

Υλοποίηση ΛΑ με το *flex* (i)

- ▶ **Μεταεργαλείο *flex***: γεννήτορας ΛΑ
- ▶ **Είσοδος**: μεταπρόγραμμα που περιγράφει τις λεκτικές μονάδες
- ▶ **Έξοδος**: πρόγραμμα σε C
 - ▶ Η συνάρτηση ***yylex*** υλοποιεί το ΛΑ
 - ▶ Επιστρέφει τον **κωδικό** της λεκτικής μονάδας που αναγνωρίστηκε, ή **0** στο τέλος της συμβολοσειράς εισόδου
 - ▶ Τοποθετεί στη μεταβλητή ***yytext*** την αντίστοιχη ακολουθία χαρακτήρων (lexeme)

Υλοποίηση ΛΑ με το flex (ii)

► Δομή του μεταπρογράμματος

Μέρος Α

%%

Μέρος Β

%%

Μέρος Γ

Και τα τρία μέρη μπορούν να είναι κενά

Υλοποίηση ΛΑ με το flex (iii)

- ▶ **Μέρος A**, περιέχει
 - ▶ **Σχόλια**, όπως στη C
 - ▶ **Κώδικα C**, μέσα σε `%{` και `%}`
 - ▶ **Μνημονικά ονόματα** ως συντομογραφίες κανονικών εκφράσεων
 - ▶ Δηλώσεις **αρχικών καταστάσεων**

Υλοποίηση ΛΑ με το flex (iv)

► **Μέρος A**, παράδειγμα

```
%{  
#define T_eof          0  
#define T_id          1  
    ...  
#define T_while       52  
  
void ERROR (const char msg []);  
%}  
  
L [A-Za-z]           /* letters      */  
D [0-9]              /* digits      */  
W [ \t\n]            /* white space */
```

Υλοποίηση ΛΑ με το flex (v)

- ▶ **Μέρος B**, περιέχει κανόνες της μορφής
κανονική έκφραση ενέργεια
- ▶ Κάθε ενέργεια είναι μια εντολή της C
- ▶ **Λειτουργία:**
 - ▶ Διαβάζεται το **μακρύτερο πρόθεμα** της συμβολοσειράς εισόδου που μπορεί να αναγνωριστεί από κάποια κανονική έκφραση
 - ▶ Εκτελείται η αντίστοιχη ενέργεια

Υλοποίηση ΛΑ με το flex (vi)

► Κανονικές εκφράσεις

- a Ο χαρακτήρας a.
- . Οποιοσδήποτε χαρακτήρας εκτός της αλλαγής γραμμής.
- \x Αν x ένα από τα a, b, f, n, r, t, v ή θ , τότε όπως στη C, αλλιώς ο ίδιος ο χαρακτήρας x .
- \123 Ο χαρακτήρας ASCII με οκταδική τιμή 123.
- \x3f Ο χαρακτήρας ASCII με δεκαεξαδική τιμή 3F.
- "abc" Η συμβολοσειρά abc.
- [abc] Ένας από τους χαρακτήρες a, b ή c.
- [a-z] Ένας από τους χαρακτήρες a έως z.
- [ac-fs] Ένας από τους χαρακτήρες a, c έως f, ή s.
- [^a-z] Ένας από τους χαρακτήρες εκτός όσων ανήκουν στην περιοχή a έως z.

Υλοποίηση ΛΑ με το flex (vii)

► Κανονικές εκφράσεις (συνέχεια)

$\{name\}$ Η κανονική έκφραση με μνημονικό όνομα name.

rs Η παράθεση των r και s .

$r|s$ Η διάζευξη των r και s .

(r) Η κανονική έκφραση r . Οι παρενθέσεις χρησιμοποιούνται για ομαδοποίηση.

r^* Η r επαναλαμβάνεται μηδέν ή περισσότερες φορές.

r^+ Η r επαναλαμβάνεται μια ή περισσότερες φορές.

$r?$ Η r είναι προαιρετική (επαναλαμβάνεται μηδέν ή μια φορά).

$r\{7\}$ Η r επαναλαμβάνεται ακριβώς 7 φορές.

$r\{3,5\}$ Η r επαναλαμβάνεται από 3 έως 5 φορές.

$r\{4,\}$ Η r επαναλαμβάνεται 4 ή περισσότερες φορές.

Υλοποίηση ΛΑ με το flex (viii)

► Κανονικές εκφράσεις (συνέχεια)

$\wedge r$

Η r αλλά μόνο στην αρχή μιας γραμμής.

$r\$$

Η r αλλά μόνο στο τέλος μιας γραμμής.

$\langle\langle\text{EOF}\rangle\rangle$

Το τέλος του αρχείου εισόδου.

r/s

Η κανονική έκφραση r αλλά μόνο αν ακολουθεί η κανονική έκφραση s .

$\langle S \rangle r$

Η r αλλά μόνο όταν η τρέχουσα αρχική κατάσταση είναι η S .

$\langle S_1, S_2, S_3 \rangle r$

Η r , αλλά μόνο όταν η τρέχουσα αρχική κατάσταση είναι μια από τις S_1 , S_2 ή S_3 .

$\langle * \rangle r$

Η r σε οποιαδήποτε αρχική κατάσταση.

Υλοποίηση ΛΑ με το flex (ix)

► Μέρος Β, παράδειγμα

```
"and"           { return T_and;   }
    ...
"while"         { return T_while; }

":="           { return T_assign; }
":"            { return T_colon;  }

{L}({L}|{D}|_)* { return T_id;    }
{D}+(\.{D}* (e\-?{D}+)?)? { return T_const; }
{W}+           { /* nothing */   }
"(*"([^\*]+|\*+[\^\*]))*\*+" { /* nothing */   }

.               { ERROR("illegal token"); }
```

Υλοποίηση ΛΑ με το flex (x)

- ▶ **Μέρος Γ**, περιέχει κώδικα C
- ▶ Παράδειγμα

```
void ERROR (const char msg [])  
{  
    fprintf(stderr, "ERROR: %s\n", msg);  
    exit(1);  
}
```

Υλοποίηση ΛΑ με το flex (xi)

- ▶ Παράδειγμα (συνέχεια)

```
int main ()
{
    int token;

    do {
        token = yylex();
        printf("token=%d, lexeme=\"%s\"\n",
            token, yytext);
    } while (token != T_eof);

    return 0;
}
```

Υλοποίηση ΛΑ με το flex (xii)

- ▶ Παράδειγμα: Αρίθμηση γραμμών

```
int lineno = 1;
```

```
[ \t]+    { /* nothing */ }  
\n        { lineno++;    }
```

```
void ERROR (const char msg [])  
{  
    fprintf(stderr, "ERROR, line %d: %s\n",  
                lineno, msg);  
    exit(1);  
}
```

- ▶ Πρόβλημα: Λάθος αρίθμηση σε σχόλια

Υλοποίηση ΛΑ με το flex (xiii)

- ▶ Αρχικές καταστάσεις
 - ▶ Κοινές: %s
 - ▶ Αποκλειστικές: %x
- ▶ Ενεργοί κανόνες σε κάποια κατάσταση
- ▶ Μετάβαση μεταξύ καταστάσεων: **BEGIN(s)**
- ▶ Αρχική κατάσταση κατά την έναρξη λειτουργίας του ΛΑ:
INITIAL

Υλοποίηση ΛΑ με το flex (xiv)

- ▶ Παράδειγμα: Αρίθμηση γραμμών (διόρθωση)

```
%x COMMENT
```

```
"(*"           { BEGIN(COMMENT); }  
<COMMENT>"*)" { BEGIN(INITIAL); }  
<COMMENT>\n      { lineno++;           }  
<COMMENT>"**"  { /* nothing */          }  
<COMMENT>[^*\n]+ { /* nothing */          }
```