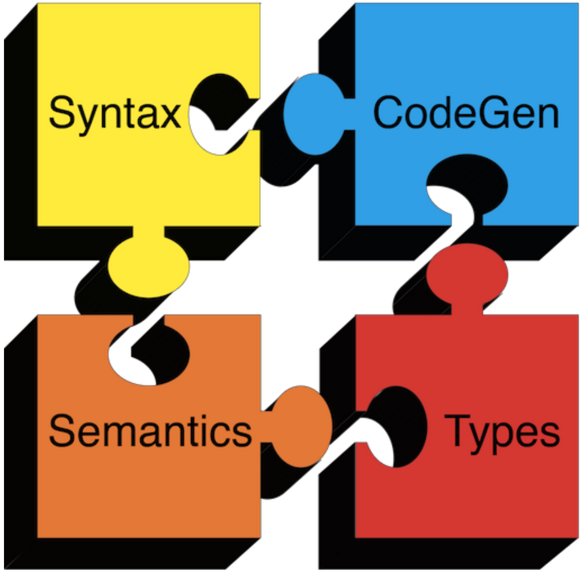

Compiladores

Regular Languages

Alex Aiken

- Estructura léxica = Clases de token
- Debemos decir qué conjunto de cadenas hay en una clase de token
  - Usar ***lenguajes regulares***

Regular Expressions

- Carácter único

'c' = {"c"}

- Épsilon

ε = {""}

≠

∅

Las expresiones regulares son una herramienta poderosa y flexible utilizada para identificar, buscar y manipular secuencias específicas de caracteres dentro de textos, basándose en patrones definidos.

Los metacaracteres son caracteres no alfabéticos que poseen un significado especial en las expresiones regulares. El conjunto de metacaracteres que se pueden utilizar en sintaxis de expresión regular ampliada es el siguiente:

* + ? $ ^ . () | \ {} [

- Unión

$A + B = \{a \mid a \in A\} \cup \{b \mid b \in B\}$

Esto significa que el lenguaje resultante incluirá todas las cadenas que sean parte de A o parte de B

- Concatenación

$(A\ B) = \{ab \mid a \in A \wedge b \in B\}$

Esta operación crea un nuevo conjunto de cadenas combinando cada cadena posible de A con cada cadena posible de B.

- Iteración

$(A)^* = \bigcup (i \geq 0)\ A^i$

$A^i = \underbrace{A \ldots A}_{i \text{ times}}$

$A^0 = \epsilon$

- Unión

$A + B = \{a \mid a \in A\} \cup \{b \mid b \in B\}$

Definición: La unión de dos lenguajes regulares A y B es el lenguaje que contiene todas las cadenas que son parte de A, de B, o de ambos.

Ejemplo: Si A es el lenguaje que contiene las cadenas {"apple", "orange"} y B contiene {"banana", "mango"}, entonces A + B incluirá {"apple", "orange", "banana", "mango"}.

Uso en expresiones regulares: En una expresión regular, el metacarácter '|' representa la unión. Por ejemplo, apple|orange coincidiría con "apple" o "orange".

- Concatenación

$(A\ B) = \{ab \mid a \in A \wedge b \in B\}$

Definición: La concatenación de dos lenguajes A y B es el lenguaje que contiene todas las cadenas que pueden formarse tomando una cadena de A y uniéndola con una cadena de B.

Ejemplo: Si A = {"a", "b"} y B = {"1", "2"}, la concatenación AB produciría {"a1", "a2", "b1", "b2"}.

Uso en expresiones regulares: La concatenación es implícita en las expresiones regulares. Por ejemplo, la expresión ab coincidiría con la cadena "ab".

- Iteración (A)*:

$(A)^* = \cup\ (i \geq 0)\ A^i$

Definición: La estrella de Kleene o iteración aplica a un lenguaje A y produce un lenguaje que contiene todas las posibles cadenas formadas por la repetición de cero o más veces de las cadenas en A.

Ejemplo: Si A es el lenguaje que contiene la cadena "ab", entonces A* incluiría "", "ab", "abab", "ababab", y así sucesivamente.

Uso en expresiones regulares: El metacarácter '*' indica la iteración de Kleene. Por ejemplo, ab* coincidiría con "a", "ab", "abb", "abbb", etc.

- **Def.** Las expresiones regulares sobre Σ son el conjunto más pequeño de expresiones, que incluye:

$$
R = \epsilon \mid \text{'}c\text{'} \mid R + R \mid RR \mid R^*
$$

$c \in \Sigma$

grammar

Regular Languages

$\Sigma = \{0,1\}$

$1^* = \bigcup_{i \ge 0} 1^i = \text{""} + 1 + 11 + 111 + 1111 + \dots = \text{""} + 1 + \dots + \underbrace{1 \cdots 1}_{i} + \dots$ = all strings of 1's

$1^* + 1$

$(1+0)1 = \{ab \mid a \in 1+0 \wedge b \in 1\} = \{11, 10\}$

$11 + 10$

$0^* + 1^* = \{0^i \mid i \ge 0\} \cup \{1^i \mid i \ge 0\}$

$(0+1)^* = \bigcup_{i \ge 0} (0+1)^i = \text{""}, 0+1, (0+1)(0+1), \dots, \underbrace{(0+1)\cdots(0+1)}_{i \text{ times}} =$ all strings of 0's and 1's

$= \Sigma^*$

Regular Languages

Alfabeto (Σ): {0,1}
Este alfabeto consiste en dos dígitos: 0 y 1.

Expresiones Regulares:

1*: Se define como la unión (U) de '1' repetido 'i' veces para i ≥ 0, lo que se traduce en todas las posibles cadenas que contienen solo el carácter '1'. Esto incluye la cadena vacía (por '1' elevado a 0), '1', '11', '111', y así sucesivamente.

(1+0): Corresponde al conjunto {a | a ∈ 1+0 ∧ b ∈ 1}. Este es el conjunto que resulta de tomar cada elemento de la unión de '1' y '0' y concatenarlo con '1'. En términos de conjuntos, esto da como resultado {11, 01}.

0* + 1*: Esta es una expresión que representa la unión de todas las posibles cadenas de ceros y la unión de todas las posibles cadenas de unos. En términos de conjuntos, incluiría {0^i | i ≥ 0} ∪ {1^i | i ≥ 0}, que es equivalente a todas las posibles cadenas formadas exclusivamente por '0's o por '1's.

Alfabeto (Σ): {0,1}

Este alfabeto consiste en dos dígitos: 0 y 1.

Expresiones Regulares:

(0+1)*: Esta expresión representa la unión de todas las cadenas posibles que se pueden formar a partir de la concatenación de '0' y '1' en cualquier orden y longitud, comenzando desde la cadena vacía y continuando con todas las combinaciones posibles de '0' y '1'. Esto es equivalente a Σ*, que es el conjunto de todas las cadenas posibles sobre el alfabeto Σ.

Regular Languages

- Las expresiones regulares especifican idiomas regulares

syntax　　　　set of strings

- Cinco constructos
  - Dos casos base
    - Cadenas vacías y de 1 carácter

ϵ

Tres expresiones compuestas

- unión, concatenación, iteración

Elija los idiomas habituales que son equivalentes al idioma regular dado: $(0 + 1)^*1(0 + 1)^*$

$\Sigma = \{ 0, 1 \}$

- ☐ $(01 + 11)^*(0 + 1)^*$
- ☐ $(0 + 1)^*(10 + 11 + 1)(0 + 1)^*$
- ☐ $(1 + 0)^*1(1 + 0)^*$
- ☐ $(0+ 1)^*(0 + 1)(0 + 1)^*$