



Tecnológico de Monterrey

Tecnológico de Monterrey Campus

Guadalajara

Implementación de Métodos Computacionales (Gpo 601)

Actividad 3.2

Programando un DFA

Manual de Usuario

Joel Isaías Solano Ocampo

A01639289

Índice:

Índice	1
Instalación del programa	2
Tipos de instalación	2
Descargar	2
Ejecución del programa	3
Tipos de ejecución	3
Archivo de entrada	3
Resultado del programa	4
Consola	4
Formato	4
Lenguaje de programación	5
C++	5
Python	5
Autómata Finito Determinístico	6
Transición	6
Variables	7
Enteros	8
Reales	8
Comentarios	9

Instalación del programa:

Tipos de instalación:

Existen dos opciones viables para instalar el programa, ya sea 1) el código fuente o 2) el archivo .exe; Antes de proceder a instalar el programa de tu preferencia te mencionamos las ventajas y desventajas de cada tipo de instalación a continuación:

- **Código fuente:**

El código fuente te permitirá ver como fue programado el DFA con el IDE de tu preferencia, podrás hacer modificaciones al mismo código como base para que se pueda adaptar a tus expectativas y realizar un código totalmente diferente. Es necesario hacer énfasis en que para poder utilizar de forma adecuada este código, es necesario compilarlo con algún tipo de compilador de C++ (el lenguaje de programación elegido para el desarrollo del DFA) ya sea por medio de consola o por algún IDE que facilite este proceso.

- **Archivo .exe:**

El archivo .exe permite usar de forma directa el DFA tal y cual como fue programado, no es posible realizar ningún tipo de cambio al código fuente pero permite ejecutarse con solo abrir el archivo, por lo que es perfecto para la practicidad del mismo. Con solo instalar el archivo .exe, debería ser más que suficiente para poder correrlo sin la necesidad de algún compilador, interpretador o IDE de por medio para su funcionamiento. Es una solución sencilla y elegante para los que gusten usarlo según lo que fue programado en un inicio, un DFA.

Descargar:

Para descargarlo, es necesario ir al repositorio de [Github](#), donde se podrá descargar el archivo de tu preferencia o inclusive ambos, los que te resulten más útiles a fin de cuentas.

Ejecución del programa:

Tipos de ejecución:

Para ejecutar el programa, existen dos formas diferente de realizarlo según el tipo de descarga que hayas realizado en el punto anterior, las cuales son las siguientes:

- **Código fuente:**

El código fuente necesita de un compilador para poder ejecutarse, estos compiladores suelen estar presente en IDE's enfocados en C++ (como DevC++) o aquellos que pueden adaptarse a C++ (como Visual Code o Visual Studio). También puede solo descargarse el mismo compilador sin la necesidad de un IDE y ejecutarlo mediante comandos por consola.

- **Archivo .exe:**

Los archivos .exe son muy prácticos de usar, ya que para ejecutarlo solo es necesario correr el programa en una computadora con el SO de Windows, MS-DOS, ReactOS, OpenVMS, entre otros que lo soportan. Su ejecución empezará de inmediato.

Archivo de entrada:

Para la ejecución correcta del programa (independientemente del tipo de instalación y ejecución), es necesario de un archivo de entrada llamado "input.txt" que se encuentre al mismo nivel del programa, es decir, que su path en el explorador de archivos sean los mismos. Esto puede modificarse a conveniencia en el código fuente, pero el programa ya viene establecido de esta manera por default, por lo que para que funcione de forma correcta debe existir dicho archivo de entrada en su path correspondiente. Es recomendable poner ambos archivos (el programa principal e "input.txt") en una sola carpeta para no comprometer otros archivos en otras localizaciones.

El archivo de entrada puede ser modificado a voluntad, en Github viene un ejemplo por default de cómo se podría estructurar el mismo pero siéntete con la libertad de probar tus propios ejemplos.

Resultado del programa:

Consola:

El resultado se imprimirá por la misma consola, ya sea que hayas compilado el programa por la misma o con ayuda de algún IDE. Se decidió este método ya que no compromete la sobreescritura de archivos de salida que podrían afectar a resultados previos y que de igual forma se esperaba tener un registro de los mismos. Aun así, con el código fuente, es posible realizar estos cambios a voluntad si es lo que se espera como resultado del programa.

Formato:

De igual forma el resultado no debería variar en lo más mínimo, mostrando siempre una columna reflejando la fila en la que se encontró el token, otra para señalar el token en sí y una última para declarar el tipo de token que es.

Linea	Token	Tipo
1	b	Variable
1	=	Asignacion
1	7	Entero
2	a	Variable
2	=	Asignacion
2	32.4	Real
2	*	Multiplicacion
2	(Parentesis que abre
2	-8.6	Real
2	-	Resta
2	b	Variable
2)	Parentesis que cierra
2	/	Division
2	6.1E-8	Real
3	d	Variable
3	=	Asignacion
3	a	Variable
3	^	Exponente
3	b	Variable
4	e	Variable
4	=	Asignacion
4	076	Octal
5	f	Variable
5	=	Asignacion
5	0xAF68	Hexadecimal
6	// c=35*19 ^ 3.	Comentario
7	_a25 // error	Error

Lenguaje de programación:

C++:

El lenguaje de programación seleccionado para el desarrollo de este programa fue C++. Es un lenguaje de programación compilado, multiparadigma, principalmente de tipo imperativo y orientado a objetos, incluyendo también programación genérica y funcional. Este lenguaje es muy completo, se eligió para el desarrollo del programa debido a los siguientes criterios:

- Ha sido la evolución de C, un lenguaje de programación mejorado y muy vigente hasta la fecha.
- Es un lenguaje muy completo, la solución de este programa se pudo haber hecho de diferentes formas con diferentes paradigmas.
- Gran variedad de librerías presentes para la resolución del problema.
- Mucha documentación en internet en base a métodos de resolución, implantación de código, solución de errores, gran apoyo de la comunidad, etc.
- Sugerencia del profesor de la materia (no nos especificó el motivo pero garantizo una recompensa para los que lo implementan de esta forma).

Python:

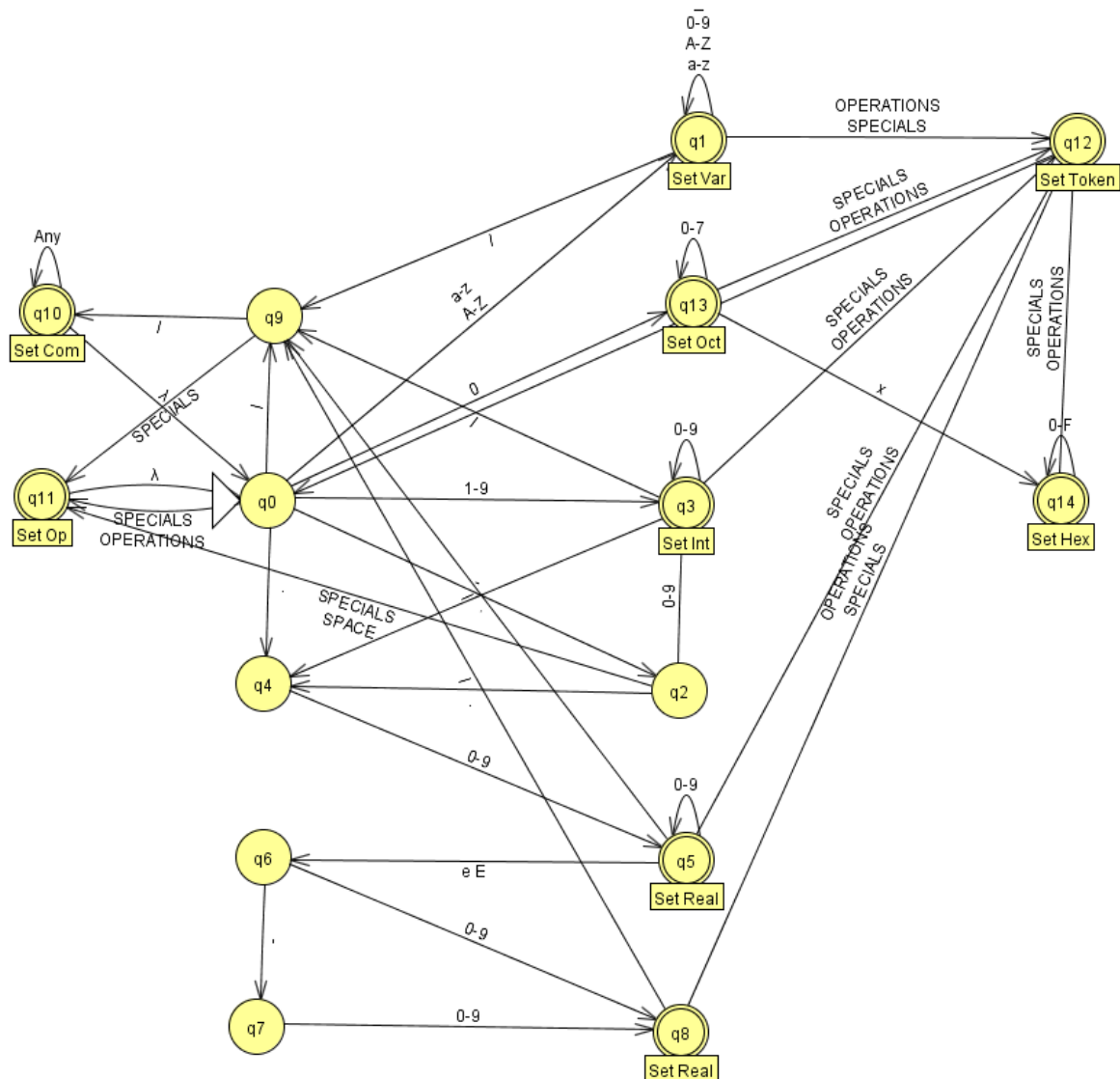
Python es un lenguaje de alto nivel de programación interpretado cuya filosofía hace hincapié en la legibilidad de su código. Python es un lenguaje que ha aumentado en popularidad, su sintaxis es bastante sencilla y amigable, ha tenido un gran apoyo de la comunidad y gracias a su increíble capacidad de resolución e implementación de problemas y código, es un lenguaje de programación muy usado en la industria y en las grandes empresas.

Aun así, este lenguaje de programación no fue elegido porque se tomó en preferencia la sugerencia del profesor. De todas maneras, como desarrollador creemos que existen mejores herramientas disponibles en Python que en C++, y aunque en ambos se pudiese desarrollar con una aproximación similar, Python es un lenguaje más moderno que aún le queda bastante por explotar a beneficio y que de forma personal nos ayude a crear estas herramientas necesarias para un mejor desempeño como programadores.

Autómata Finito Determinístico:

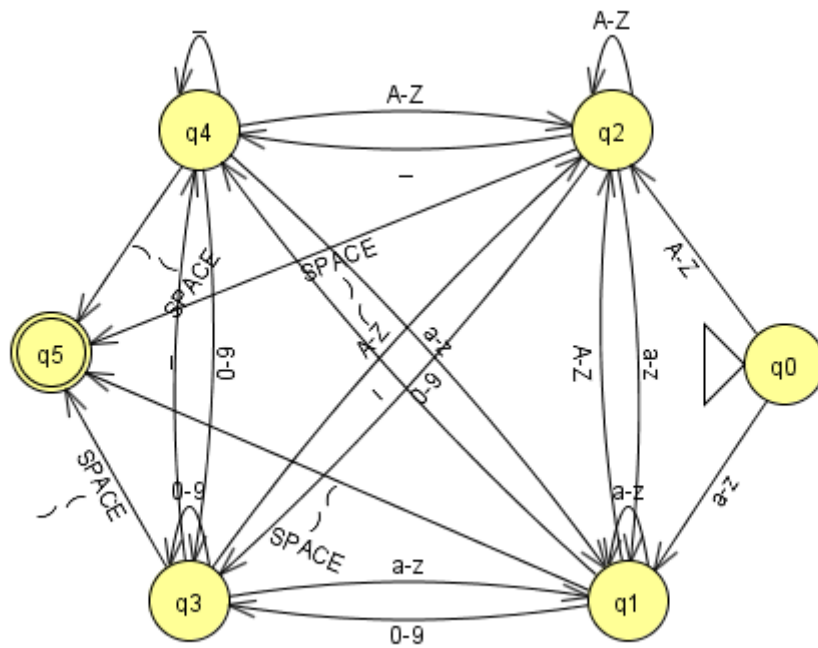
Transición:

A continuación se tiene el DFA de transición del programa. Este autómata finito determinístico es capaz de detectar variables, enteros, reales, operadores, caracteres especiales, comentarios, octales y hexadecimales.



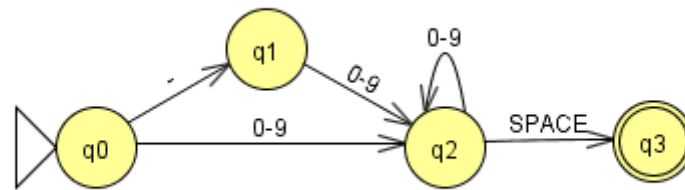
A continuación se mostrarán otros DFA's de transición como ejemplos y más resumidos que el visto previamente, que buscan solo un tipo de expresión para mejor entendimiento y su funcionamiento dentro del programa principal.

Variables:



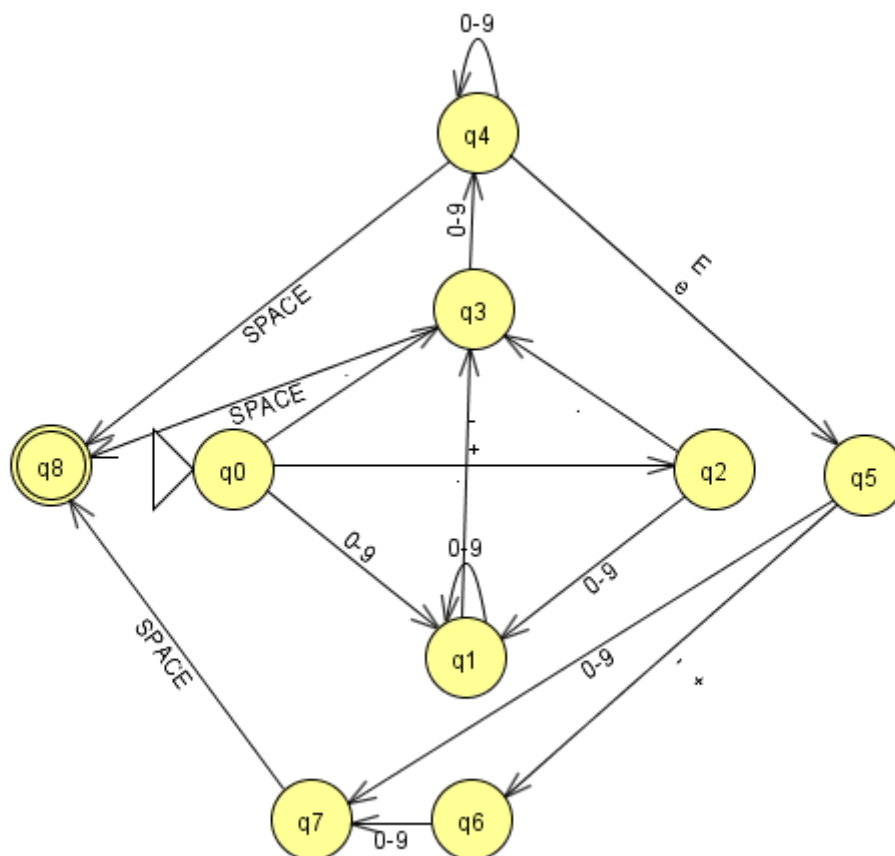
Variables					
Estado	a-z	A-Z	0-9	-	SPACE
>Q0	{q1}	{q2}	{q3}	{q4}	∅
Q1	{q1}	{q2}	{q5}	{q6}	{q7}
Q2	{q1}	{q2}	{q5}	{q6}	{q7}
Q3	{q1}	{q2}	{q5}	{q6}	{q7}
Q4	{q1}	{q2}	{q5}	{q6}	{q7}
*Q5	∅	∅	∅	∅	∅

Enteros:



Enteros			
Estado	-	0-9	SPACE
>Q0	{q2}	{q3}	\emptyset
Q1	\emptyset	{q3}	\emptyset
Q2	\emptyset	{q2}	{q3}
*Q3	\emptyset	\emptyset	\emptyset

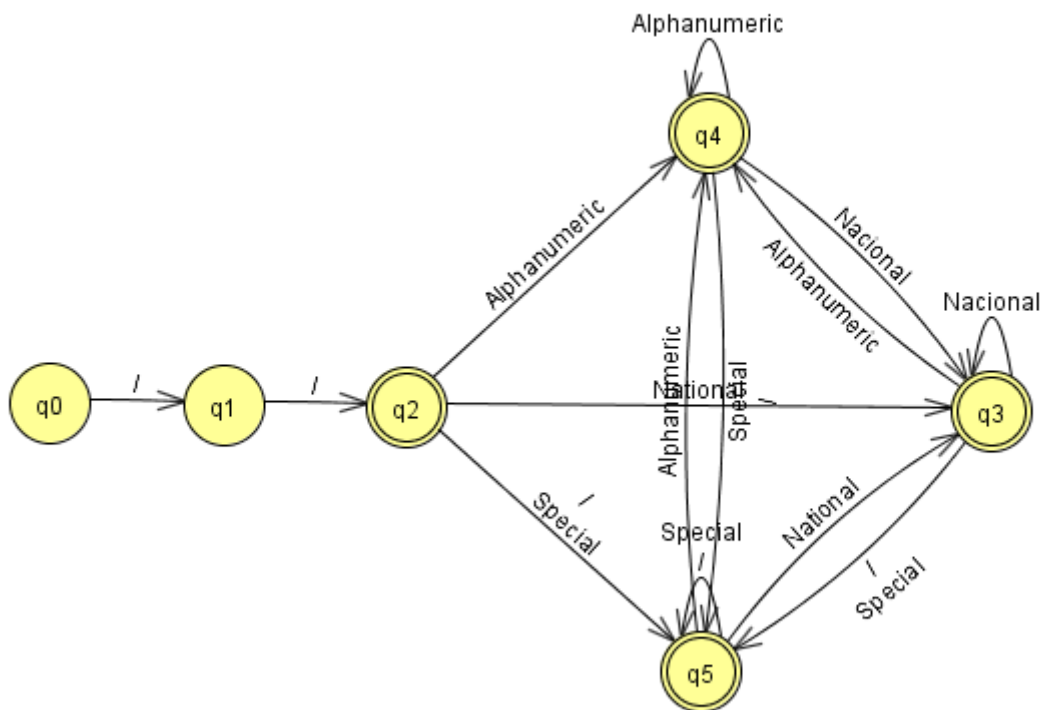
Reales:



Reales					
Estado	-	0-9	.	e\E	SPACE
>Q0	{q3}	{q1}	{q4}	\emptyset	\emptyset
Q1	\emptyset	{q1}	{q4}	\emptyset	\emptyset
Q2	\emptyset	{q1}	{q4}	\emptyset	\emptyset

Q3	\emptyset	{q1}	{q4}	\emptyset	\emptyset
Q4	\emptyset	{q4}	\emptyset	{q5}	{q9}
Q5	{q8}	{q6}	\emptyset	\emptyset	\emptyset
Q6	\emptyset	{q7}	\emptyset	\emptyset	\emptyset
Q7	\emptyset	\emptyset	\emptyset	\emptyset	{q8}
*Q8	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset

Comentarios:



Comentarios					
Estado	/	Alfanumérico	Nacional	Especial	ϵ
>Q0	{q1}	\emptyset	\emptyset	\emptyset	\emptyset
Q1	{q2}	\emptyset	\emptyset	\emptyset	\emptyset
*Q2	{q5}	{q4}	{q3}	{q5}	\emptyset
*Q3	{q5}	{q4}	{q3}	{q5}	\emptyset
*Q4	{q5}	{q4}	{q3}	{q5}	\emptyset
*Q5	{q5}	{q4}	{q3}	{q5}	\emptyset