Aquery to Q Compiler: Identifying columns requiring sort

Jose Cambronero

March 4, 2015

1 Motivation

Identifying columns that need sorting in an order-dependent aquery expression requires a bit of work. The algorithm below attempts to address one possible way of doing this.

2 Example Expression

We take as an example the following expression, which we assume must be executed as one (i.e. we can not first execute order independent parts).

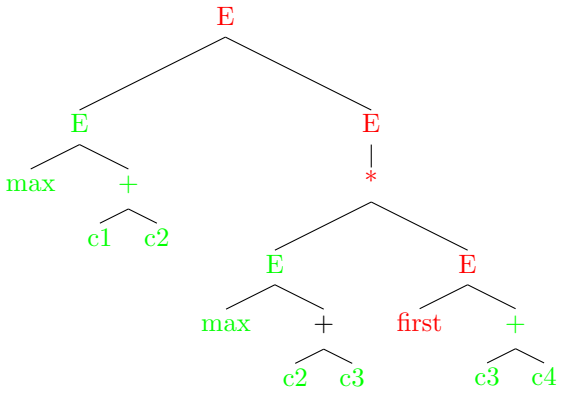
```
max(c1,c2), max(c2 + c3) * first(c3)
```

We assume `c1,c2,c3` are all columns. Notice how the only order dependent operation is `first(c3)`. However, given that we sort `c3` for this, we need to sort c2 due to `c2+c3` and for similar reasons we need to sort `c1`. The algorithm below finds all such columns.

3 AST Representation

Below an AST representation of this expression, which informs the algorithm below.

Bottom up order dependency information is shown in color, green is order-independent, red order-dependent. Note that this reflects information as we build the tree up, so e.g `first(c1)` would display `c1` as order independent, but the node calling `first` on `c1` as order dependent.



4 Proposed Algorithm

Data: Aquery Expression AST A annotated with bottom up order-dependence information
Result: List of columns requiring sort
Initialization: Create empty list S, and empty list of lists P;
During DFS traversal of A - Add to S column names in subtree annotated as order-dependent, do so until an order-independent operator (e.g. max/sum etc) is encountered;
- Let column names for order-independent nodes bubble up as a list, and when they reach the application of an order-independent operator (e.g. max/sum), append list to P, and bubble up NULL (to append to that in higher levels);
After DFS traversal added := 1;

```
while added do
    added:= 0;
    for p in P do
        if any column in p is in L then
            add p to L;
            remove p from P;
            added:= 1;
        end
    end
end
```

L now contains all columns that need to be sorted
Currently, my main concern with this is complexity.