They used above property to design PW-AC algorithm (Samaras & Woodward 2013).

Each block is supported by at most one other block.

We compute and store fine and coarse blocks at preprocessing.

### Contributions
1. Designed **PERFB**, an algorithm for enforcing $R(*,m)C$, exploiting the fact that constraints in dual CSP are piecewise functional.
2. Compared performance of **PERFB** and **PERTUPLE** (previous algorithm) to empirically establish improvements.

### Relational Consistency
$R(*,m)C$, m-wise consistency, ensures that every combination of m relations is minimal.

**PERTUPLE** enforces $R(*,m)C$ [Karakashian+ AAA10]
- Given all combinations of m relations
- For each relation in each combination
  - **SEARCH** (i.e., a backtrack search with FC) ensures each tuple can be extended to the other m-1 relations
  - If no solution is found, tuple is removed

### Piecewise Functional Constraints
Samaras & Stergiou [JAIR 05] noted that the constraints in the dual CSP are piecewise functional
1. Each relation can be partitioned into blocks of equivalent tuples
2. Each block is supported by at most one other block
They used above property to design PW-AC algorithm ($m=2$)

### From PERTUPLE To PERFB
**PERFB** makes fewer calls to **SEARCH** than **PERTUPLE**
1. **PERFB** iterates over fine blocks rather than tuples
2. At each call, it dynamically determines the intermediate blocks induced on a relation by the considered other relations.

Considering relations $R_1, R_2, R_3$
- The union of the subscopes of $R_1$, $R_2$ and $R_3$ determines the intermediate partition induced by $R_2 R_3$ on $R_1$
- Projecting a fine block over this union forms a signature of a fine block.
- Once **SEARCH** finds (or not) a support for a fine block, it reuses this result for future fine blocks with the same signature.

**Future Research**
- Extend our approach to **ALLSol**, our other algorithm for enforcing minimality of m relations [Karakashian PhD 13]