

Manuel de l'éditeur d'expressions rationnelles

Jesper K. Pedersen
Traduction française : Simon Depiets



Manuel de l'éditeur d'expressions rationnelles

Table des matières

1	Introduction	5
2	Qu'est-ce qu'une expression rationnelle ?	6
3	Utilisation de l'éditeur d'expressions rationnelles	9
3.1	L'organisation de l'écran	9
3.2	L'outil d'édition	10
3.2.1	L'outil de sélection	10
3.2.2	L'outil de texte	10
3.2.3	L'outil des caractères	10
3.2.4	L'outil n'importe quel caractère	11
3.2.5	L'outil de répétition	11
3.2.6	L'outil alternatif	11
3.2.7	L'outil de composition	12
3.2.8	Outils de début et de fin de lignes	12
3.2.9	Outils de limite	12
3.2.10	Outils de recherche d'expression rationnelle positive / négative	13
3.3	Expressions rationnelles définies par l'utilisateur	13
4	Rapport de bogues et suggestions de fonctionnalités	14
5	Foire aux questions	15
5.1	Est-ce que l'éditeur d'expressions rationnelles gère les références arrières?	15
5.2	Est-ce que l'éditeur d'expressions rationnelles gère les opérations de reconnaissance d'affichage?	15
5.3	Je suis l'auteur d'un programme KDE. Comment puis-je utiliser ce composant graphique dans mon application?	15
5.4	Je n'arrive pas à trouver le bouton <i>éditeur d'expressions rationnelles</i> dans konqueror sur une autre installation de KDE3 par exemple, pourquoi?	15
6	Remerciements et licence	16

Résumé

Ce manuel décrit le composant graphique pour l'édition d'expression rationnelle.

Chapitre 1

Introduction

L'éditeur d'expressions rationnelles est un éditeur pour modifier des expressions rationnelles de manière graphique (à l'inverse de la syntaxe ASCII). Traditionnellement, les expressions rationnelles sont écrites avec une syntaxe ASCII, ressemblant par exemple à $\hat{.}^*kde\backslash b$. Les inconvénients majeurs de ce style sont :

- il est difficilement compréhensible pour les non informaticiens ;
- il est nécessaire que vous *échappiez* certains symboles (pour trouver une étoile, par exemple, il vous faut écrire \backslash^*);
- il est nécessaire que vous vous souveniez des règles de *précédence* (à quoi correspond $x|y^*$? Un seul x ou un certain nombre de y , *OU* un certain nombre de x et de y mélangés?).

L'éditeur d'expressions rationnelles, d'un autre côté, vous permet de *dessiner* vos expressions rationnelles d'une manière non ambiguë. L'éditeur résout au moins les éléments deux et trois ci-dessus. Il ne permet pas aux non programmeurs d'accéder aux expressions rationnelles, bien que seuls des tests par des utilisateurs peuvent le dire. Donc, si vous n'êtes pas un programmeur, et que vous avez terrassé la puissance des expressions rationnelles par l'intermédiaire de cet éditeur, alors [faites-le moi savoir](#) s'il vous plaît.

Chapitre 2

Qu'est-ce qu'une expression rationnelle ?

Les expressions rationnelles sont une manière de spécifier des *conditions* devant être remplies en pensant à une situation. Normalement, lorsque vous faites une recherche dans un éditeur de texte, vous spécifiez le texte à rechercher *littéralement*, à l'opposé, en utilisant les expressions rationnelles, vous décrivez à quoi ressemble un élément donné. Les exemples illustrant cela incluent *je recherche le mot KDE, mais uniquement en début de ligne*, ou *je recherche le mot le, mais entouré d'espace*, ou bien encore *je cherche les fichiers commençant par le mot test, suivi par des numéros, par exemple test12, test107 et test007*.

Vous construisez des expressions rationnelles à partir d'expressions rationnelles plus petites, exactement comme vous construisez un immense jouet en Lego à partir de sous-parties plus petites. De même que dans le monde des Lego, il existe un certain nombre de blocs de construction de base. Dans les chapitres suivants, je décrirai chacun de ces blocs de construction de base en illustrant par un certain nombre d'exemples.

Exemple 2.1 Recherche d'un texte normal

Si vous désirez rechercher un texte donné, les expressions rationnelles ne sont vraiment pas un bon choix. La raison à cela est que les expressions rationnelles associent des significations particulières à certains caractères. Ce qui comprend les caractères suivants : `. * | $`. Ainsi, si vous désirez chercher le texte `kde.` (c'est-à-dire les caractères `kde` suivi d'un point), alors vous devrez spécifier cela de cette manière `kde\.`¹ Écrire `\.` au lieu d'un simple `.` est appelé *échappement*.

Exemple 2.2 Faire correspondre des URL

Lorsque vous sélectionnez quelque chose ressemblant à une URL dans KDE, alors le programme **klipper** proposera de lancer **konqueror** avec l'URL sélectionnée.

Klipper arrive à faire cela en associant la sélection à plusieurs expressions rationnelles différentes, lorsqu'une des expressions rationnelles correspond, la commande adaptée est proposée. L'expression rationnelle pour les URL spécifie (entre autres choses), que la sélection doit commencer par le texte `http://`. Cela est spécifié dans les expressions rationnelles en préfixant le texte `http://` par un chapeau (le caractère `^`).

La paragraphe ci-dessus est un exemple pour faire des reconnaissances par rapport à la position en utilisant des expressions rationnelles. De façon identique, la position *fin-de-ligne* peut être associée en utilisant le caractère `$` (c'est-à-dire le signe du dollar).

Exemple 2.3 Rechercher le mot `le`, mais pas `leur`, `souffle` ou `raleur`.

Deux types de positions supplémentaires peuvent être utilisés dans le paragraphe ci-dessus, nommé *placé en limite de mot* et *non placé en limite de mot*. Les positions sont spécifiées en utilisant le texte `\b` (pour *word-boundary*, limite de mot) et `\B` (pour *non-word boundary*, non-limite de mot)

Ainsi, la recherche du mot `le` peut être faite en utilisant l'expression rationnelle `\ble\b`. Ce qui spécifie que nous recherchons `le` sans lettres de chaque côté (c'est-à-dire avec une limite de mot de chaque côté).

La quatrième expression rationnelle pour des mises en correspondances en fonction de la position sont insérées dans l'éditeur d'expressions rationnelles en utilisant [l'outil des quatre positions différentes](#).

Exemple 2.4 Rechercher soit `ceci` soit `cela`.

Imaginez que vous désiriez parcourir votre document pour rechercher soit le mot `ceci` soit le mot `cela`. Avec une méthode de recherche normale, vous pouvez faire cela en deux passes : lors du premier parcours, vous recherchiez `ceci`, et la seconde fois vous recherchiez `cela`.

En utilisant la recherche avec les expressions rationnelles, vous rechercherez les deux dans la même passe. Vous pouvez réaliser cela en cherchant `ceci|cela`, c'est-à-dire en séparant les deux mots par une barre verticale.²

Dans l'éditeur d'expressions rationnelles, vous ne devez pas écrire la barre verticale vous-même, mais vous devez sélectionner plutôt [l'outil alternatif](#), et insérer les expressions rationnelles plus petites de chaque côté.

Exemple 2.5 Reconnaître n'importe quel caractère

Les expressions rationnelles sont souvent comparées à la recherche par joker des consoles - c'est la possibilité de spécifier un certain nombre de fichiers par un astérisque. Vous reconnaîtrez encore plus la recherche par jokers avec les exemples suivants :

- `ls *.txt` - ici `*.txt` est le joker de la console pour trouver chaque fichier finissant par l'extension `.txt`.
- `cat test?.res` - trouve chaque fichier commençant par `test` suivi de deux caractères quelconques, et enfin suivi par `.res`

Dans une console, les astérisques correspondent à n'importe quel caractère répété plusieurs fois. En d'autres termes, les astérisques correspondent à *n'importe quel caractère*. Cela s'écrit ainsi `*` avec la syntaxe des expressions rationnelles. Le point correspond à un seul caractère quelconque, c'est-à-dire à *un* caractère, et l'astérisque spécifie que l'expression rationnelle le précédant devra être présente de multiples fois. Réunis ensemble, cela signifie un unique caractère quelconque répété un certain nombre de fois.

Cela peut sembler énormément compliqué, mais en ayant une vue plus globale, vous découvrirez toute sa puissance. Laissez-moi vous montrer une autre expression rationnelle de base : `a`. La lettre `a` isolée est une expression rationnelle qui correspond à une lettre unique, pour ne pas la nommer, la lettre `a`. Si nous combinons cela avec l'astérisque, c'est-à-dire `a*`, alors nous avons une expression rationnelle correspondant à plusieurs `a`.

Nous pouvons combiner plusieurs expressions rationnelles l'une après l'autre, par exemple `ba(na)*`.³ Imaginons que vous ayez écrit cette expression rationnelle dans la zone de recherche d'un éditeur de texte, alors vous pourrez trouver les mots suivants (parmi tant d'autres) : `ba`, `bana`, `banana`, `banananananana`

Après vous avoir donné l'information ci-dessus, il ne devrait plus être difficile pour vous de trouver le joker de la console `test?.res` dans une expression rationnelle. Réponse : `test.\.res`. Le point seul correspond à un seul caractère. Pour trouver un seul point, vous devez écrire `\.`⁴. En d'autres mots, l'expression rationnelle `\.` correspond à un point, alors qu'un point seul correspond à n'importe quel caractère.

Dans l'éditeur d'expressions rationnelles, la répétition d'une expression rationnelle est créée en utilisant l'[outil de répétition](#).

Exemple 2.6 Remplacer `&` par `&` dans un document HTML

En HTML, le caractère spécial `&` doit être écrit `&` ; - c'est la même chose que l'échappement dans les expressions rationnelles.

Imaginez que vous ayez écrit un document HTML dans un éditeur normal (par exemple XEmacs ou Kate), et que vous ayez totalement oublié cette règle. Ce qu'il vous faudra faire lorsque vous réaliserez votre erreur sera de remplacer chaque occurrence de `&` par `&` ; .

Cela est fait facilement en utilisant une recherche normale et le remplacer, mais, cependant, il y a un pépin. Imaginez que vous vous êtes souvenu de cette règle - *juste un peu* - et que vous en avez corrigés quelques uns. Les remplacer tous sans conditions aura pour conséquence de remplacer `&` ; par `&amp;` ; .

Ce que vous désirez vraiment faire c'est que les `&` soient uniquement remplacés s'ils *ne* sont *pas* suivis des lettres `amp ;` . Vous pouvez faire cela en utilisant les expressions rationnelles avec la *préanalyse positive*.

L'expression rationnelle, qui ne cherche qu'un `&` et commercial s'il n'est pas suivi des lettres `&ressemble à : &(!amp ;)`. Ce qui est, évidemment, plus facile à lire en utilisant l'éditeur d'expressions rationnelles, où vous utiliserez les [outils de préanalyse](#).

Chapitre 3

Utilisation de l'éditeur d'expressions rationnelles

Ce chapitre vous décrira comment l'éditeur d'expressions rationnelles fonctionne.

3.1 L'organisation de l'écran



La partie la plus importante de l'éditeur est évidemment la zone d'édition, qui est la zone où vous dessinerez l'expression rationnelle. Cette zone est la zone grise la plus large au milieu.

Au-dessus de la zone d'édition, vous pouvez voir deux barres d'outils. La première contient les [actions d'édition](#) (comme les outils de dessin dans un programme de dessin). La seconde barre d'outils contient les boutons *Qu'est-ce que c'est* et les boutons pour annuler et refaire les actions précédentes.

En dessous de la zone d'édition, vous trouverez les expressions rationnelles en cours de construction, dans la syntaxe dite ASCII. La syntaxe ASCII est mise à jour pendant l'édition de l'expression rationnelle dans l'éditeur graphique. Si vous désirez plutôt mettre à jour la syntaxe ASCII alors faites-le, l'éditeur graphique sera mis à jour à la volé pour refléter les changements.

Enfin, à gauche de la zone d'édition, vous trouverez un certain nombre d'expressions rationnelles prédéfinies. Elles ont deux buts : (1) Lorsque vous chargez l'éditeur avec une expression rationnelle alors cette expression rationnelle est affichée de façon *plus belle* ou plus compréhensible en remplaçant les expressions rationnelles classiques. Dans la copie d'écran ci-dessus, vous pouvez voir comment la syntaxe ASCII « . * » a été remplacée par une zone affichant « quelque chose ». (2) Lorsque vous insérez une expression rationnelle vous trouverez des blocs fabriqués pour vos propres expressions rationnelles dans l'ensemble des expressions rationnelles préfabriquées. Veuillez consulter le chapitre sur [les expressions rationnelles définies par l'utilisateur](#) pour apprendre comment enregistrer vos propres expressions rationnelles.

3.2 L'outil d'édition

Le contenu de ce chapitre nécessite que vous ayez lu le chapitre [Qu'est-ce qu'une expression rationnelle](#), ou que vous ayez déjà des connaissances sur ce sujet.

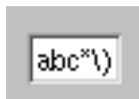
Tous les outils d'édition sont situés dans la barre d'outils au-dessus de la zone d'édition. Chacun d'entre eux sera décrit dans ce qui suit.

3.2.1 L'outil de sélection



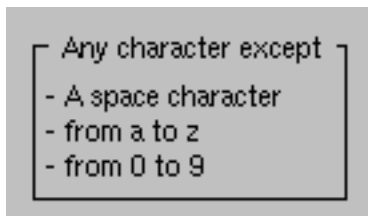
L'outil de sélection est utilisé pour marquer des éléments pour les couper-coller et pour les glisser-déposer. C'est un outil très similaire à un outil de sélection dans tout programme de dessin.

3.2.2 L'outil de texte



En utilisant cet outil, vous insérerez du texte normal à rechercher. Le texte est recherché littéralement, c'est-à-dire que vous n'avez pas à vous préoccuper d'échapper les caractères spéciaux. Dans l'exemple ci-dessous, l'expression rationnelle suivante sera construite : `abc*\)`

3.2.3 L'outil des caractères



En utilisant cet outil, vous insérerez des domaines de caractères. On peut citer comme exemple en texte ASCII `[0-9]`, `[^a-zA-Z, _]`. Lorsque vous insérez un élément avec cet outil, une boîte de dialogue apparaîtra, dans laquelle vous spécifierez les domaines de caractères.

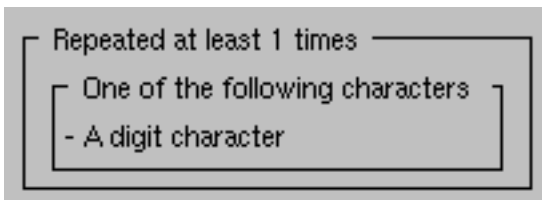
Consultez la description faite dans le chapitre [expressions rationnelles répétées](#).

3.2.4 L'outil n'importe quel caractère



Il correspond à l'expression rationnelle « point » (.). Il correspond à un unique caractère quelconque.

3.2.5 L'outil de répétition



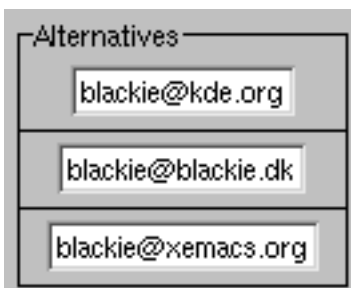
C'est l'élément de répétition. Cela inclut ce qui, en syntaxe ASCII, est représenté par l'astérisque (*), un plus (+), un point d'interrogation (?) et par des domaines ([3,5]). Lorsque vous insérez un élément en utilisant cet outil, une boîte de dialogue apparaît vous demandant le nombre de répétitions.

Vous spécifiez alors ce qu'il faut répéter en dessinant le contenu répété dans le rectangle que cet outil insère.

Des éléments répétés peuvent être à la fois construits à l'intérieur et à l'extérieur. C'est-à-dire que vous pouvez d'abord dessiner ce que vous voulez répéter, le sélectionner et utiliser l'outil de répétition pour spécifier la répétition. Ou bien, vous pouvez d'abord insérer l'élément de répétition, puis dessiner ce qui doit être répété à l'intérieur.

Consultez la description faite dans le chapitre [expressions rationnelle répétées](#).

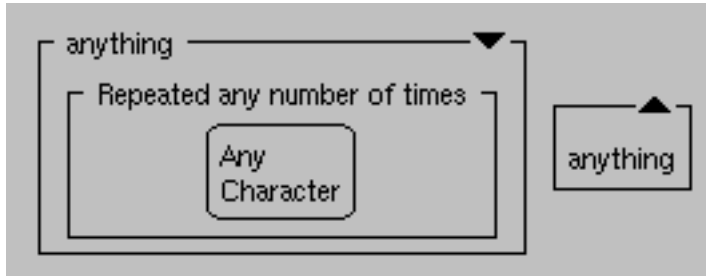
3.2.6 L'outil alternatif



Il correspond à l'expression rationnelle (|). Vous spécifiez les possibilités en dessinant chaque possibilité l'une au-dessus de l'autre à l'intérieur du rectangle que l'outil a inséré.

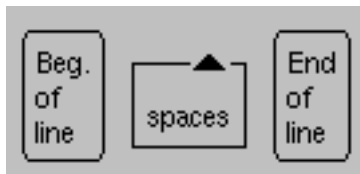
Consultez la description faite dans le chapitre [expressions rationnelles alternatives](#).

3.2.7 L'outil de composition



L'outil de composition ne représente aucune expression rationnelle. Il est utilisé pour regrouper plusieurs composants dans une boîte, qui peut être facilement réduite à son titre. Cela peut se voir dans la partie droite de la copie d'écran ci-dessus.

3.2.8 Outils de début et de fin de lignes



Les outils de début et de fin de lignes correspondent respectivement au début de ligne et à la fin de ligne. L'expression rationnelle dans la copie d'écran ci-dessous recherche donc les lignes composées d'espaces uniquement.

Consultez la description faite dans le chapitre [expressions rationnelle de positionnement](#)

3.2.9 Outils de limite



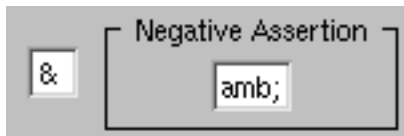
Les outils de limite correspondent à une limite de mot, respectivement à un contenu de mot. L'expression rationnelle dans la copie d'écran recherche donc des mots démarrant par le. Le mot le n'est, cependant, pas accepté.

Consultez la description faite dans le chapitre [expressions rationnelles de limite](#).

3.2.10 Outils de recherche d'expression rationnelle positive / négative

+I

-I



Les outils de recherche spécifient une expression rationnelle soit positive soit négative à rechercher.

Remarque : il vous est uniquement permis de placer la recherche à la fin d'une expression rationnelle. Le composant graphique pour l'édition d'expressions rationnelles ne l'impose pas.

Consultez la description faite dans le chapitre [expressions rationnelles de recherche arrière](#).

3.3 Expressions rationnelles définies par l'utilisateur

À gauche de la zone d'édition se situe une liste contenant les expressions rationnelles définies par l'utilisateur. Certaines expressions rationnelles sont préinstallées avec votre installation de KDE, alors que d'autres sont enregistrées par vous-même.

Ces expressions rationnelles ont deux buts ([consultez la description détaillée](#)), qui sont (1) de vous offrir un ensemble de blocs préfabriqués et (2) vous faciliter la création d'expressions rationnelles.

Il vous est possible d'enregistrer vos propres expressions rationnelles en cliquant avec le bouton droit de la souris dans la zone d'édition, et de choisir `Enregistrer l'expression rationnelle`.

Si l'expression rationnelle que vous enregistrez est à l'intérieur d'un [composant conteneur](#) alors elle prendra part plus facilement dans la création d'expressions rationnelles à venir.

Les expressions rationnelles définies par l'utilisateur peuvent être effacées ou renommées en cliquant avec le bouton droit de la souris au-dessus de l'expression rationnelle en question dans la zone contenant la liste.

Chapitre 4

Rapport de bogues et suggestions de fonctionnalités

Les rapports de bogues et de fonctionnalités sont rapportés par le [système de suivi de bogue de KDE](#). **Avant** de faire un rapport de bogues ou de suggérer une nouvelle fonctionnalité, veuillez vérifier qu'il / elle n'a pas été déjà [rapporté](#) / [suggéré](#).

Chapitre 5

Foire aux questions

5.1 Est-ce que l'éditeur d'expressions rationnelles gère les références arrières ?

Non, ce n'est pas encore géré. Mais c'est planifié pour une prochaine version.

5.2 Est-ce que l'éditeur d'expressions rationnelles gère les opérations de reconnaissance d'affichage ?

Non, mais heureusement cela sera disponible dans la prochaine version.

5.3 Je suis l'auteur d'un programme KDE. Comment puis-je utiliser ce composant graphique dans mon application ?

Consultez [la documentation de la classe KRegExpEditorInterface](#).

5.4 Je n'arrive pas à trouver le bouton *éditeur d'expressions rationnelles* dans konqueror sur une autre installation de KDE3 par exemple, pourquoi ?

Le composant d'expressions rationnelles est situé dans le paquet « utils » de KDE. Si vous n'avez pas installé ce paquet, alors les boutons de *éditeur d'expressions rationnelles* n'apparaîtront pas dans les programmes.

Chapitre 6

Remerciements et licence

La documentation est sous copyright 2001, Jesper K. Pedersen blackie@kde.org

Traduction française par Simon Depiets 2df@tuxfamily.org.

Cette documentation est soumise aux termes de la [Licence de Documentation Libre GNU \(GNU Free Documentation License\)](#).

Ce programme est soumis aux termes de la [Licence Générale Publique GNU \(GNU General Public License\)](#).