Radial Level Planarity Testing and Embedding in Linear Time

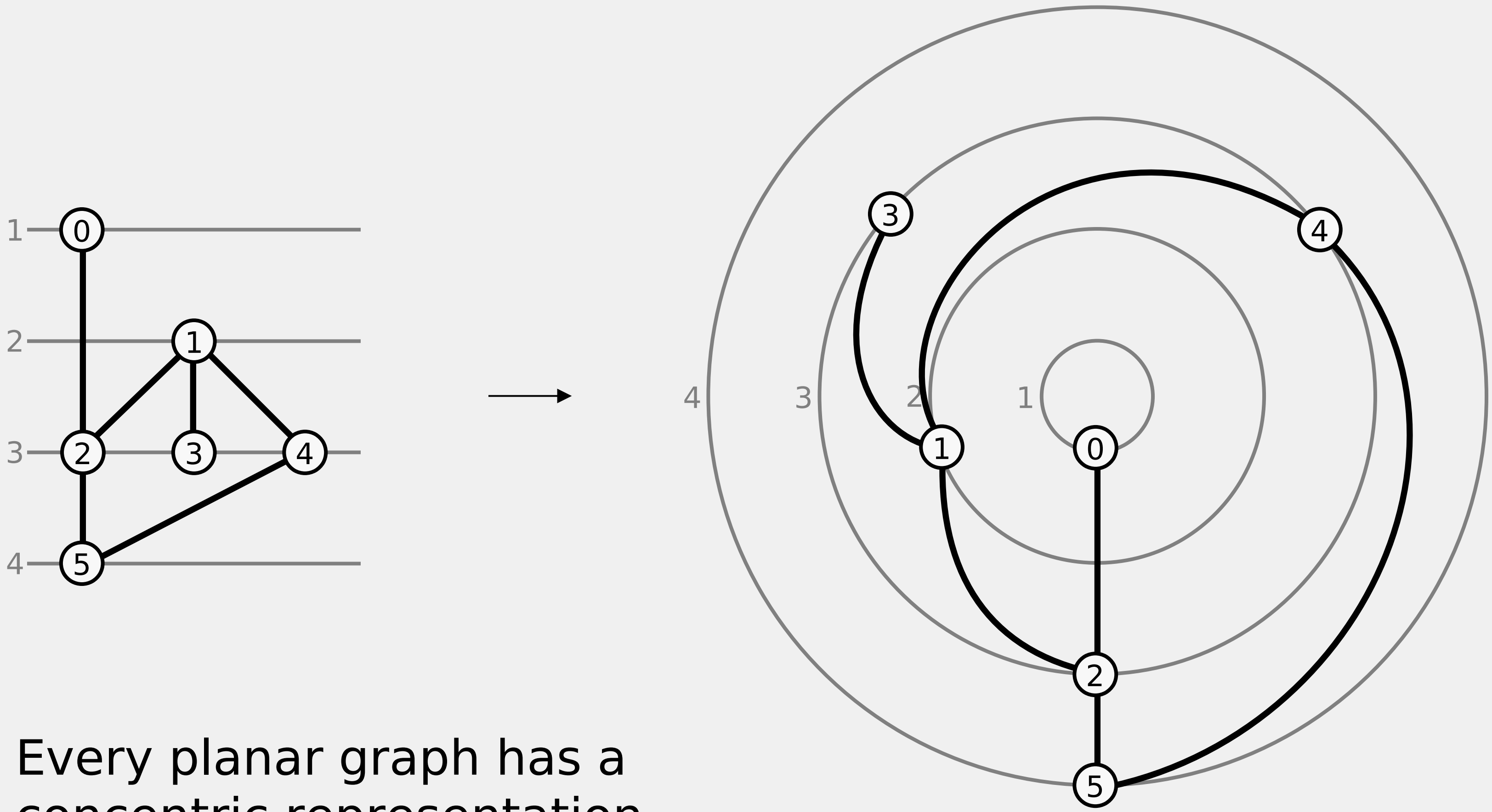
Christian Bachmaier
bachmaier@fmi.uni-passau.de

Franz J. Brandenburg
brandenb@fmi.uni-passau.de

Michael Forster
forster@fmi.uni-passau.de

University of Passau, Germany

Idea



- Every planar graph has a concentric representation
- Opposite view
  - Is a graph with vertices fixed to concentric levels planar?

Overview

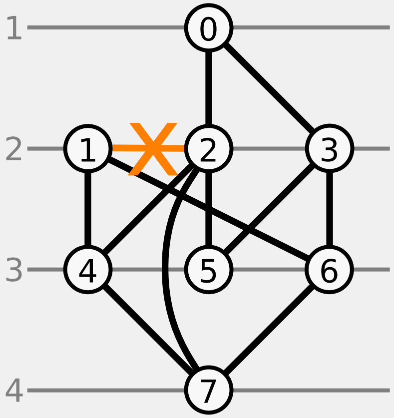
- Level planarity
  - Definition
  - Level planar graphs
  - Level planarity testing and embedding
- Radial level planarity
  - Definition
  - Concepts
- Radial level planarity testing
  - PQR-tree data structure
  - Merge of PQR-trees
- Summary

1. Level Planarity

Definition, Example, Previous Work

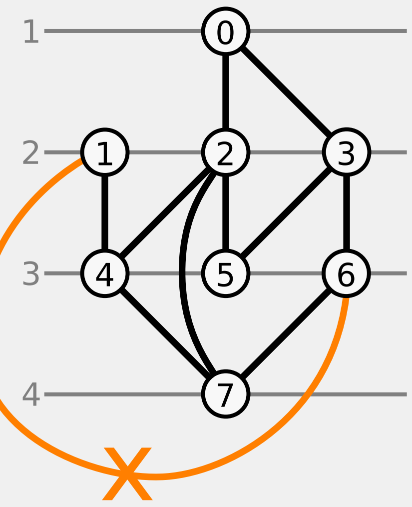
Level Graph

- G = ($V^1 \cup V^2 \cup \ldots \cup V^k$, E) is a k-level graph
  - No horizontal edges
- G is proper
  - Each edge between adjacent levels



Level Planar Graph

- G is k-level planar
  - k-level graph
  - Edges drawn strictly downwards
  - Planar

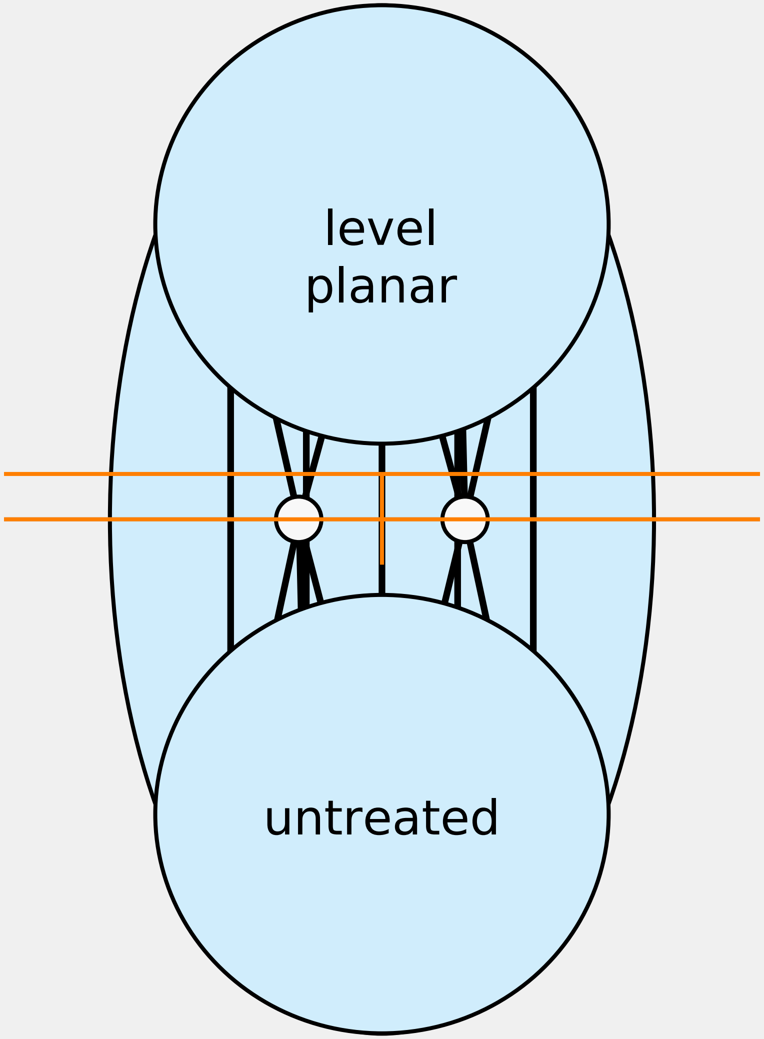


Time Line

- Planarity
  - Lempel, Even, Cederbaum (1966)
  - Booth, Lueker (1976)
    - O(n) running time
    - PQ-trees
  - Chiba, Nishizeki, Abe, Ozawa (1985)
    - Embedding
- Alternative methods
  - Hopcroft, Tarjan (1974)
  - …
  - Boyer, Cortese, Patrignani, Di Battista (last Monday)
- Level planarity
  - Di Battista, Nardelli (1988)
    - Edges are not allowed to span more than one level
    - All source vertices must lie on the first level
  - Heath, Pemmaraju (1995)
    - Extension to arbitrary level graphs (merge of PQ-trees)
  - Jünger, Leipert, Mutzel (1998/1999)
    - Adjustments and improvements
    - Embedding

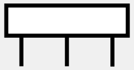
Level Planarity

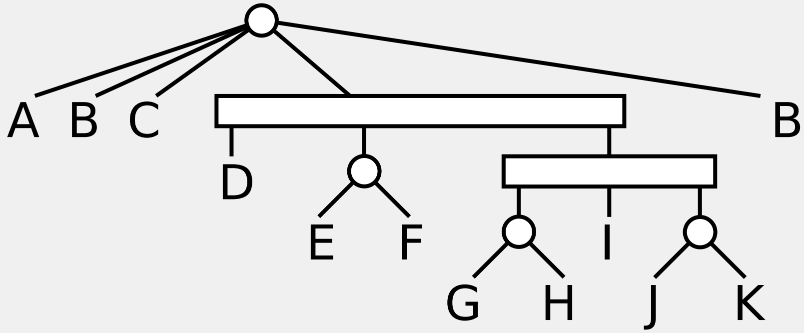
- Similar to planarity test
  (vertex addition method, LEC)
- "Sweep line"
  - Traversing the graph level by level
  - Storing "admissible" edge permutations
  - For each vertex: REDUCE
    - Test on planarity
    - Update of permutations
  - For each vertex: REPLACE
    - Insertion of new edges
  - Next level
- Running time: O(n)
- Data structure: PQ-trees
  - Merges



Storing the Permutations

- PQ-trees [Booth and Lueker 1976]
  - Leaves represent edges ("virtual nodes")
  - P-nodes:
    - Correspond to cut vertices
    - Arbitrary permutations of its children
  - Q-nodes:
    - Correspond biconnected components
    - Only reversion





PQ-trees: Operations

- REDUCE
  - Restricts permutation set
    - Leaves of a set S must lie side by side
  - Traversing tree from leaves to the root
    - Templates: P0-P6 and Q0-Q3
  - Failure ⇔ graph not (level) planar
- REPLACE
  - Replaces consecutive leaves

Complexity

- Theorem (JLM)
  - There is an O(n) time algorithm for
    - level planarity testing
    - level planar embedding

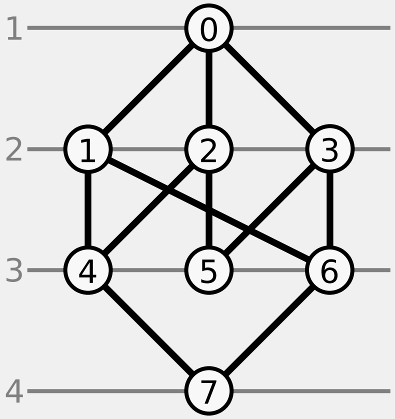
2. Radial Level Planarity

Definition, Example, Differences, Concepts

Radial Level Planar Graphs

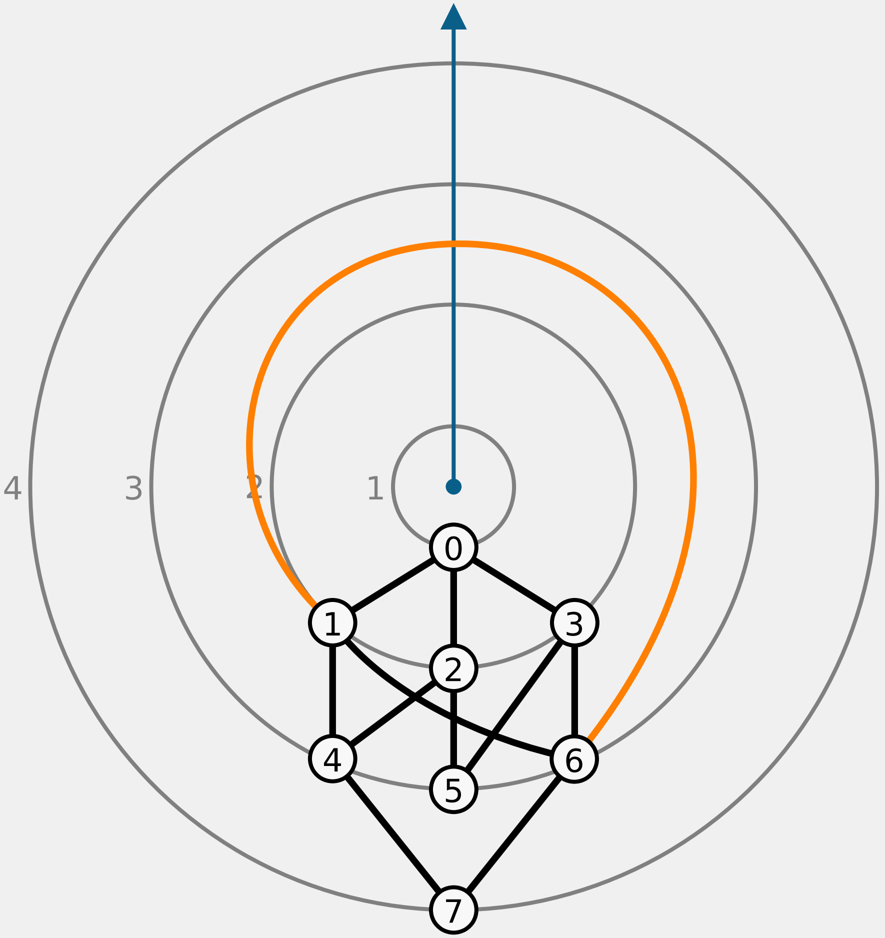
- Generalisation of level planar graphs
- G is radial k-level planar if it can be drawn such that
  - Vertices of each level lie on a concentric circle
  - Edges drawn strictly outwards
  - Planar

- 4-level graph
- Not level planar
- Radial 4-level planar



Transformation

- 4-level graph
- Not level planar
- Radial 4-level planar
- Bend level lines to circles
- Planar possibility to route edge (1, 6)
- Ray through connection points
- Cut edges
- Level planar ⇔ no cut edges
- Test?

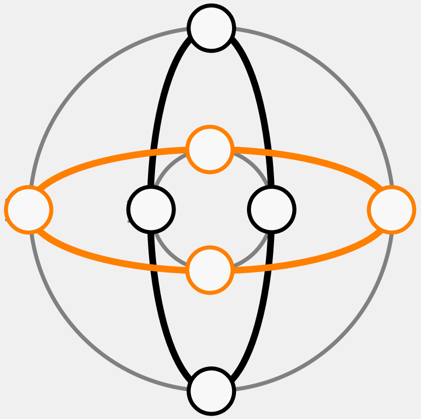


Radial Level Planarity Testing

- [Dujmović, Fellows, Hallet, Kitching, Liotta, McCartin, Nishimura, Ragde, Rosamond, Suderman, Whitesides, Wood 01]:
  - Detection of radial level planarity is fixed parameter tractable
  - $k \in O(1) \Rightarrow O(n)$ running time
  - Large constants
- Our algorithm
  - $O(n)$ running time for any k
  - Practical constants, same as in JLM

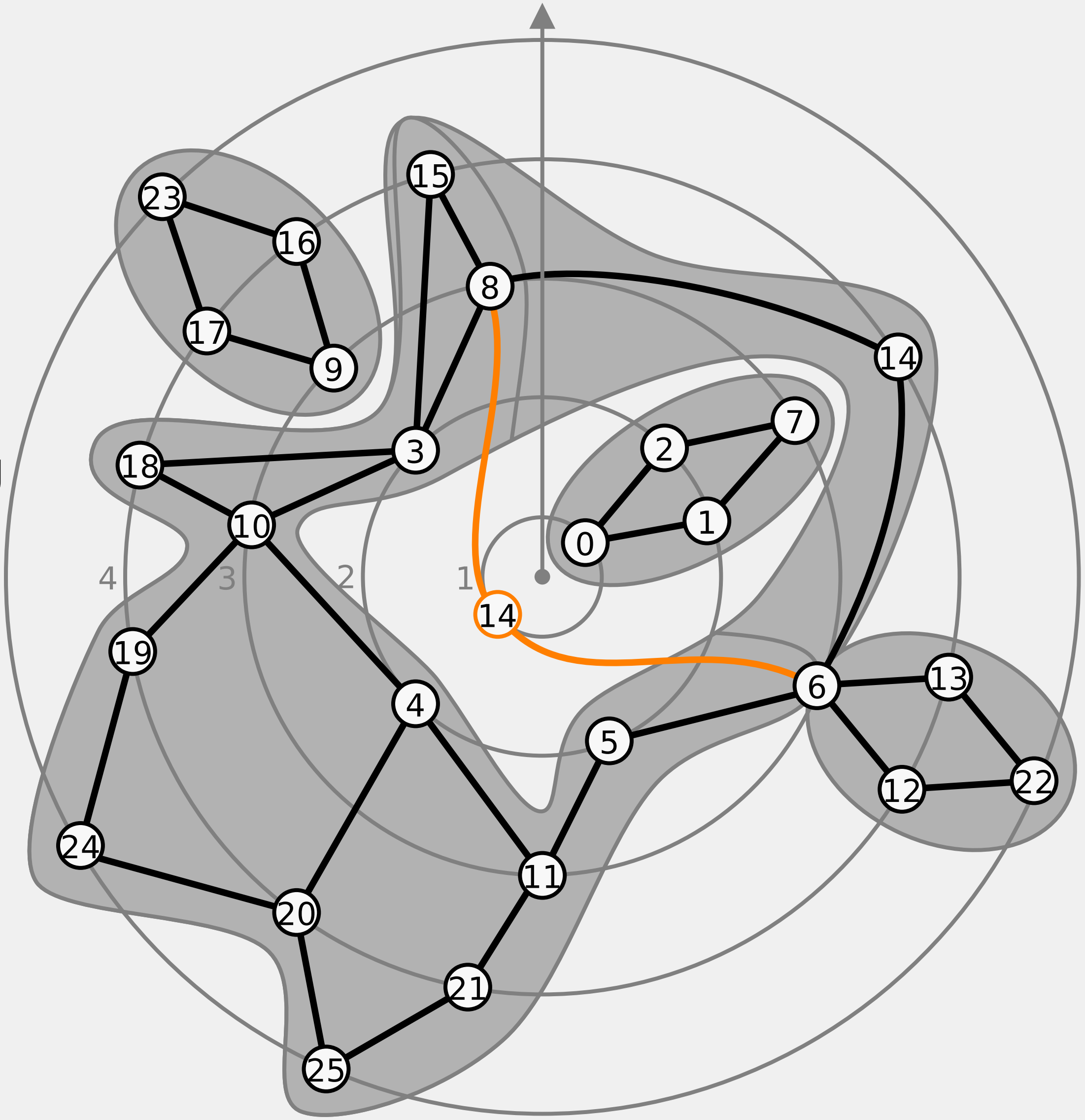
Concepts

- Differences: level vs. radial
  - Level planar ⇔ each component level planar
  - Not for radial level planarity



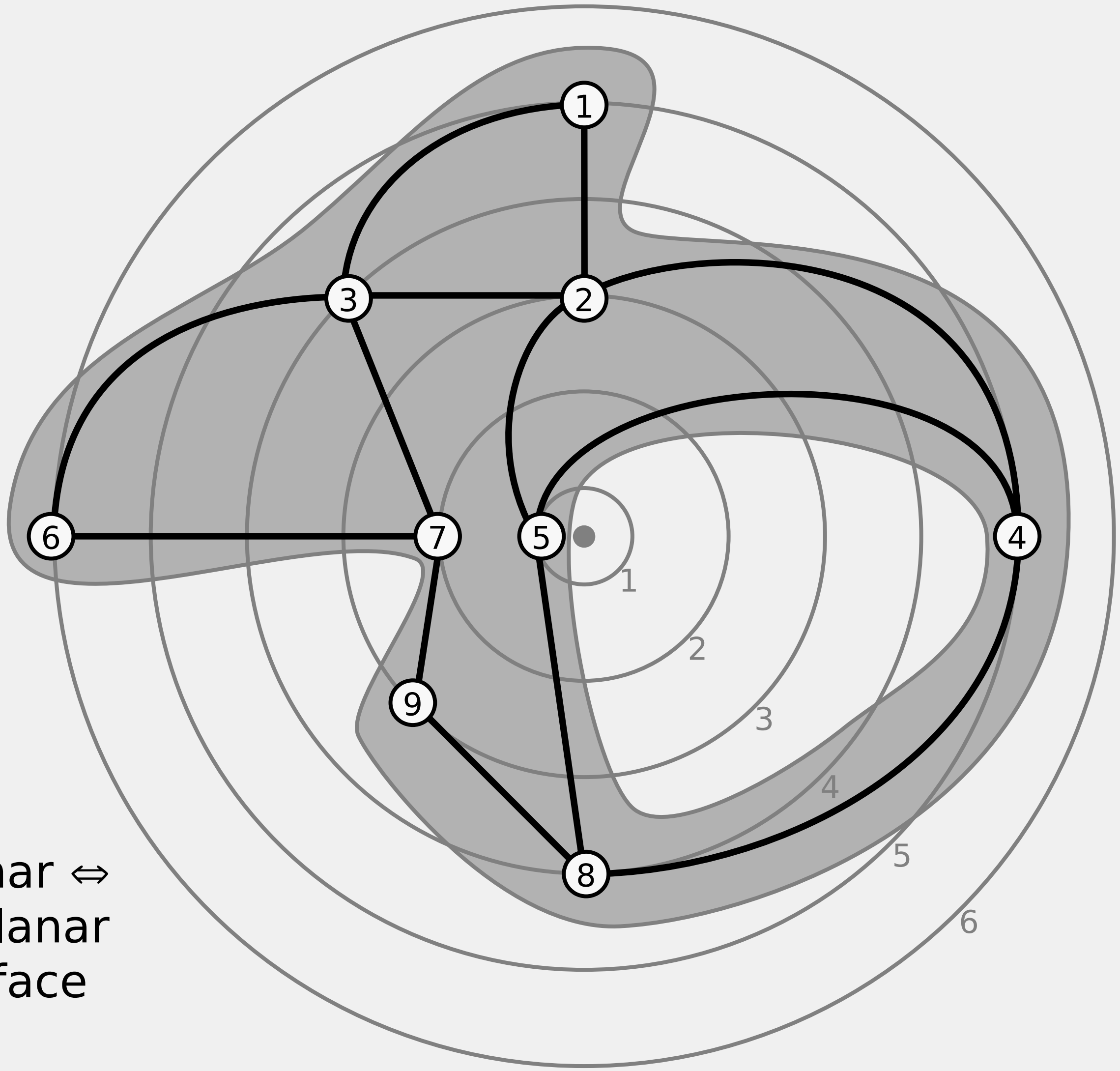
Rings

- Ring:
  - Biconnected component of a level graph
  - Not level planar
  - Not level planar
  - Depends on levelling



Nesting of Rings

- Ring extremes
  - Minimum level
  - Inner radius
  - Outer radius
  - Maximum level



- Level graph G
  with two rings R and S
- Lemma
  - G is radial level planar ⇔
    R and S are radial planar
    and R fits in centre face
    of S or vice versa

Level Optimality

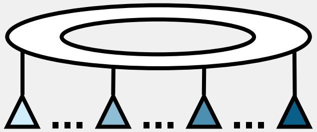
- Depend on embedding
  - Inner radius
  - Outer radius
- For leaving maximum space
  - Maximise inner radius
  - Minimise outer radius
- Level optimal embeddings
  - Existence?
  - Our algorithm computes level optimal embeddings

3. Radial Level Planarity Testing

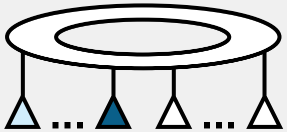
PQR-trees, Algorithm

R-Nodes

- Wavefront sweep
- New R-nodes in PQ-trees to represent rings
- R-nodes similar to Q-nodes
  - Represent a biconnected component
  - Reverse children



- New
  - Rotate children



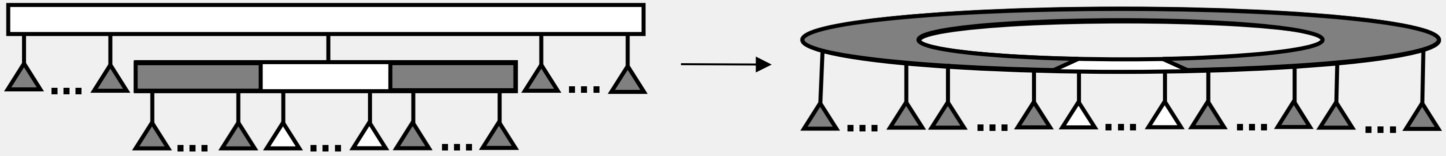
  - Only at the root

New Templates

- PQR-trees
- Additional templates for PQR-trees which treat R-nodes
  - P-templates: P7 – P9
  - Q-templates: Q4 – Q7
  - R-templates: R0 – R4

New Templates...

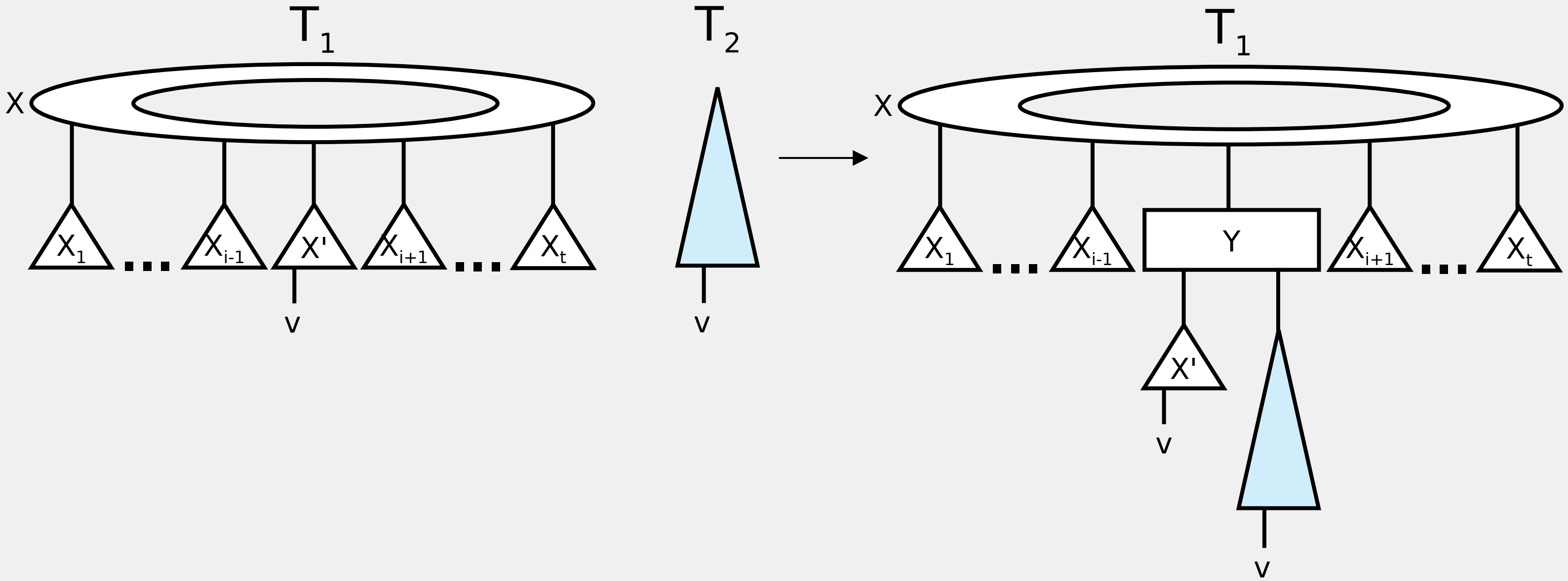
- Template Q6
  - Grey shading: pertinent leaves or pertinent sequence
  - Grey parts must be made consecutive
  - Applicable to the Q-root of a PQR-tree



- Boundary partial Q-nodes admissible
  - Root is/becomes an R-node during this reduction step
  - Impossible with PQ-tree templates
- Templates which can only be applied to the root

Merge Operations on PQR-trees

- Differences when merging two PQR-trees
  - Additional merge operations C^R and D^R
  - Treating admissible merges into PQR-trees at an R-root
- Example C^R:

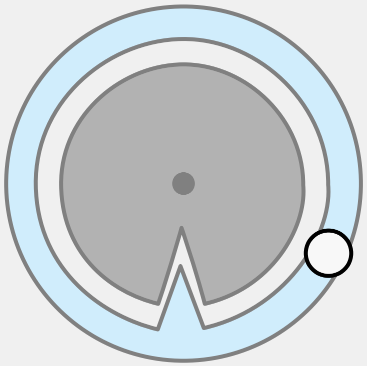


Merge of Processed Non-Rings

- Presence of ring R
  - Placing components side by side impossible
  - Other processed components must fit in some face of R
- Check whenever an inner face is closed
- Storing minimum level over all processed components
- Same mechanism as JLM use for v-singular forms

Merge of Processed Rings

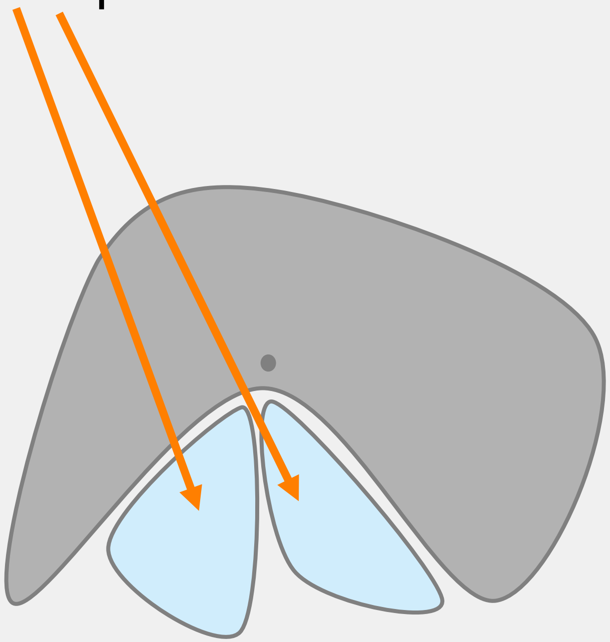
- A processed component may contain a ring
- It must fit in centre face of an outer ring
  - Highest jag
- Checked whenever a ring is closed



- Invariant
  - At most one PQR-tree with an R-root

Radial Level Planarity

- Radial level planarity ⇔
  - No REDUCE operation failed
  - No merge operation failed
  - All remaining components fit in outer face

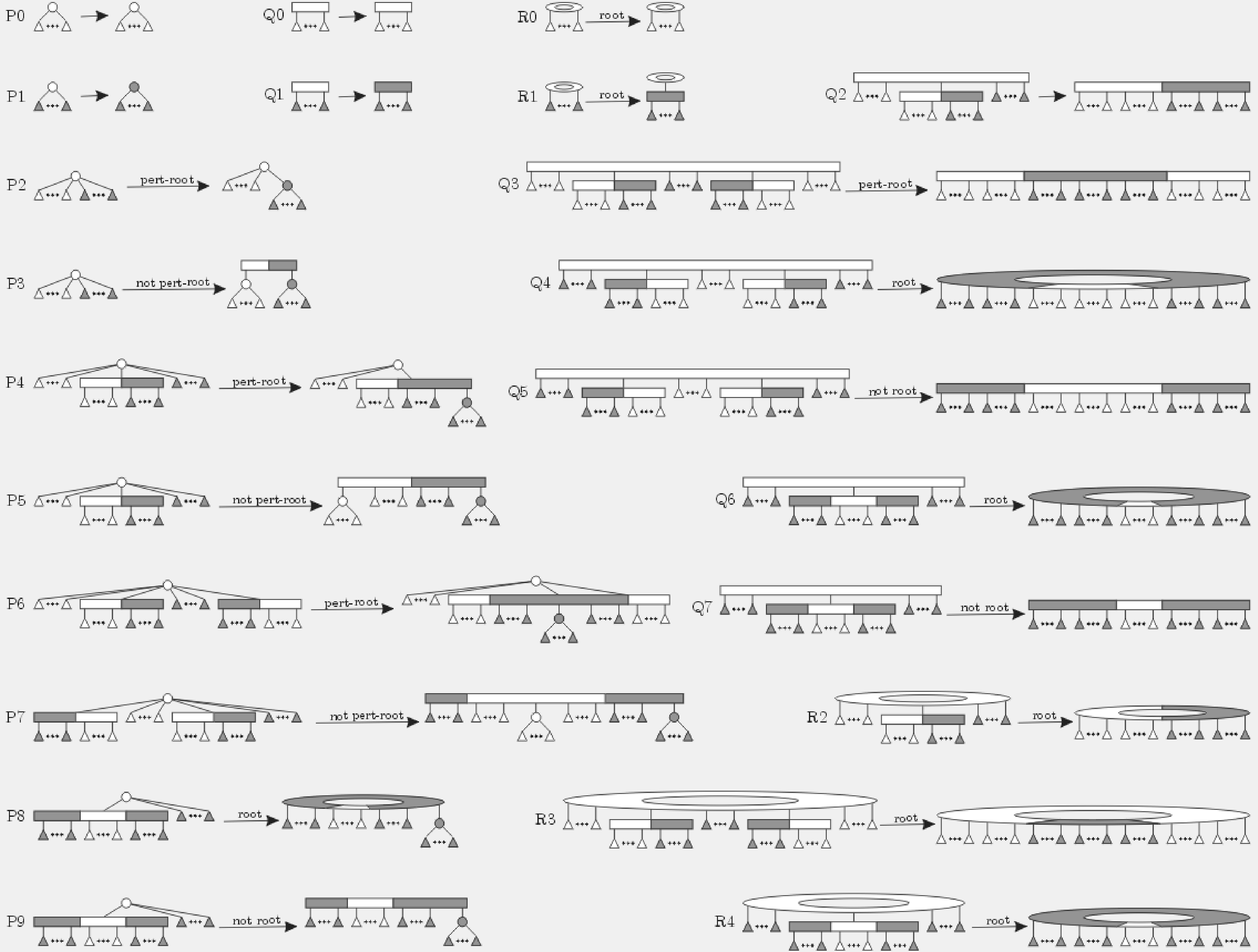


Summary

Past and Future Work

Results

- Theorem
  - There is an O(n) time algorithm for radial level planarity testing
- Theorem
  - There is an O(n) time algorithm for radial level planar embedding
- Prototypic implementation in C++ using GTL
- Future work
  - Detection of forbidden subgraphs for (radial) level planarity
  - Drawings


P0
Q0
R0
root
P1
Q1
R1
root
Q2
P2
pert-root
Q3
pert-root
P3
not pert-root
Q4
root
P4
pert-root
Q5
not root
P5
not pert-root
Q6
root
P6
pert-root
Q7
not root
P7
not pert-root
R2
root
P8
root
R3
root
P9
not root
R4
root