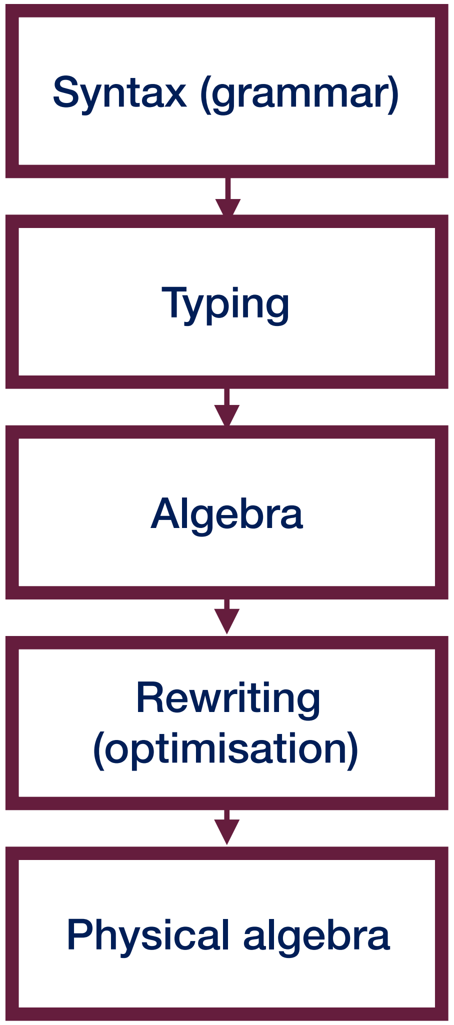
OQL Physical Algebra

Getting down to the libs

Matematikos
ir informatikos
fakultetas

Arūnas Janeliūnas
Object Databases

Behind the query



Syntax (grammar)

Typing

Algebra

Rewriting (optimisation)

Physical algebra

Is the query well written?

Will the query output its result?

What result query will output?

Can it be done faster?

How it will be done?

O_2 physical algebra

- Just an example
- C programming language

get

```
get (monoid, extent_name, range_variable, predicate)
```

implements

$$\sigma_{\texttt{predicate}}(\texttt{extent\_name[range\_variable]})$$

Actually, it’s

```
select *
  from range_variable in extent_name
 where predicate
```

reduce

`reduce (monoid, expr, variable, head, predicate)`

implements

$$\mathtt{MAP}_{\mathtt{variable:head}}(\sigma_{\mathtt{predicate}}(\mathtt{expr}))$$

join

```
join (monoid, left, right, predicate, keep)
```

implements

$$\texttt{left} \bowtie_{\texttt{predicate}} \texttt{right}$$

Is it the outer join?

```
keep = left
keep = right
keep = none
```

unnest

unnest (monoid, expr, variable, path, predicate, keep)

implements

$\sigma_{\texttt{predicate}}$(expr[path_root]<path[variable]>)

Path may be of the form *path_root.path_links*

```
l.courses
```

Is it the outer d-join?

```
keep = true
```

```
keep = false
```

nest

```
nest (monoid, expr, var, head, groupby, nestvars, predicate)
```

implements

$$\mathrm{MAP}_{\mathtt{nestvars:nestvars}}(\Gamma_{\mathtt{var,\ groupby,\ \{=,=,\ldots\},\ head}}(\mathtt{expr}))$$

Here the attribute `var` which is added to every combination of `groupby` attributes is

```
var = reduce (monoid, expr, var, head, predicate)
```

map

`map` `(monoid, expr, variable, function)`

implements

$\mathtt{MAP}_{\mathtt{variable:function}}(\mathtt{expr})$

merge

```
merge (monoid, left, right)
```

implements an union of the two collections: `left` **and** `right`

Example

```
select struct ( age: a,
                 cnt: count(partition)
               )
  from l in Lecturers,
       c in d.courses
 where c.title = “ODBS”
group by a: l.age()
```

```
reduce ( set,
         nest ( bag,
                unnest ( bag
                         unnest ( bag,
                                  get ( set,
                                        Lecturers,
                                        l,
                                        and()
                                      ),
                                  c,
                                  l.courses,
                                  and( c.title=”ODBS” )
                                ),
                         a,
                         l.age(),
                         and()
                       )
                partition,
                d,
                vars(a),
                vars(),
                and()
              )
         result,
         struct( age:a, cnt:count(partition) ),
         and()
       )
```

```
get    : set ([d: Destytjas])
unnest : bag ([d: Destytjas, k: Kursas])
unnest : bag ([d: Destytjas, k: Kursas, a: integer])
nest   : bag ([a: integer, partirion: bag(Destytojas)])
reduce : set ([amzius: integer, kiekis: integer])
```