McScM - Feature # 27: Convert QDDs to regular expressions for display

Status:	Closed	**Priority:**	Normal
Author:	Grégoire Sutre	**Category:**	
Created:	07/22/2010	**Assignee:**	Tristan Le Gall
Updated:	10/20/2011	**Due date:**	10/12/2011
Subject:	Convert QDDs to regular expressions for display		
Description:	The debugging output currently displays recognizable languages of queue contents as lists of states and transitions of the corresponding QDDs. In many cases, regular expressions would be easier to read. _Note_: the development of this feature is not worth the effort if QDDs are only displayed for debugging.		

History

07/22/2010 07:52 pm - Grégoire Sutre

- Assignee set to Tristan Le Gall

07/23/2010 03:29 pm - Tristan Le Gall

- Status changed from New to In Progress

- % Done changed from 0 to 20

Working on this

04/08/2011 01:49 pm - Grégoire Sutre

- Target version set to 1.x

06/23/2011 10:47 am - Tristan Le Gall

- % Done changed from 20 to 100

SCM library allow us to display regular expressions insted of QDD, but they are not easier to read and there is no known algorithm to always find "simple" regular expression.

Not: this was implemented months ago. I just forgot to update this feature.

06/23/2011 12:33 pm - Tristan Le Gall

- Status changed from In Progress to Resolved

08/10/2011 12:01 pm - Grégoire Sutre

- Status changed from Resolved to In Progress

I find the output difficult to use, but maybe this is because I'm used to the automaton output.

Comparison for simple sets:

* For { ## }:

<pre>
{ nodes = 0 => attrvertex = 0; pred = []; succ = [1];;
1 => attrvertex = 1; pred = [0]; succ = [];;
arcs = (0,1) => {(##,Top)};
info = { initial = [|[0]; [1]|];
final = [|[0]; [1]|];
det=true; min=true; counter=2;
partition={(##,Top); (c,Top); (d,Top); (o,Top)}; }; }
</pre>

versus

<pre>

```
  E.{(##,Top)}+{(##,Top)}).E+E.{(##,Top)}+{(##,Top)}
```

* For { o## }:

```
  { nodes = 0 => attrvertex = 0; pred = []; succ = [2];;
            1 => attrvertex = 1; pred = [2]; succ = [];;
            2 => attrvertex = 0; pred = [0]; succ = [1];;
    arcs = (0,2) => {(o,Top)};
           (2,1) => {(##,Top)};
    info = { initial = [|[0]; [1]|];
             final = [|[2]; [1]|];
             det=true; min=false; counter=3;
             partition={(##,Top); (c,Top); (d,Top); (o,Top)}; }; }
```

versus

```
  (E.{(o,Top)}.{(##,Top)}+{(o,Top)}.{(##,Top)}).E
  +
  E.{(o,Top)}.{(##,Top)}+{(o,Top)}.{(##,Top)}
```

* For { oc## }:

```
  { nodes = 0 => attrvertex = 0; pred = []; succ = [1];;
            1 => attrvertex = 0; pred = [0]; succ = [3];;
            2 => attrvertex = 1; pred = [3]; succ = [];;
            3 => attrvertex = 0; pred = [1]; succ = [2];;
    arcs = (0,1) => {(o,Top)};
           (1,3) => {(c,Top)};
           (3,2) => {(##,Top)};
    info = { initial = [|[0]; [2]|];
             final = [|[3]; [2]|];
             det=true; min=false; counter=4;
             partition={(##,Top); (c,Top); (d,Top); (o,Top)}; }; }
```

versus

```
  (E.{(o,Top)}.{(c,Top)}.{(##,Top)}
  +
  {(o,Top)}.{(c,Top)}.{(##,Top)}).E
  +
  E.{(o,Top)}.{(c,Top)}.{(##,Top)}
  +
  {(o,Top)}.{(c,Top)}.{(##,Top)}
```

* For { (oc)^+## }:

```
  { nodes = 0 => attrvertex = 0; pred = []; succ = [1];;
            1 => attrvertex = 0; pred = [0 2]; succ = [2];;
            2 => attrvertex = 0; pred = [1]; succ = [1 3];;
            3 => attrvertex = 1; pred = [2]; succ = [];;
```

```
    arcs = (0,1) => {(o,Top)};
           (1,2) => {(c,Top)};
           (2,1) => {(o,Top)};
           (2,3) => {(##,Top)};
    info = { initial = [|[0]; [3]|];
             final = [|[2]; [3]|];
             det=true; min=false; counter=4;
             partition={(##,Top); (c,Top); (d,Top); (o,Top)}; }; }
```

versus

```
  (E.{(o,Top)}.{(c,Top)}.({(o,Top)}.{(c,Top)})*.{(##,Top)}
  +
  {(o,Top)}.{(c,Top)}.({(o,Top)}.{(c,Top)})*.{(##,Top)}).E
  +
  E.{(o,Top)}.{(c,Top)}.({(o,Top)}.{(c,Top)})*.{(##,Top)}
  +
  {(o,Top)}.{(c,Top)}.({(o,Top)}.{(c,Top)})*.{(##,Top)}
```

08/16/2011 12:31 pm - Grégoire Sutre

- Priority changed from Low to Normal

- Target version changed from 1.x to 1.2

08/23/2011 06:24 pm - Tristan Le Gall

+ même syntaxe que les expressions régulières du langage d'entrée

09/07/2011 02:57 pm - Tristan Le Gall

- Status changed from In Progress to Resolved

output regexp have the same syntax as input regexp

10/20/2011 02:02 am - Grégoire Sutre

- Due date set to 10/12/2011

- Status changed from Resolved to Closed

It is not clear at this time how to obtain a better output, i.e., simpler regular expressions. So let's close this issue for now, since it is probably not worth spending more development time on this.