How to use queries & co.

Katarína Smoleňová

GroIMP Summer School,

University of Göttingen,

June 2, 2025





Graph operations

 To search for a specific pattern (query), collect values (aggregators), analyse relations between nodes

Find:

all leaves total leaf area stem length

first internode below bud









- To analyse the actual structure (graph), i.e. search for a specific pattern
- XL syntax: enclosed in asterisked parentheses (* *)

The elements are given in their expected order, e.g.:

(* A A B *) searches for a subgraph that consists of a sequence of nodes of the types A A B, connected by *successor* edges



Examples

- Find all internodes
- ... and print them out
- Find all newly created internodes (with age 0)
- Search for all internodes with diameter > 0.01
- Find all nodes B, connected to A with a branching edge (* A +> B
- Find all pairs with distance < 1

```
(* Internode *)
println((* Internode *));
(* i:Internode, (i[age] == 0) *)
(* i:Internode, (i[diameter] > 0.01) *)
(* A +> B *)
(* f:F, g:F, ((f != g) && (distance(f, g) < 1)) *)
```





- Collect multiple values when traversing the graph and return one single value as result
- Standard aggregate operations:

count, sum, empty, exist, forall, first, last, max, min, mean, selectRandomly, selectWhereMin, selectWhereMax, ...

Can be applied to results of a query

```
count((* Leaf *))
first((* l:Leaf, (l[order] == 1 && l[rank] == 1) *))
selectRandomly((* F *))
selectWhereMax((* f:F *), (f[diameter]))
```



Examples

Count all segments F,

longer than 1

- Search for all leaves and sum up their surface areas
- Sum up potential growth rate of all growing leaves

```
count((* f:F, (f[length] > 1) *))
sum((* Leaf *)[area]) I
sum((* l:Leaf, (l.isGrowing()) *).pgr())
```



Derived relations

- Relation between successive nodes connected by several edges (one after the other) of the same type
- Example: find all descendants of some given node that are of type Internode
- "Transitive hull" of the original relation (iteration of edges)





XL syntax

- 1-to-n repetitions: +
 - A (-edgetype->)+ B
- O-to-n repetitions: *
 - A (-edgetype->)* B
- Minimal elements (stop searching once a match has been found):
 - A (-edgetype->)+ :(B)
 - A (-edgetype->)* :(B)

- Examples:
- Find all the internodes connected to the bud
 - (* Bud (<--)+ Internode *)

- Find the first internode connected to the bud
 - (* Bud (<--)+ :(Internode) *)





Edge types

 Edge notation: Arbitrary User-defined Successor Branch Decomposition

 Edge direction: Forward Backward Forward or backward Forward and backward -r->
(blank) or > or -successor->
+> or [or -branch->
/>



-r-<-r->

-r->

<-r-

r = relation

-->

Summary

- Provide possibilities to connect structure and function
- Example: search for all leaves that are successors of node c and sum up their surface areas







<u>tutorials:xl-queries-and-operators [GroIMP wiki]</u>

- Distance checking binaryTree_distance.gsz
- Relations nodeRelations.gsz





