Lenguajes formales y autómatas

Docentes

Ariel Rossanigo (arielrossanigo@gmail.com)

Mariano Ferrero (marianoferrero.mf@gmail.com)

Grupo de trabajo

Dirección web del grupo: https://groups.google.com/group/lenguajes_formales_y_automatas_ucse_dar

Condiciones de la materia

Para regularizar:

- Aprobar el examen con 60% o más
- Aprobar el TP (entrega y defensa)

Para promocionar

- Aprobar el examen con 75% o más
- Aprobar el TP (entrega y defensa)

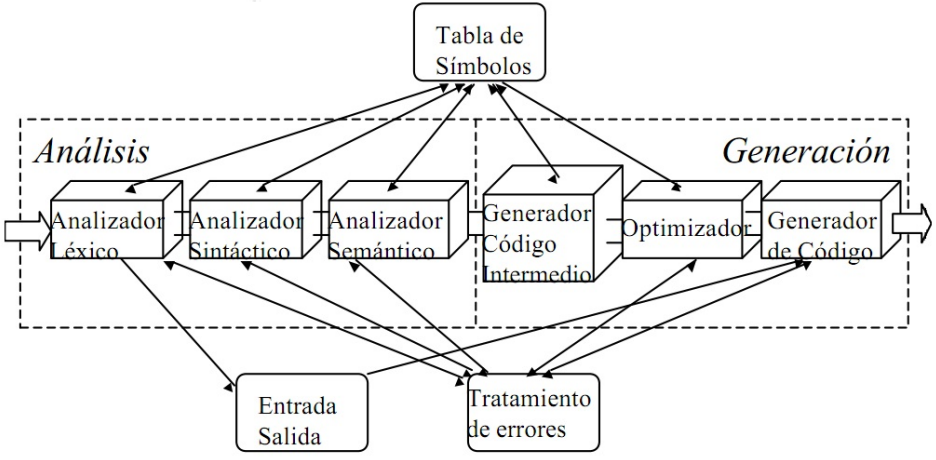
¿Qué es un traductor?

- Programa que transforma código fuente en código objeto
- Los hay de 1 pasada y de múltiples pasadas

Tipos de traductores:

- **Compilador:** traducción y ejecución están separados; ej: C, Pascal
- **Intérprete:** la traducción se hace a medida que se ejecuta; ej: PHP, javascript
- **Compilador JIT:** compilación a *bytecode* y luego interpretación de *bytecode* a codigo nativo; ej: Java, .Net, Python

¿Estructura básica de un compilador?



Fase de análisis

- **Analizador léxico:** aisla la entrada y genera una lista de **tokens**
- **Analizador sintáctico:** verifica si la secuencia de tokens es correcta según reglas sintácticas (gramática). Genera el árbol sintáctico.
- **Analizador semántico:** recibe una entrada bien formada y verifica que tenga sentido. Reúne información para fases posteriores.

Fase de síntesis

- **Generador de código intermedio:** genera una representación intermedia de más bajo nivel pero independiente del hardware
- **Optimizador:** análisis y modificación del código intermedio para optimizar el procesamiento.
- **Generador de código:** generación de código nativo

Tabla de símbolos

- Reúne información sobre los distintos atributos (tipo de atributo, tipo de datos, cant. y tipos de parámetros, etc)
- Se completa en las fases de análisis léxico y sintáctico
- Es utilizada para diversas comprobaciones en el análisis semántico (comprobación de tipos, existencia de variables, etc)
- Es consultada para la generación de código.

Detección de errores.

cada fase es capaz de reconocer distintos tipos de errores:

- léxico: caracteres que no forman ningún componente léxico
- sintáctico: secuencias de caracteres mal formadas
- semántico: errores de significado

Ante la presencia de errores se puede:

- detener la compilación
- tratar de recuperar el error y continuar compilando

Teoría de lenguajes

Alfabeto

- Conjunto no vacío de **letras** o **símbolos**.
- Un alfabeto lo representamos como Σ
- Las letras griegas minúsculas denota símbolos genéricos
- Los símbolos puntuales se denotan con letras latinas minúsculas

Ejemplos alfabetos:

- Σ_1= {a, b, c, .., z}
- Σ_2= {0, 1}

Palabra

- Se llama **palabra**, formada con los símbolos de un alfabeto, a todo secuencia finita de las letras de ese alfabeto.

Ejemplos de palabras sobre Σ_1: *juan*, *pedro* y *coche*.

Para representar palabras se usarán las últimas letras del alfabeto latino; ej: *x* = *pedro*

Las palabras se construyen a partir de letras, es necesario:

- palabra vacía λ
- constructor de palabras (concatenación): a partir de σy *x* forma una palabra σ. *x*
- analizadores **inicial** y **resto**

Longitud de una palabra

- Se llama longitud de una palabra al número de letras que la componen.
- La longitud se representa con la notación |x|.
  - |juan| = 4
  - $|\lambda| = 0$

Universo del discurso o Lenguaje universal

- Es el conjunto de todas las palabras que se pueden construir con las letras de un alfabeto Σ
- Se representa con Σ^*
- Σ^* es un conjunto infinito

Operaciones con palabras

- **Concatenación**
  - Sean *x* e *y* palabras que pertenecen a Σ^*, se representa como *xy* o *x.y* a *z*, tal que *z* está formada por las letras de *x* seguidas de las letras de *y*
- **Potencia**
  - Se llama potencia i-ésima de una palabra a la operación que consiste en concatenarla **i** veces consigo misma.
  - Ej: x^i = xxx...x (i veces)
- **Palabra refleja o inversa**
  - Sea $x = \sigma_1 \sigma_2 ... \sigma_n$, se denomina inversa a $x^R = \sigma_n ... \sigma_2 \sigma_1$

Lenguajes

- Se denomina **lenguaje** sobre el alfabeto Σa cualquier subconjunto del lenguaje universal Σ^*
- El **lenguaje vacío** Φes distinto de $\{\lambda\}$. $|\Phi| = 0$ **vs** $|\{\lambda\}| = 1$

Operaciones con lenguajes

Sean $L_1 = \{x, y\}$ y $L_2 = \{v, w\}$lenguajes que pertenecen a Σ^*

- **Union** $L_1 + L_2 = \{x | x \in L_1 \vee x \in L_2\}$
  - Ej: $L_1 + L_2 = \{x, y, v, w\}$
- **Concatenación** $L_1 L_2 = \{xy | x \in L_1, y \in L_2\}$
  - Ej: $L_1 L_2 = \{xv, xw, yv, yw\}$
- **Potencia**: consiste en concatenarlo consigo mismo. $L^0 = \lambda$

- **Cierre** o **clausura** de **L** al lenguaje que se obtiene uniendo **L** con todas sus potencias.
  - $L^* = \bigcup_{i=0}^{\infty} L_i$
- **clausura positiva** de **L** es $L^+ = \bigcup_{i=1}^{\infty} L_i$
- **lenguaje inverso** de **L** es $L^R = \{x^R | x \in L\}$

Bibliografía y enlaces útiles.

- Alfonseca Cubero y otros - Teoría de autómatas y lenguajes formales - McGRAW-HILL
- García y otros - Teoría de autómatas y lenguajes formales - ALFAOMEGA GRUPO EDITOR S.A.
- Aho Alfred y Ullman Jeffrey - Compiladores, principios, técnicas y herramientas - PEARSON EDUCACION