# Improved Booth-Lueker algorithm

- Does not stop when C1P violation is found
- Goes on to build PQR-tree instead
- Time complexity: almost linear
- Extra $O(\alpha(f))$ factor

# Union-find (disjoint set) structure

- *make_set(x)* *O(1)*
  - Creates new singleton set
- *find(x): r* *O(α(f))*
  - Finds representative of set containing x
- *union(r, s): t* *O(1)*
  - Gets two representatives, unites their sets

- f = number of elements involved
- (or, number of make_set operations)

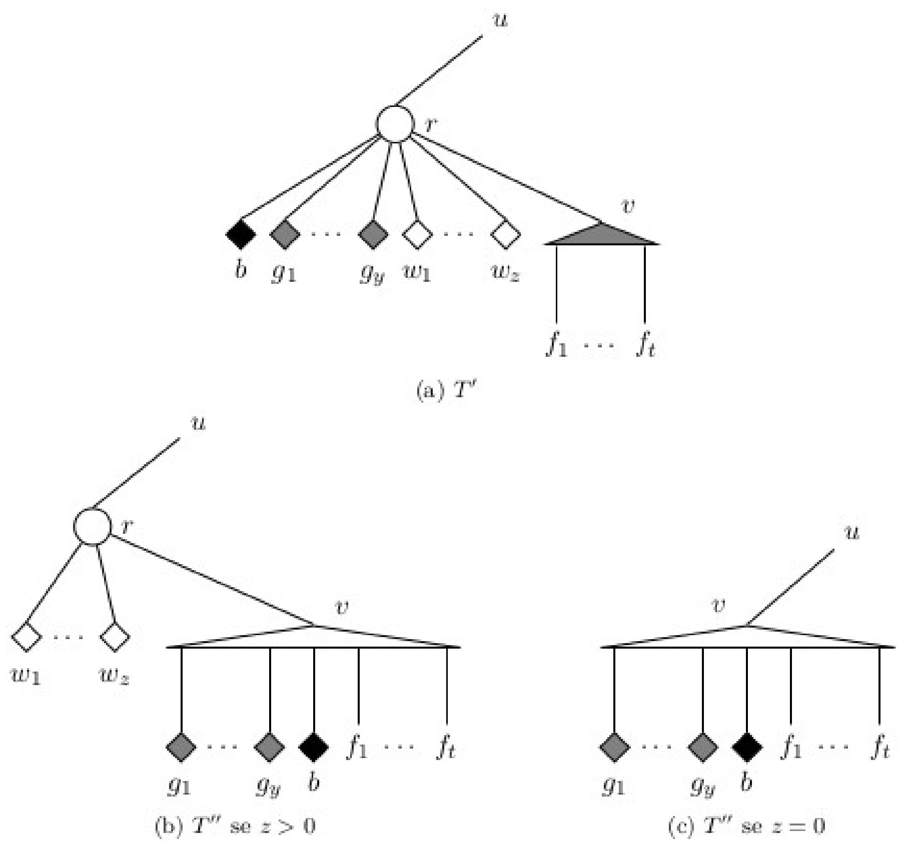
# Pointers to parent

- Children of P-nodes
  - Point to their parents
- Children of Q- and R-nodes
  - Use union-find structure
  - Only representative has pointer to parent
- Advantage
  - Merging nodes with one union-find operation
- Price to pay
  - Extra *find* operation to get parent

# Templates

- Less cases
- All templates applied to ROOT(T, S)
- Can be seen as “eliminating partial nodes” while keeping consecutiveness restrictions
- If there is a partial node, ROOT(T, S) is partial
- Only full or partial nodes are moved

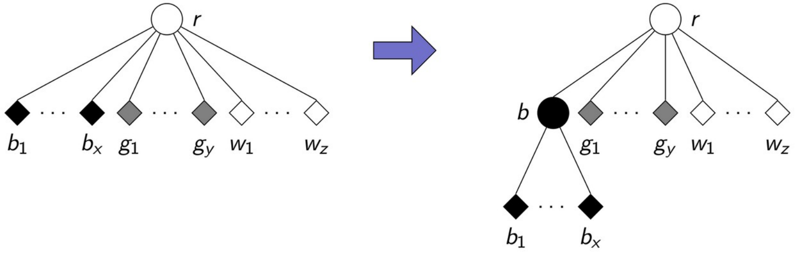
# Template: P root, Q/R partial child



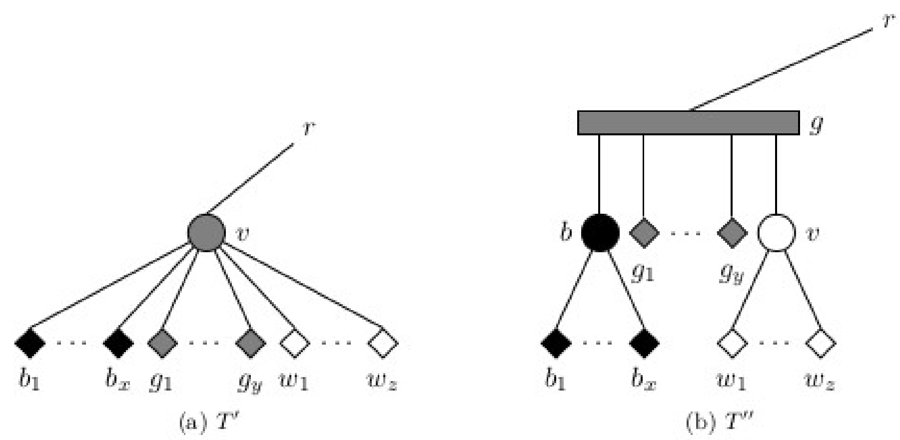

- At most one full child *b* in root
- Node *v*'s children must be ordered “darkest first”

# Template: P root, Q/R partial child

- If more than one full child *b* in root:

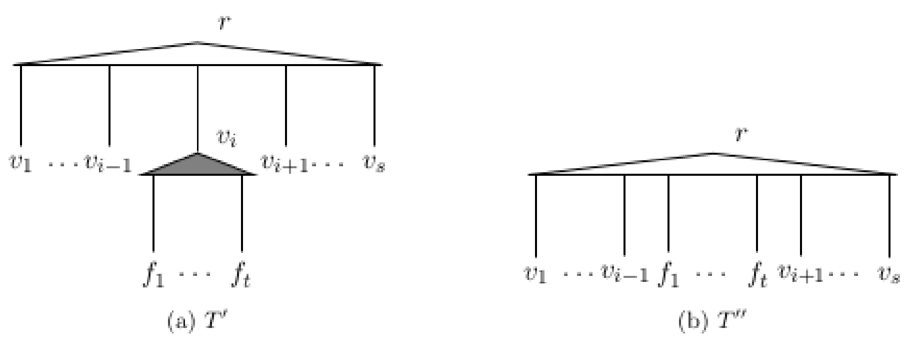


# Template: P root, P partial child



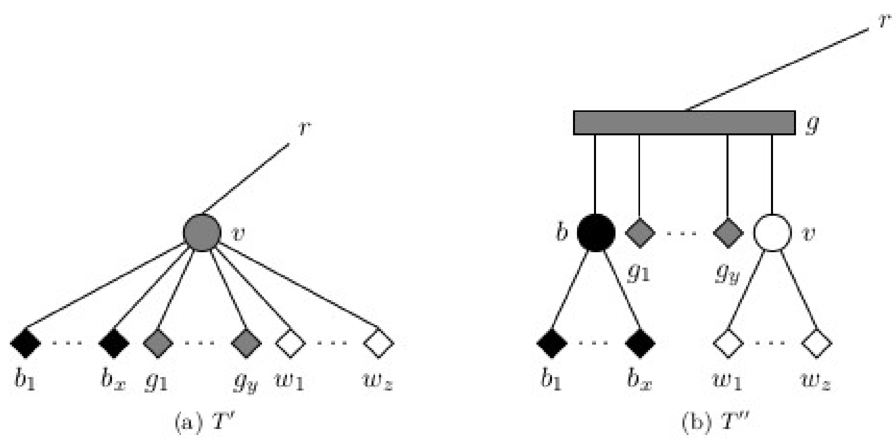

- Then apply “P root, Q/R partial child” template

# Template: Q/R root, Q/R partial child




- Nodes $v_{i-1}$ , $v_{i+1}$ ordered “darkest first”
- Node $v_i$'s children ordered “darkest first”

# Template: Q/r root, P partial child




- Then apply “Q/R root, Q/R partial child” template

# Implementation details

- Nodes deleted from the tree must be kept for the sake of the union-find structure
- First pass (called bubble by Booth and Lueker) essentially kept, but goes on reagrdless of C1P: the goal is to “color” prunned nodes and find ROOT(T, S)
- NORM(T) still applies for amortized analysis
- NORM(T) = # of Q/R nodes +
  # of nodes with P parent