Compiladores
Análise Léxica
Expressões Regulares

Prof. Dr. Luiz Eduardo G. Martins
(adaptado por Profa Dra Ana Carolina Lorena)
UNIFESP

Expressões Regulares

- No processo de análise léxica, os *tokens* válidos da linguagem normalmente são especificados por expressões regulares
- Expressões regulares representam padrões de cadeias de caracteres
- Uma expressão regular *r* é completamente definida pelo conjunto de cadeias de caracteres com as quais ela "casa"
  - Esse conjunto é chamado de linguagem gerada pela expressão regular, e é denotado como *L(r)*

Expressões Regulares Básicas

- Expressões regulares básicas
  - São os caracteres em separado do alfabeto
    - Alfabeto: conjunto de símbolos legais da linguagem, usualmente denotado pelo símbolo grego ∑
  - Dado um caractere *a* do alfabeto ∑ indicamos que a expressão regular **a** casa com o caractere *a* escrevendo $L(\mathbf{a}) = \{a\}$

Expressões Regulares Básicas

- Expressões regulares básicas
  - Precisamos de símbolos adicionais para situações especiais
    - Utilizamos o meta-símbolo ε para denotar a cadeia vazia (uma cadeia sem caracteres)

$$L(\boldsymbol{\varepsilon}) = \{\varepsilon\}$$

    - Utilizamos o meta-símbolo Φ para denotar a linguagem que não casa com nenhuma cadeia de caracteres (linguagem vazia)

$$L(\boldsymbol{\Phi}) = \{\}$$

Expressões Regulares: Operações

- Escolha entre alternativas
  - Se *r* e *s* são expressões regulares, então *r* | *s* é uma expressão regular que casa com qualquer cadeia que case com *r* ou com *s*
  - Em termos de linguagem, a linguagem de *r* | *s* é a união de *r* e *s*

$$L(\boldsymbol{r}|\boldsymbol{s}) = L(\boldsymbol{r}) \cup L(\boldsymbol{s})$$

Exemplos:

Para a expressão regular ***a*** **|** ***b***,

$L(\boldsymbol{a}|\boldsymbol{b}) = L(\boldsymbol{a}) \cup L(\boldsymbol{b}) = \{a\} \cup \{b\} = \{a, b\}$

Para a expressão regular ***a*** **|** ***ε***

$L(\boldsymbol{a}|\boldsymbol{\varepsilon}) = \{a, \varepsilon\}$

Para a expressão regular ***a*** **|** ***b*** **|** ***c*** **|** ***d***

$L(\boldsymbol{a}|\boldsymbol{b}|\boldsymbol{c}|\boldsymbol{d}) = \{a, b, c, d\}$

Expressões Regulares: Operações

- Concatenação
  - A concatenação de duas expressões regulares *r* e *s* é denotada por *rs*, e casa com qualquer cadeia de caracteres que seja a concatenação de duas cadeias, desde que a 1ª case com *r* e a 2ª case com *s*

$$L(\mathbf{rs}) = L(\mathbf{r})L(\mathbf{s})$$

Exemplos:

Para a expressão regular **(a|b)c**,

$L(\mathbf{(a|b)c}) = L(\mathbf{a|b})L(\mathbf{c}) = \{a,b\}\{c\} = \{ac, bc\}$

Para a expressão regular *rs*, em que $r = (a|b)$ e $s = (c|d)$, *temos*

$L(\mathbf{rs}) = L(\mathbf{(a|b)(c|d)}) = L(\mathbf{a|b})L(\mathbf{c|d}) = \{a, b\}\{c, d\} = \{ac, ad, bc, bd\}$

Expressões Regulares: Operações

- Repetição ou fecho
  - Seja uma expressão regular *r*, a operação de repetição é denotada por *r**
  - A expressão *r** casa com qualquer concatenação finita de cadeias de caracteres (inclusive a cadeia vazia) , desde que cada cadeia case com *r*

    Seja a expressão regular a*,

    $L(a^*) = L(a)^* = \{a\}^* = \{\varepsilon, a, aa, aaa, ...\}$

Exemplo:

Considere a expressão regular (a|bb)*

L(**(a|bb)***) = L(**a|bb**)* = {a, bb}* = {ε, a, bb, aa, abb, bba, bbbb, aaa, aabb, abba, bbaa, ...}

Expressões Regulares: Operações

- Precedência de operações
  - Repetição (maior)
  - Concatenação
  - Escolha (menor)

- Uso de parênteses
  - Quando queremos indicar uma precedência diferente, devemos utilizar parênteses
  - Exemplos:

    (a|b)c a escolha terá precedência sobre a concatenação

    (a|b)* a escolha terá precedência sobre a repetição

Exercício 1: quais são as linguagens geradas pelas expressões regulares anteriores?

Expressões Regulares: Nomes

- Nomes para expressões regulares
  - É útil simplificar a notação com nomes significativos para expressões regulares

Exemplo:

dígito dígito*

onde

dígito = 0|1|2|...|9

dizemos que o nome *dígito* é uma definição regular

Expressões Regulares: definição

- **Definição** de expressão regular
  - Uma expressão regular é uma das seguintes:
    1. Uma expressão regular básica, composta por um único caractere **a**, onde *a* pertence a um alfabeto ∑ de caracteres legais; o metacaractere ε; ou o metacaractere Φ
       - No 1º caso, L(**a**) = {a}
       - No 2º caso L(**ε**) = {ε}
       - No 3º caso L(**Φ**} = {}
    2. Uma expressão da forma *r*|*s*, onde *r* e *s* são expressões regulares

       $$L(\mathbf{r}|\mathbf{s}) = L(\mathbf{r}) \cup L(\mathbf{s})$$

    3. Uma expressão da forma *rs*, onde *r* e *s* são expressões regulares

       $$L(\mathbf{rs}) = L(\mathbf{r})L(\mathbf{s})$$

Expressões Regulares: definição

- **Definição** de expressão regular (cont.)
  1. Uma expressão da forma *r**, onde *r* é uma expressão regular

     $L(\mathbf{r}^*) = L(\mathbf{r})^*$
  2. Uma expressão da forma *(r)*, onde *r* é uma expressão regular

     $L((\mathbf{r})) = L(r)$

Os parênteses não modificam a linguagem, são utilizados apenas para ajustar a precedência dos operadores

Expressões Regulares: exemplo

Σ = {a, b, c}, ER para cadeias que contêm somente um b

(a|c)*b(a|c)*

Exercício 2:

a) Significa que b está sempre no meio da expressão?

b) Dê exemplos de cadeias que casem com essa ER

c) Faça uma ER para representar conjunto de cadeias de caracteres que contêm no máximo um b

Expressões Regulares: extensões

- Operações que estendem o conjunto básico
  - Uma ou mais repetições
    - Seja *r* uma expressão regular, *r+* indica uma ou mais repetições de *r*
    - Impede a cadeia vazia ε
  - Qualquer caractere
    - Casamento com qualquer caractere do alfabeto
    - Denotado pelo metacaractere ponto (.)
    - Ex.: cadeias que contêm ao menos um b: .***b***.*

Expressões Regulares: extensões

- Operações que estendem o conjunto básico

  - Intervalo de caracteres
    - Utiliza-se os metacaracteres colchetes e hífen

      Exemplos: [a – z] [0 – 9] [a-zA-Z] [abc]

      [abc] é equivalente a a|b|c

      Essa notação com colchetes é chamada de classe de caracteres

  - Qualquer caractere fora de um conjunto
    - Para excluir um ou mais caracteres de um conjunto
    - Utiliza-se os metacaracteres til (~) ou circunflexo (^), que indicam "não" ou o complemento

      Exemplos: *[^a]* expressão regular para um caractere no alfabeto que não seja *a*

      ~(a|b|c) ou [^abc] qualquer caractere que não seja a, b ou c

Expressões Regulares: extensões

- Operações estendidas
  - Subexpressões opcionais
    - Cadeias com partes opcionais
    - Utiliza-se o metacaractere interrogação (?)
    - *r?* indica que as cadeias que casam com *r* são opcionais

    Exemplo:
    natural = [0-9]+
    naturalComSinal = (+|-)? *natural*

Expressões Regulares: extensões

- Operações estendidas
  - Caractere de escape
    - Utiliza-se o metacaractere barra invertida "\" seguido do caractere literal que pretende-se especificar
    - O caractere de escape permite que um metacaractere seja interpretado de forma literal

    Exemplo: r = [0-9]\.[0-9]

    *L(**r**) = {0.0, 0.1, 0.2..., 1.0, 1.1, 1.2...}*

ERs em Linguagens de Programação

- Usadas para especificar marcas
  - Categorias de marcas comuns: palavras reservadas (ou chave), símbolos especiais, identificadores, literais ou constantes (ex. constantes numéricas)
  - Exemplos:
    - Números: sequências de dígitos, com ou sem casa decimal e com ou sem expoente

      natural = [0-9]+

      naturalSinal = (+|-)?natural

      número = naturalSinal(“.”natural)?(E naturalSinal)?

      ↓

      Para diferenciar do metacaractere ponto

ERs em Linguagens de Programação

- Exemplos:
  - Palavras reservadas: são sequências fixas de caracteres
    reservadas = if|while|do|...

  - Identificadores: são sequências de caracteres não fixas, que devem iniciar com letra e depois conter apenas letras e dígitos
  - letra = [a-zA-Z]
  - dígito = [0-9]
  - identificador = letra(letra|dígito)+

ERs em Linguagens de Programação

- Exemplos:
  - Comentários: devem ser reconhecidos e descartados
  - É simples escrever uma ER para comentários de delimitadores únicos
  - Ex.: Pascal – comentários entre chaves
  - {(~})*}
  - Mas é difícil escrever ER para delimitadores com mais de um caractere
  - Ex.: C – comentários entre /* e */
  - Normalmente é tratado por métodos *ad hoc* nos sistemas de varredura

Expressões Regulares

- Ambiguidade
  - Ocorre quando uma cadeia pode casar com mais do que uma expressão regular
  - Formas de resolver
    - Palavras reservadas
      - Deve haver uma expressão regular para cada palavra reservada
      - Exemplos: **if** **else** **do** **while**
      - Ou então a interpretação de palavra-chave é preferida sobre a de identificador (por isso o nome palavra reservada)
    - Princípio da subcadeia mais longa
      - A expressão regular que casa com a cadeia mais longa deve ser adotada
      - Exemplos: **>=** **==** **<=** **<>** **!=**

Expressões Regulares: delimitadores

- Delimitadores de Lexemas
  - Espaço em branco
  - Caractere de tabulação
  - Caractere de mudança de linha
  - Caracteres que não casam com a expressão regular em análise
  - Exemplos
    - x = 10 (delimitados por espaços em branco)
    - x=10 (delimitados pelo caractere =)
    -

Expressões Regulares: delimitadores

- Verificação à frente: às vezes é necessário voltar um ou mais caracteres
- Ex.: x=10 → x é delimitado pelo caractere =, que deve ser reconhecido

Expressões Regulares: observações

- O conjunto de cadeias que é linguagem para uma ER é chamado conjunto regular
  - Linguagem do tipo 3 na hierarquia de Chomsky

- Ocasionalmente pode haver conjuntos não regulares como cadeias válidas em linguagens de programação
  - São tratados de forma diferenciada na varredura

Expressões Regulares: curiosidade

Testador de expressão regular

Regex Tester: http://regexpal.com

Expressões Regulares

- Bibliografia consultada

Capítulo 2 LOUDEN, K. C. **Compiladores: princípios e práticas.** São Paulo: Pioneira Thompson Learning, 2004