

## 6. Γενετική Δύναμη των Γραμματικών Φορμαλισμών

Οι γραμματικοί φορμαλισμοί μπορούν να κατηγοριοποιηθούν σύμφωνα με την γενετική (παραγωγική) τους δύναμη (generative capacity). Ο Chomsky (το 1957) περιγράφει τέσσερεις κλάσεις γραμματικών φορμαλισμών, που διαφέρουν μόνο ως προς τη μορφή των κανόνων τους. Οι κλάσεις αυτές μπορούν να ταξινομηθούν σε μια ιεραρχία, όπου κάθε κλαση μπορεί να χρησιμοποιηθεί για να περιγράψει όλες τις γλώσσες που περιγράφονται από λιγότερο ισχυρή κλάση και μερικές επιπλέον γλώσσες. Η ιεραρχία αυτή είναι:

	κλάση	δείγμα κανόνα	δείγμα γλώσσας
1	αναδρομικά αριθμήσιμη	A B --> C	όποια
2	συμφραστικώς εξηρτημένη (context sensitive)	A B --> B A	a <sup>n</sup> b <sup>n</sup> c <sup>n</sup>
3	συμφραστικώς ανεξάρτητη (context free)	S --> a S b	a <sup>n</sup> b <sup>n</sup>
4	κανονική (regular)	S --> a S	a* b*

### Γραμματικές Ορισμένης Πρότασης (Definite Clause)

Ψάχνουμε να βρούμε τον πιο κατάλληλο φορμαλισμό για την επικοινωνία ανθρώπου - μηχανής (ή και μηχανής - μηχανής) μέσω γλώσσας (φυσικής ή τυπικής). Ας χρησιμοποιήσουμε τον φορμαλισμό BNF. Δύο είναι τα προβλήματα που συναντάμε:

Ο φορμαλισμός αυτός χειρίζεται μόνο συμβολοακολουθίες (strings) και δεν επιτρέπει τον συνδυασμό τους με κάποια “σημασία” (meaning). Από την άλλη, θέλουμε να περιγράψουμε γραμματικές με συμφραζόμενα (context sensitive) και η BNF είναι αυστηρά συμφραστικώς ανεξάρτητη (context free). Γιαυτό, επιστρατεύτηκε η λογική πρώτης τάξης για να μπορούμε να αναφερόμαστε στις συμβολοακολουθίες μαζί με τα νοήματα (που εμείς δίνουμε σ' αυτές).

BNF	Λογική Πρώτης Τάξης
S --> NP VP	NP (s1) Λ VP (s2) ==> S(append(s1, s2))
...	
Det --> ‘η’	Det (‘η’)
Noun --> ‘αρκούδα’   ‘παιδί’   ...	(s = ‘αρκούδα’ V ‘παιδί’ ...) ==> Noun (s)

Ο πρώτος κανόνας λέει ότι αν υπάρχει συμβολοακολουθία s1 που είναι ονοματική φράση και μια άλλη s2 που είναι ρηματική, τότε, η συμβολοακολουθία που δημιουργείται από την συμπαράθεση (append) της μιας στην άλλη, αποτελεί πρόταση. Ο δεύτερος κανόνας λέει ότι αν η συμβολοακολουθία s είναι μία από τις λέξεις ‘αρκούδα’, ‘παιδί’ ... τότε η συμβολοακολουθία s είναι όνομα. Η γραμματική που είναι γραμμένη με λογικές προτάσεις ονομάζεται **λογική γραμματική**. Επειδή είναι υπολογιστικά δαπανηρή η απεριόριστη συνεπαγωγή (inference) επιστρατεύεται μια πιο περιορισμένη μορφή. Η πιο κοινή είναι η **γραμματική ορισμένης πρότασης** (definite clause grammar : DCG). Είναι μια μορφή προτάσεων Horn με ένα ή περισσότερα άτομα στην συνεπάγουσα και ένα μόνο στην συνεπαγόμενη:

$$A_1 \Lambda A_2 \Lambda A_3 \Lambda \dots ==> C_1$$

Το πλεονέκτημα είναι ότι γράφουμε κάτι σ' αυτήν και το αντιλαμβανόμαστε καλύτερα (λογική πρώτης τάξης). Το μειονέκτημα ότι είναι “φλύαρη”.

Η μετάφραση των τριών κανόνων από BNF σε DCG είναι:

**Μετάφραση Κανόνων από BNF σε Γραμματική Ορισμένης Πρότασης (DCG).**

**BNF**

X --> Y Z

X --> 'λέξη'

X --> Y | Z | ...

**DCG**

Y (s1) Λ Z (s2) ==> X(append(s1, s2)).

X (['λέξη']).

Y'(s) V Z'(s) V ... ==> X(s)

(όπου Y' είναι η μετάφραση της έκφρασης Y της DCG σε λογική έκφραση.)

Θα δούμε τώρα πώς μπορούμε να **επεκτείνουμε (augment)** τον συμβολισμό μας για να συμπεριλάβουμε και γραμματικές που δεν μπορούν να παρασταθούν με BNF:

- Μη **τερματικά σύμβολα** μπορούν να “επεκταθούν” (augment) με επιπλέον παραμέτρους (ορίσματα). Ενώ στην BNF διαθέτουμε μία μόνο παράμετρο, την συμβολοακολουθία (π.χ. NP(s) ), στον εκτεταμένο συμβολισμό DCG γράφουμε NP(sem) υπονοώντας λογουχάρη ότι “η σημασία της Ονοματικής Φράσης NP είναι sem“. Αυτό αποδίδεται στη λογική ως κατηγόρημα με δύο παραμέτρους: NP(sem, s)
- Μια **μεταβλητή** μπορεί να βρεθεί στο δεξιό μέλος ενός κανόνα της DCG. Αυτή η μεταβλητή παριστάνει ένα αντικείμενο μέσα στην εισαγόμενη συμβολοακολουθία χωρίς να λέμε τι είναι ακριβώς. Έτσι μπορούμε να τη χρησιμοποιήσουμε για να ορίσουμε την εμφάνιση κάποιου αντικειμένου (π.χ. λέξης) δύο φορές.

**BNF**

Double --> w w

**DCG**

(s1 = [w] Λ s2 = [w]) ==> Double ( Append (s1,s2) )

- Στη γραμματική DCG ο οποιοσδήποτε **λογικός έλεγχος** μπορεί να διατυπωθεί. Αυτός μπαίνει σε άγκιστρα μέσα στην DCG, ως εντολή της Prolog.

**Ασκηση:**

Να γραφεί σε DCG η γραμματική που περιγράφει τους ακεραίους διψήφιους αποτιμώντας την αριθμητική αξία τους:

*DCG:*

*Digit(sem) --> sem { 'sem' μεταξύ 0 και 9 }*

*Number(sem) --> Digit(sem)*

*Number(sem) --> (Number(sem1) Digit(sem2) { sem = 10 sem1 + sem2 }.*

## Εφαρμογή στην Γενετική Δύναμη των Αυξημένων Γραμματικών

### ΓΡΑΜΜΑΤΙΚΕΣ σε φορμαλισμό PATR II

```

; 1. context free grammar
;
; a^n . b^n
; sentence grammar for PC-PATR
; { } to include optional(s)
; / to indicate disjunction
; ( ) to indicate inseparable structure
; S_1, S_2 ... to differentiate multiple occurrences of the same ; variable in a rule

; sole
RULE
; syntactic skeleton without feature structures (augmentation)
S_1 -> ( Aterm { S_2 } Bterm )
; to instantiate equal number of a's, b's .
END

; 2. context sensitive grammar
;
; a^n. b^n. c^n
; sentence grammar for PC-PATR

RULE
S -> A B C
<A count> = <B count>
<A count> = <C count>
; same number of a, b and c.

RULE
A_1 -> Aterm A_2
<A_1 count count> = <A_2 count>
;

RULE
B_1 -> Bterm B_2
<B_1 count count> = <B_2 count>
;

RULE
C_1 -> Cterm C_2
<C_1 count count> = <C_2 count>
;

RULE
A -> Aterm
<A count> = end
;

RULE
B -> Bterm
<B count> = end
;
```

```
;  
RULE  
C -> Cterm  
<C count> = end  
;  
END
```

### **; 1. context free grammar lexicon**

```
;  
\w a  
\c Aterm  
  
\w b  
\c Bterm
```

### **; 1. context sensitive grammar lexicon**

```
;  
\w a  
\c Aterm  
  
\w b  
\c Bterm  
  
\w c  
\c Cterm
```