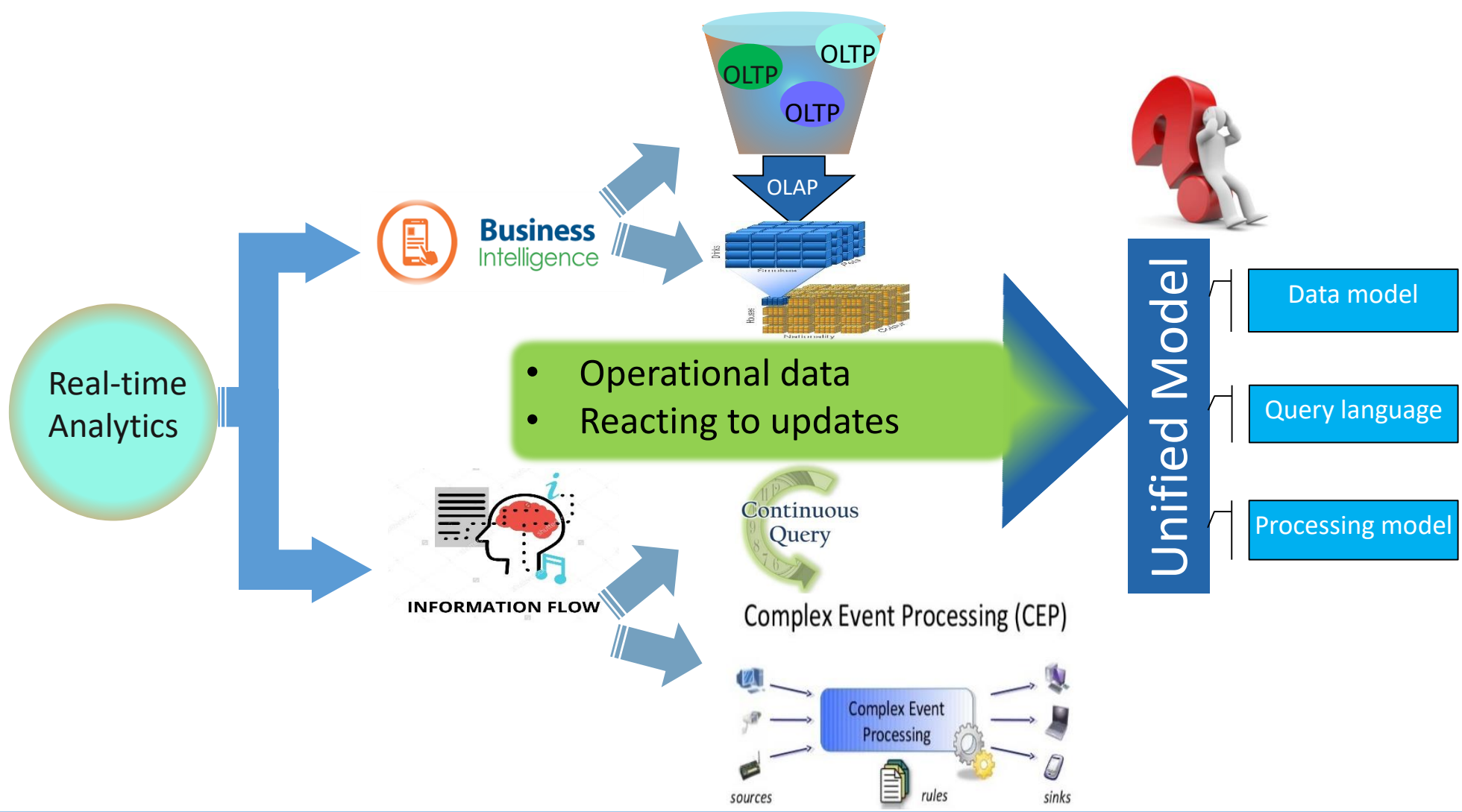
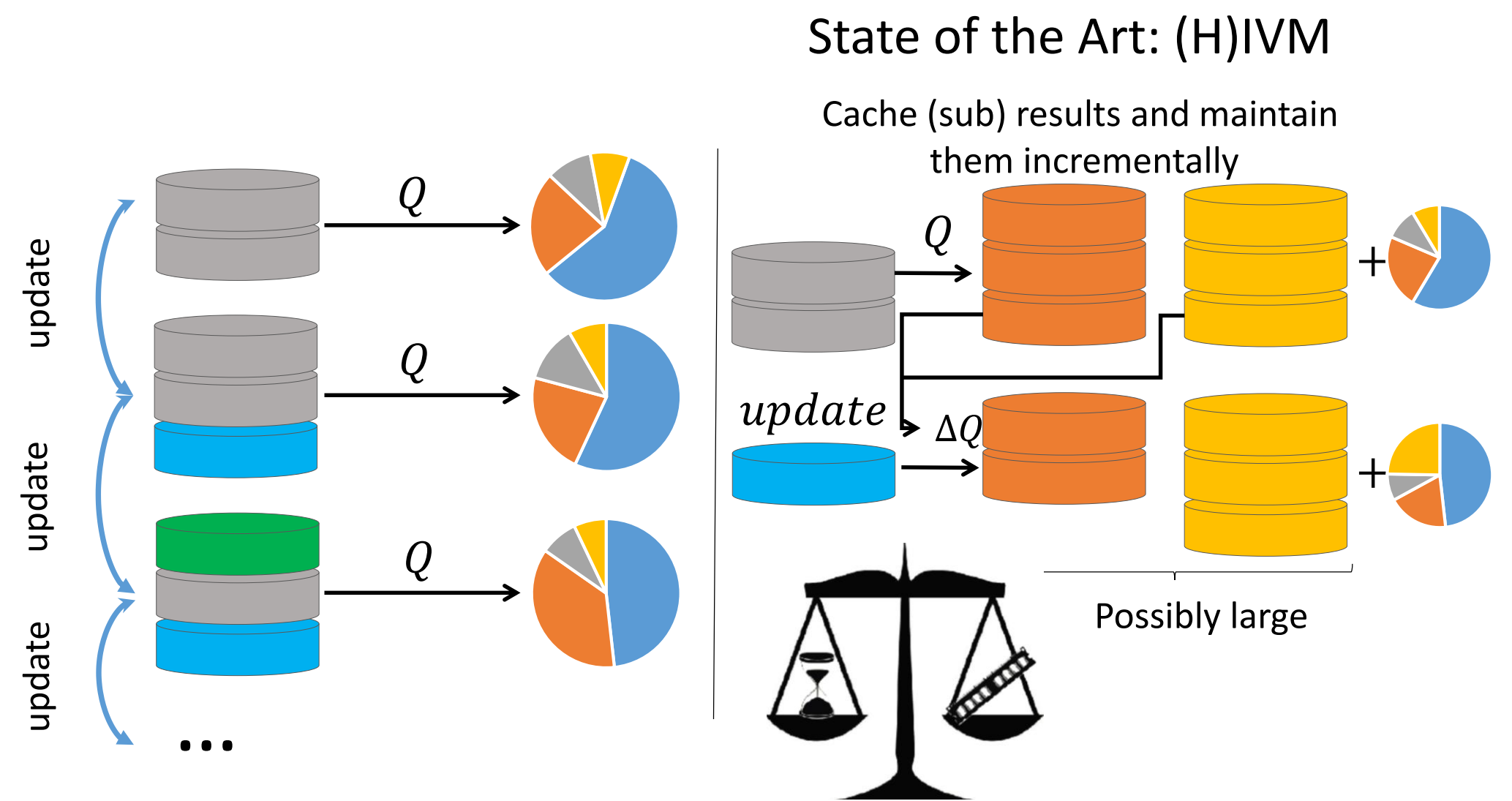


Motivation

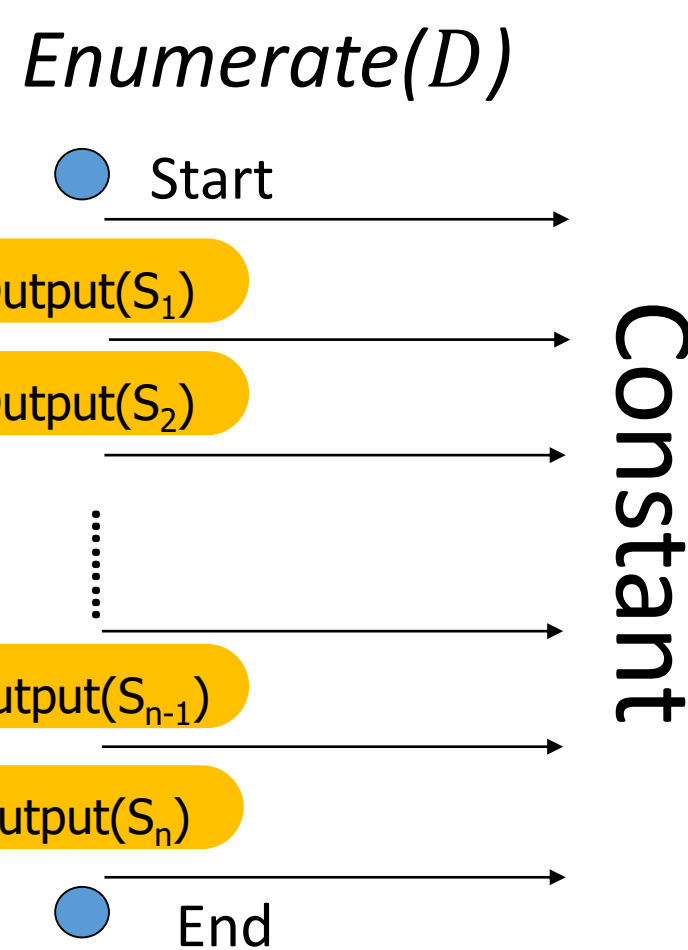


Dynamic Query Processing



Our approach

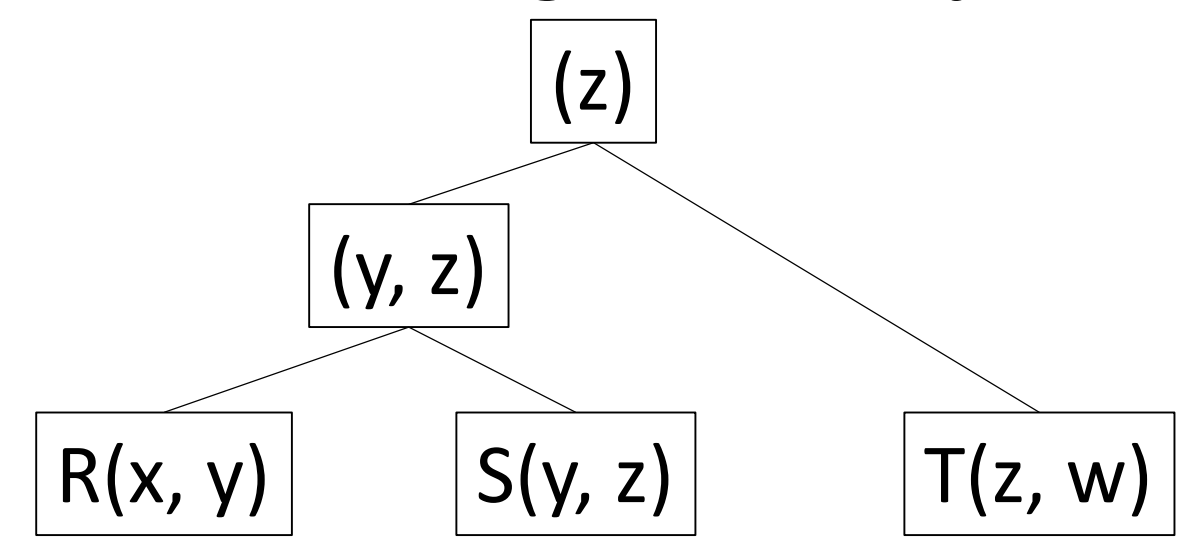
- Give up on full materialization of (sub) results
- Instead: maintain a data structure D of size $O(|DB|)$
- $Q(DB)$ can be generated with constant delay from D (independent of $|D|, |DB|, |Q(DB)|$)
- D can be efficiently maintained under updates



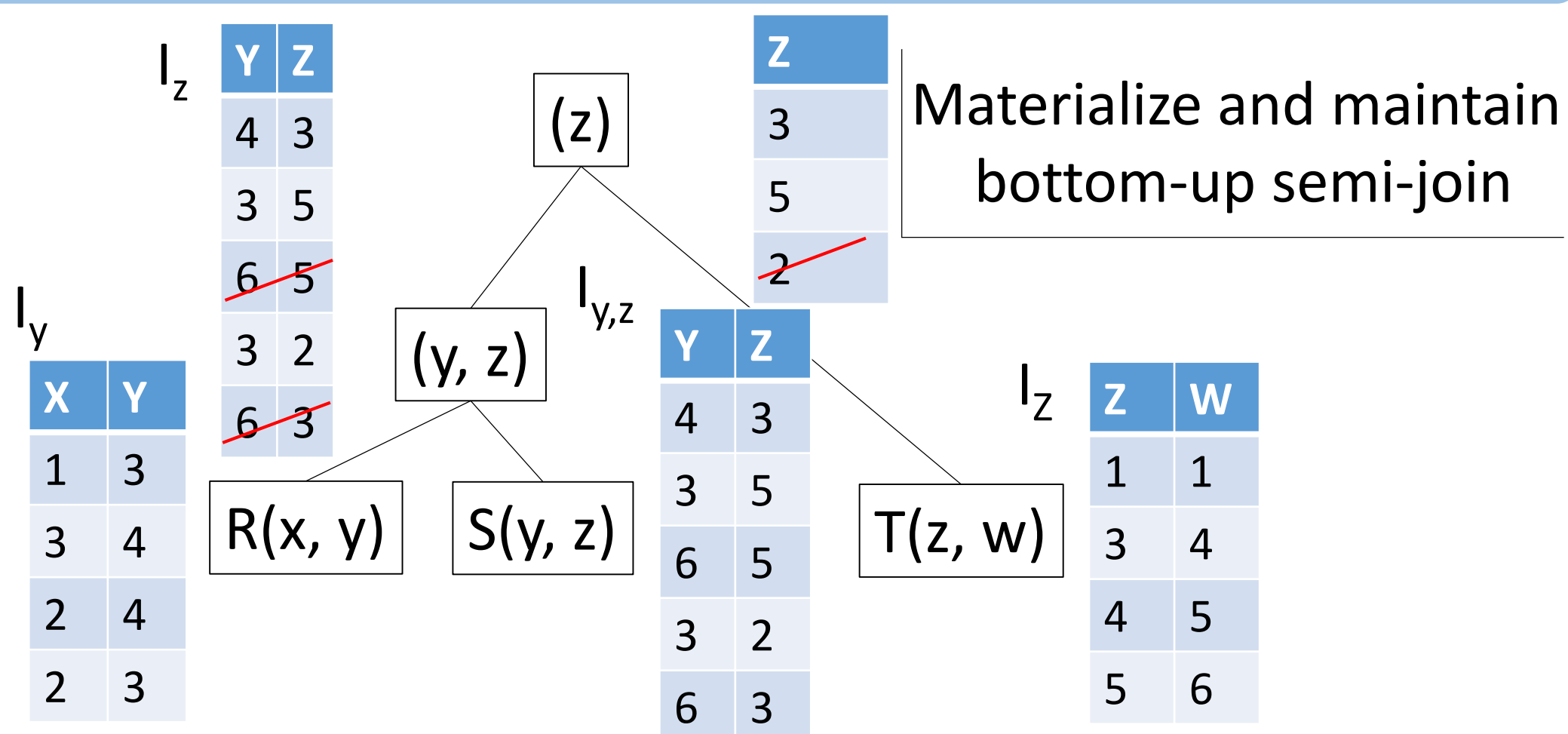
Conjunctive Acyclic Aggregate Queries

SELECT sum(Z), Y **FROM** R, S, T
WHERE R.y = S.y **AND** S.z = T.z
 $\sum_Z \pi_{z,y}(R(x,y) \bowtie S(y,z) \bowtie T(z,w))$

A join query is **acyclic** if it has a join tree.
 We work with **generalized join trees**.



Dynamic Yannakakis



Query Planning | Enumeration

How fast are single tuple updates?

