Item 172
git_comments:

git_commits:

1. **summary:** Tests for JENA-989
   message: Tests for JENA-989

github_issues:

github_issues_comments:

github_pulls:

github_pulls_comments:

github_pulls_reviews:

jira_issues:

1. **summary:** Round tripping property paths from SPARQL into Sexpressions and back yields a syntax error
   description: The following SPARQL query: {noformat} SELECT ?uri WHERE { ?uri <http://www.w3.org/2000/01/rdf-schema#subPropertyOf>+ <http://purl.org/linked-data/sdmx/2009/dimension#refArea> } {noformat} When converted into an sse: {noformat} (path ?uri (path+ <http://www.w3.org/2000/01/rdf-schema#subPropertyOf>) <http://purl.org/linked-data/sdmx/2009/dimension#refArea>) {noformat} and converted back into a SPARQL query via OpAsQuery is syntactically invalid. Note the misplaced `.`: {noformat} SELECT * WHERE { . ?uri (<http://www.w3.org/2000/01/rdf-schema#subPropertyOf>)+ <http://purl.org/linked-data/sdmx/2009/dimension#refArea> } {noformat} The same seems to occur for all other property paths (e.g. those that use +, *, /, ^ etc...) This seems related to issues resolved in JENA-963.
   label: code-design
2. **summary:** Round tripping property paths from SPARQL into Sexpressions and back yields a syntax error
   description: The following SPARQL query: {noformat} SELECT ?uri WHERE { ?uri <http://www.w3.org/2000/01/rdf-schema#subPropertyOf>+ <http://purl.org/linked-data/sdmx/2009/dimension#refArea> } {noformat} When converted into an sse: {noformat} (path ?uri (path+ <http://www.w3.org/2000/01/rdf-schema#subPropertyOf>) <http://purl.org/linked-data/sdmx/2009/dimension#refArea>) {noformat} and converted back into a SPARQL query via OpAsQuery is syntactically invalid. Note the misplaced `.`: {noformat} SELECT * WHERE { . ?uri (<http://www.w3.org/2000/01/rdf-schema#subPropertyOf>)+ <http://purl.org/linked-data/sdmx/2009/dimension#refArea> } {noformat} The same seems to occur for all other property paths (e.g. those that use +, *, /, ^ etc...) This seems related to issues resolved in JENA-963.
3. **summary:** Round tripping property paths from SPARQL into Sexpressions and back yields a syntax error
   description: The following SPARQL query: {noformat} SELECT ?uri WHERE { ?uri <http://www.w3.org/2000/01/rdf-schema#subPropertyOf>+ <http://purl.org/linked-data/sdmx/2009/dimension#refArea> } {noformat} When converted into an sse: {noformat} (path ?uri (path+ <http://www.w3.org/2000/01/rdf-schema#subPropertyOf>) <http://purl.org/linked-data/sdmx/2009/dimension#refArea>) {noformat} and converted back into a SPARQL query via OpAsQuery is syntactically invalid. Note the misplaced `.`: {noformat} SELECT * WHERE { . ?uri (<http://www.w3.org/2000/01/rdf-schema#subPropertyOf>)+ <http://purl.org/linked-data/sdmx/2009/dimension#refArea> } {noformat} The same seems to occur for all other property paths (e.g. those that use +, *, /, ^ etc...) This seems related to issues resolved in JENA-963.
4. **summary:** Round tripping property paths from SPARQL into Sexpressions and back yields a syntax error
   description: The following SPARQL query: {noformat} SELECT ?uri WHERE { ?uri <http://www.w3.org/2000/01/rdf-schema#subPropertyOf>+ <http://purl.org/linked-data/sdmx/2009/dimension#refArea> } {noformat} When converted into an sse: {noformat} (path ?uri (path+ <http://www.w3.org/2000/01/rdf-schema#subPropertyOf>) <http://purl.org/linked-data/sdmx/2009/dimension#refArea>) {noformat} and converted back into a SPARQL query via OpAsQuery is syntactically invalid. Note the misplaced `.`: {noformat} SELECT * WHERE { . ?uri (<http://www.w3.org/2000/01/rdf-schema#subPropertyOf>)+ <http://purl.org/linked-

data/sdmx/2009/dimension#refArea> } {noformat} The same seems to occur for all other property paths (e.g. those that use +, *, /, ^ etc...) This seems related to issues resolved in JENA-963.

5. **summary:** Round tripping property paths from SPARQL into Sexpressions and back yields a syntax error
   description: The following SPARQL query: {noformat} SELECT ?uri WHERE { ?uri <http://www.w3.org/2000/01/rdf-schema#subPropertyOf>+ <http://purl.org/linked-data/sdmx/2009/dimension#refArea> } {noformat} When converted into an sse: {noformat} (path ?uri (path+ <http://www.w3.org/2000/01/rdf-schema#subPropertyOf>) <http://purl.org/linked-data/sdmx/2009/dimension#refArea>) {noformat} and converted back into a SPARQL query via OpAsQuery is syntactically invalid. Note the misplaced `.`: {noformat} SELECT * WHERE { . ?uri (<http://www.w3.org/2000/01/rdf-schema#subPropertyOf>)+ <http://purl.org/linked-data/sdmx/2009/dimension#refArea> } {noformat} The same seems to occur for all other property paths (e.g. those that use +, *, /, ^ etc...) This seems related to issues resolved in JENA-963.
6. **summary:** Round tripping property paths from SPARQL into Sexpressions and back yields a syntax error
   description: The following SPARQL query: {noformat} SELECT ?uri WHERE { ?uri <http://www.w3.org/2000/01/rdf-schema#subPropertyOf>+ <http://purl.org/linked-data/sdmx/2009/dimension#refArea> } {noformat} When converted into an sse: {noformat} (path ?uri (path+ <http://www.w3.org/2000/01/rdf-schema#subPropertyOf>) <http://purl.org/linked-data/sdmx/2009/dimension#refArea>) {noformat} and converted back into a SPARQL query via OpAsQuery is syntactically invalid. Note the misplaced `.`: {noformat} SELECT * WHERE { . ?uri (<http://www.w3.org/2000/01/rdf-schema#subPropertyOf>)+ <http://purl.org/linked-data/sdmx/2009/dimension#refArea> } {noformat} The same seems to occur for all other property paths (e.g. those that use +, *, /, ^ etc...) This seems related to issues resolved in JENA-963.
   label: code-design
7. **summary:** Round tripping property paths from SPARQL into Sexpressions and back yields a syntax error
   description: The following SPARQL query: {noformat} SELECT ?uri WHERE { ?uri <http://www.w3.org/2000/01/rdf-schema#subPropertyOf>+ <http://purl.org/linked-data/sdmx/2009/dimension#refArea> } {noformat} When converted into an sse: {noformat} (path ?uri (path+ <http://www.w3.org/2000/01/rdf-schema#subPropertyOf>) <http://purl.org/linked-data/sdmx/2009/dimension#refArea>) {noformat} and converted back into a SPARQL query via OpAsQuery is syntactically invalid. Note the misplaced `.`: {noformat} SELECT * WHERE { . ?uri (<http://www.w3.org/2000/01/rdf-schema#subPropertyOf>)+ <http://purl.org/linked-data/sdmx/2009/dimension#refArea> } {noformat} The same seems to occur for all other property paths (e.g. those that use +, *, /, ^ etc...) This seems related to issues resolved in JENA-963.
8. **summary:** Round tripping property paths from SPARQL into Sexpressions and back yields a syntax error
   description: The following SPARQL query: {noformat} SELECT ?uri WHERE { ?uri <http://www.w3.org/2000/01/rdf-schema#subPropertyOf>+ <http://purl.org/linked-data/sdmx/2009/dimension#refArea> } {noformat} When converted into an sse: {noformat} (path ?uri (path+ <http://www.w3.org/2000/01/rdf-schema#subPropertyOf>) <http://purl.org/linked-data/sdmx/2009/dimension#refArea>) {noformat} and converted back into a SPARQL query via OpAsQuery is syntactically invalid. Note the misplaced `.`: {noformat} SELECT * WHERE { . ?uri (<http://www.w3.org/2000/01/rdf-schema#subPropertyOf>)+ <http://purl.org/linked-data/sdmx/2009/dimension#refArea> } {noformat} The same seems to occur for all other property paths (e.g. those that use +, *, /, ^ etc...) This seems related to issues resolved in JENA-963.
9. **summary:** Round tripping property paths from SPARQL into Sexpressions and back yields a syntax error
   description: The following SPARQL query: {noformat} SELECT ?uri WHERE { ?uri <http://www.w3.org/2000/01/rdf-schema#subPropertyOf>+ <http://purl.org/linked-data/sdmx/2009/dimension#refArea> } {noformat} When converted into an sse: {noformat} (path ?uri (path+ <http://www.w3.org/2000/01/rdf-schema#subPropertyOf>) <http://purl.org/linked-data/sdmx/2009/dimension#refArea>) {noformat} and converted back into a SPARQL query via OpAsQuery is syntactically invalid. Note the misplaced `.`: {noformat} SELECT * WHERE { . ?uri (<http://www.w3.org/2000/01/rdf-schema#subPropertyOf>)+ <http://purl.org/linked-data/sdmx/2009/dimension#refArea> } {noformat} The same seems to occur for all other property paths (e.g. those that use +, *, /, ^ etc...) This seems related to issues resolved in JENA-963.

jira_issues_comments:

1. **body:** This is a problem caused by JENA-972 (Improve the printing of basic graph patterns) but OpAsQuery tends to create queries that are likely to trigger this. OpAsQuery tends to create group elements (between {{\{...\}}}) with adjacent {{ElementPathBlocks}} - blocks for any basic graph pattern

and blocks for each path. {{ElementPathBlocks}} can hold both and the parser will not produce adjacent blocks, but instead a single combined block. Both wend up as the same algebra. JENA-972 improved the prettiness of output but has a bug whereby "no triple patterns before the path" causes the stray {{.}}. OpAsQuery is not necessary to cause this though JENA-963 makes the situation more common.
label: code-design

2. Commit 85ed839b0d184995810a3fe0e232c184d47ede36 in jena's branch refs/heads/master from [~andy.seaborne] [https://git-wip-us.apache.org/repos/asf?p=jena.git;h=85ed839] Tests for JENA-989
3. Commit 20ac8e6f768888981f312a5ce6008ffdbc9f2fd9 in jena's branch refs/heads/master from [~andy.seaborne] [https://git-wip-us.apache.org/repos/asf?p=jena.git;h=20ac8e6] JENA-989 : Output '.' between items in an ElementPathBlock.
4. Commit a31c7d2718b85779d38c00abc74f8ab573dce41b in jena's branch refs/heads/master from [~andy.seaborne] [https://git-wip-us.apache.org/repos/asf?p=jena.git;h=a31c7d2] Follow-on from JENA-989: more cases of producing a simpler query.
5. Thanks for this Andy! I can confirm that with these patches the queries now appear to work for us.
6. **body:** Thanks for the confirmation. While in the area, I cleaned up query production for various unnecessary nesting cases as well and expanded the test coverage. Adding in syntax elements into a syntax group can result in extra {}. Output is correct (equivalent query; same algebra) from OpAsQuery, just looks different. There are quite a few cases but also some where it can't be done and the code is on the cautious side of identified safe cases, rather than trying to apply a general rule, which needs deeper thought/checking..
   label: code-design
7. Jena 3.0.0 Release