Query Optimization
Exercise Session 6

Bernhard Radke

December 12, 2016
Generating Permutations

ConstructPermutationsRec\((P, R, B)\)

**Input:** a prefix \(P\), remaining relations \(R\), best plan \(B\)

**Output:** side effects on \(B\)

if \(|R| = 0\) {

  if \(B = \epsilon \lor C(B) > C(P)\) {
    \(B = P\)
  }
}

else {

  for each \(R_i \in R\) {
    if \(C(P \circ < R_i >) \leq C(P[1:|P| - 1] \circ < R_i, P[|P|]>)\) {
      ConstructPermutationsRec\((P \circ < R_i >, R \setminus \{R_i\}, B)\)
    }
  }
}
Generating Permutations

- Keep current prefix and the rest of relations
- Extend the prefix only if exchanging the last two relations does not result in a cheaper sequence
Memoization

- DP: bottom-up construction of the join tree
- Memoization: top-down construction
- Memoize already generated join tree to avoid duplicate work
- Sometimes more efficient
Algorithms: Roadmap

- Deterministic
  - Exact (IKKBZ, DP, Permutations, Memoization,...)
  - Heuristics (GOO, MVP, Query Simplification,...)
- Probabilistic
- Hybrid
Random left-deep trees with cross products

- there are \( n! \) trees (every tree - permutation)
- let’s generate a random number in \([0, n!]\)
- \textit{unranking} - for a generated number construct a tree
- \textit{ranking} - for a tree define it’s number
Generating random permutations

\[\text{for each } k \in [0, n[ \text{ descending} \]
\[\text{swap}(\pi[k], \pi[\text{random}(k)])\]

Array \(\pi\) initialized with elements \([0, n[. \)
\(\text{random}(k)\) generates a random number in \([0, k].\)
Unranking

Unrank\((n, r)\)

**Input:** the number \(n\) of elements to be permuted and the rank \(r\) of the permutation to be constructed

**Output:** a permutation \(\pi\)

for each \(0 \leq i < n\)

\[\pi[i] = i\]

for each \(n \geq i > 0\) descending \{ 
    swap\((\pi[i - 1], \pi[r \mod i])\)
    \[r = \lfloor r/i \rfloor\]
\}

return \(\pi\);
Random join trees with cross products

- Generate a tree, then generate a permutation: $C(n - 1)$ trees, $n!$ permutations
- Pick a random number $b \in [0, C(n - 1)]$, unrank $b$
- Pick a random number $p \in [0, n![$, unrank $p$
- Attach the permutation to the leaves
Unranking

- every tree is a word in \{ (, ) \}
- map such words to the grid, every step up is (, down)
Unranking

- every tree is a word in \{(, )\}
- map such words to the grid, every step up is (, down )
Unranking

- every tree is a word in \{(, )\}
- map such words to the grid, every step up is (, down )
- the number of different paths \( q \) can be computed (see lectures)
- Procedure: start in \((0,0)\), walk up as long as rank is smaller than \( q \). When it is bigger, step down, \( rank = rank - q \)
Example

- Bushy tree number 56, 8 leaves
Random Join Tree Selection
Random Join Tree Selection
Random Join Tree Selection

429
Random Join Tree Selection
Random Join Tree Selection

Diagram of a random join tree with points labeled 429 and 297.
Random Join Tree Selection
Random Join Tree Selection
Random Join Tree Selection
Random Join Tree Selection
Random Join Tree Selection
Random Join Tree Selection
Random Join Tree Selection

Diagram showing a tree structure with numbers at various nodes, including 429, 297, 165, 75, 27, 20.
Random Join Tree Selection
Random Join Tree Selection
Random Join Tree Selection
Random Join Tree Selection
Random Join Tree Selection
Random Join Tree Selection
Random Join Tree Selection
Random Join Tree Selection
Random Join Tree Selection
Random Join Tree Selection

![Random Join Tree Selection Diagram]
Random Join Tree Selection
Random Join Tree Selection
Random Join Tree Selection
Info

- Exercises due: 9 AM, December 19, 2016