

Clase #21 de 27

AFN, Árbol de Derivación & Lex

Octubre 22, Lunes

Lenguajes Regulares

- Ejercicio #59 de [MUCH2012] V1 p95:
 - Dado el alfabeto de los dígitos decimales, obtenga una metaER que represente al LR *“Todas las palabras que terminan con 22 o con 47 y tienen longitud mayor o igual que 3”*.
- Representar el lenguaje de las siguientes maneras
 - Operaciones de Conjuntos
 - AF
 - Definición formal de un AFN
 - Diferencias entre la función T de un AFN y AFD
 - ER
 - Regex
 - Gramática
 - Definición Formal de la Gramática
 - Árbol de Derivación

Intervalo

20 minutos

Lex

Generador de Analizadores Léxicos

Lex

- Proceso
 - `def.l` (*reglas*) \rightarrow **[Lex]** \rightarrow `lex.yy.c`
 - `lex.yy.c` \rightarrow **[Compilador C]** \rightarrow `lex.yy.exe`
 - Input \rightarrow **[yy.lex.exe]** \rightarrow output
- Lex versus Flex
- Instalación de Flex
 - <https://josemariasola.wordpress.com/ssl/reference/>

Secciones de un programa Lex

- General

- ... definiciones...

- %%

- ... reglas...

- %%

- ... subrutinas...

- Secciones opcionales

- Segundo %%

- Regla por defecto

- Enviar todo lo recibido

- Mínimo Lex

- %%

- Mínimo Flex

- %option main

- %%

RegExen que usa Lex

[Lex - A Lexical Analyzer Generator], [MUCH2012] v1c5p92

- `x` the character "x"
- `"x"` an "x", even if x is an operator.
- `\x` an "x", even if x is an operator.
- `[xy]` the character x or y. [MUCH2012]
- `[x-z]` the characters x, y or z. [MUCH2012]
- `[^x]` any character but x.
- `.` any character but newline. [MUCH2012]
- `^x` an x at the beginning of a line.
- `<y>x` an x when Lex is in start condition y.
- `x$` an x at the end of a line.
- `x?` an optional x. [MUCH2012]
- `x*` 0,1,2, ... instances of x. [MUCH2012]
- `x+` 1,2,3, ... instances of x. [MUCH2012]
- `x|y` an x or a y. [MUCH2012]
- `(x)` an x. [MUCH2012]
- `x/y` an x but only if followed by y.
- `{xx}` the translation of xx from the definitions section.
- `x{n}` n occurrences of x [MUCH2012]
- `x{m,n}` m through n occurrences of x [MUCH2012]

Programa Mínimo Lex:

Copiar entrada a salida; 2da versión [K&R1988] 1.5

```
#include <stdio.h>
```

```
%option main
```

```
%%
```

```
// copy input to output; 2nd version
```

```
int main(void){
```

```
    for (int c; (c = getchar()) != EOF;)
```

```
        putchar(c);
```

```
}
```


Función yylex()

Conteo de caracteres; 2da versión [K&R1988] 1.5.2

```
#include <stdio.h>

// count characters in input; 2nd
// version
int main(void){
    double nc;

    for (nc = 0 ; getchar() != EOF; ++nc)
        ;
    printf("%.0f\n", nc);
}
```

```
%option noyywrap
%{
double nc;
%}
%%
.|\\n  ++nc;
%%

int main(void){
    yylex();
    printf("%.0f\n", nc);
}
```

Conteo de líneas [K&R1988] 1.5.3

```
#include <stdio.h>

// count lines in input
int main(void){
    int c, nl;

    for (nl=0;(c=getchar())!= EOF;)
        if (c == '\n')
            ++nl;
    printf("%d\n", nl);
}
```

```
%option noyywrap
%{
double nl;
%}
%%
```

```
\n    ++nl;
.      ;
%%
```

```
int main(void){
    yylex();
    printf("%.0f\n", nl);
}
```

Longitud Promedio de Líneas

```
#include <stdio.h> // getchar EOF printf
```

```
int main(void) {  
    int nl=0, nc=0; // la cantidad de líneas y de caracteres  
  
    for(int c; (c = getchar()) != EOF; ++nc)  
        if(c == '\n')  
            ++nl;  
  
    printf("%f\n", nl ? (nc - nl) / (double)nl : 0);  
}
```

```
%option noyywrap
```

```
{  
    int nl, nc;  
}
```

```
%%
```

```
\n++nl;
```

```
. ++nc;
```

```
%%
```

```
int main(void){
```

```
    yylex();
```

```
    printf("%.1f\n", nl?nc/(double)nl:0);
```

```
}
```

Conteo de números y palabras

Ejemplo 53 [MUCH2012] V1, págs 96-97

```
%option noyywrap
```

```
%{
```

```
#include <stdio.h>
```

```
int nnums, nwords;
```

```
%}
```

```
%%
```

```
[0-9]+      nnums++;
```

```
[a-zA-Z]+  nwords++;
```

```
|\n        ;
```

```
%%
```

```
int main(void) {
```

```
    yylex();
```

```
    printf("Números: %d\nPalabras: %d\n", nnums, nwords);
```

```
}
```

Definiciones Regulares

Basado en ejemplo 53 [MUCH2012] V1, págs 96-97

```
%option noyywrap
%{
#include <stdio.h>
int nnums, nwords;
%}
```

```
NUMBER  [0-9]+
WORD    [a-zA-Z]+
```

```
%%
```

```
{NUMBER} nnums++;
{WORD}   nwords++;
.|\n    ;
```

```
%%
```

```
int main(void) {
    yylex();
    printf("Hay %d números y %d palabras.\n", nnums, nwords);
}
```

Operador ^ y Variable yyleng

Conteo de Palabras [K&R1988] 1.5.4

```
#include <stdio.h>

enum{IN, OUT}; // inside or outside a word

// count lines, words, and characters in input
int main(void){
    int nl, nw, nc, state;

    state = OUT;
    nl = nw = nc = 0;
    for(int c; (c = getchar()) != EOF; ++nc){
        if (c == '\n')
            ++nl;
        if (c == ' ' || c == '\n' || c == '\t')
            state = OUT;
        else if (state == OUT) {
            state = IN;
            ++nw;
        }
    }
    printf("%d %d %d\n", nl, nw, nc);
}
```

```
%option noyywrap
%{
    int nl, nc, nw;
}%
WORD      [^ \t\n]+
%%
{WORD}    ++nw, nc+=yyleng;
\n        ++nl, ++nc;
[ \t]     ++nc;
%%
int main(void){
    yylex();
    printf("%d %d %d\n",
           nl, nw, nc);
}
```

Variable yytext

Conteo de dígitos, blancos y otros [K&R1988] 1.6

```
#include <stdio.h>

// count digits, white space, others
int main(){
    int nwhite, nother;
    int ndigit[10];

    nwhite = nother = 0;
    for(int i = 0; i < 10; ++i)
        ndigit[i] = 0;

    for(int c; (c = getchar()) != EOF;)
        if(c >= '0' && c <= '9')
            ++ndigit[c-'0'];
        else if (c==' ' || c=='\n' || c=='\t')
            ++nwhite;
        else
            ++nother;

    printf("digits =");
    for(int i = 0; i < 10; ++i)
        printf(" %d", ndigit[i]);
    printf(", whitespace=%d, other=%d\n",
        nwhite, nother);
}
```

```
%option noyywrap
%{
    int nwhite, nother, ndigit[10];
}%

%%
    ++ndigit[ *yytext  - '0' ];
[0-9]    ++ndigit[ yytext[0] - '0' ];
[ \n\t]  ++nwhite;
.        ++nother;
%%

int main(void){
    int i;
    yylex();
    printf("digits =");
    for(i = 0; i < 10; ++i)
        printf(" %d", ndigit[i]);
    printf(", whitespace=%d, other=%d\n",
        nwhite, nother);
}
```

¿Consultas?



Fin de la clase