειρισμού πολλαπλών τιμών

(first$ <multivalue>)

- Επιστρέφει το πρώτο στοιχείο μιας πολλαπλής τιμής, αλλά σε μορφή λίστας.

```
CLIPS> (first$ (create$ 3 4 5))
(3)
```

(rest$ <multivalue>)

- Επιστρέφει τα υπόλοιπα στοιχεία εκτός από το πρώτο στοιχείο μιας πολλαπλής τιμής.

```
CLIPS>(rest$ (create$ 3 4 5))
(4 5)
```

Συναρτήσεις εισόδου - εξόδου

(printout <device> <expression>)

- Αποστέλλει την έκφραση **<expression>** στη συσκευή **<device>**
  - Η συσκευή μπορεί να είναι ένα αρχείο ή η οθόνη
  - Για οθόνη, χρησιμοποιούμε **t** (terminal)

```
(printout t "The day was " ?type crlf)
The day was sunny
```

- Το σύμβολο **crlf** δηλώνει αλλαγή γραμμής

Συναρτήσεις εισόδου - εξόδου

(read)

- Εισάγει σύμβολο από το πληκτρολόγιο
- Συνήθως χρησιμοποιείται σε συνδυασμό με την εντολή **bind**, για ανάθεση τιμής σε μεταβλητή στις ενέργειες ενός κανόνα

```
(defrule get-user-answer
  (initial-fact)
  =>
  (printout t "What's your name: ")
  (bind ?name (read))
  (assert (user-name ?name))
)
```

Ανάθεση τιμής σε μεταβλητή

(bind <variable> <value>)

- Ανατίθεται η τιμή **<value>** σε μια μεταβλητή **<variable>** στις **ενέργειες** των κανόνων

```
(defrule rule1 "example rule"
  (oldcost ?oldcost)
  (newcost ?newcost)
  =>
  (bind ?total_cost (+ ?newcost
  ?oldcost))
  (assert (cost ?total_cost))
  (printout t "The total cost is "
  ?total_cost crlf)
)
```

Συναρτήσεις εισόδου - εξόδου

- Η **`(read)`** επιστρέφει SYMBOL
- Αν πληκτρολογηθούν λέξεις με κενό μεταξύ τους, επιστρέφει μόνο την πρώτη και αγνοεί τις υπόλοιπες
  - Π.χ. **`CLIPS> (read)`**
  - Πληκτρολογούμε: **`hello there`**
  - Επιστρέφει: **`hello`**

Συναρτήσεις εισόδου - εξόδου

`(readline)`

- Διαβάζει οτιδήποτε πληκτρολογηθεί μέχρι να πατηθεί enter και επιστρέφει ένα string
  - Π.χ. **`CLIPS> (readline)`**
  - Πληκτρολογούμε: **`hello there`**
  - Επιστρέφει: **`"hello there"`**

Αρχεία σε CLIPS

- Είσοδος/έξοδος μπορεί να γίνει και από/προς εξωτερικά αρχεία (κειμένου)
- Πριν την πρόσβαση σε ένα αρχείο, αυτό πρέπει να ανοιχθεί κατάλληλα

```
(open "mydata.dat" data "r")
```

  - "**mydata.dat**": το όνομα του αρχείου
    - Μπορεί να προστεθεί και path
  - **data**: identifier (*λογικό όνομα*) του αρχείου για όση ώρα θα είναι ανοιχτό
    - Χρησιμοποιείται σε συνδυασμό με εντολές read / printout
  - "**r**" – το αρχείο ανοίχθηκε μόνο για ανάγνωση (read)

Αρχεία σε CLIPS

- Η συνάρτηση **`open`** επιστρέφει:
  - **`TRUE`** αν ανοίξει το αρχείο κανονικά
  - **`FALSE`** αν όχι (π.χ. δεν υπάρχει)
- Τρόποι προσπέλασης αρχείου
  - **`r`**: read only – ανάγνωση μόνο
  - **`w`**: write only – εγγραφή μόνο
  - **`r+`**: read/write - ανάγνωση και εγγραφή
  - **`a`**: append only – μόνο προσθήκη εγγραφών στο τέλος του αρχείου

Αρχεία σε CLIPS

- Συνάρτηση **(close)**
  - Αν δεν εκτελεστεί μπορεί να χαθούν εγγραφές στο αρχείο
  - Χρήση: **(close data)**
- Εγγραφή σε αρχείο
  - **(printout data "Hello world" crlf)**
- Ανάγνωση από αρχείο
  - **(bind ?line (readline data))**
  - **(bind ?word (read data))**

Έλεγχος ροής προγράμματος
Συνάρτηση while

```
(while (condition) do          ;συνθήκη
  (command 1)                  ;εντολή 1
  ...
  (command n)                  ;εντολή n
)
```

- Όσο ικανοποιείται μία συνθήκη, εκτελείται ένα σύνολο συναρτήσεων.
- Είναι απαραίτητη η **`bind`** για να αλλάξει τις τιμές μεταβλητών που συμμετέχουν στη συνθήκη

Έλεγχος ροής προγράμματος

Συνάρτηση while

- Έστω ότι υπάρχει ένα γεγονός **(num 1)**
- Τύπωσε όλους τους αριθμούς από το 1 μέχρι το 9, με τη φράση "**the num is:**" σαν πρόθεμα

```
(defrule test
  (num ?n)
  =>
  (while (< ?n 10)
     do
   (printout t "the num is: " ?n crlf)
   (bind ?n (+ 1 ?n))
  )
)
```

Έλεγχος ροής προγράμματος

Συνάρτηση if-then-else

```
(if     (condition) ;συνθήκη
  then
     (command 1) ;εντολή 1
     ...
     (command M) ;εντολή M
  else
     (command A) ;εντολή A
     ...
     (command N) ;εντολή N
)
```

- Εκτελεί υπό συνθήκη κάποια σύνολα εντολών

Έλεγχος ροής προγράμματος

Συνάρτηση if-then-else

- Τύπωσε **positive, negative** ή **zero**, αν ο αριθμός **?n** είναι μεγαλύτερος, μικρότερος ή ίσος με μηδέν

```
(defrule sign
      (num ?n)
  =>
    (if (> ?n 0)
     then (printout t "positive" crlf)
     else (if (< ?n 0)
              then  (printout  t  "negative"
  crlf)
              else (printout t "zero" crlf))
    )
)
```

Ορισμός Συναρτήσεων

```
(deffunction <function name> (<variables>)
   (command 1)
   ...
   (command n)
)
```

- Η συνάρτηση επιστρέφει την τιμή της τελευταίας εντολής που εκτελείται

```
(deffunction mean-value (?v1 ?v2 ?v3 ?v4)
      ( / (+ ?v1 ?v2 ?v3 ?v4) 4)
)
CLIPS> (mean-value 4 5 6 7)
5.5
```

Ορισμός Συναρτήσεων

Παράδειγμα

```
(deffunction deviation (?v1 ?v2 ?v3
  ?v4)
  (bind ?avg (mean-value ?v1 ?v2 ?v3
  ?v4))
  (bind ?d1 (- ?avg ?v1))
  (bind ?d2 (- ?avg ?v2))
  (bind ?d3 (- ?avg ?v3))
  (bind ?d4 (- ?avg ?v4))
  (create$  ?avg ?d1 ?d2 ?d3 ?d4)
)

CLIPS > (deviation 5 6 7 8)
(6.5 1.5 0.5 -0.5 -1.5)
```

Παράδειγμα Επιλογής Δώρου
Συνάρτηση Ερώτησης προς Χρήστη

```
(deffunction ask-question
             (?question $?allowed-values)
   (printout t ?question)
   (bind ?answer (read))
   (if (lexemep ?answer)
       then (bind ?answer (lowcase
 ?answer)))
   (while (not (member ?answer ?allowed-
 values)) do
      (printout t ?question)
      (bind ?answer (read))
      (if (lexemep ?answer)
         then (bind ?answer (lowcase
 ?answer))))
    ?answer)
```

Παράδειγμα Επιλογής Δώρου
Κανόνας Εισαγωγής Δεδομένων

```
(defrule questions "ask questions to user"
   (initial-fact)
=>
   (bind ?music
       (ask-question "Does he/she likes music
                                  (yes/no)?" yes no) )
   (if (eq ?music yes) then (assert (like music
   ))
   (bind ?educated  (ask-question "Is he/she
                          educated (yes/no)?" yes
   no) )
   (assert (educated ?educated))
   (bind ?expen  (ask-question "Do you want an
                  expensive gift (yes/no)?" yes
   no) )
   (if (eq ?expen yes)
      then (assert (price expensive)) )
   (printout t "How many years old is he/she?")
   (bind ?age (read))
   (assert (age ?age))
)
```

Παράδειγμα Επιλογής Δώρου
Κανόνας Εκτύπωσης Αποτελεσμάτων

```
(defrule printresult
   "a rule to print out the
  results"
    (gift ?x)
    =>
    (printout t "A possible gift is
  " ?x crlf)
)
```

- Λόγω της μεταβλητής **?x** θα τυπωθούν όλα τα δώρα

Παράδειγμα Επιλογής Δώρου

Νέος Τρόπος Αλληλεπίδρασης

```
CLIPS> (load "gift_adv.clp")
TRUE
CLIPS> (reset)
CLIPS> (facts)
f-0     (initial-fact)
For a total of 1 fact.
CLIPS> (run)
Does he/she likes music (yes/no)? yes
Is he/she educated (yes/no)? yes
Do you want an expensive gift
  (yes/no)? no
How many years old is he/she? 19
A possible gift is CD
A possible gift is book
```

Παράδειγμα Επιλογής Δώρου
Νέος Τρόπος Αλληλεπίδρασης

```
CLIPS> (facts)
f-0     (initial-fact)
f-1     (music yes)
f-2     (educated yes)
f-3     (age 19)
f-4     (agegroup middle)
f-5     (gift CD)
f-6     (gift book)
For a total of 7 facts.
```

Παράδειγμα Επιλογής Δώρου
Κανόνες Ηλικιακών Ομάδων με `test`

Παλιός Κανόνας

```
(defrule middle "middle age rule"
   (age ?x)
  =>
   (if (and (> ?x 14) (< ?x 35))
   then (assert (agegroup middle))))
```

Νέος Κανόνας

```
(defrule middle "middle age rule"
   (age ?x)
   (test (and (> ?x 14) (< ?x 35)))
  =>
   (assert (agegroup middle)))
```

ΑΡΙΣΤΟΤΕΛΕΙΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΟΝΙΚΗΣ

ΑΝΟΙΚΤΑ
ΑΚΑΔΗΜΑΙΚΑ
ΜΑΘΗΜΑΤΑ

Τέλος Ενότητας

Επεξεργασία: Εμμανουήλ Ρήγας

Θεσσαλονίκη, 17/3/2014

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο

ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης