Mulgara - Bug #36

Basic query failure

11/20/2006 11:50 PM - ronald -

Status:	Closed	**Start date:**	
Priority:	High	**Due date:**	
Assignee:	Andrae Muys -	**% Done:**	0%
Category:	Mulgara	**Estimated time:**	0.00 hour
Target version:			
Resolution:	fixed		

Description

```
The following setup demonstrates the problem:
<br/>

<br/>
    drop &lt;<a href="local:///topazproject#test&gt;;">
local:///topazproject#test&amp;gt;;</a>
<br/>
    create &lt;<a href="local:///topazproject#test&gt;;">
local:///topazproject#test&amp;gt;;</a>
<br/>

<br/>
    insert
<br/>
      &lt;foo:bar&gt; &lt;foo:set&gt; &lt;user:joe&gt;
<br/>
      &lt;foo:set&gt; &lt;topaz:implies&gt; &lt;bar:set&gt;
<br/>
      &lt;foo:set&gt; &lt;topaz:implies&gt; &lt;bar:get&gt;
<br/>
    into &lt;<a href="local:///topazproject#test&gt;;">
local:///topazproject#test&amp;gt;;</a>
<br/>

<br/>
    select $s $p $o from &lt;<a href="local:///topazproject#test">
local:///topazproject#test</a>&gt; where (
<br/>
      $s $p $o
<br/>
      and ($s &lt;mulgara:is&gt; &lt;foo:bar&gt;)
<br/>
    ) or (
<br/>
      ($s $impliedBy $o and $impliedBy &lt;topaz:implies&gt; $p)
<br/>
      and ($s &lt;mulgara:is&gt; &lt;foo:bar&gt;)
<br/>
    );
<br/>

<br/>
    select $s $p $o from &lt;<a href="local:///topazproject#test">
local:///topazproject#test</a>&gt; where (
<br/>
      $s $p $o
<br/>
      or ($s $impliedBy $o and $impliedBy &lt;topaz:implies&gt; $p)
<br/>
    ) and ($s &lt;mulgara:is&gt; &lt;foo:bar&gt;);
<br/>
```


The two queries are logically equivalent (the <mulgara:is>
 term is duplicated in the first query, and factored out in the second query). However, while the
first query returns three statements, the second query only returns one. Specifically, the first q
uery returns

    <foo:bar> <foo:set> <user:joe>

    <foo:bar> <bar:set> <user:joe>

    <foo:bar> <bar:get> <user:joe>

but the second returns only

    <foo:bar> <foo:set> <user:joe>

History

#1 - 11/21/2006 05:00 AM - Paula Gearon

I need confirm this, but I have a theory...

In the second query the initial part is:

$s $p $o or

($s $impliedBy $o and $impliedBy <topaz:implies> $p)

Since the ($s $p $o) triple is all that is being selected then the part in parentheses would appear redundant.
  Perhaps there is code that recognizes this and does the optimization before the mulgara:is predicate is appl
ied?

#2 - 11/21/2006 05:14 AM - Paula Gearon

The theory isn't looking too good. However, the problem appears to be resolution order, if the constraint ord
er is reversed, then the bug does not appear:

select $s $p $o

from <rmi://localhost/server1#test>

where (

  ($s $impliedBy $o and $impliedBy <topaz:implies> $p)

  or $s $p $o

) and $s <mulgara:is> <foo:bar>;

Result:

[foo:bar, foo:set, user:joe]

[foo:bar, bar:get, user:joe]

[foo:bar, bar:set, user:joe]

#3 - 11/21/2006 05:35 AM - Andrae Muys -

Looks like non-union compatible disjunctions strike again.

Note that the second query is attempting a join between the assignment and a NUC-disjunction.

Paul, note the use of the $p in the object position - this does mean the parenthesised expression is not redun
dant.

#4 - 11/21/2006 07:07 AM - ronald -

Thanks to both of you for checking this out so quickly. So should we

just be avoiding using 'or'? That would be a bit difficult, though.

#5 - 11/21/2006 02:54 PM - Paula Gearon

I had a typo in the first comment....

When I said "would appear redundant", I should have said "could appear redundant"
. The current behaviour is clearly a bug. I wondered if the resolver made this mistake. I find that scenari
o less likely now that I know that it works if the terms are reversed.

I was speculating out loud in case it prompted anyone to have an "A-Ha" moment. :-)

#6 - 12/11/2006 06:05 AM - Andrae Muys -

No need to avoid 'or', but it is worth avoiding non-union-compatible 'or'.

What this means is that there are variables in one term of the disjunction that do not appear in all of the ot
hers.

So

$a <:is> <foo> or $a <:is> <foo> is fine. but

$a <:is> <foo> or $a <pred> $b is not.

The work around is to use the distributive-law to convert any POSOP... form query into SOP as per your first e
xample above.

The only problem with this work around is that the query fragment will not join against other constraints cont
aining $b properly as unbound should join successfully with anything, while with an explicit placeholder it wi
ll eliminate. However for SOP form queries handle UNBOUND correctly.

#7 - 02/24/2007 02:13 AM - Andrae Muys -

Downgraded from critical to major as workaround does exist (just provide suitable <mulgara:is>
 bindings to avoid non-union-compatible disjunction).

#8 - 04/17/2007 05:42 AM - Andrae Muys -

Got it - this is not a non-union-compatible disjunction bug after all.

This is a bug in [[UnorderedProjection]] which can only be triggered in the presence of a conjunction/disjunct
ion combination.

Specifically projection is being used to reorder the variables of its operand, without taking into considerati
on the affect this has on prefixing. To see this, look at the implementation of beforeFirst() in [[UnorderedP
rojection]], while considering that the order of variables in the projection may not be the same as the order
in its operand.

An initial fix is to materialise the operand when a reordering of variables is required. The ultimate fix is
to do this only if the reordering will invalidate a defined-prefix, as we don't care if the defined-prefix is
reordered as long as any prefix variable is not reordered with respect to non-prefix variables.

It would be good to take the time to fully exploit the Annotations to provide transparent access to defineInde
x and reresolve.

#9 - 04/17/2007 10:13 AM - Andrae Muys -

Resovled in branches/nuc-disj revision 231.

Please test - if there are no reports of regressions in the next few days I'll merge over to trunk.

#10 - 04/20/2007 05:27 AM - ronald -

This works for us. We tested the original query that lead to this bug report. Many thanks.

#11 - 04/20/2007 05:29 AM - ronald -

Ooops, scratch that last comment - wrong bug report. Sorry. We're still in the process of verifying this one.

#12 - 04/20/2007 10:38 PM - ronald -

Ok, for real now: we tested our full original query and the seems to be fixed. Thanks!