

## Η Γλώσσα Uni-CLIPS

### Εισαγωγή

Η γλώσσα Uni-CLIPS είναι ένα μικρό υποσύνολο της γλώσσας που χρησιμοποιεί το περιβάλλον ανάπτυξης εμπείρων συστημάτων CLIPS.

Η Uni-CLIPS περιλαμβάνει κυρίως συναρτήσεις για τον ορισμό γεγονότων, τα οποία περιγράφουν την κατάσταση στην οποία βρίσκεται ένα πρόβλημα σε μια δεδομένη στιγμή, και τον ορισμό κανόνων, οι οποίοι περιγράφουν το πώς μπορούν τα γεγονότα αυτά να τροποποιηθούν ώστε να αλλάξει μια τρέχουσα κατάσταση του προβλήματος. Επιπλέον, διαθέτει συναρτήσεις σύγκρισης τιμών και αριθμητικών πράξεων.

Η γραμματική περιγράφει τα απαραίτητα στοιχεία της γλώσσας ώστε να μπορεί κανείς να συντάξει εύκολα ένα στοιχειώδες πηγαίο αρχείο Uni-CLIPS και να πειραματιστεί με τη δημιουργία και χρήση λεκτικού και συντακτικού αναλυτή της γλώσσας για τις ανάγκες του εργαστηρίου του μαθήματος των Μεταγλωττιστών.

### ΜΕΡΟΣ 1ο: Λεκτική Ανάλυση

Ένα πηγαίο πρόγραμμα Uni-CLIPS διαβάζεται από έναν λεκτικό αναλυτή – ΛΑ (lexical analyzer) ως μια μεγάλη συμβολοσειρά εισόδου. Ο ΛΑ διαχωρίζει ένα-ένα τα λεξήματα που περιέχονται μέσα στη συμβολοσειρά εισόδου και τα αναγνωρίζει ως λεκτικές μονάδες κάθε φορά που του στέλνει ένα αίτημα ο συντακτικός αναλυτής – ΣΑ (parser). Αυτό το κεφάλαιο περιγράφει πώς ο λεκτικός αναλυτής πραγματοποιεί τον διαχωρισμό και την αναγνώριση των λεξημάτων του πηγαίου κώδικα.

### Το αλφάβητο

Το αλφάβητο της γλώσσας αποτελείται από τους ακόλουθους ASCII χαρακτήρες:

Πεζοί και κεφαλαίοι λατινικοί χαρακτήρες: a-z A-Z

Αριθμητικά ψηφία: 0–9

Ειδικοί χαρακτήρες: [ ; ? ( ) + - \* / = > \_ ] .

Whitespace χαρακτήρες: κενό (space \s), tab (\t), νέα-γραμμή (NEWLINE \n), τέλος-αρχείου (EOF)

Η γλώσσα κάνει διάκριση μεταξύ πεζών και κεφαλαίων (case sensitive language).

Η λεκτική δομή της Uni-CLIPS είναι αρκετά απλή. Αποτελείται από λεκτικές μονάδες, κρατημένες λέξεις, σχόλια και διαχωριστές μεταξύ τους:

### Λεκτικές μονάδες (tokens) σχόλια και διαχωριστές

Τα tokens που πρέπει να διαχωρίσει και αναγνωρίσει ο λεκτικός αναλυτής της γλώσσας είναι τα παρακάτω:

- αριθμητικά κυριολεκτικά (ακέραιοι και αριθμοί κινητής υποδιαστολής)
- ονόματα ορισμών και άλλων στοιχείων μέσα σε γεγονότα
- ονόματα μεταβλητών
- λεκτικά κυριολεκτικά (συμβολοσειρές)

Επίσης, πρέπει να είναι σε θέση να αναγνωρίσει τα σχόλια και τις συμβολοσειρές μεταξύ των λέξεων ενός πηγαίου κώδικα που αποτελούν διαχωριστές τους. Τα σχόλια και οι διαχωριστές δεν αποτελούν tokens της γλώσσας και κατά την αναγνώρισή τους, ο λεκτικός αναλυτής δεν επιστρέφει κάποια λεκτική μονάδα στον συντακτικό αναλυτή.

Ακολουθεί η περιγραφή σύνταξης των παραπάνω.

### **Ακέραιοι αριθμοί**

Οι ακέραιοι αριθμοί της γλώσσας πρέπει να αρχίζουν από ένα μη μηδενικό ψηφίο (1-9) και να ακολουθούν κανένα, ένα ή περισσότερα ψηφία (0-9). Μπορεί να είναι προσημασμένοι ή όχι. Ειδική περίπτωση το ίδιο το μηδέν (0) το οποίο είναι αποδεκτό.

Π.χ.: 5, +8, -115, 1234567, 0

### **Αριθμοί κινητής υποδιαστολής**

Οι αριθμοί κινητής υποδιαστολής (floating point) περιγράφονται ως εξής: Ένας δεκαδικός αριθμός αποτελείται από το ακέραιο μέρος και το δεκαδικό μέρος που διαχωρίζονται με μία τελεία '.'. Τόσο το ακέραιο μέρος όσο και το δεκαδικό ορίζονται σύμφωνα με τους κανόνες που περιγράφηκαν παραπάνω για τους απλούς ακραίους του δεκαδικού αριθμητικού συστήματος. Υπάρχει επίσης η δυνατότητα ορισμού δυνάμεων με χρήση του χαρακτήρα 'e' ή 'E'. Στην περίπτωση αυτή το ακέραιο ή/και δεκαδικό μέρος υψώνεται στην ακέραια δύναμη που ακολουθεί μετά τον χαρακτήρα 'e' ή 'E'. Όπως και οι ακέραιοι, μπορεί να είναι προσημασμένοι ή όχι.

Π.χ.: 3.14 -10.0 +0.001 1e100 3.14e-10 0e0 0.25 0.9

### **Ονόματα ορισμών και στοιχείων γεγονότων**

Τα ονόματα των ορισμών που αφορούν γεγονότα ή κανόνες, όπως επίσης και ονόματα άλλων στοιχείων γενικά που χρησιμοποιούνται κατά τους ορισμούς γεγονότων αποτελούνται από πεζούς και κεφαλαίους λατινικούς χαρακτήρες και μπορούν μετά τον πρώτο λατινικό χαρακτήρα να περιέχουν χαρακτήρες, ψηφία ή τα σύμβολα - και \_.

Π.χ.: static-facts, MoveUp, CUBES, sum-1, table, packman, , A-21-b.

### **Μεταβλητές**

Τα ονόματα των μεταβλητών αρχίζουν υποχρεωτικά με τον χαρακτήρα ? ακολουθούμενο από έναν ή περισσότερους λατινικούς χαρακτήρες ή/και ψηφία.

Π.χ.: ?x, ?X, ?3, ?ad, ?X1b23, ?32AbC, ?ABcd1234de

## Συμβολοσειρές (ή Λεκτικά κυριολεκτικά)

Τα λεκτικά κυριολεκτικά ή απλώς συμβολοσειρές (string literals ή strings) περικλείονται μέσα σε διπλές " αποστρόφους και περιλαμβάνουν οποιονδήποτε χαρακτήρα εκτός του backslash \, της νέας γραμμής \n ή της διπλής αποστρόφου " που για τη χρήση τους απαιτούν χρήση σειράς διαφυγής. Αναγνωρισμένοι συνδυασμοί σειρών διαφυγής (escape sequence) μέσα στα strings είναι:

| Escape Sequence | Meaning             | Notes   |
|-----------------|---------------------|---------|
| \\              | Backslash (\)       |         |
| \"              | Double quote (")    |         |
| \n              | ASCII Linefeed (LF) | NEWLINE |

Π.χ.: "" "Test" "Hello World" "Mark said, \"Boo!\""

## Σχόλια

Τα σχόλια αρχίζουν με το σύμβολο ; (ελληνικό ερωτηματικό) και ολοκληρώνονται στην ίδια γραμμή με το end-of-line. Τα σχόλια μπορούν να περιλαμβάνουν μετά το ; οποιονδήποτε χαρακτήρα εντός ή εκτός αλφαβήτου.

## Διαχωριστές

Η γλώσσα δέχεται ως διαχωριστές συμβολοσειρές ενός ή περισσότερων **κενών χαρακτήρων** (blanks), ενός ή περισσότερων tabs, το τέλος γραμμής (end-of-line) και το τέλος αρχείου (end-of-file). Οι δυο παρενθέσεις και ορισμένα από τα υπόλοιπα σύμβολα της αλφαβήτου της γλώσσας χρησιμοποιούνται επίσης ως διαχωριστές κατά περίπτωση μέσα στη σύνταξη.

## ΚΡΑΤΗΜΕΝΕΣ ΛΕΞΕΙΣ

Στην Uni-CLIPS, τα ακόλουθα ονόματα χρησιμοποιούνται ως λέξεις κλειδιά (reserved words, ή keywords) και δεν απαιτούν πρότυπο αναγνώρισης γιατί είναι ήδη καταχωρημένα μέσα στον πίνακα συμβόλων, όμως επιστρέφονται κανονικά ως tokens με αντίστοιχο αναγνωριστικό όνομα.

### Πρωταρχικές συναρτήσεις ορισμών

Κρατημένες λέξεις για τις πρωταρχικές συναρτήσεις της γλώσσας είναι:

**def facts:** χρησιμοποιείται για τον ορισμό γεγονότων,

**def rule:** χρησιμοποιείται για τον ορισμό ενός κανόνα.

### Πρωταρχικές συναρτήσεις για συγκρίσεις, αναθέσεις, ανάγνωση και εκτυπώσεις

**test** : που χρησιμοποιείται για την πραγματοποίηση σύγκρισης μεταξύ δυο ή περισσότερων ακεραίων ή/και μεταβλητών

**bind** : ανάθεση τιμής σε μεταβλητή

**read** : ανάγνωση από τη μονάδα εισόδου

***printout*** : που χρησιμοποιείται για την πραγματοποίηση εκτύπωσης στην οθόνη

### ***Τελεστές***

Κρατημένες λέξεις για συγκρίσεις και τελεστές σύγκρισης

= : που χρησιμοποιείται ως ο μοναδικός τελεστής σύγκρισης,

+, -, \*, / : που χρησιμοποιούνται για ως τελεστές αριθμητικών πράξεων.

Σημείωση: Οι κρατημένες λέξεις για τους τελεστές αποτελούν επίσης πρωταρχικές συναρτήσεις για την Uni-CLIPS.

## Μέρος 2ο: Συντακτική Ανάλυση

Ο ρόλος του συντακτικού αναλυτή – ΣΑ (syntax analyser-parser) είναι να διαπιστώσει τη συντακτική ορθότητα των δομών του πηγαίου κώδικα βάσει της γραμματικής της γλώσσας. Η είσοδος στον συντακτικό αναλυτή - ΣΑ είναι μια ακολουθία από λεκτικές μονάδες (tokens), που παράγονται από τον λεκτικό αναλυτή - ΛΑ (lexical analyzer-scanner), μετά το διαχωρισμό και την αναγνώριση των λεξημάτων της συμβολοσειράς εισόδου. Τα tokens επιστρέφονται από τον ΛΑ ένα-ένα μετά από συνεχή αιτήματα του ΣΑ κατά τη διάρκεια των ελέγχων που πραγματοποιεί.

Αυτό το κεφάλαιο περιγράφει τη γραμματική για τη δομή ενός πηγαίου κώδικα και τη σύνταξη των εκφράσεων (expressions) της γλώσσας Uni-CLIPS.

### Γραμματική για τη σύνταξη των προτάσεων της γλώσσας

Η γλώσσα Uni-CLIPS αποτελείται κυρίως από ορισμούς γεγονότων με την `deffacts` και ορισμούς κανόνων με την `defrule`.

Μέσα στους κανόνες μπορούν να παρουσιαστούν επιπλέον συναρτήσεις που στην περίπτωση μας θα είναι η συνάρτηση ελέγχου τιμών `test`, οι αριθμητικές πράξεις και η συνάρτηση για εκτύπωση στην οθόνη `printout`.

#### Σύνταξη ενός γεγονότος

Τα γεγονότα περιγράφονται πάντα μέσα σε παρενθέσεις και μπορεί να περιέχουν ένα ή περισσότερα στοιχεία, π.χ.

```
(color A is blue)
(cube A on table)
(color B is white)
(cube B on A)
(number_of_cubes 2)
```

#### Σύνταξη ορισμού γεγονότων

Για να πραγματοποιηθεί ο ορισμός ενός γεγονότος χρησιμοποιείται η συνάρτηση **deffacts** της οποίας η σύνταξη απαιτεί πριν το `deffacts` να υπάρχει παρένθεση, μετά να ακολουθεί ένα όνομα ορισμού, και στη συνέχεια να υπάρχουν ένα ή περισσότερα γεγονότα. Η σύνταξη κλείνει με παρένθεση, π.χ.

```
(deffacts CUBES
  (color A is blue)
  (cube A on table)
  (color B is white)
  (cube B on A)
  (number_of_cubes 2) )
```

#### Σύνταξη ορισμού ενός κανόνα

Για να πραγματοποιηθεί ο ορισμός ενός κανόνα χρησιμοποιείται η συνάρτηση **defrule** της οποίας η σύνταξη απαιτεί να περιέχεται μέσα σε παρενθέσεις, μετά το `defrule` να υπάρχει το

όνομα του κανόνα και να ακολουθούν ένα ή περισσότερα γεγονότα και η εντολή test. Με το τέλος των γεγονότων πρέπει να παρουσιάζεται ο συνδυασμός συμβόλων -> (2 σύμβολα η παύλα και το > ) και να ακολουθεί μια ή περισσότερες printout εντολές , π.χ.

**(defrule** move-up

(cube A on ?something)

(cube B on A)

(number\_of\_cubes ?num)

(test (= ?num 2))

->

(printout t (?something " is under A") ("A is under B") ("there are 2 cubes")) )

### **Αριθμητικές πράξεις**

Μια πρωταρχική συνάρτηση υπολογισμού αριθμητικής πράξης συντάσσεται με την αναγραφή του ονόματος της συνάρτησης (+, -, \*, /) ακολουθούμενου από δυο ή περισσότερους τελεστέους που μπορεί να είναι ή μεταβλητές ή ακέραιοι αριθμοί,

π.χ.: (+ 1 sum), (- +3 -20 ?x1) (\* ?id1 ?id2), (/ 10 -2 1)

### **Σύγκριση**

Μια σύγκριση αφορά μόνο ισότητες μεταξύ ακεραίων και μεταβλητών ή ακεραίων και αριθμητικών πράξεων ή μεταξύ μεταβλητών και αριθμητικών πράξεων,

π.χ.: (= ?num 2) (= 100 (+ ?num 2)) (= (+ ?num 2) ?x)

### **Έλεγχος τιμών με την test**

Η test περιέχεται μεταξύ παρενθέσεων και μετά την εμφάνισή της πρέπει να ακολουθεί μια σύγκριση,

π.χ. (test (= ?num 2))

### **Ανάθεση τιμής σε μεταβλητή με την bind**

Η bind περιέχεται μεταξύ παρενθέσεων και μετά την εμφάνισή της πρέπει να ακολουθεί το όνομα μιας μεταβλητής και ένα από τα ακόλουθα:

α) η σύνταξη (read) που υπονοεί ότι η μεταβλητή θα δεχθεί τη τιμή που θα δοθεί από το input

β) μια αριθμητική πράξη

π.χ. (bind ?x (read))

(bind ?var 15)

(bind ?x (+ 4 5))

## Πλήρες παράδειγμα κώδικα της Uni-CLIPS

Το παρακάτω παράδειγμα παρουσιάζει ένα μέρος του Uni-CLIPS κώδικα για το πρόβλημα του rac-man. Το rac-man βρίσκεται μέσα σε ένα πλέγμα και μπορεί να κινείται προς τις τέσσερις διαφορετικές κατευθύνσεις, πάνω, κάτω, αριστερά και δεξιά. Στο παράδειγμά μας, δίνεται η θέση του χαρακτήρα και οι θέσεις όπου υπάρχει τροφή. Ο κανόνας move-up περιγράφει την κίνηση του rac-man προς τα επάνω στην περίπτωση που συγκεκριμένη θέση υπάρχει τροφή.

|   |   |
|---|---|
| <i>;;ορισμός των γεγονότων του προβλήματος</i><br><br><b>(defacts static-facts</b><br><i>;;; food declarations</i><br><b>(food-at 4 2)</b><br><b>(food-at 5 2)</b><br><b>)</b><br><br><b>(defacts dynamic-facts</b><br><i>;;;pacman declaration</i><br><b>(pacman-at 6 2)</b><br><b>)</b> | <i>;;κανόνας όπου ο χαρακτήρας κινείται θόρεια</i><br><i>;;μόνο αν υπάρχει τροφή εκεί</i><br><br><b>(defrule move-UP</b><br><b>(pacman-at ?x ?y)</b><br><b>(food-at ?z ?y)</b><br><b>(test (= ?z (- ?x 1))</b><br><br><b>-&gt;</b><br><i>;;; prints just a message</i><br><b>(printout t ("pacman has reached food"))</b><br><b>)</b> |
|---|---|