



**UNIVERSIDAD TECNOLÓGICA NACIONAL**  
**Facultad Regional Buenos Aires**

*Departamento Ingeniería en Sistemas de Información*  
*SINTAXIS y SEMÁNTICA de los LENGUAJES*

**2021**

**DOCENTE:**  
**ING. ROXANA LEITUZ**

**TRABAJO PRÁCTICO N° 1**  
**«AUTÓMATAS Y CADENAS»**

**CURSO: K2055**  
**ALUMNO: CARRILLO FRANCO NICOLAS**

*"Dada una cadena que contenga varios números que pueden ser decimales, octales o hexadecimales, con o sin signo para el caso de los decimales, separados por el carácter '&', reconocer los tres grupos de constantes enteras, indicando si hubo un error léxico, en caso de ser correcto contar la cantidad de cada grupo."*

## Análisis:

Se deben de poder reconocer los números pertenecientes a la: base 8 (números octales), base 10 (decimales) y base 16 (hexadecimales), para el caso de los números decimales estos pueden o no ser signados.

Teniendo esto en cuenta procedo a definir el formato de los números que pertenezcan a cada base:

Los números de base 8 comenzarán con un 0, y será precedido por los números enteros comprendidos entre 0 y 7 una cantidad indefinida de veces con cualquier combinación entre los mismos.

Los números de base 10 podrán comenzar con un signo y su primer dígito debe de ser uno distinto de 0, además los números que conforman dicha base son los números enteros entre el 0 y el 9, por lo cual reconoce para los decimales signados que comienzan con un signo seguidos de cualquier combinatoria de los elementos de la base siempre y cuando el primer dígito NO sea 0 (cero), mientras que para los enteros NO SIGNADOS, puede tener cualquier combinatoria de los elementos siempre y cuando NO empiece con 0. Teniendo esto en cuenta se debe de aclarar que para la interpretación del 0 decimal, se produce solamente si la cadena leída contiene únicamente al dígito en cuestión, de ser precedido por otro dígito, esta cadena será descartada como un número decimal.

Los números de base 16 deberán comenzar con un 0 seguido de una "x" minúscula (o mayúscula), posteriormente deberá tener cualquier combinatoria de los dígitos comprendidos en el intervalo de números enteros [0; 9] además de sus caracteres correspondientes para representar los números del 10 al 15 con las letras dispuestas entre la "a" y la "f" (propongo que sus mayúsculas sean tratadas de igual manera).

Sea la base que sea, pondré la restricción de que ningún número podrá tener 2 ceros consecutivos como dígitos iniciales, lógicamente excluyendo los ceros utilizados como identificadores como el 0 inicial del octal.

A continuación, se mostrará un cuadro con la interpretación de algunos números importantes (como la interpretación del 0 en cada base) y ejemplos para una mejor aclaración:

<i>Lenguaje Natural</i>	<i>Decimal S/Signo</i>	<i>Decimal Signado</i>	<i>Octal</i>	<i>Hexadecimal</i>
0	0	No tiene	00	0x0 / 0X0
235	235	+235	0235	0x235 / 0X235
-82	No tiene	-82	No tiene	No tiene

## Resolución

Realizadas dichas aclaraciones en la etapa de análisis, conformaré los grupos de caracteres que serán leídos e interpretados:

- Dígitos octales (OCT): {0,1,2,3,4,5,6,7}
- Dígitos decimales (DEC): {0,1,2,3,4,5,6,7,8,9}
- Dígitos hexadecimales (HEX): {0,1,2,3,4,5,6,7,8,9,a,b,c,d,e,f,A,B,C,D,E,F}
- Signo (S): {+,-}
- Carácter especial para interpretar hexadecimales: {x,X}
- Delimitador de cadenas (Delimitador): {&}
- Fin del cadena (fdc): { \0 }.
- Otros: cualquier otro carácter distinto a los descriptos anteriormente.

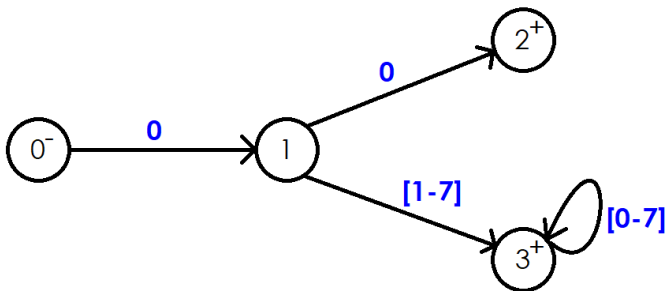
Una vez definidos los grupos de caracteres que han de ser leídos, para poder desarrollar los autómatas encargados de reconocer las diferentes bases, escribiré lo que ya se explicó en lenguaje natural, pero con una expresión regular extendida las cadenas que puede reconocer cada una de las bases:

<i>Bases</i>	<i>Expresión Regular Extendida</i>
Octal (Base 8)	0 (0   [1-7] [0-7]*)
Decimal (Base 10)	0   ([1-9] [0-9]*)   (S [1-9] [0-9]*)
Hexadecimal (Base 16)	0(x   X) (0   ([1-9, a-f, A-F] [0-9, a-f, A-F]*))

Con las expresiones descriptas, se construyen los autómatas para cada base, resultando en los siguientes autómatas:

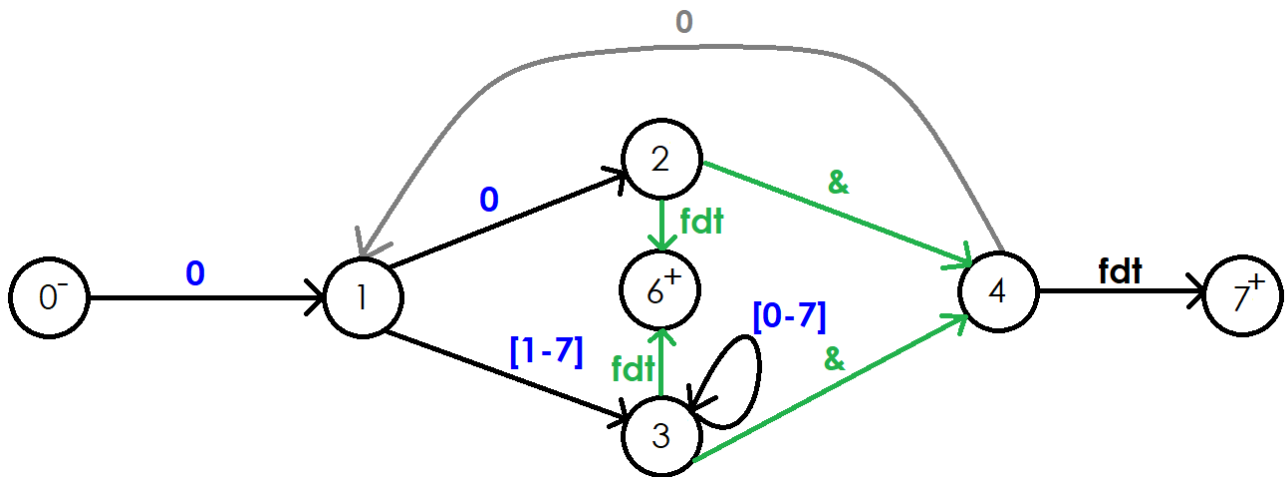
## Octal:

Este autómata reconoce los números octales con las características propuestas, pero para que nos sirva para resolver la situación planteada debemos de agregar algunas transiciones y con sus respectivos estados correspondientes. Para determinar estas nuevas transiciones y estos nuevos estados definiré una tabla que contenga todas las posibles transiciones con una descripción al respecto y cambiando los estados finales:



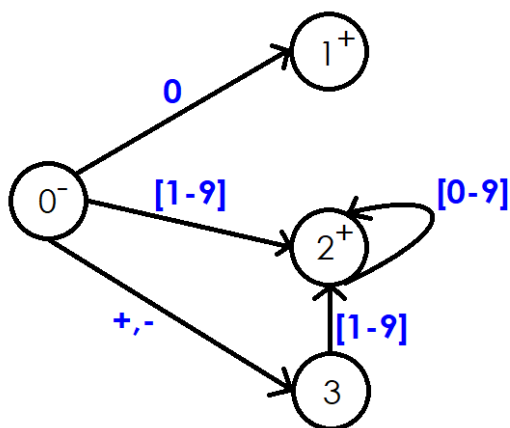
Transiciones			Descripción
Estado de Partida	Caracter Leído	Estado de Llegada	
0-	0	1	Primer carácter leído, indicador de octal, posible N.º Octal.
1	0	2	Primer dígito leído, es el 0, posible N.º Octal.
2	&	4	Delimitador de cadena, ES N.º Octal, hay más cadenas a analizar.
2	fdc	6+	ES N.º Octal, y fin de cadena.
2	[0-9, a-f, A-F], x, X, Signo, Otro	5	No es un N.º Octal, ya que tiene al menos 2 ceros iniciales u otro carácter inapropiado.
1	&	0	NO ES N.º Octal, hay más cadenas para analizar.
1	[1-7]	3	Primer dígito leído, posible N.º Octal.
3	[0-7]	3	Enésimo dígito leído, posible N.º Octal.
3	&	4	Delimitador de cadena, ES N.º Octal, hay más cadenas a analizar.
3	fdc	6+	ES N.º Octal, y fin de cadena.
3	[8-9, a-f, A-F], x, X, Signo, Otro	5	No es un N.º Octal.
1	[8-9, a-f, A-F], x, X, Signo, Otro	5	No es un N.º Octal.
0-	&	0	Cadena Vacía.
0-	fdc	7+	Cadena vacía, fin de cadena.
0-	[1-9, a-f,A-F], x, X, Signo, Otro	5	No es un N.º Octal.
4	0	1	Primer carácter leído, indicador de octal, posible N.º Octal.
4	&	0	Analizar próxima cadena.
4	fdc	7+	Cadena vacía, fin de cadena.
4	[1-9, a-f,A-F], x, X, Signo, Otro	5	No es un N.º Octal.
5	&	0	NO es N.º Decimal, analizar próxima cadena.
5	fdc	7+	Fin de cadena.
5	[0-9, a-f,A-F], x, X, Signo, Otro	5	NO es un N.º Octal.

Teniendo en cuenta la tabla generada podemos diseñar un autómata que reconozca los números octales que nos aproxime un poco más a la resolución del problema, siendo este el siguiente autómata (sin el estado 5 de rechazo ni las transiciones del mismo, a su vez se omitieron las transiciones del delimitador (" & ") dada la cantidad de transiciones que serían y sería un despropósito para la legibilidad del autómata):



Terminado el desarrollo del autómata para los números octales, se realiza el mismo procedimiento para los números decimales.

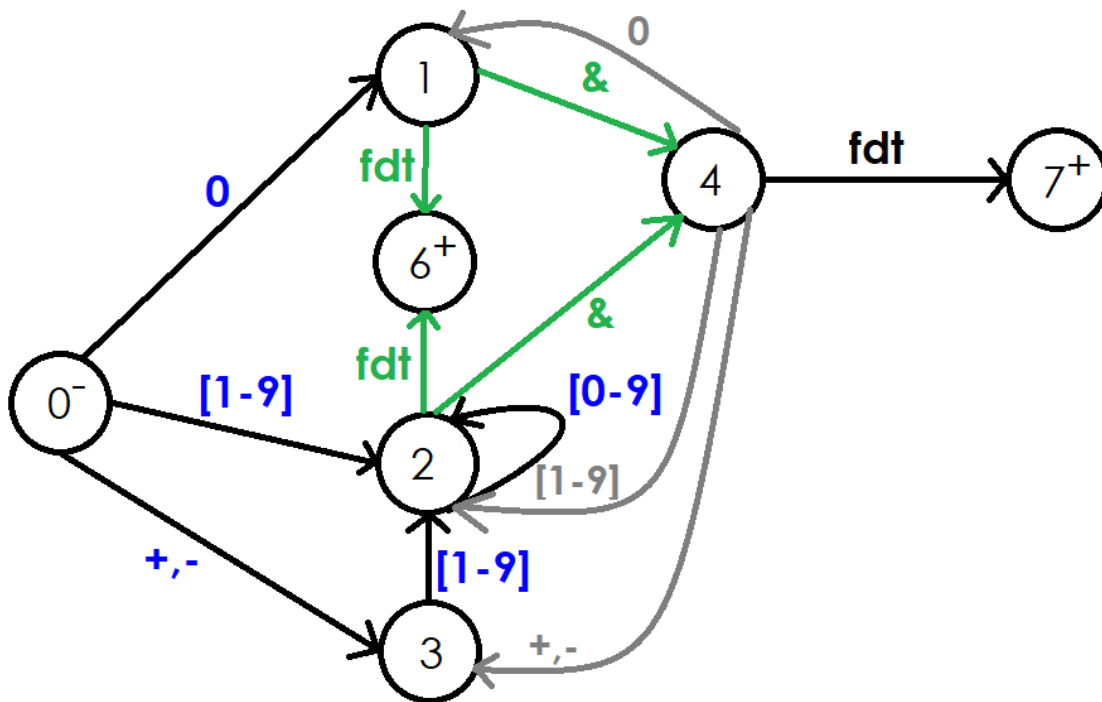
## Decimal:



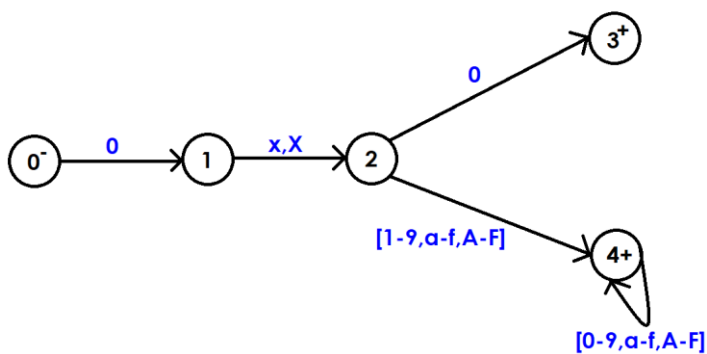
Con el autómata que reconoce números decimales signados o no signados, se genera la tabla con todas las posibles transiciones y una descripción de su razón de ser.

Transiciones			Descripción
Estado de Partida	Caracter Leído	Estado de Llegada	
0-	0	1	Primer dígito leído, posible N.º Decimal.
1	&	4	ES N.º Decimal, hay más para analizar.
1	fdc	6+	ES N.º Decimal, fin de cadena.
1	[0-9, a-f, A-F], x, X, Signo, Otro	5	No es N.º Decimal.
0-	[1-9]	2	Primer dígito leído, posible N.º Decimal.
2	[0-9]	2	Enésimo dígito leído, posible N.º Decimal.
2	&	4	ES N.º Decimal, hay más para analizar.
2	fdc	6+	ES N.º Decimal, fin de cadena.
2	[a-f, A-F], x, X, Signo, Otro	5	No es N.º Decimal.
0-	Signo	3	Signo leído, posible N.º Decimal.
3	[1-9]	2	Primer dígito leído, posible N.º Decimal.
3	&	0	NO es N.º Decimal, analizar siguiente cadena.
3	0, [a-f, A-F], x, X, Signo, Otro	5	No es N.º Decimal.
4	0	1	Primer dígito leído, posible N.º Decimal.
4	[1-9]	2	Primer dígito leído, posible N.º Decimal.
4	Signo	3	Signo leído, posible N.º Decimal.
4	&	0	Analizar próxima cadena.
4	fdc	7+	Cadena vacía, fin de cadena.
4	[a-f, A-F], x, X, Otro	5	No es N.º Decimal.
0-	&	0	Cadena Vacía.
0-	fdc	7+	Cadena vacía, fin de cadena.
0-	[a-f, A-F], x, X, Otro	5	No es N.º Decimal.
5	&	0	No es N.º Decimal, analizar la siguiente cadena.
5	fdc	7+	No es N.º Decimal, fin de cadena.
5	[0-9, a-f, A-F], x, X, Signo, Otro	5	No es N.º Decimal.

Mediante la tabla desarrollada con todas las transiciones para el autómata capaz de reconocer al conjunto de números decimales con y sin signo, desarrollamos un esquema del autómata, (de igual manera que se hizo con el autómata para los números octales, este tampoco posee el estado 5 de rechazo ni las transiciones del mismo, a su vez se omitieron las transiciones del delimitador (" & ") dada la cantidad de transiciones que serían):



## Hexadecimal:

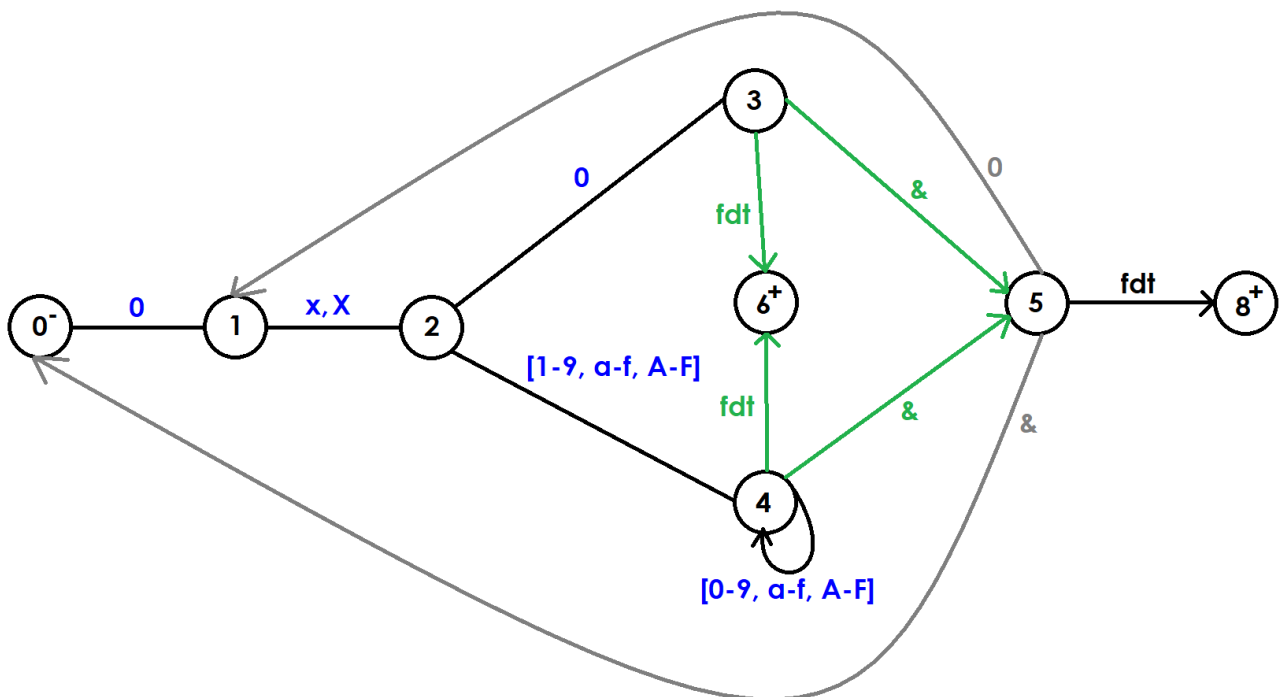


Este autómata reconoce a cualquier número que cumpla las siguientes condiciones:

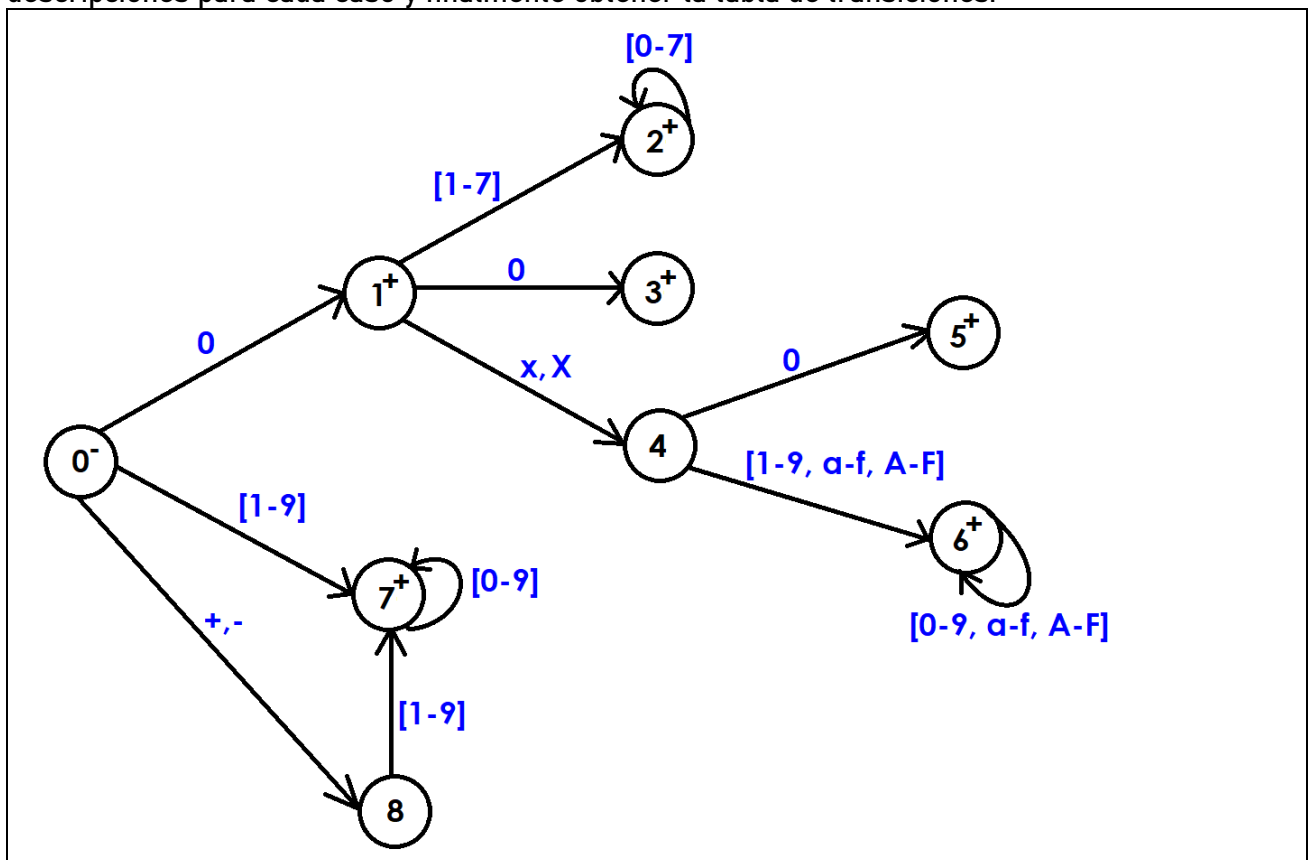
- Empiece con 0x o bien 0X
- El tercer carácter es 0 y es el último dígito (dado que es la representación del 0) o bien es un dígito hexadecimal distinto del 0 seguido de cualquier combinación de los dígitos hexadecimales.

Transiciones			Descripción
Estado de Partida	Caracter Leído	Estado de Llegada	
0-	0	1	Primer caracter leído, posible N.º Hexadecimal.
1	x,X	2	Segundo caracter leído, posible N.º Hexadecimal.
2	0	3	Tercer caracter leído, posible N.º Hexadecimal.
3	&	5	ES N.º Hexadecimal, hay más para leer.
3	fdc	6+	ES N.º Hexadecimal, fin de cadena.
3	[0-9,a-f,A-F], x, X, Signo, Otro	7	No es N.º Hexadecimal.
2	[1-9,a-f,A-F]	4	Posible N.º Hexadecimal.
4	[0-9,a-f,A-F]	4	Posible N.º Hexadecimal.
4	&	5	ES N.º Hexadecimal, hay más para leer.
4	fdc	6+	ES N.º Hexadecimal, fin de cadena.
4	x, X, Signo, Otro	7	No es N.º Hexadecimal.
2	&	0	No es N.º Hexadecimal, hay más para leer.
2	fdc	8+	No es N.º Hexadecimal, fin de cadena.
2	x, X, Signo, Otro	7	No es N.º Hexadecimal.
1	&	0	No es N.º Hexadecimal, hay más para leer.
1	fdc	8+	No es N.º Hexadecimal, fin de cadena.
1	[0-9,a-f,A-F], Signo, Otro	7	No es N.º Hexadecimal.
0-	&	0	Cadena vacía.
0-	fdc	8+	Cadena vacía, fin de cadena.
0-	[1-9,a-f,A-F], x, X, Signo, Otro	7	No es N.º Hexadecimal.
5	0	1	Primer caracter leído de la nueva cadena, posible N.º Hexadecimal.
5	&	0	Leo próxima cadena.
5	fdc	8+	Fin de cadena.
5	[1-9,a-f,A-F], x, X, Signo, Otro	7	No es N.º Hexadecimal.
7	&	0	No es N.º Hexadecimal, analizar próxima cadena.
7	fdc	8+	Fin de cadena.
7	[0-9,a-f,A-F], x, X, Signo, Otro	7	No es N.º Hexadecimal.





Utilizando todos los datos previstos mediante las tablas y los autómatas para cada base, desarrollo un autómata que reconozca cualquiera de las 3 bases para poder obtener una tabla con las descripciones para cada caso y finalmente obtener la tabla de transiciones.



- ❖ Si logra terminar en los estados 7 y 1, será un número decimal.
- ❖ De terminar en los estados 2 y 3, es un número octal.
- ❖ Será un número hexadecimal si finaliza en los estados 5 o 6.

Transiciones			Descripción
Estado de Partida	Caracter Leído	Estado de Llegada	
0-	0	1	Posible N.º Decimal/Octal/Hexadecimal.
1	[1-7]	2	Posible N.º Octal.
2	[0-7]	2	Posible N.º Octal.
2	&	9	ES un N.º Octal, hay más para leer.
2	fdc	10+	ES un N.º Octal, fin de cadena.
2	[8-9,a-f,A-F], x, X, Signo, Otro.	15	No pertenece a ninguna base.
1	0	3	Posible cero Octal.
3	&	9	ES un N.º Octal, hay más para leer.
3	fdc	10+	ES un N.º Octal, fin de cadena.
3	[0-9,a-f,A-F], x, X, Signo, Otro.	15	No pertenece a ninguna base.
1	x, X	4	Posible N.º Hexadecimal.
4	0	5	Posible cero Hexadecimal.
5	&	13	ES un N.º Hexadecimal, hay más para leer.
5	fdc	14+	ES un N.º Hexadecimal, fin de cadena.
5	[0-9,a-f,A-F], x, X, Signo, Otro.	15	No pertenece a ninguna base.
4	[1-9,a-f,A-F]	6	Posible N.º Hexadecimal.
6	[0-9,a-f,A-F]	6	Posible N.º Hexadecimal.
6	&	13	ES un N.º Hexadecimal, hay más para leer.
6	fdc	14+	ES un N.º Hexadecimal, fin de cadena.
6	x, X, Signo, Otro	15	No pertenece a ninguna base.
4	&	0	No pertenece a ninguna base, leo la próxima cadena.
4	fdc	16+	No pertenece a ninguna base, fin de cadena.
4	x, X, Signo, Otro	15	No pertenece a ninguna base.
1	&	11	ES un N.º Decimal, hay más para leer.
1	fdc	12+	ES un N.º Decimal, fin de cadena.
1	[8-9,a-f,A-F], Signo, Otro	15	No pertenece a ninguna base.
0-	[1-9]	7	Posible N.º Decimal.
7	[0-9]	7	Posible N.º Decimal.
7	&	11	ES un N.º Decimal, hay más para leer.
7	fdc	12+	ES un N.º Decimal, fin de cadena.
7	[a-f,A-F], x, X, Signo, Otro	15	No pertenece a ninguna base.
0-	Signo	8	Posible N.º Decimal signado.
8	[1-9]	7	Posible N.º Decimal.
8	&	0	No pertenece a ninguna base, leo la próxima cadena.
8	fdc	16+	No pertenece a ninguna base, fin de cadena.
8	0, [a-f,A-F], x, X, Signo, Otro	15	No pertenece a ninguna base.
0-	&	0	Cadena vacía.
0-	fdc	16+	Fin de cadena.
0-	[a-f,A-F], x, X, Otro	15	No pertenece a ninguna base.

9, 11, 13	0	1	Posible N.º Decimal/Octal/Hexadecimal.
9, 11, 13	[1-9]	7	Posible N.º Decimal.
9, 11, 13	Signo	8	Posible N.º Decimal signado.
9, 11, 13	&	0	Cadena Vacía
9, 11, 13	fdc	16+	Fin de cadena.
9, 11, 13	[a-f,A-F], x, X, Otro	15	No pertenece a ninguna base.
10+			La última cadena fue un Octal.
12+			La última cadena fue un Decimal.
14+			La última cadena fue un Hexadecimal.
15	&	0	Salgo del estado de rechazo, leo la próxima cadena.
15	fdc	16	La última cadena no pertenece a ninguna base.
15	[0-9,a-f,A-F], x, X, Signo, Otro	15	No pertenece a ninguna base.
16+			La última cadena no pertenece a ninguna base.

De acuerdo con esta tabla y el autómata que reconoce si dada una cadena es un número de alguna de las bases octal, decimal y hexadecimal, puedo crear mi tabla de transiciones:

	0	[1-7]	[8-9]	[a-f, A-F]	x, X	Signo	&	fdc	Otro
0-	1	7	7	15	15	8	0	16+	15
1	3	2	15	15	4	15	11	12+	15
2	2	2	15	15	15	15	9	10+	15
3	15	15	15	15	15	15	9	10+	15
4	5	6	6	6	15	15	0	16+	15
5	15	15	15	15	15	15	13	14+	15
6	6	6	6	6	15	15	13	14+	15
7	7	7	7	15	15	15	11	12+	15
8	15	7	7	15	15	15	0	16+	15
9	1	7	7	15	15	8	0	16+	15
10+									
11	1	7	7	15	15	8	0	16+	15
12+									
13	1	7	7	15	15	8	0	16+	15
14+									
15	15	15	15	15	15	15	0	16+	15
16+									15

- ❖ Los estados terminales 10,12 y 14 no tienen transiciones debido a que ya no hay caracteres para leer, estos son estados de aceptación de las bases Octal, Decimal y Hexadecimal respectivamente.
- ❖ El estado terminal 16 es un estado de rechazo para los casos en que la última cadena no pertenezca a ninguna base numérica.
- ❖ El estado 15 es un estado de rechazo.
- ❖ Los estados 9,11 y 13 son los estados donde ya leí toda la cadena al leer el delimitador de cadena y se determina que es un número que pertenece a alguna de las bases Octal, Decimal y Hexadecimal respectivamente, por este motivo tienen las mismas transiciones que el estado inicial 0, ya que al comenzar a leer una nueva cadena deberán poder verificar las mismas opciones.

A la hora de programarlo, gracias a la utilidad de la función strtok que me da la habilidad de particionar una cadena de caracteres conociendo un delimitador de la misma, de esta manera pudiendo reducir la tabla de transiciones a la siguiente:

	0	[1-7]	[8-9]	[a-f, A-F]	x, X	Signo	Otro
0-	1	7	7	9	9	8	9
1	3	2	9	9	4	9	9
2	2	2	9	9	9	9	9
3	9	9	9	9	9	9	9
4	5	6	6	6	9	9	9
5	9	9	9	9	9	9	9
6	6	6	6	6	9	9	9
7	7	7	7	9	9	9	9
8	9	7	7	9	9	9	9
9	9	9	9	9	9	9	9

<https://github.com/carrilloFN/SSL/blob/main/AutomatasBasesNumericas.c>

*"Utilizando el ejercicio 1 ingresar una cadena que represente una operación simple con enteros decimales y obtener su resultado, se debe operar con +, y \*. Ejemplo =  $3+4*7+3-5 = 29$ "*

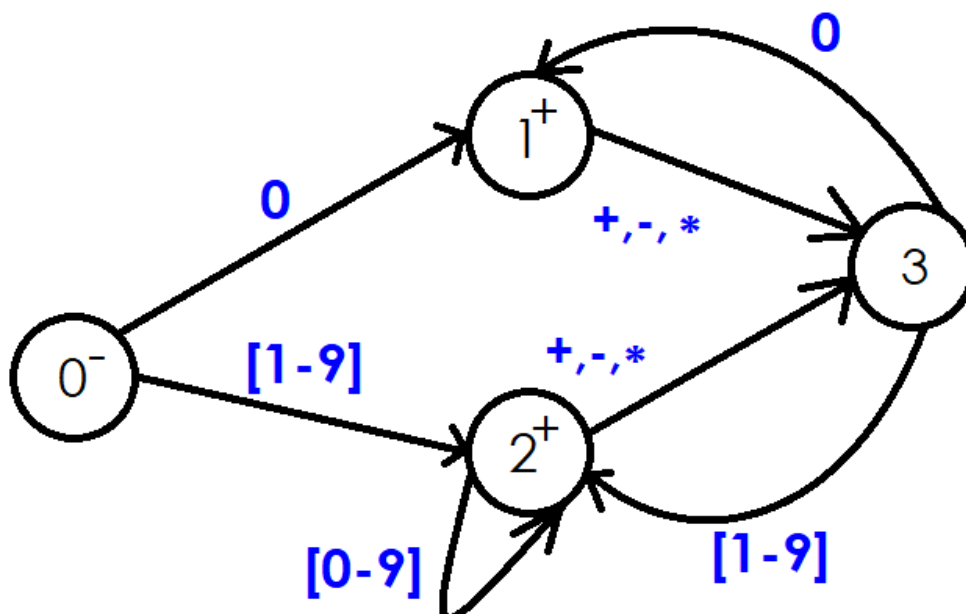
### Análisis:

Se recibirá una cadena formada por una sucesión de números decimales y operadores matemáticos básicos, tales como la suma, resta y la multiplicación. Una operación será considerada como tal siempre y cuando tenga a ambos lados del operador sus respectivos operandos.

La cadena no puede ser leída carácter a carácter y realizar su correspondiente operación según el operador dado en la cadena de manera inmediata, dado que la multiplicación tiene mayor precedencia sobre la suma y resta.

### Resolución:

Dado que las operaciones serán con números decimales (no signados) puedo utilizar el autómata diseñado para reconocer números decimales y sumarle la posibilidad de lograr reconocer operaciones decimales, de esta manera se puede verificar si las operaciones son posibles en un principio.



Transiciones			Descripción
Estado de Partida	Caracter Leído	Estado de Llegada	
0-	0	1	Primer dígito leído, 0, posible cuenta decimal válida.
1	+, -, *, /	3	Operando leído.
3	0	1	Dígito leído, posible cuenta decimal válida.
0-	[1-9]	2	Primer dígito leído, posible cuenta decimal válida.
2	[0-9]	2	Decimal leído, posible cuenta decimal válida.
2	+, -, *, /	3	Operando leído.
3	[1-9]	2	Decimal leído, posible cuenta decimal válida.
1	fdc	4+	Cuenta decimal válida.
2	fdc	4+	Cuenta decimal válida.
0-	fdc	5+	Cuenta decimal NO válida.
0-	+, -, *, /, Otro	5+	Rechazo.
1	[0-9]	5+	Rechazo.
2	Otro	5+	Rechazo.
3	fdc	5+	Cuenta decimal NO válida.
3	+, -, *, /, Otro	5+	Rechazo.
4	fdc	5+	Cuenta decimal NO válida.
4	Otro	5+	Rechazo.

Tabla de Transiciones completa

	0	[1-9]	+, -, *	fdc	Otro
0-	1	2	5	5	5
1	5	5	3	4	5
2	2	2	3	4	5
3	1	2	5	5	5
4+	5	5	5	5	5
5+	5	5	5	5	5

Como personas, al momento de realizar una operación combinacional debemos de separar en términos utilizando los operadores suma (+) y resta (-), y realizar en primer lugar las multiplicaciones, por lo cual debiera de poder diferenciar al momento de leer un operador, y realizar acciones distintas. Para poder resolver esta problemática el algoritmo que se diseñó es el siguiente:

1. Ver si la cadena ingresada es palabra, es decir, es una serie de operaciones realizable.
  - a. Empezar a recorrer la cadena desde el final hacia el inicio, de esta manera al realizar las sumas o restas puedo directamente realizar dicha operación, de leer de izquierda a derecha, al encontrar una resta, el número decimal a restar no va a ser el leído, sino el que viene después del operador, recorriendo de derecha a izquierda evito este problema.
  - b. Mientras la posición de la cadena a evaluar sea distinta de -1 (dado que dicha posición se irá restando):
    - 1) Obtener el siguiente estado.
    - 2) Realizar una determinada acción conociendo el estado en el que me encuentro, o armo el número decimal o realizo una operación matemática.