



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

UNIVERSITY OF PIRAEUS

ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΩΝ  
ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΩΝ  
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

**Επεξεργασία Φυσικής Γλώσσας**  
Απαλλακτική Εργασία

Όνομα: Γεώργιος Άγγελος Νικητάκης

Αριθμός μητρώου: Π19120

[Τηλ.επικοινωνίας:6984241364]

[[email:nikitasgio21@gmail.com](mailto:nikitasgio21@gmail.com)]

**B.**

**B1) Λύση σε PROLOG**

**ΘΕΜΑ 1**

**(α)**

s --> np, vp.

np --> det, noun.

pp --> prep,np.

vp --> verb, np, pp.

det --> [the].

verb --> [brought].

noun --> [waiter].

noun --> [meal].

prep --> [to].

noun --> [table].

**Και κάνετε την ερώτηση:**

s([the,waiter,brought,the,meal,to,the,table],[]).

**(β)**

s(s(NP,VP)) --> np(NP),vp(VP).

np(np(D,N)) --> det(D),noun(N).

pp(pp(PREP,NP) --> prep(PREP),np(NP).

vp(vp(V,NP,PP) --> verb(V),np(NP),pp(PP).

det(det(the)) --> [the].

noun(noun(waiter)) --> [waiter].

verb(verb(brought)) --> [brought].

noun(noun(meal)) --> [meal].

**Και κάνετε την ερώτηση:**

s(S,[the,waiter,brought,the,meal,to,the,table],[]).

## ΘΕΜΑ 2

expression(Value) --> number(Value).  
expression(Value) --> number(X), [+], expression(V), {Value is X+V}.  
expression(Value) --> number(X), [-], expression(V), {Value is X-V}.  
expression(Value) --> number(X), [\*], expression(V), {Value is X\*V}.  
expression(Value) --> number(X), [/], expression(V), {V\=0, Value is X/V}.  
expression(Value) --> left\_parenthesis, expression(Value), right\_parenthesis.  
left\_parenthesis --> ['('].  
right\_parenthesis --> [')'].

number(X) --> digit(X).  
number(Value) --> digit(X), number(Y), {numberofdigits(Y,N), Value is X\*2^N+Y}.  
digit(0) --> [0].  
digit(1) --> [1].

numberofdigits(Y,1) :- Z is Y/2, Z<1, !.  
numberofdigits(Y,N) :-  
    Z is (Y - mod(Y,2))/2,  
    numberofdigits(Z,N1),  
    N is N1+1, !.

## ΘΕΜΑ 3

Χρησιμοποιώντας τον κώδικα που μου έχει δωθεί και κάνοντας τις κατάλληλες αλλαγές μπορεί πλέον να «κατανοήσει» μια ιστορία. Η ιστορία που χρησιμοποιώ εγώ στην εργασία μου είναι η παρακάτω:  
(Η παρακάτω ιστορία είναι εμπνευσμένη απο τις ταινίες Star wars).

```
'anakin is jedi. anakin has a lightsaber. anakin hates obiwan. anakin loves padme. anakin is scary.  
obiwan is wise. obiwan is jedi. obiwan has a lightsaber. obiwan is blonde. obiwan loves anakin.  
padme is senator. padme is pretty. padme loves anakin.  
obiwan chased anakin. obiwan fought anakin. obiwan defeated anakin.'
```

### Εκτέλεση προγράμματος

Αρχικά γράφουμε στην prolog το «understand('sw.txt').» (το αρχείο sw.txt είναι το αρχείο όπου έχω την ιστορία που έδειξα παραπάνω). Με το οποίο η prolog κατανοεί την ιστορία κάνοντας με την σειρά την λεκτική, συντακτική και σημασιολογική ανάλυση και ύστερα ανανεώνει την βάση γνώσης.

Προχωρίστε παρακάτω για να δείτε screenshot απο την εκτέλεση του understand('sw.txt').

```

?- understand('sw.txt').
[anakin,is,jedi]
[anakin,has,a,lightsaber]
[anakin,hates,obiwan]
[anakin,loves,padme]
[anakin,is,scary]
[obiwan,is,wise]
[obiwan,is,jedi]
[obiwan,has,a,lightsaber]
[obiwan,is,blonde]
[obiwan,loves,anakin]
[padme,is,senator]
[padme,is,pretty]
[padme,loves,anakin]
[obiwan,chased,anakin]
[obiwan,fought,anakin]
[obiwan,defeated,anakin]

the lexical analysis completed!

```

```

s(np(pn(anakin)),vp(av(is),adj(jedi)))
s(np(pn(anakin)),vp(v(has),np(det(a),n(lightsaber))))
s(np(pn(anakin)),vp(v(hates),np(pn(obiwan))))
s(np(pn(anakin)),vp(v(loves),np(pn(padme))))
s(np(pn(anakin)),vp(av(is),adj(scary)))
s(np(pn(obiwan)),vp(av(is),adj(wise)))
s(np(pn(obiwan)),vp(av(is),adj(jedi)))
s(np(pn(obiwan)),vp(v(has),np(det(a),n(lightsaber))))
s(np(pn(obiwan)),vp(av(is),adj(blonde)))
s(np(pn(obiwan)),vp(v(loves),np(pn(anakin))))
s(np(pn(padme)),vp(av(is),adj(senator)))
s(np(pn(padme)),vp(av(is),adj(pretty)))
s(np(pn(padme)),vp(v(loves),np(pn(anakin))))
s(np(pn(obiwan)),vp(v(chased),np(pn(anakin))))
s(np(pn(obiwan)),vp(v(fought),np(pn(anakin))))
s(np(pn(obiwan)),vp(v(defeated),np(pn(anakin))))

the syntactic analysis completed!

```

```

jedi(anakin)
has(anakin,lightsaber)
hates(anakin,obiwan)
loves(anakin,padme)
scary(anakin)
wise(obiwan)
jedi(obiwan)
has(obiwan,lightsaber)
blonde(obiwan)
loves(obiwan,anakin)
senator(padme)
pretty(padme)
loves(padme,anakin)
chased(obiwan,anakin)
fought(obiwan,anakin)
defeated(obiwan,anakin)

the semantic analysis completed!

```

```

kb_fact(jedi(anakin)) asserted
kb_fact(has(anakin,lightsaber)) asserted
kb_fact(hates(anakin,obiwan)) asserted
kb_fact(loves(anakin,padme)) asserted
kb_fact(scary(anakin)) asserted
kb_fact(wise(obiwan)) asserted
kb_fact(jedi(obiwan)) asserted
kb_fact(has(obiwan,lightsaber)) asserted
kb_fact(blonde(obiwan)) asserted
kb_fact(loves(obiwan,anakin)) asserted
kb_fact(senator(padme)) asserted
kb_fact(pretty(padme)) asserted
kb_fact(loves(padme,anakin)) asserted
kb_fact(chased(obiwan,anakin)) asserted
kb_fact(fought(obiwan,anakin)) asserted
kb_fact(defeated(obiwan,anakin)) asserted

Knowledge Base Updated!

```

```

true.

?-

```

Τώρα που έχουμε ανανεώσει την βάση γνώσης μπορούμε να κάνουμε ερωτήσεις με το **ask**, Για παράδειγμα:

```
?- ask([is, anakin, jedi]).  
Yes.  
true.  
  
?- ■
```

Εδώ ρωτάμε αν ο anakin είναι jedi και η prolog βλέπει απο την βάση γνώσης που έχει δηλώσει απο την ιστορία μας οτι ο anakin είναι jedi οπότε επιστρέφει **Yes**.

Ακόμη ενα παράδειγμα:

```
?- ask([who, defeated, anakin]).  
obiwan  
true.  
  
?-
```

Εδώ κάνουμε την ερώτηση ποιός κέρδισε τον anakin το οποίο γνωρίζει η prolog σύμφωνα με την ιστορία μας οτι είναι ο obiwan.

Μερικά ακομα παραδείγματα:

```
?- ask([is, anakin, wise]).  
No.  
true.  
  
?- ■
```

Σύμφωνα με την ιστορία μας ο anakin δεν είναι σοφός οπότε η απάντηση στην ερώτηση είναι ο anakin σοφός? Επιστρέφει **No**.

```
?- ask([who, is, pretty]).  
padme  
true.  
  
?- ask([is, padme, jedi]).  
No.  
true.  
  
?-  
  
?- ask([does, obiwan, love, anakin]).  
Yes.  
true.  
  
?- ■
```

Τώρα θα αλλάξουμε την βάση γνώσης κατευθείαν απο το πληκτρολόγιο χωρίς να φτιάχνουμε καινούργια ιστορία και να την κάνουμε understand. Θα χρησιμοποιήσουμε το **tell**.

Όπως είδαμε στα προηγούμενα παραδείγματα ο Anakin δεν είναι σοφός όμως κάνοντας το `tell([Anakin,is,wise])`. Ανανεώνουμε την βάση γνώσης και τώρα αμα κάνουμε την ερώτηση `ask([is,Anakin,wise])`. Παίρνουμε πλέον την απάντηση Yes.

```
?- tell([anakin,is,wise]).
```

```
kb_fact(wise(anakin))  
  added to knowledge base.  
true.
```

```
?- ask([is,anakin,wise]).  
Yes.  
true.
```

```
?- ■
```

Μερικά ακόμα παραδείγματα:

```
?- tell([padme,has,a,lightsaber]).
```

```
kb_fact(has(padme,lightsaber))  
  added to knowledge base.  
true.
```

Κάνουμε την ερώτηση αν ο anakin αγαπάει τον obiwan πριν και μετά απο την tell και βλέπουμε οτι αλλάζει η βάση γνώσης και η prolog απαντάει ανάλογα.

```
?- ask([does,anakin,love,obiwan]).  
No.  
true.
```

```
?- tell([anakin,loves,obiwan]).
```

```
kb_fact(loves(anakin,obiwan))  
  added to knowledge base.  
true.
```

```
?- ask([does,anakin,love,obiwan]).  
Yes.  
true.
```

```
?- ■
```

