Análisis Léxico

Diseño de Compiladores

Dr. Víctor de la Cueva
vcueva@itesm.mx

Scanner

- Programa fuente (texto) → TOKENS:
  - Palabras de un lenguaje natural
  - Secuencia de caracteres → unidad de información:
    - Palabras reservadas
    - Identificadores
    - Símbolos especiales
    - Constantes
- Métodos (algoritmos) principales (reconocimiento de patrones):
  - Expresiones regulares
  - Autómatas finitos

Tokens

- Son entidades lógicas que por lo general se definen como un tipo enumerado.
  - e.g. en C:

```
typedef enum
    {IF, THEN, ELSE, PLUS, MINUS, NUM, ID, …}
    TokenType;
```

  - En Python: clase que hereda de la clase `Enum` (importar `from enum import Enum`).
  - Los tokens, como unidades lógicas, se deben distinguir claramente de las cadenas de caracteres (lexema) que representan.
    - IF representa a "if", PLUS representa a "+".

Atributos

- Cualquier valor asociado con un token.
  - NUM:
    - Valor de cadena → "4537"
    - Valor numérico → 4537
  - PLUS
    - Valor de cadena → "+"
    - Operador aritmético → +
- Es útil recolectar los atributos de un token en un registro (`TokenRecord`):

```
Typedef struct {
    TokenType tokenval;
    char * stringval;
    int numval;
} TokenRecord;
```

```
Typedef struct {
      TokenType tokenval;
      union {
            char * stringval;
            int numval;
      } attribute;
} TokenRecord;
```

Repaso

EXPRESIONES REGULARES

Expresiones regulares

- Representan patrones de cadenas de caracteres.
- r → L(r):
  - Σ → Alfabeto → Símbolos → Conjunto de caracteres
  - Normalmente ASCII
  - Metacaracteres: \ y ‘ (también son símbolos)

Operaciones de ER

- Tres operaciones básicas:
  - Selección (entre alternativas): |
  - Concatenación: yuxtaposición
  - Repetición (cerradura): *
- Extensiones:
  - Yuxtaposición n veces de a: a^n
  - Al menos una vez: +
  - Cadena vacía: ε
  - Agrupamiento: ( )
  - Cualquier carácter: .
  - Intervalo (clases de caracteres): [a-z]
  - Cualquier carácter que no esté (not): ~a, ~(a|b|c) ≡ ^a, [^abc]
  - Opcionales: ? (+,-)?

38

ER para tokens en lenguajes

- Tienden a caer en categorías:
  - Palabras reservadas
  - Símbolos especiales
  - Identificadores
  - Contantes o literales (numéricas o de cadena)

43

Ambigüedad

- Algunas cadenas se puede definir con varias expresiones regulares: `then`, `<>`.
- Se deben establecer: reglas de no ambigüedad:
  - Identificador o palabra reservada → palabra reservada
  - Principio de la subcadena más larga
  - Delimitadores de token (¿se eliminan o no?)

Repaso

AUTÓMATAS FINITOS

Autómatas finitos

- O máquinas de estados finitos, son una clase particular de algoritmos.
- Uso → proceso de reconocimiento de patrones en cadenas → construir analizadores léxicos.
- Fuerte relación entre AF y ER

Nomenclatura

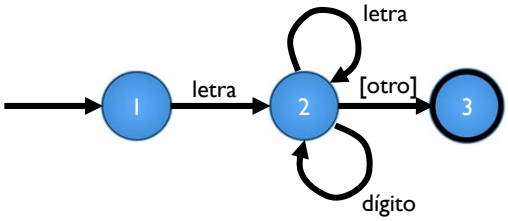
- Estados: círculos
- Estados finales: círculos dobles
- Transiciones: aristas dirigidas
- Estado inicial: con una arista sin círculo previo
- Etiquetas: en los estados o en las aristas

48

51

DFA o NFA en código

- Existen varias formas.
- No todas son útiles para un analizador léxico.
- Ejemplo: identificadores



59

- Con anidación
- Manteniendo el estado
- Tabla

ANALIZADOR DE LÉXICO PARA TINY

Dos implementaciones

- Directa usando la variable "estado".
- Tabla: generándolo automáticamente mediante Lex.

Referencias

- A.V. Aho, M. S. Lam, R. Sethi, and J. D. Ullman. *Compilers: Principles, Techniques, and Tools*. 2nd Pearson (2012).
- K.C. Louden. *Contrucción de Compiladores: principios y práctica*. Thomson (2004).
- Alex Aiken. Compilers. Stanford Online (2018).
  - https://lagunita.stanford.edu/courses/Engineering/Compilers/Fall2014/about