Manual técnico

Manual Técnico sobre AFN y AFD

Introducción

Los Autómatas Finitos son una parte esencial de la teoría de la computación y son utilizados para modelar y reconocer lenguajes regulares. En este manual, nos enfocaremos en dos tipos específicos de autómatas finitos: el Autómata Finito No Determinista (AFN) y el Autómata Finito Determinista (AFD). Estos conceptos son fundamentales en la teoría de lenguajes formales y son ampliamente utilizados en la construcción de compiladores, análisis léxico y en la resolución de problemas relacionados con la teoría de la computación.

Autómata Finito No Determinista (AFN)

Definición

Un Autómata Finito No Determinista (AFN) es un modelo matemático de cómputo que consiste en un conjunto finito de estados, un alfabeto de entrada, una función de transición, un estado inicial y un conjunto de estados de aceptación. A diferencia de un AFD, en un AFN, las transiciones desde un estado pueden tener múltiples destinos posibles para un mismo símbolo de entrada.

Componentes

Estados: Conjunto finito de estados que representa el comportamiento del autómata.

Alfabeto: Conjunto finito de símbolos de entrada.

Función de Transición: Reglas que especifican cómo el autómata cambia de un estado a otro para cada símbolo de entrada.

Estado Inicial: El estado desde el cual comienza la ejecución del autómata.

Estados de Aceptación: Conjunto de estados que indica si una cadena de entrada es aceptada.

Autómata Finito Determinista (AFD)

Definición

Un Autómata Finito Determinista (AFD) es similar a un AFN, pero con la restricción de que, para cada símbolo de entrada, existe solo una transición posible desde un estado dado. Esto hace que el comportamiento del autómata sea determinista.

Componentes

Estados: Conjunto finito de estados que representa el comportamiento del autómata.

Alfabeto: Conjunto finito de símbolos de entrada.

Función de Transición: Reglas que especifican cómo el autómata cambia de un estado a otro para cada símbolo de entrada.

Estado Inicial: El estado desde el cual comienza la ejecución del autómata.

Estados de Aceptación: Conjunto de estados que indica si una cadena de entrada es aceptada.

Ejemplo

Consideremos el siguiente AFD:

AFD Ejemplo

Estados: Q = {q0, q1, q2}

Alfabeto: Σ = {0, 1}

Función de Transición: δ(q0, 0) = q1, δ(q1, 1) = q2, δ(q2, 0) = q0

Estado Inicial: q0

Estados de Aceptación: {q2}

Comparación entre AFN y AFD

Determinismo:

AFN: No determinista, múltiples transiciones posibles para un símbolo de entrada y estado.

AFD: Determinista, una única transición posible para cada símbolo de entrada y estado.

Complejidad:

AFN: Puede ser más compacto y fácil de diseñar.

AFD: Más determinista y puede ser más eficiente en términos de espacio y tiempo de ejecución.

Diseño:

AFN: Puede ser más natural para diseñar en algunos casos.

AFD: Más estructurado y fácil de entender en muchos casos.

Aplicaciones:

AFN: Ampliamente utilizado en la especificación de expresiones regulares.

AFD: Preferido en implementaciones de compiladores y análisis léxico debido a su determinismo.

Conclusión

Tanto los AFN como los AFD son herramientas poderosas en la teoría de lenguajes formales y automatas. La elección entre ellos depende de los requisitos específicos de diseño y del problema a resolver. En la práctica, se pueden convertir entre sí según sea necesario para aprovechar las fortalezas de cada tipo de autómata.

inicio> ::= LLAVEA <declaraciones> LLAVEC

<declaraciones> ::= <declaraciones> <declaracion>

| <declaracion>

<declaracion> ::= <conj>

| <expresiones>

<conj> ::= CONJ DOS\_PUNTOS IDENTIFICADOR IGUAL <notacion> PUNTO\_COMA

<notacion> ::= <letra\_mayuscula>

| <letra\_minuscula>

| <secuencia\_numero>

| <secuencia\_letras>

| <especial>

| <numeros>

<letra\_mayuscula> ::= LETRA\_MAYUSCULA GUION\_N LETRA\_MAYUSCULA

<letra\_minuscula> ::= LETRA\_MINUSCULA GUION\_N LETRA\_MINUSCULA

<numeros> ::= NUMERO GUION NUMERO

<secuencia\_numero> ::= NUMERO COMA <secuencia\_numero>

| NUMERO

<secuencia\_letras> ::= <letras\_MM> COMA <secuencia\_letras>

| <letras\_MM>

<letras\_MM> ::= LETRA\_MAYUSCULA

| LETRA\_MINUSCULA

<expresiones> ::= IDENTIFICADOR IGUAL <operacion> PUNTO\_COMA

<operacion> ::= OR <operacion> <operacion>

| CONCATENACION <operacion> <operacion>

| POSITIVO <operacion>

| KLEENE <operacion>

| OPCIONAL <operacion>

| <operando>

<operando> ::= PARENTESIS\_A IDENTIFICADOR PARENTESIS\_C

| STRING

| LETRA\_MAYUSCULA

| LETRA\_MINUSCULA

| NUMERO

| SALTO

| COMILLA\_DOBLE

| COMILLA\_SIMPLE

<simbolos> ::= ASCII

| OR

| CONCATENACION

| POSITIVO

| KLEENE

| OPCIONAL

<especial> ::= <simbolos> GUION\_N <simbolos>