



**INSTITUTO POLITÉCNICO
NACIONAL**



**Práctica 2:
Operaciones entre lenguajes**

Alumno:
González Tetuán Héctor David

Materia:
Teoría de la computación

Grupo:
4CM4

Introducción:

La práctica consiste en mostrar una serie de operaciones que pueden realizarse entre lenguajes como la unión, concatenación, potencia, la cerradura positiva y de Kleene y la reflexión.

Todo esto nos puede generar nuevos lenguajes que surgen a partir de las operaciones mencionadas anteriormente, lenguajes con los cuales también se pueden usar para interactuar con el resto de lenguaje y generar nuevas cadenas.

Código:

```
int main(){

    int rept=1;
    int opt;
    int choose[2];

    Nodo *words[3];
    const char *filename[3] = {"arch1.txt","arch2.txt","arch3.txt"};

    for(int i=0; i<3; i++){
        words[i] = NULL;
    }

    for(int i=0; i<3; i++){
        printf("%s",filename[i]);
        words[i] = getFile(filename[i]);
    }

    while(rept){

        //Operaciones de cadenas
        printf("\n-----Seleccione una de las siguientes operaciones-----\n");
        printf("\n1. Union\n2. Concatenacion\n3. Cerradura de Kleene\n4. Cerradura Positiva\n5. Potencia\n6. Reflexion\n0. Salir\n--> ");

        scanf("%d",&opt);

        switch( opt ){
```

```
case 1:
    printf("----Union----\n");

    printf("Seleccione dos lenguajes:\n");

    scanf("%d %d",&choose[0],&choose[1]);
    join( words[choose[0]-1],words[choose[1]-1] );

    break;

case 2:
    printf("----Concatenacion----\n");

    printf("Seleccione dos lenguajes:\n");
    scanf("%d %d",&choose[0],&choose[1]);

    printf("Concatenacion de los lenguajes %d y %d:\n",choose[0],choose[1]);

    concatenateShow( words[choose[0]-1],words[choose[1]-1] );

    break;

case 3:
    printf("----Cerradura de Kleene----\n");

    printf("Seleccione un lenguaje:\n");
    scanf("%d",&choose[0]);

    cerraduraKleene( words[choose[0]-1] );

    break;
```

```
case 4:
    printf("----Cerradura Positiva----\n");

    printf("Seleccione un lenguaje:\n");
    scanf("%d",&choose[0]);

    cerraduraPositiva( words[choose[0]-1] );

    break;

case 5:
    printf("----Potencia----\n");

    printf("Seleccione un lenguaje:\n");
    scanf("%d",&choose[0]);

    printf("Escribir potencia: ");
    scanf("%d", &choose[1]);

    potencia( words[choose[0]-1],choose[1] );

    break;

case 6:
    printf("----Reflexion----\n");

    printf("Seleccione un lenguaje:\n");
    scanf("%d",&choose[0]);

    reflexion( words[choose[0]-1] );

    break;
```

```

    Nodo* getFile(const char* filename) {

        FILE *file = fopen(filename, "r");
        if (!file) {
            printf("Error al abrir el archivo: %s\n", filename);
            exit(1);
        }

        Nodo *aux = NULL;
        char word[100];

        while (fscanf(file, "%s", word) != EOF) {

            Nodo *newNode = (Nodo*)malloc(sizeof(Nodo));
            newNode->c = strdup(word);
            newNode->sig = NULL;

            if(aux == NULL){
                aux = newNode;
            }
            else{
                Nodo *temp = aux;
                while(temp->sig != NULL){
                    temp = temp->sig;
                }
                temp->sig = newNode;
            }

        }

        showCadenas(aux);

        fclose(file);
        return aux;
    }

```

```

void join(Nodo *n1, Nodo *n2){

    Nodo *temp1 = n1;
    Nodo *temp2 = n2;

    // Add unique words from n2 to n1
    while (temp2 != NULL) {
        if (!existsInResult(n1, temp2->c)) {
            Nodo *last = n1;
            while (last->sig != NULL) {
                last = last->sig;
            }
            last->sig = temp2;
            Nodo *next = temp2->sig;
            temp2->sig = NULL;
            temp2 = next;
        } else {
            Nodo *next = temp2->sig;
            temp2 = next;
        }
    }

    showCadenas(n1);
}

```

```

Nodo* concatenate(Nodo *n1, Nodo *n2) {

    Nodo *result = NULL;
    Nodo *temp1 = n1;

    while (temp1 != NULL) {
        Nodo *temp2 = n2;
        while (temp2 != NULL) {

            //printf("%s\n", temp1->c, temp2->c);

            Nodo *newNode = (Nodo*)malloc(sizeof(Nodo));
            char *concatenatedWord = (char*)malloc(strlen(temp1->c) + strlen(temp2->c) + 1);

            strcpy(concatenatedWord, temp1->c);
            strcat(concatenatedWord, temp2->c);

            //Nueva nodo
            newNode->c = concatenatedWord;
            newNode->sig = NULL;

            if (result == NULL) {
                result = newNode;
            }

            else {
                Nodo *aux = result;
                while( aux->sig != NULL ) {
                    aux = aux->sig;
                }
                aux->sig = newNode;
            }

            temp2 = temp2->sig;
        }
        temp1 = temp1->sig;
    }

    return result;
}

```

```

void potencia(Nodo *n, int pot){

    Nodo *aux = n;

    for(int i=0; i<pot-1; i++){
        aux = concatenate(n,aux);
    }

    showCadenas(aux);
}

```

```

void cerraduraKleene(Nodo *n){

    for(int i=1; i<MAX; i++){
        potencia(n,i);
    }
}

```

```

void reflexion(Nodo *n){

    Nodo *current = n;

    while (current != NULL) {
        char *temp = strdup(current->c);
        invertirCadena(temp);

        printf("%s\n", temp);

        free(temp);
        current = current->sig;
    }
}

```

Resultados:

```
-----Seleccione una de las siguientes operaciones-----
1. Union
2. Concatenacion
3. Cerradura de Kleene
4. Cerradura Positiva
5. Potencia
6. Reflexion
0. Salir
-->
```

----Cerradura de Kleene----

Selecciona un lenguaje:

1

ab

cd

abab

abcd

cdab

cdcd

ababab

ababacd

abacdab

abacdcd

abcdabab

abcdabcd

cdcdab

cdcdcd

abababab

abababcd

ababacdab

ababacdcd

abcdabab

abcdabcd

abcdcdab

abcdcdcd

cdababab

cdababcd

cdabacdab

cdabacdcd

cdcdabab

cdcdabcd

cdcdcdab

cdcdcdcd

----Union----

Selecciona dos lenguajes:

1

2

ab

cd

perro

gato

conejo

tortuga

mariposa

raton

leon

----Cerradura Positiva----

Selecciona un lenguaje:

1

0

ab

cd

abab

abcd

cdab

cdcd

ababab

ababcd

abcdab

abcdcd

cdabab

cdabcd

cdcdab

cdcdcd

----Concatenacion----

Selecciona dos lenguajes:

1

1

Concatenacion de los lenguajes 1 y 1:

abab

abcd

cdab

cdcd

----Potencia----

Selecciona un lenguaje:

1

Escribir potencia: 3

ababab

ababcd

abcdab

abcdcd

cdabab

cdabcd

cdcdab

cdcdcd

----Reflexion----

Selecciona un lenguaje:

2

orrep

otag

ojenoc

agutrot

asopiram

notar

noel