

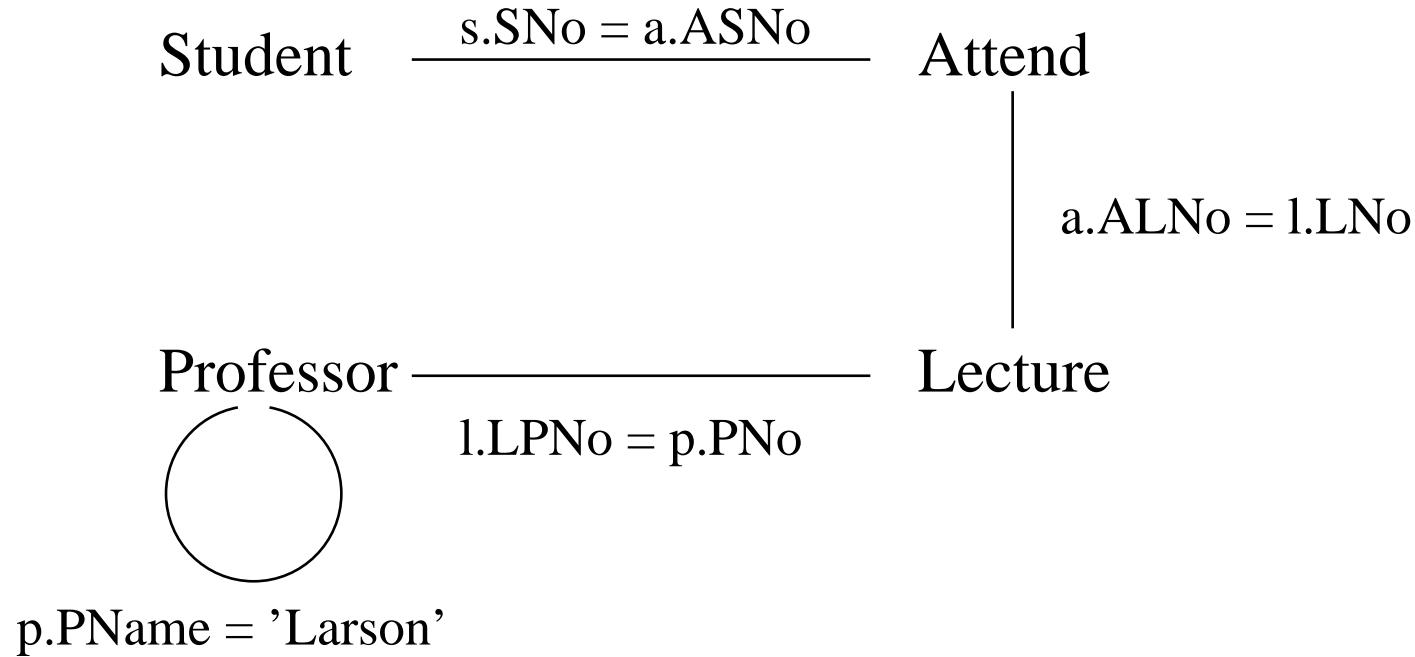
# **Dynamic Programming for Join Ordering Revisited**

Guido Moerkotte

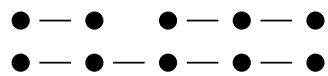
# Contents

1. Introduction
2. Algorithm DPsize
3. Algorithm DPsub
4. Analysis
5. Algorithm DPCCP

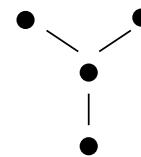
## Query Graph



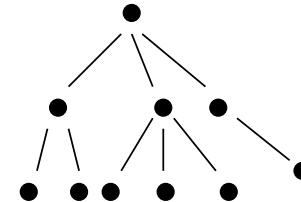
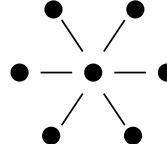
# Query Graph Types



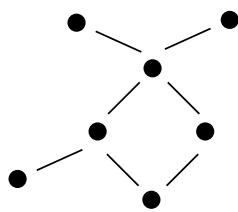
chain queries



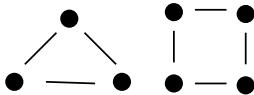
star queries



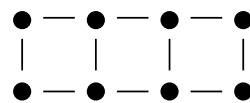
tree query



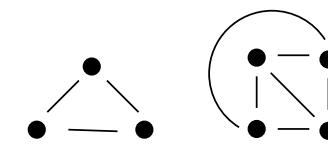
cyclic query



cycle queries



grid query



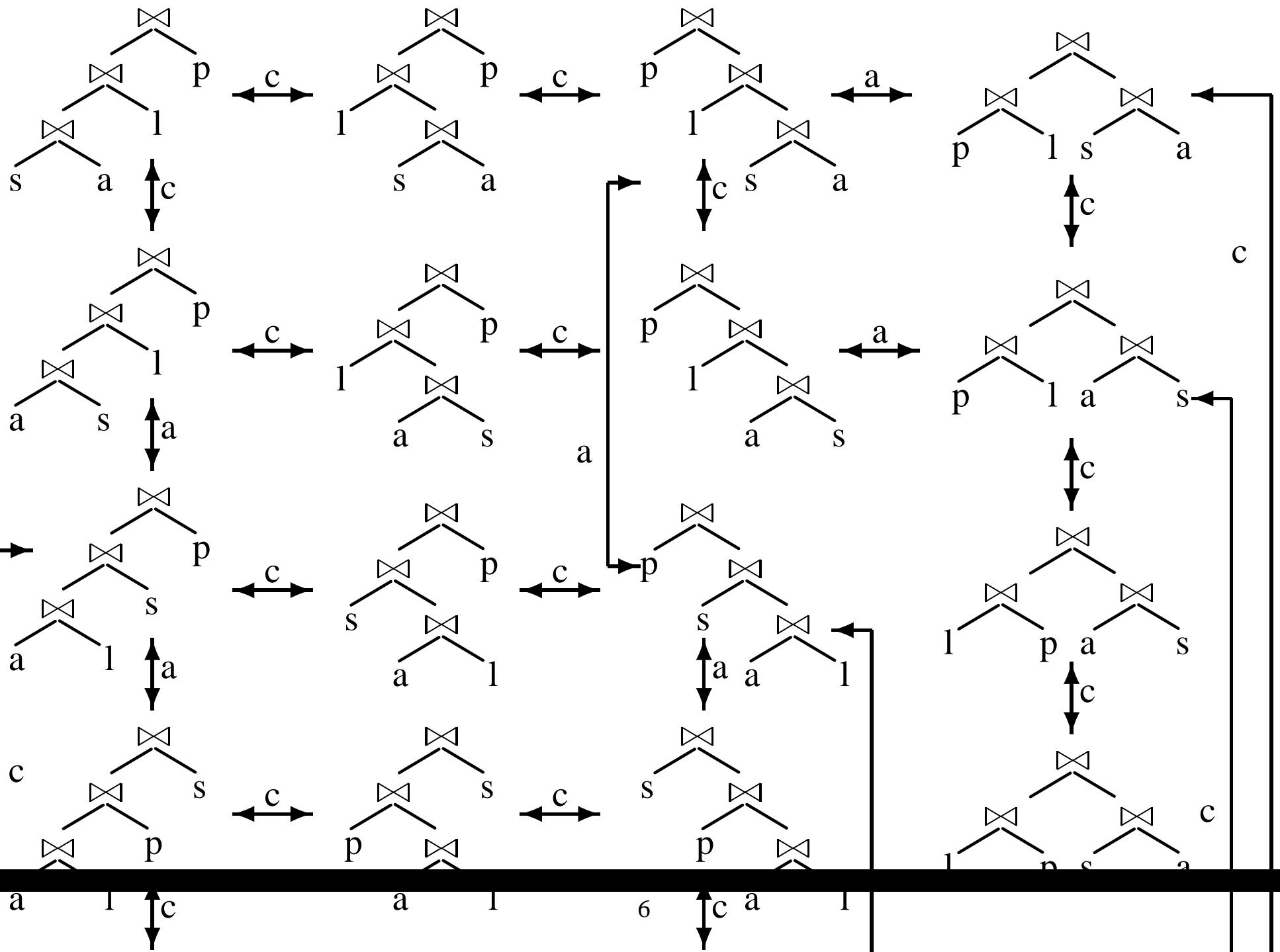
clique queries

## **Problem Definition**

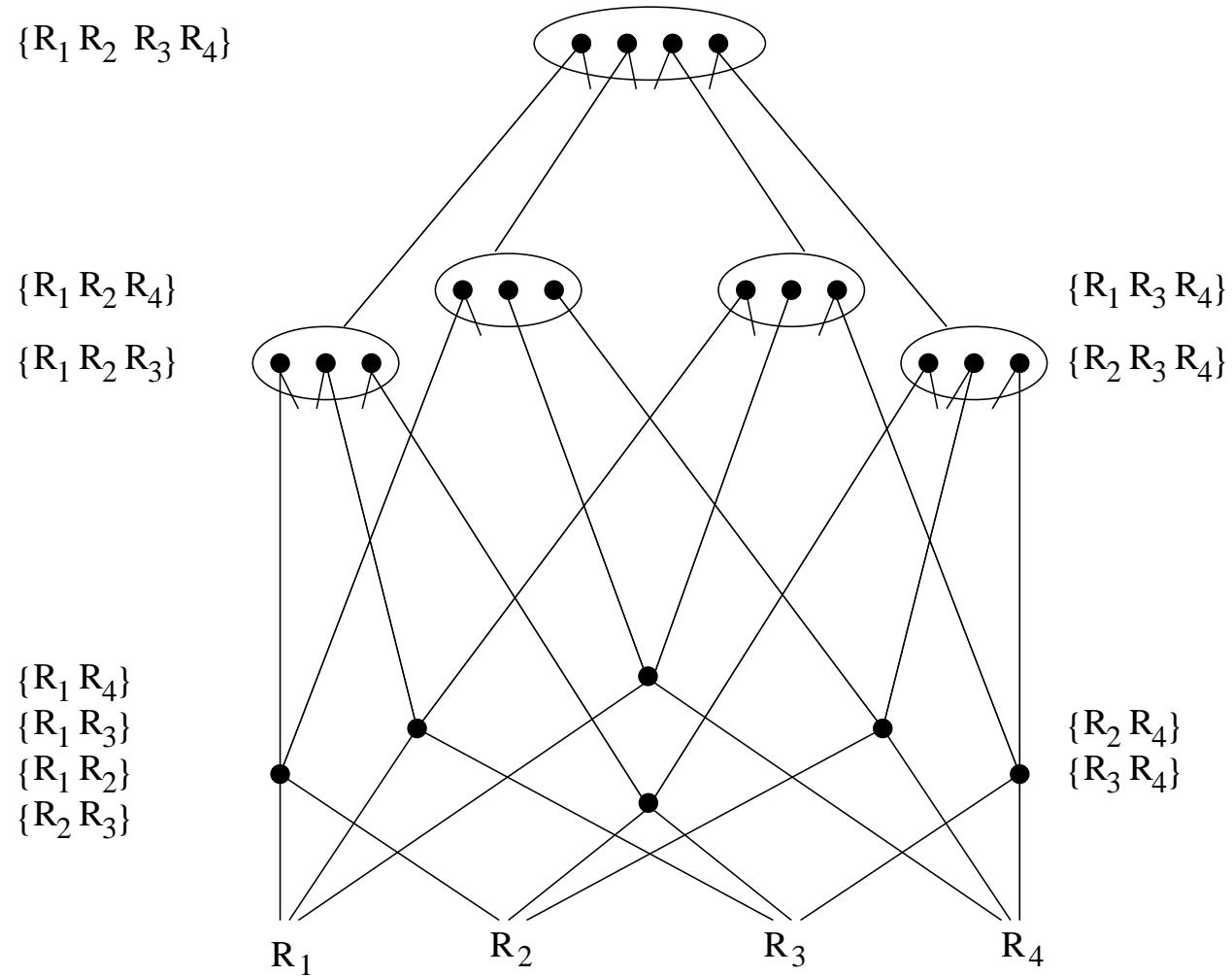
For a connected query graph:

Generate the best bushy join tree not containing a cross product.

Possible join trees (plans):



# Dynamic Programming



## csg-cmp-pairs

Let  $G = (V, E)$  be a connected graph. Let  $S_1, S_2 \subseteq V$  be non-empty. Then  $(S_1, S_2)$  is a *csg-cmp-pair* if

1.  $S_1$  is connected,
2.  $S_2$  is connected,
3.  $S_1 \cap S_2 = \emptyset$ ,
4. there exist nodes  $v_1 \in S_1$  and  $v_2 \in S_2$  such that there is an edge between  $v_1$  and  $v_2$  in the query graph.

**#csg and #ccp**

We denote by  $\#csg$  the number of connected subgraphs and by  $\#ccp$  the number of csg-cmp-pairs ( $\#ccp$ ).

- $\#csg$  is the number of plans to be stored in the DP-table
- $\#ccp$  is lower bound for DP

## Introduction: #csg and #ccp (Ono/Lohman)

$$\#csg^{\text{chain}}(n) = \frac{n(n+1)}{2}$$

$$\#csg^{\text{cycle}}(n) = n^2 - n + 1$$

$$\#csg^{\text{star}}(n) = 2^{n-1} + n - 1$$

$$\#csg^{\text{clique}}(n) = 2^n - 1$$

$$\#ccp^{\text{chain}}(n) = \frac{1}{3}((n+1)^3 - (n+1)^2 + 2 * (n+1))$$

$$\#ccp^{\text{cycle}}(n) = n^3 - 2n^2 + n$$

$$\#ccp^{\text{star}}(n) = (n-1)2^{n-2}$$

$$\#ccp^{\text{clique}}(n) = 3^n - 2^{n+1} + 1$$

## Algorithm DPsizE

```
for all  $R_i \in R$  BestPlan( $\{R_i\}$ ) =  $R_i$ ;  
for all  $1 < s \leq n$  ascending // size of plan  
for all  $1 \leq s_1 < s$  // size of left subplan  
     $s_2 = s - s_1$ ; // size of right subplan  
    for all  $p_1 = \text{BestPlan}(S_1 \subset R : |S_1| = s_1)$   
         $p_2 = \text{BestPlan}(S_2 \subset R : |S_2| = s_2)$   
        ++InnerCounter;  
        if ( $\emptyset \neq S_1 \cap S_2$ ) continue;  
        if not ( $S_1$  connected to  $S_2$ ) continue;  
        ++CsgCmpPairCounter;  
        CurrPlan = CreateJoinTree( $p_1, p_2$ );  
        if ( $\text{cost}(\text{BestPlan}(S_1 \cup S_2)) > \text{cost}(\text{CurrPlan})$ ) BestPlan( $S_1 \cup S_2$ ) = CurrPlan;  
    OnoLohmanCounter = CsgCmpPairCounter / 2;  
return BestPlan( $\{R_0, \dots, R_{n-1}\}$ );
```

## Algorithm DPsub

```
for all  $R_i \in R$  BestPlan( $\{R_i\}$ ) =  $R_i$ ;  
for  $1 \leq i < 2^n - 1$  ascending  
     $S = \{R_j \in R | (\lfloor i/2^j \rfloor \bmod 2) = 1\}$   
    if not (connected  $S$ ) continue;  
    for all  $S_1 \subset S$ ,  $S_1 \neq \emptyset$  do  
        ++InnerCounter;  $S_2 = S \setminus S_1$ ;  
        if ( $S_2 = \emptyset$ ) continue;  
        if not (connected  $S_1$ ) continue;  
        if not (connected  $S_2$ ) continue;  
        if not ( $S_1$  connected to  $S_2$ ) continue;  
        ++CsgCmpPairCounter;  $p_1 = \text{BestPlan}(S_1)$ ,  $p_2 = \text{BestPlan}(S_2)$ ;  
        CurrPlan = CreateJoinTree( $p_1$ ,  $p_2$ );  
        if ( $\text{cost}(\text{BestPlan}(S)) > \text{cost}(\text{CurrPlan})$ ) BestPlan( $S$ ) = CurrPlan;  
    OnLochmanCounter = CsgCmpPairCounter / 2;  
return BestPlan( $\{R_0, \dots, R_{n-1}\}$ );
```

## Analysis: DPsize

$$\begin{aligned}
 I_{\text{DPsize}}^{\text{chain}}(n) &= \begin{cases} 1/48(5n^4 + 6n^3 - 14n^2 - 12n) & n \text{ even} \\ 1/48(5n^4 + 6n^3 - 14n^2 - 6n + 11) & n \text{ odd} \end{cases} \\
 I_{\text{DPsize}}^{\text{cycle}}(n) &= \begin{cases} \frac{1}{4}(n^4 - n^3 - n^2) & n \text{ even} \\ \frac{1}{4}(n^4 - n^3 - n^2 + n) & n \text{ odd} \end{cases} \\
 I_{\text{DPsize}}^{\text{star}}(n) &= \begin{cases} 2^{2n-4} - 1/4 \binom{2(n-1)}{n-1} + q(n) & n \text{ even} \\ 2^{2n-4} - 1/4 \binom{2(n-1)}{n-1} + 1/4 \binom{n-1}{(n-1)/2} + q(n) & n \text{ odd} \end{cases} \\
 &\text{with } q(n) = n2^{n-1} - 5 * 2^{n-3} + 1/2(n^2 - 5n + 4) \\
 I_{\text{DPsize}}^{\text{clique}}(n) &= \begin{cases} 2^{2n-2} - 5 * 2^{n-2} + 1/4 \binom{2n}{n} - 1/4 \binom{n}{n/2} + 1 & n \text{ even} \\ 2^{2n-2} - 5 * 2^{n-2} + 1/4 \binom{2n}{n} + 1 & n \text{ odd} \end{cases}
 \end{aligned}$$

## Analysis: DPsub

$$I_{\text{DPsub}}^{\text{chain}}(n) = 2^{n+2} - n^2 - 3n - 4$$

$$I_{\text{DPsub}}^{\text{cycle}}(n) = n2^n + 2^n - 2n^2 - 2$$

$$I_{\text{DPsub}}^{\text{star}}(n) = 2 * 3^{n-1} - 2^n$$

$$I_{\text{DPsub}}^{\text{clique}}(n) = 3^n - 2^{n+1} + 1$$

## Sample Numbers

	Chain			Cycle		
$n$	#ccp	DPsub	DPsize	#ccp	DPsub	DPsize
5	20	84	73	40	140	120
10	165	3962	1135	405	11062	2225
15	560	130798	5628	1470	523836	11760
20	1330	4193840	17545	3610	22019294	37900
	Star			Clique		
$n$	#ccp	DPsub	DPsize	#ccp	DPsub	DPsize
5	32	130	110	90	180	280
10	2304	38342	57888	28501	57002	306991
15	114688	9533170	57305929	7141686	14283372	307173877
20	4980736	2323474358	59892991338	1742343625	3484687250	309338182241

## Algorithm DPCCP

```
for all ( $R_i \in \mathcal{R}$ ) BestPlan( $\{R_i\}$ ) =  $R_i$ ;  
forall csg-cmp-pairs  $(S_1, S_2)$ ,  $S = S_1 \cup S_2$   
    ++InnerCounter;  
    ++OnoLohmanCounter;  
     $p_1 = \text{BestPlan}(S_1)$ ;  
     $p_2 = \text{BestPlan}(S_2)$ ;  
    CurrPlan = CreateJoinTree( $p_1, p_2$ );  
    if ( $\text{cost}(\text{BestPlan}(S)) > \text{cost}(\text{CurrPlan})$ )  
        BestPlan( $S$ ) = CurrPlan;  
    CurrPlan = CreateJoinTree( $p_2, p_1$ );  
    if ( $\text{cost}(\text{BestPlan}(S)) > \text{cost}(\text{CurrPlan})$ )  
        BestPlan( $S$ ) = CurrPlan;  
CsgCmpPairCounter = 2 * OnoLohmanCounter;  
return BestPlan( $\{R_0, \dots, R_{n-1}\}$ );
```

## Notation

Let  $G = (V, E)$  be an undirected graph.

For a node  $v \in V$  define the *neighborhood*  $\mathcal{N}(v)$  of  $v$  as

$$\mathcal{N}(v) := \{v' | (v, v') \in E\}$$

For a subset  $S \subseteq V$  of  $V$  we define the *neighborhood* of  $S$  as

$$\mathcal{N}(S) := \cup_{v \in S} \mathcal{N}(v) \setminus S$$

The neighborhood of a set of nodes thus consists of all nodes reachable by a single edge.

Note that for all  $S, S' \subset V$  we have  $\mathcal{N}(S \cup S') = (\mathcal{N}(S) \cup \mathcal{N}(S')) \setminus (S \cup S')$ . This allows for an efficient bottom-up calculation of neighborhoods.

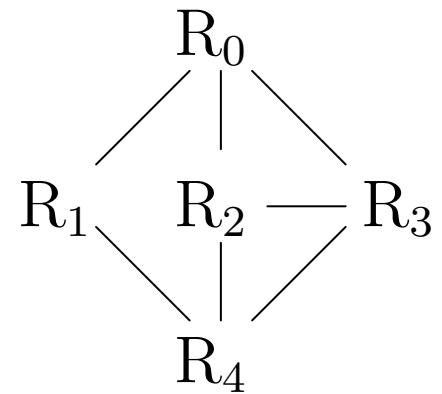
$$\mathcal{B}_i = \{v_j | j \leq i\}$$

## Algorithm EnumerateCsg

```
for (i = n - 1; i ≥ 0; --i) {  
    emit {vi};  
    EnumerateCsgRec(G, {vi}, Bi);  
}
```

```
EnumerateCsgRec(G, S, X)  
N = N(S) \ X;  
for all (S' ⊆ N; S' ≠ ∅) {  
    emit (S ∪ S');  
}  
for all (S' ⊆ N; S' ≠ ∅) {  
    EnumerateCsgRec(G, (S ∪ S'), (X ∪ N));  
}
```

## Example



$S$	$X$	$N$	$\text{emit}/S$
$\{4\}$	$\{0, 1, 2, 3, 4\}$	$\emptyset$	
$\{3\}$	$\{0, 1, 2, 3\}$	$\{4\}$	$\{3, 4\}$
$\{2\}$	$\{0, 1, 2\}$	$\{3, 4\}$	$\{2, 3\}$
			$\{2, 4\}$
			$\{2, 3, 4\}$
$\{1\}$	$\{0, 1\}$	$\{4\}$	$\{1, 4\}$
$\rightarrow \{1, 4\}$	$\{0, 1, 4\}$	$\{2, 3\}$	$\{1, 2, 4\}$
			$\{1, 3, 4\}$
			$\{1, 2, 3, 4\}$

## Algorithm EnumerateCmp

EnumerateCmp

**Input:** a connected query graph  $G = (V, E)$ , a connected subset  $S_1$

**Precondition:** nodes in  $V$  are numbered according to a breadth-first search

**Output:** emits all complements  $S_2$  for  $S_1$  such that  $(S_1, S_2)$  is a csg-cmp-pair

$X = \mathcal{B}_{\min(S_1)} \cup S_1;$

$N = \mathcal{N}(S_1) \setminus X;$

**for all**  $(v_i \in N \text{ by descending } i)$  {

**emit**  $\{v_i\}$ ;

    EnumerateCsgRec( $G, \{v_i\}, X \cup N$ );

}

where  $\min(S_1) := \min(\{i | v_i \in S_1\})$ .

## Evaluation

n	DPSIZE	DPSUB	DPCCP
chain queries			
5	7.7e-6	9.7e-6	9.2e-6
10	5.8e-5	0.00018	6.4e-5
15	0.0013	0.0056	0.0013
20	0.048	0.22	0.048
cycle queries			
5	1.1e-5	1.5e-5	1.4e-5
10	0.0001	0.00031	0.00012
15	0.001	0.01	0.0015
20	0.049	0.47	0.048

## Evaluation

n	DPSIZE	DPsub	DPCCP
star queries			
5	9.8e-6	1.2e-5	1.0e-5
10	0.00069	0.0008	0.00044
15	0.71	0.1	0.022
20	4791	42.7	1.00
clique queries			
5	2.1e-5	2.4e-5	2.4e-5
10	0.0058	0.0048	0.005
15	4.6	1.2	1.3
20	21294	439	529

## Conclusion

DPCCP is the algorithm of choice.

Paper: <http://pi3.informatik.uni-mannheim.de> -> Publications