

Manual do Editor de Expressões Regulares

Jesper K. Pedersen
Tradução: Marcus Gama



Manual do Editor de Expressões Regulares

Conteúdo

1	Introdução	5
2	O Que é uma Expressão Regular	6
3	Usando o Editor de Expressões Regulares	9
3.1	A organização do tela	9
3.2	Ferramentas de Edição	10
3.2.1	Ferramenta de Seleção	10
3.2.2	Ferramenta de Texto	10
3.2.3	Ferramenta de Caracteres	10
3.2.4	Ferramenta para Qualquer Caractere	11
3.2.5	Ferramenta de Repetição	11
3.2.6	Ferramenta de Alternativas	11
3.2.7	Ferramenta de Composição	12
3.2.8	Ferramentas do Início/Fim da Linha	12
3.2.9	Ferramentas de (Não-)Limite das Palavras	12
3.2.10	Ferramentas de Leitura Antecipada Positiva/Negativa	13
3.3	Expressões Regulares Definidas pelo Usuário	13
4	Relatando erros e Sugerindo Funcionalidades	14
5	Perguntas frequentes	15
5.1	O editor de expressões regulares suporta referências anteriores?	15
5.2	O editor de expressões regulares suporta a apresentação das ocorrências?	15
5.3	Eu sou o autor de um programa do KDE, como eu posso usar este item na minha aplicação?	15
5.4	Não consigo encontrar o botão <i>Editar a Expressão Regular</i> numa instalação do KDE 3 ou no 'konqueror', por exemplo. Porquê?	15
6	Créditos e licença	16

Resumo

Este manual descreve o componente gráfico do Editor de Expressões Regulares

Capítulo 1

Introdução

O editor de expressões regulares é um editor para criar uma expressão regular de uma forma gráfica (em contraste com a sintaxe em ASCII). Tradicionalmente, as expressões regulares têm sido escritas com a sintaxe de ASCII, a qual se assemelha, por exemplo, a $\hat{.}^*kde\backslash b$. A maior desvantagem deste estilo é:

- É difícil de compreender pelos não-programadores.
- Obriga a que você *escape* ou *comente* alguns símbolos (para corresponder com um asterisco, por exemplo, você tem que escrever \backslash^*).
- Obriga a que você se recorde das regras para a *precedência* (A quem corresponde o $x|y^*$? A um único x ou um conjunto de ys , *OU*, por sua vez, um conjunto de xs e ys misturados?)

O editor de expressões regulares, por outro lado, permite-lhe *desenhar* a sua expressão regular de uma forma não-ambígua. O editor resolve, pelo menos, os itens 2 e 3 descritos acima. Ele pode não tornar as expressões regulares disponíveis para os não-programadores, ainda que só os testes pelos usuários possam confirmar isso. Por isso, se você for um não-programador e tiver ganho o poder das expressões regulares com este editor, [comunique-me esse fato](#).

Capítulo 2

O Que é uma Expressão Regular

A expressão regular é uma forma de especificar as *condições* a cumprir para uma determinada situação em mente. Normalmente, quando você procura num editor de texto, você indica o texto a procurar *literalmente*; se usar expressões regulares, por outro lado, você diz como a ocorrência poderá se parecer. Alguns exemplos disto incluem *Estou à procura da palavra KDE, mas só no início da linha* ou *Estou à procura da palavra a, mas tem que existir isolada* ou ainda *Estou à procura dos arquivos que comecem pela palavra teste, seguida de um conjunto de algarismos, como por exemplo teste12, teste107 e teste007*

Você constrói as expressões regulares a partir expressões menores, tal como você constrói os brinquedos grandes de Lego a partir de peças pequenas. Tal como no mundo do Lego, também existe um conjunto de peças básicas. A seguir será descrita cada uma dessas “peças de construção” usando um conjunto de exemplos.

Example 2.1 Procurando por texto normal.

Se você quiser procurar apenas um determinado texto, então a expressão regular não é de fato a melhor escolha. A razão para isso é que as expressões regulares atribuem um significado especial a alguns caracteres. Isto inclui os seguintes caracteres: `. * | $`. Por isso, se quiser procurar pelo texto `kde.` (i.e. os caracteres `kde` seguidos de um ponto), então você teria de indicar isto assim: `kde\.`¹ Se escrever `\.` em vez de `.` é chamado de *escape*.

Example 2.2 Correspondência com URLs

Quando você seleciona algo que se assemelha a um URL no KDE, aí o programa **klipper** se oferecerá para iniciar o **konqueror** com o URL selecionado.

O **Klipper** faz isto ao fazer uma correspondência do texto selecionado ou da área de transferência com várias expressões regulares diferentes; quando uma das expressões regulares fizer correspondência, o comando correspondente será disparado.

A expressão regular para os URLs diz (entre outras coisas), que o texto deverá começar por `http://`. Isto é descrito com base nas expressões regulares antecedendo o texto `http://` com um acento circunflexo (o caracter `^`).

O exemplo acima serve para marcar as posições com expressões regulares. De forma semelhante, a posição do *fim de linha* poderá ser correspondida com o `$` (isto é, um símbolo de cifrão).

Example 2.3 Procurando pela palavra *a*, mas não por *aí*, *andar* ou outras

Dois tipos adicionais de posições podem ser correspondências na forma acima, nomeadamente *a posição no limite de uma palavra* e *a posição no limite de uma não-palavra*. As posições são indicadas usando o texto `\b` (para os limites das palavras), e o `\B` (para os limites de sequências que não são palavras)

Assim, procurando pela palavra *a*, pode fazê-lo com a expressão regular `\ba\b`. Isto indica que você está procurando a palavra *a* sem nenhuma letra de cada lado dela (isto é, com o limite de uma palavra de cada lado)

As quatro de expressões regulares de correspondências de posições são introduzidas no editor de expressões regulares usando a [ferramenta de quatro posições diferentes](#)

Example 2.4 Procurando tanto por isto como por aquilo

Imagine que deseja percorrer o seu documento à procura tanto da palavra *isto* como da palavra *aquilo*. Com um método de procura normal, você conseguiria fazer em duas tentativas, sendo que na primeira iria procurar por *isto* e na segunda iria procurar por *aquilo*.

Ao usar as pesquisas usando as expressões regulares, você iria procurar por ambas as sequências de uma vez. Você faria isso ao procurar por `isto|aquilo`, isto é, separando as sequências por uma barra vertical.²

No editor de expressões regulares você não escreve por si próprio a barra vertical, mas sim seleciona a [ferramenta de alternativas](#), introduzindo as pequena expressões regulares uma por cima da outra.

Example 2.5 Corresponder a tudo

As expressões regulares são frequentemente comparadas com os caracteres especiais existentes também na linha de comando - esta é a capacidade de indicar um conjunto de arquivos com o asterisco. Você irá provavelmente reconhecer a correspondência com os caracteres especiais a partir dos seguintes exemplos:

- `ls *.txt` - aqui o `*.txt` corresponderá a todos os arquivos que terminem na extensão `.txt`.
- `cat teste??.res` - corresponde a todos os arquivos que comecem por `teste`, seguido de dois caracteres quaisquer, terminando finalmente em `.res`

Na linha de comando o asterisco corresponde a qualquer caracter, segundo uma quantidade qualquer. Em outras palavras, o asterisco corresponde a *tudo*. Isto é escrito com o formato `.*` na sintaxe das expressões regulares. O ponto corresponde a um caracter qualquer, individualmente, isto é, a apenas *um* caracter, enquanto que o asterisco indica que a expressão anterior a ele deverá ter uma correspondência repetida por várias vezes. Em conjunto, isto diz que corresponde a qualquer conjunto de caracteres arbitrários com tamanho também arbitrário.

Isto poderá parecer mais ou menos complicado, mas quando tiver uma ideia global, irá ver o seu poder. Deixe-me mostrar outra expressão regular básica: `o e`. A letra `e` por si só é uma expressão regular que corresponde a uma única letra, nomeadamente a letra `e`. Se combinar isto com o asterisco, isto é, `e*`, então obter-se-á uma expressão regular que irá corresponder a uma quantidade arbitrária de `e`'s.

Podem-se combinar várias expressões regulares, umas após às outras, como por exemplo `ba(na)*`.³ Imagine que você escreveu esta expressão regular no campo de procura de um editor de texto; nesse caso iria obter as seguintes palavras (entre outras): `ba`, `bana`, `banana`, `banananananana`

Fornecida a informação acima, espera-se que não seja difícil para você escrever a sequência da linha de comando `teste??.res` como uma expressão regular. A resposta é: `teste.\.res`. O ponto por si só significa qualquer caracter. Para corresponder a um único ponto você precisa de escrever `\.`⁴. Em outras palavras, a expressão regular `\.` corresponde a um ponto, enquanto que um ponto sozinho corresponde a qualquer caracter.

No editor de expressões regulares, a expressão regular repetida é criada através da [ferramenta de repetição](#)

Example 2.6 Substituir o `&` por `&` num documento de HTML

No HTML, o caracter especial `&` deverá ser escrito como `&` - isto assemelha-se a 'escapar' nas expressões regulares.

Imagine que você escreveu um documento em HTML num editor normal (p.ex., o XEmacs ou o Kate), e se esqueceu por completo desta regra. O que você teria de fazer ao descobrir o seu erro seria substituir todas as ocorrências de `&` por `&`;

Isto pode ser feito facilmente usando a procura e substituição normais, só que existe, contudo, um pormenor. Imagine que você se lembrou de fato desta regra - *só um bocadinho* - e que resultava em alguns pontos. A substituição incondicional iria resultar em que o `&` seria substituído por `& amp;`

O que você queria mesmo dizer era que o `&` só deveria ser colocado se *não* fosse seguido pelas letras `amp;`. Você pode fazer isto usando expressões regulares com *leitura antecipada positiva*.

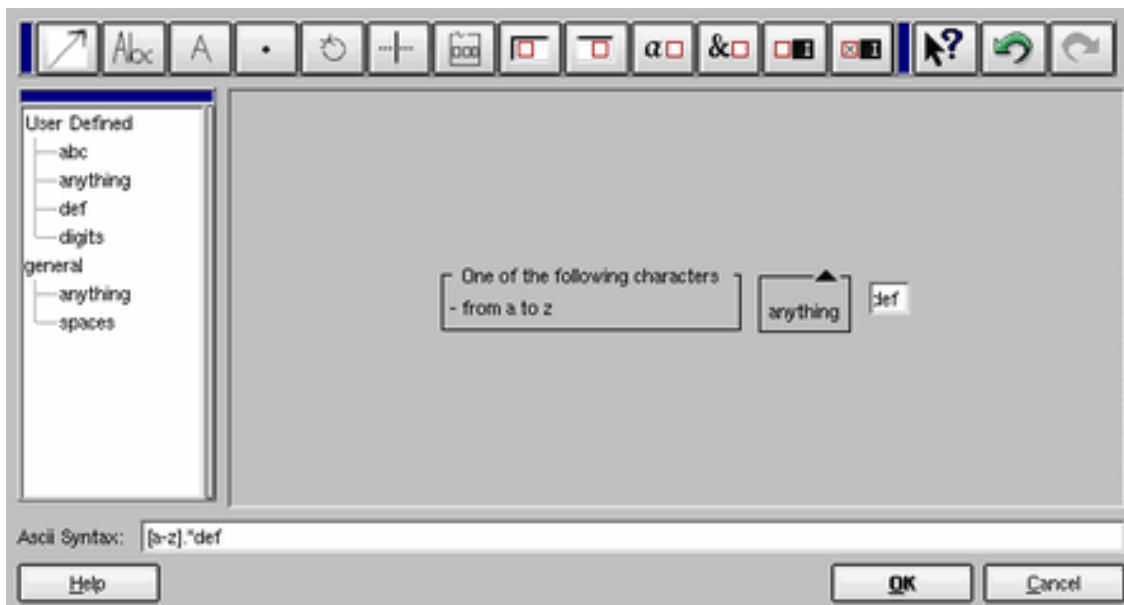
A expressão regular que só faria corresponder o 'e comercial' se não for seguido das letras `amp;` assemelha-se ao seguinte: `&(?!amp;)`. Isto é, obviamente, mais simples de ler usando o editor de expressões regulares, onde você iria utilizar as [ferramentas de leitura antecipada](#).

Capítulo 3

Usando o Editor de Expressões Regulares

Este capítulo explicará como o editor de expressões regulares funciona.

3.1 A organização do tela



A componente mais importante do editor é, como era de se esperar, a área de edição; esta é a área onde você desenha a sua expressão regular. A área em questão é a zona acinzentada no meio.

Acima da área de edição existem duas barras de ferramentas, onde a primeira contém as [ações de edição](#) - muito parecidas com as ferramentas de desenho num programa de desenho. A segunda barra de ferramentas contém o botão de *O Que é Isto?*, assim como os botões para desfazer e refazer.

Abaixo da área de edição é possível encontrar a expressão regular construída atualmente, descrita na sintaxe denominada ASCII. A sintaxe ASCII é atualizada enquanto você edita a expressão regular no editor gráfico. Se você quiser, por outro lado, atualizar a sintaxe ASCII fique à vontade para o fazê-lo; nesse caso, o editor gráfico atualiza-se para refletir as suas alterações.

Finalmente, à esquerda da área de edição, você encontra um conjunto de expressões regulares pré-concebidas. Elas servem para dois fins: (1) Quando você carregar o editor com uma expressão regular, então esta é tornada *mais bonita* ou mais compreensível se substituir as expressões regulares comuns. No resultado da tela acima, você pode ver como a sintaxe ASCII foi substituída por uma caixa que diz "tudo". (2) Quando você insere expressões regulares, poderá encontrar um conjunto de peças de construção para a sua própria expressão regular a partir do conjunto de expressões regulares predefinidas. Veja a seção sobre as [expressões regulares definidas pelo usuário](#) para aprender como salvar as suas próprias expressões regulares.

3.2 Ferramentas de Edição

O texto nesta seção espera que você tenha lido o capítulo sobre [o que é uma expressão regular](#) ou que tenha algum conhecimento prévio sobre o assunto.

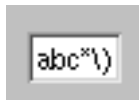
Todas as ferramentas de edição estão localizadas na barra de ferramentas acima da área de edição. Cada uma delas será descrita a seguir.

3.2.1 Ferramenta de Seleção



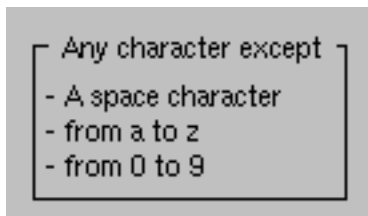
A ferramenta de seleção é usada para marcar os elementos para as operações de cortar-e-colar e de arrastar-e-soltar. Isto é muito semelhante à ferramenta de seleção num programa de desenho.

3.2.2 Ferramenta de Texto



Se usar esta ferramenta você irá introduzir algum texto normal com o qual irá fazer a correspondência. O texto é correspondido literalmente, isto é, você não terá que escapar os caracteres especiais. No exemplo acima, a seguinte expressão regular será criada: `abc*\\)`

3.2.3 Ferramenta de Caracteres



Se usar esta ferramenta, você irá introduzir gamas de caracteres. Os exemplos incluem que o texto ASCII diz `[0-9]` e `[^a-zA-Z, _]`. Ao introduzir um item com esta ferramenta, aparecerá uma janela na qual você poderá indicar as gamas ou intervalos de caracteres.

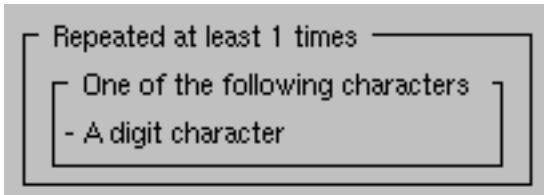
Veja a descrição das [expressões regulares repetidas](#).

3.2.4 Ferramenta para Qualquer Caractere



Esta é a expressão regular do "ponto" (.). Ela aceita qualquer caractere isolado.

3.2.5 Ferramenta de Repetição



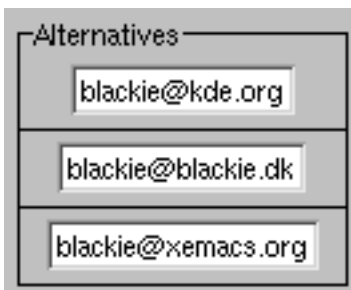
Isto corresponde aos elementos repetidos. Isto inclui o que, na sintaxe ASCII, é representado usando um asterisco (*), um sinal de mais (+), um ponto de interrogação (?) e os intervalos ({3,5}). Quando introduzir um item com esta ferramenta, aparecerá uma janela que irá perguntar quantas vezes ocorrerá a repetição.

Você especifica o que deseja repetir ao desenhar o conteúdo repetido dentro da caixa que esta ferramenta introduz.

Os elementos repetidos podem tanto ser construídos de dentro para fora como vice-versa. Isto significa que você pode desenhar primeiro o que será repetido, selecioná-lo e usar a ferramenta de repetição para o replicar ou, em alternativa, pode indicar primeiro o número de repetições e desenhar o que se pretende repetir dentro dele.

Veja a descrição sobre as [expressões regulares repetidas](#).

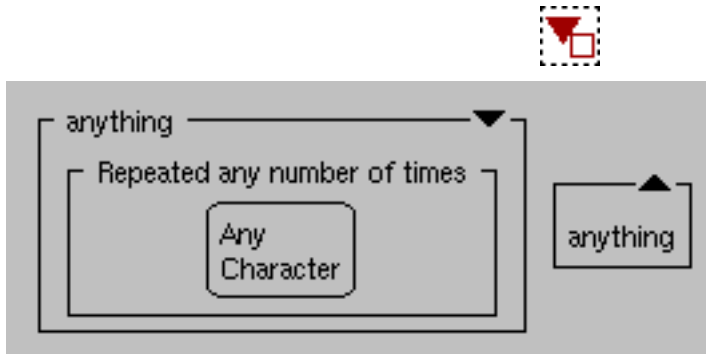
3.2.6 Ferramenta de Alternativas



Esta é a expressão regular para as alternativas (|). Você indica as alternativas quando desenha cada uma delas por cima da anteriores dentro da caixa que esta ferramenta introduz.

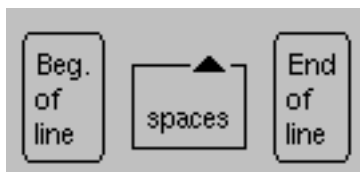
Veja a descrição sobre as [expressões regulares para as alternativas](#)

3.2.7 Ferramenta de Composição



A ferramenta de composição não representa nenhuma expressão regular. É usada para agrupar outros subcomponentes em conjunto dentro de uma caixa, que poderá facilmente ser fechada, restringindo-se apenas ao seu título. Isto poderá ser visto na parte direita do resultado da tela acima.

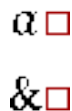
3.2.8 Ferramentas do Início/Fim da Linha



As ferramentas do início e do fim de linha correspondem, respectivamente ao início da linha ou ao seu fim, como seria de esperar. A expressão regular no resultado da tela acima corresponde deste modo às linhas que só tenham espaços.

Veja a descrição sobre as [expressões regulares de posição](#).

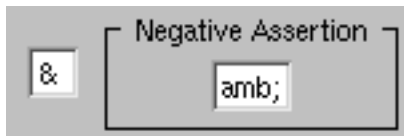
3.2.9 Ferramentas de (Não-)Limite das Palavras



As ferramentas dos limites fazem a sua correspondência com os limites das palavras ou das outras sequências que não representem palavras, respectivamente. A expressão regular no resultado da tela acima corresponderá às palavras que comecem por the. A palavra the em si, todavia, não fará correspondência.

Veja a descrição sobre as [expressões regulares dos limites](#).

3.2.10 Ferramentas de Leitura Antecipada Positiva/Negativa



As ferramentas de leitura antecipada tanto poderão corresponder a expressões positivas ou negativas. A correspondência não faz, contudo, parte da correspondência total.

Nota: Só é permitido colocar leituras antecipadas no fim das expressões regulares. O Editor de Expressões Regulares não obriga a isto.

Veja a descrição sobre as [expressões regulares com leitura antecipada](#).

3.3 Expressões Regulares Definidas pelo Usuário

À esquerda a área de edição existe uma lista que contém as expressões regulares definidas pelo usuário. Algumas das expressões regulares vêm incluídas na sua instalação do KDE, enquanto que outras poderão ser gravadas por você.

Estas expressões regulares servem dois propósitos ([ver uma descrição detalhada](#)), nomeadamente (1) oferecer ao usuário um conjunto de peças de construção e (2) tornar as expressões regulares comuns mais bonitas.

Você poderá gravar as suas próprias expressões regulares ao clicar com o botão direito do mouse na área de edição e ao escolher a opção Salvar a Expressão Regular.

Se a expressão regular que você salvar estiver dentro de um [bloco de composição](#), então a expressão regular irá ajudar a tornar as expressões regulares subsequentes mais bonitas.

A expressão regular definida pelo usuário pode ser removida ou ver o seu nome alterado ao clicar com o botão direito na lista, mais precisamente em cima da expressão regular em questão.

Capítulo 4

Relatando erros e Sugerindo Funcionalidades

Os erros e funcionalidades comunicados são submetidos no [Sistema de Registro de Erros do KDE](#). **Antes** de comunicar um erro ou sugerir uma funcionalidade, verifique primeiro se esta não foi já comunicada/sugerida.

Capítulo 5

Perguntas frequentes

5.1 O editor de expressões regulares suporta referências anteriores?

Não, atualmente isto não é suportado. Está planejado para a próxima versão.

5.2 O editor de expressões regulares suporta a apresentação das ocorrências?

Não, mas esperamos que isto esteja disponível na próxima versão.

5.3 Eu sou o autor de um programa do KDE, como eu posso usar este item na minha aplicação?

Veja na [documentação da classe KRegExpEditorInterface](#).

5.4 Não consigo encontrar o botão *Editar a Expressão Regular* numa instalação do KDE 3 ou no 'konqueror', por exemplo. Porquê?

O elemento de expressões regulares está localizado no pacote 'KDE-utils'. Se você não tiver este pacote instalado, então os botões para *editar expressões regulares* não irão aparecer nos programas.

Capítulo 6

Créditos e licença

Direitos autorais da documentação de 2001, Jesper K. Pedersen blackie@kde.org

Tradução de Marcus Gama marcus.gama@gmail.com

Esta documentação é licenciada sob os termos da [Licença de Documentação Livre GNU](#).

Este programa é licenciado sob os termos da [Licença Pública Geral GNU](#).