Saxon - Bug #4765

NPE in LoopLifter (from StackOverflow)

2020-09-29 11:47 - Norm Tovey-Walsh

Status:	Closed	**Start date:**	2020-09-29
Priority:	Normal	**Due date:**	
Assignee:	Michael Kay	**% Done:**	100%
Category:	Internals	**Estimated time:**	0:00 hour
Sprint/Milestone:		**Spent time:**	0:00 hour
Legacy ID:		**Fixed in Maintenance Release:**	9.9.1.8, 10.3
Applies to branch:	10, 9.9, trunk	**Platforms:**	
Fix Committed on Branch:	10, 9.9, trunk		

Description

From StackOverflow, https://stackoverflow.com/questions/64114488/

```
declare function local:test($input as element(input)) as element(output)
{
  <output>
      <details>
          <service>
          {
              for $i in 1 to count($input/foo)
              return
                  for $j in 1 to count($input/bar)
                  return if ($input/foo[$i]/data = 'A' and $input/baz[$i]/type = 'B')
                         then <result/>
                         else ()
          }
          </service>
      </details>
  </output>
};

declare variable $input as element(input) external := <input>
</input>;

local:test($input)
```

raises NPE:

```
java.lang.NullPointerException
        at net.sf.saxon.expr.parser.LoopLifter.markDependencies(LoopLifter.java:168)
        at net.sf.saxon.expr.parser.LoopLifter.gatherInfo(LoopLifter.java:112)
        at net.sf.saxon.expr.parser.LoopLifter.gatherInfo(LoopLifter.java:122)
        at net.sf.saxon.expr.parser.LoopLifter.gatherInfo(LoopLifter.java:122)
        at net.sf.saxon.expr.parser.LoopLifter.gatherInfo(LoopLifter.java:122)
...
```

History

#1 - 2020-09-29 16:28 - Michael Kay

- Priority changed from Low to Normal

Running with -explain triggers additional verification of the expression tree at various stages during optimisation, leading to earlier failure:

```
java.lang.IllegalStateException: Invalid parent pointer in ($input/(child::element(Q{}foo)[$i]))/data = "A" an
d ($input/(child::element(Q{}baz)[$i]))/type = "B" subexpression ($input/(child::element(Q{}foo)[$i]))/data =
"A"
 at net.sf.saxon.expr.Expression.verifyParentPointers(Expression.java:229)
```

```
at net.sf.saxon.expr.Expression.verifyParentPointers(Expression.java:239)
at net.sf.saxon.expr.parser.Optimizer.trace(Optimizer.java:555)
at net.sf.saxon.expr.ForExpression.optimize(ForExpression.java:132)
at net.sf.saxon.expr.flwor.FLWORExpression.rewriteForOrLet(FLWORExpression.java:782)
at net.sf.saxon.expr.flwor.FLWORExpression.optimize(FLWORExpression.java:495)
at net.sf.saxon.expr.flwor.FLWORExpression.optimize(FLWORExpression.java:488)
at net.sf.saxon.expr.Operand.optimize(Operand.java:228)
at net.sf.saxon.expr.Expression.optimizeChildren(Expression.java:618)
at net.sf.saxon.expr.instruct.ParentNodeConstructor.optimize(ParentNodeConstructor.java:191)
at net.sf.saxon.expr.instruct.FixedElement.optimize(FixedElement.java:107)
at net.sf.saxon.expr.Operand.optimize(Operand.java:228)
at net.sf.saxon.expr.Expression.optimizeChildren(Expression.java:618)
at net.sf.saxon.expr.instruct.ParentNodeConstructor.optimize(ParentNodeConstructor.java:191)
at net.sf.saxon.expr.instruct.FixedElement.optimize(FixedElement.java:107)
at net.sf.saxon.expr.Operand.optimize(Operand.java:228)
at net.sf.saxon.expr.Expression.optimizeChildren(Expression.java:618)
at net.sf.saxon.expr.instruct.ParentNodeConstructor.optimize(ParentNodeConstructor.java:191)
at net.sf.saxon.expr.instruct.FixedElement.optimize(FixedElement.java:107)
at net.sf.saxon.query.XQueryFunction.optimize(XQueryFunction.java:448)
at net.sf.saxon.query.XQueryFunctionLibrary.optimizeGlobalFunctions(XQueryFunctionLibrary.java:327)
at net.sf.saxon.query.QueryModule.optimizeGlobalFunctions(QueryModule.java:1207)
at net.sf.saxon.expr.instruct.Executable.fixupQueryModules(Executable.java:458)
at net.sf.saxon.query.XQueryParser.makeXQueryExpression(XQueryParser.java:177)
at net.sf.saxon.query.StaticQueryContext.compileQuery(StaticQueryContext.java:568)
at net.sf.saxon.query.StaticQueryContext.compileQuery(StaticQueryContext.java:630)
at net.sf.saxon.s9api.XQueryCompiler.compile(XQueryCompiler.java:609)
at net.sf.saxon.Query.compileQuery(Query.java:804)
at net.sf.saxon.Query.doQuery(Query.java:317)
at net.sf.saxon.Query.main(Query.java:97)
```

#2 - 2020-09-29 16:44 - Michael Kay

The rewrite that corrupts the tree is in ForExpression.rewriteWhereClause() (there isn't actually a where clause here, but the "return if..." compiles to exactly the same tree as if a where clause had been used). The rewrite extracts all the terms (=operands of AND) in the condition, and moves any of them that don't depend on the control variable of the "for" (that is, $j) up a level, recombining the extracted terms (in this case, all of them) into a new AndExpression. It then notices that all of the terms of the (imaginary) where clause have been promoted, which means the where clause itself can be dropped, and it replaces the entire construct for $i in X return if C then Y else () with if C then (for $i in X return Y) else (). However the C that it uses here is the original AndExpression, rather than the reconstituted AndExpression, and this is no longer valid, because its subexpressions have been transferred to the reconstituted AndExpression, which means their parent pointers are now corrupt.

#3 - 2020-09-29 16:59 - Michael Kay

- Category set to Internals

- Status changed from New to In Progress

- Assignee set to Michael Kay

- Applies to branch 9.9, trunk added

The simplest fix is to use the reconstituted AndExpression rather than the original, that is, change ForExpression#281 from

```
return Choose.makeConditional(condition, this);
```

to

```
return Choose.makeConditional(promotedCondition, this);
```

#4 - 2020-09-29 17:25 - Michael Kay

In trying to write a regression test for this, I've found there is another precondition for the bug to trigger: the "where" clause must not be rewritten as a predicate by an earlier phase of optimisation, specifically in FLWORExpression.rewriteWhereClause(). This means it must not actually be a where clause (where C return X), it has to be written as return if C then X else () which looks like a where clause but isn't caught be the earlier optimisation phase.

QT3 Test case WhereExpr035 created: this fails without the patch, and succeeds with it.

#5 - 2020-09-29 17:50 - Michael Kay

On the 9.9 branch, my first attempt to run the new test didn't fail: this turned out to be because I was running the test with option -unfolded, which complicates the query sufficiently to prevent the optimization taking place. Running without that option, I confirmed the patch is needed and works.

However, regression testing the patch under 9.9, I get a new failure in test -s:prod-WhereClause -t:cbcl-left-outer-join-004. The failure occurs whether or not the new patch is present. The failure occurs because we're trying to rewrite the "order-by" expression of a FLWORExpression with something that isn't a TupleExpression.

The problem is in OrderByClause#31, which reads

```
this.sortKeysOp = new Operand(flwor, new SortKeyDefinitionList(sortKeys), OperandRole.REPEAT_NAVIGATE_CONSTRAI
NED);
```

whereas the 10 branch has, correctly,

```
this.sortKeysOp = new Operand(flwor, new SortKeyDefinitionList(sortKeys), SORT_KEYS_ROLE);
```

I'm having difficulty searching the history here because of the change from Subversion to Git.

#6 - 2020-09-29 18:12 - Michael Kay

- Status changed from In Progress to Resolved

- Fix Committed on Branch 10, 9.9, trunk added

The bug in OrderByClause was fixed in the 10 branch somewhere between 10.0 and 10.1, but I haven't been able to track it down to a particular bug entry. I have retrofitted the change to the 9.9 branch, however, so 9.9 now passes these tests.

#7 - 2020-10-22 18:21 - O'Neil Delpratt

- Fixed in Maintenance Release 9.9.1.8 added

Bug fix applied on the Saxon 9.9.1.8 maintenance release. Leaving open until applied on the Saxon 10 maintenance release.

#8 - 2020-10-28 18:58 - O'Neil Delpratt

Bug fix applied in the Saxon 10.3 maintenance release

#9 - 2020-10-28 19:12 - O'Neil Delpratt

- Status changed from Resolved to Closed

- % Done changed from 0 to 100

- Fixed in Maintenance Release 10.3 added