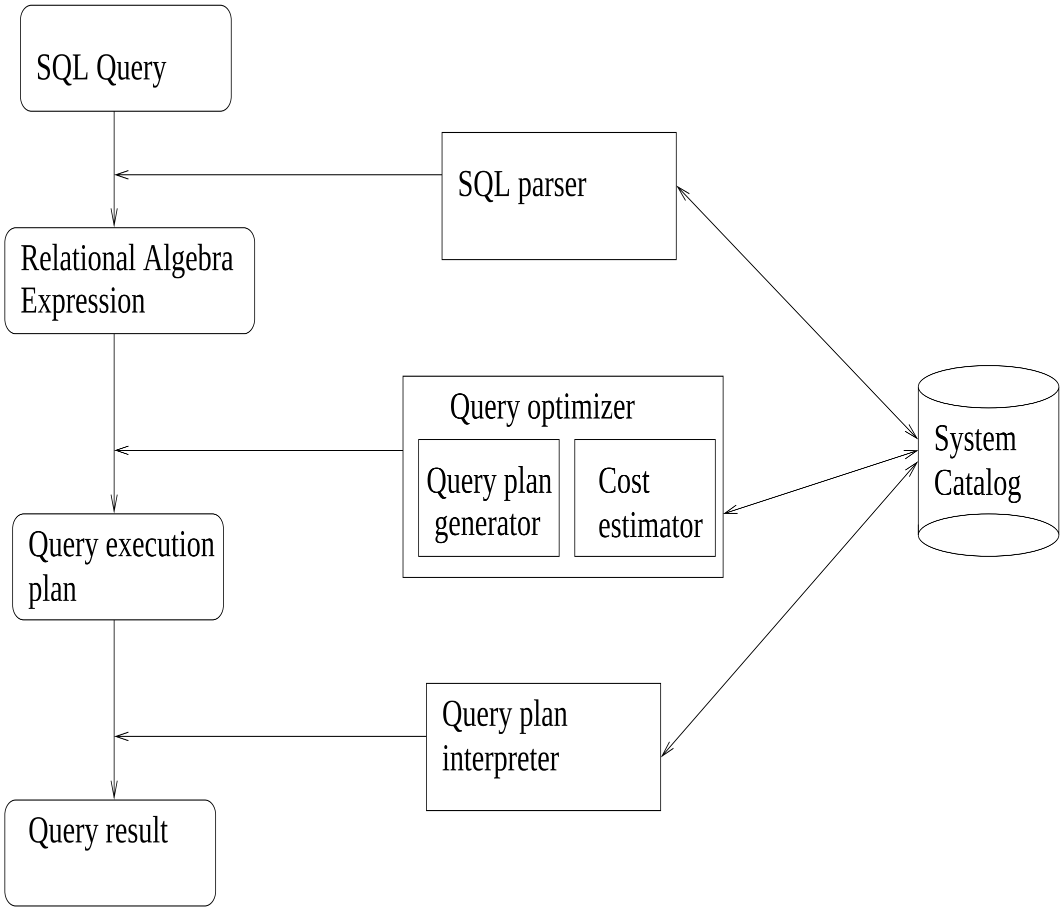
Relational Query Optimization

University of Toronto at Scarborough

CSCD34 - Data Management Systems

A. Vaisman

Query Processing and Optimization



Query Optimization Strategies

- System R optimization; works well for < 10 joins.
- A combination of heuristics and cost optimization.
- Statistics, maintained in system catalogs, used to estimate cost of operations and result sizes.
- Defines a good plan for accessing data: an 'Access Plan'.
- Plan Space: Too large, must be pruned.
- Only the space of left-deep plans is considered.
- Cartesian products avoided.

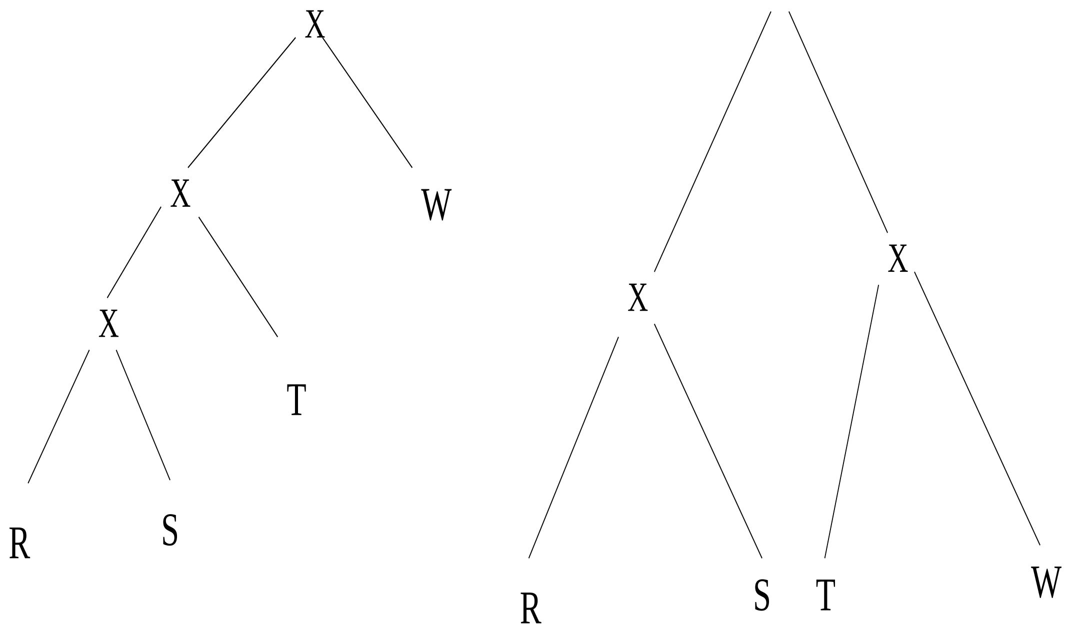
Query Optimization Strategies(cont.)

- Plan: Tree of R.A. ops, with choice of alg for each op.
- Each operator typically implemented using a 'pull interface: when an operator is 'pulled' for the next output tuples, it 'pulls' on its inputs and computes them. Two main issues:
  - For a given query, what plans are considered?
    - Algorithm to search plan space for cheapest (estimated) plan.
  - How is the cost of a plan estimated?
- Ideally: Want to find best plan. Practically: Avoid worst plans!
- We will study the System R approach.

Example

- Three relations: R(A,B,C,D); S(B,C,D,E); T(C,D,E,G).
- R:1,000 tuples;S: 100,000 tuples; T: 100 tuples.
- Image $I_{R.A} = \#$ of distinct values for attribute A in relation R
  - If there are three different brands in relation 'Product',$I_{Product.brand} = 3$.
- $I_{S.C} = 1{,}000$; $I_{T.E} = 100$.

Only Left-Deep Plans Considered

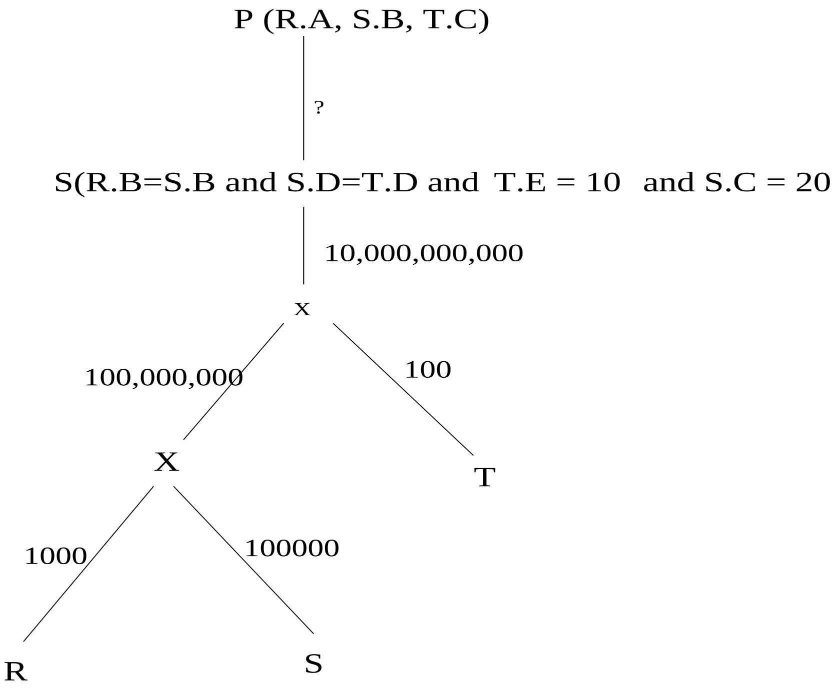


Canonical Query Tree

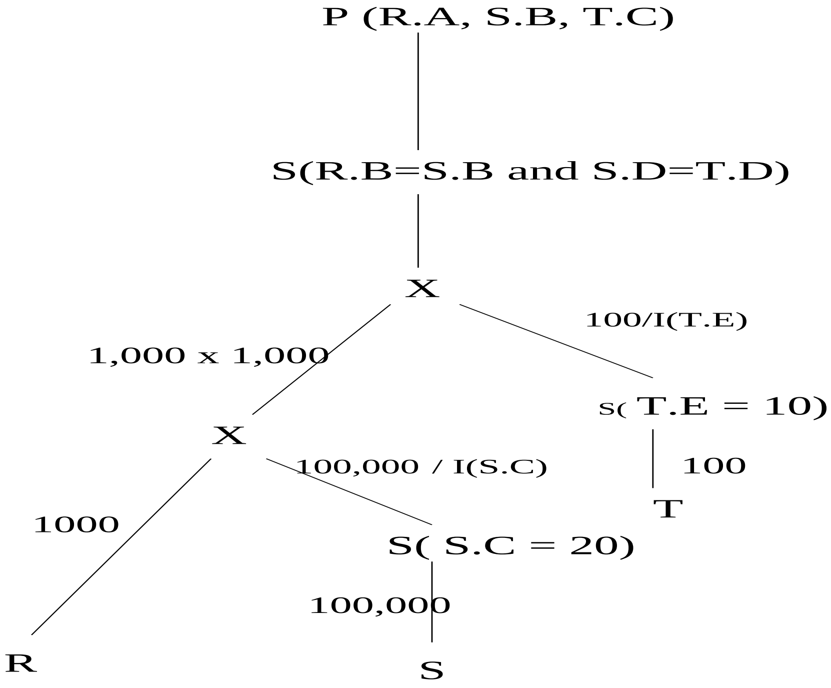
```
SELECT R.A,S.B,T.C

FROM R, S, T

WHERE R.B = S.B AND S.D = T.D AND T.E =10 AND S.C = 20
```

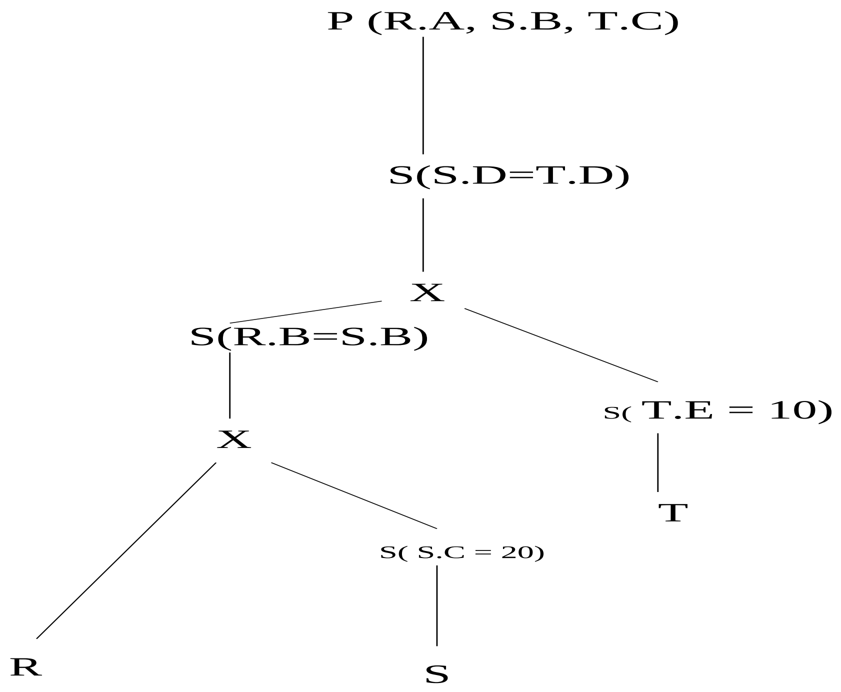


Push Down Selections



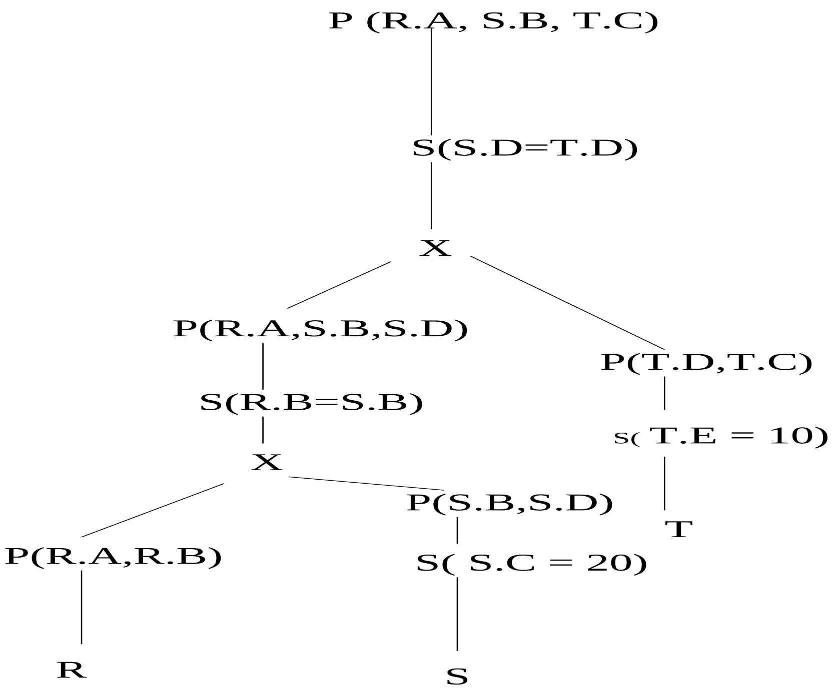
- Reduce the flow of tuples in the tree applying R.A equivalences.
- $\sigma_{c_1 \wedge \ldots \wedge c_n}(R) = \sigma_{c_1}(\ldots\sigma_{c_n}(R))$.
- $\sigma_{c_1}(\sigma_{c_2}(R)) = \sigma_{c_2}(\sigma_{c_1}(R))$

Push Down Selections(cont.)



- $(R \bowtie S) \bowtie T = R \bowtie (S \bowtie T)$.

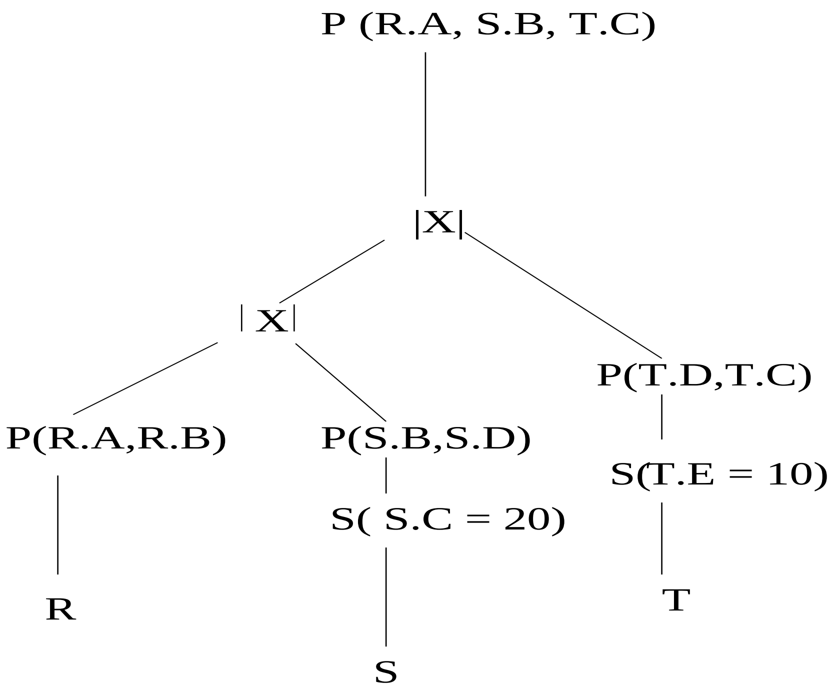
Push Down Projections



- $\Pi_{R.A,S.B}(R \bowtie_{A=B} S) = \Pi_A(R) \bowtie_{A=B} \Pi_B(S)$.

- $\Pi_{A,B,C}\sigma_{A=B \wedge B=C}(R) = \sigma_{A=B \wedge B=C}(\Pi_{A,B,C}(R)$.

Final Step



- Replace $\Pi, \sigma, \times$ by $\bowtie$.

Alternative Plans

- Single-relation plans.
- Multiple-relation plans.
- Queries over single relation:a combination of selects, projects, and aggregate ops.
  - Each available access path (file scan / index) is considered, and the one with the least estimated cost is chosen..

Alternative Plans(cont.)

- For each plan considered, must estimate cost:
  - Must estimate cost of each operation in plan tree.
    - Depends on input cardinalities.
    - We will discuss how to estimate the cost of operations (sequential scan, index scan, joins, etc.)
  - Must also estimate size of result for each operation in tree.
    - Use information about the input relations.

Alternative Plans(cont.)

- Left-deep plans differ only in the order of relations, the access method for each relation, and the join method for each join.
- Enumerated using N passes (if N relations joined):
  - Pass 1: Find best 1-relation plan for each relation.
  - Pass 2: Find best way to join result of each 1-relation plan to another relation. (all 2-relation plans.)
  - Pass N: Find best way to join result of a (N-1)-relation plan (as outer) to the Nth relation. (all N-relation plans.)
- For each subset of relations, retain only the cheapest plan overall.