DÍA MES AÑO NOTA /

Escuela Politécnica Nacional

Nombre: Fernando Elicio Huilca Villagómez Fecha:

Especificación de tokens con expresiones regulares.

Los tokens se definen mediante expresiones regulares, una notación formal compacta y poderosa. Se introducen conceptos fundamentales de teoría de lenguajes:

- Alfabeto: conjunto de símbolos (como letras o dígitos).
- Cadena: secuencia de símbolos
- Lenguaje: conjunto de cadenas sobre un alfabeto.

Autómatas finitos y reconocimientos de patrones.

El capítulo profundiza en cómo lex, una herramienta generadora de analizadores léxicos convierte las expresiones regulares en autómatas finito (AF) Estos modelos matemáticos son capaces de reconocer cadenas válidas de entrada y aceptar o rechazar según los patrones definidos

Tipos de Autómatas

Autómata finito no determinista: Pueden tener multiples transiciones para un mismo símbolo, incluso transiciones con cada cadena vacía (ε) Es más facil de construir, pero no tan directo de implementar.

Autómata finito determinista (AFD) para cada estado o símbolo, hay exactamente una transición. Aunque más rigido, es más eficiente para la interpretación real en compiladores.

Ambos equivalentes en capacidad ya que reconocen los mismos lenguajes

DÍA MES AÑO NOTA 7

Conversión entre expresiones regulares y autómatas

Para construir analizadores léxicos eficientes se convierte una expresión regular en un AFN mediante el algoritmo de Thompson. Posteriormente, este AFN se transforma en un AFD usando la construcción de subconjuntos, lo que permite simular el autómata de manera eficiente sobre cadenas de entrada.

Además, se presenta el proceso inverso: convertir un AFD en una expresión regular útil para análisis teóricos o depuración.

3. Implementación y simulación.

Se introducen algoritmos para simular un AFD, que verifica si una cadena pertenece al lenguaje aceptado. El analizador léxico generado simula este autómata al reconocer los caracteres del programa fuente y devolver el token correspondiente cuando encuentra una coincidencia. Se discute también la eficiencia temporal de estos algoritmos, destacando que el uso de AFDs es generalmente más rápido para cadenas largas, mientras que simular directamente un AFN puede ser más eficiente en casos simples o de uso único.

Lex permite describir analizadores léxicos de forma modular y automatizada, a partir de expresiones regulares extendidas. El compilador lex traduce esas expresiones en un programa en C.

Fernando Llop