

O que é RegEx e qual a importância dela para a linguagem de programação?

Uma expressão regular é um método formal de se especificar um padrão de texto. Mais detalhadamente, é uma composição de símbolos, caracteres com funções especiais, que, agrupados entre si e com caracteres literais, formam uma sequência, uma expressão. Essa expressão é interpretada como uma regra, que indicará sucesso se uma entrada de dados qualquer casar com essa regra, ou seja, obedecer exatamente a todas as suas condições. Servem para você dizer algo abrangente de forma específica. Definido seu padrão de busca, você tem uma lista (finita ou não) de possibilidades de casamento.

A origem das expressões regulares está na teoria dos autômatos e na teoria das linguagens formais, e ambas fazem parte da teoria da computação. Esses campos estudam modelos de computação (autômatos) e formas de descrição e classificação de linguagens formais. Na década de 1950, o matemático Stephen Cole Kleene descreveu tais modelos usando sua notação matemática chamada de "conjuntos regulares", formando a álgebra de Kleene. A linguagem SNOBOL foi uma implementação pioneira de casamento de padrões, mas não era idêntica às expressões regulares. Ken Thompson construiu a notação de Kleene no editor de texto QED como uma forma de casamento de padrões em arquivos de texto. Posteriormente, ele adicionou essa funcionalidade no editor de texto Unix `ed`, que resultou no uso de expressões regulares na popular ferramenta de busca `grep`. Desde então, diversas variações da adaptação original de Thompson foram usadas em Unix e derivados, incluindo `expr`, `AWK`, `Emacs`, `vi` e `lex`.

As expressões regulares de Perl e Tcl foram derivadas da biblioteca escrita por Henry Spencer, e no Perl a funcionalidade foi expandida posteriormente.[1] Philip Hazel desenvolveu a PCRE (Perl Compatible Regular Expressions), uma biblioteca usada por diversas ferramentas modernas como PHP e o servidor Apache. Parte do desenvolvimento do Perl 6 foi melhorar a integração das expressões regulares de Perl, e aumentar seu escopo e funcionalidade para permitir a definição de gramáticas de expressão de analisadores sintáticos.[2] O resultado foi uma mini-linguagem, as regras do Perl 6, usada para definir a gramática do Perl 6 assim como fornecer uma ferramenta para programadores da linguagem. Tais regras mantiveram as funcionalidades de expressões regulares do Perl 5.x, mas também permitiram uma definição BNF de um analisador sintático descendente recursivo.

O uso de expressões regulares em normas de informação estruturada para a modelagem de documentos e bancos de dados começou na década de 1960, e expandiu na década de 1980 quando normas como a ISO SGML foram consolidadas.

Expressões regulares são usadas por diversos editores de texto, utilitários e linguagens de programação para procurar e manipular texto baseado em padrões. Por exemplo, Perl e Tcl possuem suporte a expressões regulares nativamente. Diversos utilitários de distribuições Unix incluem o editor de texto `ed`, que popularizou o conceito de expressão regular, e o filtro `grep`. Outro uso é a validação de formatos de texto (validação de protocolos ou formatos digitais). Por exemplo, ao receber a entrada dum campo de formulário numa aplicação que supõe receber um endereço de email, pode-se usar uma expressão regular para garantir que o que foi recebido de fato é um endereço de email. Mais um uso é a implementação interna dum sistema de realce de sintaxe, como encontrado em ambientes de desenvolvimento integrado. Expressões regulares podem ser usadas para encontrar palavras reservadas, literais e outros tokens específicos, e para alterar a formatação do texto de acordo com o casamento feito.