
Rewriting within a graph structure

— Towards fast and easy rewriter implementations —

Queries as trees

- Lucene
- Elasticsearch / Open Search Query DSL
- Querqy

```
{  
  should: [  
    must: [  
      term: dishwasher  
      score: [max|mean]  
    ]  
    must: [  
      term: dish  
      term: washer  
      score: [max|mean]  
    ]  
  ]  
}
```

Applying synonyms (nested approach)

```
must(mini, notebook)
```

mini notebook =>
SYNONYM: ultrabook

```
should(  
  must(mini, notebook),  
  ultrabook  
)
```

Applying synonyms (repeated approach)

```
must(mini, notebook)
```

mini notebook =>
SYNONYM: ultrabook

```
must(  
    should(mini,      ultrabook) ,  
    should(notebook,  ultrabook)  
)
```

Multiple words to multiple words

```
must(convertible, notebook)
```

convertible notebook =>
SYNONYM: tablet computer

```
must(  
  should(  
    convertible,  
    must(tablet, computer)),  
  should(  
    notebook,  
    must(tablet, computer))  
)
```

Synonym and delete interactions

```
must(mini, notebook)
```

mini notebook =>
SYNONYM: ultrabook
DELETE: mini

```
must(  
  should(ultrabook),  
  should(notebook, ultrabook)  
)
```

Synonym and delete interactions

```
must(mini, notebook)
```

mini notebook =>
SYNONYM: ultrabook
DELETE: mini notebook

```
must(  
  should(ultrabook),  
  should(ultrabook)  
)
```

Rewriter interactions

```
must(mini, note, book)
```

note book => (Wordbreak) => notebook

```
must(  
    mini, should(note, notebook), should(book, notebook)  
)
```

mini notebook =>
SYNONYM: ultrabook

```
must(  
    should(mini, ultrabook),  
    should(note, notebook, ultrabook),  
    should(book, notebook, ultrabook)  
)
```

Conclusions

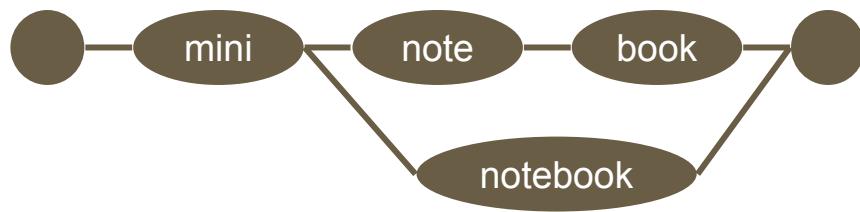
- Repeated clauses complicate handling
 - Synonym and delete interactions
 - Rewriter interactions
 - Subsequence extraction (e. g. for rule lookups)
 - Difficult to define rewriting actions across different nesting levels
- Querqy handles a lot of complexity Lucene does not take care about, but
 - Unexpected side effects
 - Implementing custom rewriters has high barriers

Rethinking querqy-core



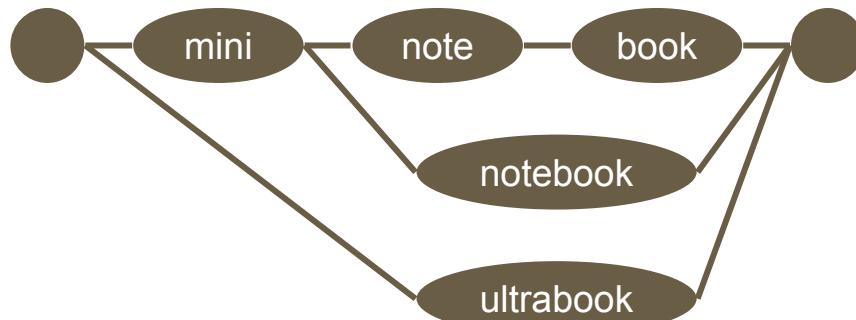
Rethinking querqy-core

note book => (WB) => notebook



Rethinking querqy-core

mini notebook =>
SYNONYM: ultrabook

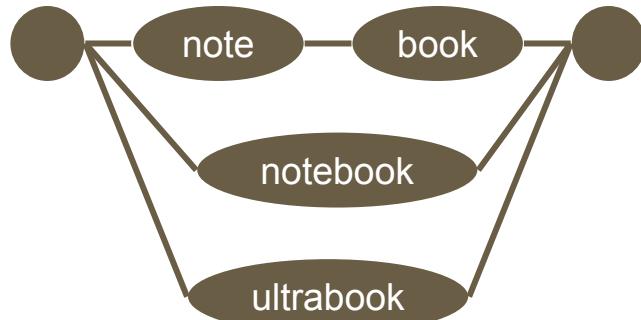


Rethinking querqy-core

mini notebook =>

SYNONYM: ultrabook

DELETE: mini



Conclusions

- Considering a query as a graph
 - Makes rewriting easier to understand
 - Helps avoiding repeated clauses
 - Helps avoiding side effects
- Querqy handles a lot of complexity Lucene does not take care about, but
 - Unexpected side effects
 - Implementing custom rewriters has high barriers