Motivation

- What is a matrix model?
- What are row and column symmetries?
- Why bother?

What is a Matrix Model?

- Constraint program that contains (one or more) matrices of decision variables.
- Benefits
  - Effective representation of a problem
  - Efficient solving of the model
  - Captures common modelling patterns

Example: Sports Scheduling

- Given **n** weeks and **n/2** periods for every week
  - schedule a match for every week and period.

	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7
Period 1	0 vs 1	0 vs 2	4 vs 7	3 vs 6	3 vs 7	1 vs 5	2 vs 4
Period 2	2 vs 3	1 vs 7	0 vs 3	5 vs 7	1 vs 4	0 vs 6	5 vs 6
Period 3	4 vs 5	3 vs 5	1 vs 6	0 vs 4	2 vs 6	2 vs 7	0 vs 7
Period 4	6 vs 7	4 vs 6	2 vs 5	1 vs 2	0 vs 5	3 vs 4	1 vs 3

- Each <w,p> corresponds to the match played on period p of week w

Diversity of Matrix Models

- Many problems in diverse domains can naturally be modelled and effectively solved using matrix models
- Combinatorial problems
  - BIBDs, magic squares, projective planes, …
- Design
  - Rack configuration, template and slab design, …
- Scheduling
  - Classroom, social golfer, sports scheduling, …
- Assignment
  - Warehouse location, progressive party, …
- …

What are Row and Column Symmetries?

- In a CSP
  - Symmetry involves the variables, the values, or both
  - Maps each search state (partial assignment, solution, failure etc) into an equivalent one.
- In a matrix model
  - Rows and/or columns can represent objects which are indistinguishable and are therefore symmetric.
- We can permute any two rows or columns (or both).

Example: Sports Scheduling

- Weeks are indistinguishable
- Periods are indistinguishable

	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7
Period 1	0 vs 1	0 vs 2	4 vs 7	3 vs 6	3 vs 7	1 vs 5	2 vs 4
Period 2	2 vs 3	1 vs 7	0 vs 3	5 vs 7	1 vs 4	0 vs 6	5 vs 6
Period 3	4 vs 5	3 vs 5	1 vs 6	0 vs 4	2 vs 6	2 vs 7	0 vs 7
Period 4	6 vs 7	4 vs 6	2 vs 5	1 vs 2	0 vs 5	3 vs 4	1 vs 3

Example: Sports Scheduling

- Weeks are indistinguishable
- Periods are indistinguishable

	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7
Period 1	0 vs 1	0 vs 2	4 vs 7	3 vs 6	3 vs 7	1 vs 5	2 vs 4
Period 2	2 vs 3	1 vs 7	0 vs 3	5 vs 7	1 vs 4	0 vs 6	5 vs 6
Period 3	4 vs 5	3 vs 5	1 vs 6	0 vs 4	2 vs 6	2 vs 7	0 vs 7
Period 4	6 vs 7	4 vs 6	2 vs 5	1 vs 2	0 vs 5	3 vs 4	1 vs 3

Example: Sports Scheduling

- Weeks are indistinguishable
- Periods are indistinguishable

	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7
Period 1	0 vs 1	0 vs 2	4 vs 7	3 vs 6	3 vs 7	1 vs 5	2 vs 4
Period 2	2 vs 3	1 vs 7	0 vs 3	5 vs 7	1 vs 4	0 vs 6	5 vs 6
Period 3	4 vs 5	3 vs 5	1 vs 6	0 vs 4	2 vs 6	2 vs 7	0 vs 7
Period 4	6 vs 7	4 vs 6	2 vs 5	1 vs 2	0 vs 5	3 vs 4	1 vs 3

Example: Sports Scheduling

- Weeks are indistinguishable
- Periods are indistinguishable

	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7
Period 1	0 vs 1	3 vs 7	4 vs 7	3 vs 6	0 vs 2	1 vs 5	2 vs 4
Period 2	2 vs 3	1 vs 4	0 vs 3	5 vs 7	1 vs 7	0 vs 6	5 vs 6
Period 3	4 vs 5	2 vs 6	1 vs 6	0 vs 4	3 vs 5	2 vs 7	0 vs 7
Period 4	6 vs 7	0 vs 5	2 vs 5	1 vs 2	4 vs 6	3 vs 4	1 vs 3

Example: Sports Scheduling

- Weeks are indistinguishable
- Periods are indistinguishable

	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7
Period 1	0 vs 1	3 vs 7	4 vs 7	3 vs 6	0 vs 2	1 vs 5	2 vs 4
Period 2	2 vs [illegible]	1 vs 4	0 vs 3	5 vs 7	1 vs 7	0 vs 6	5 vs 6
Period 3	4 vs 5	2 vs 6	1 vs 6	0 vs 4	3 vs 5	2 vs 7	0 vs 7
Period 4	6 vs 7	0 vs 5	2 vs 5	1 vs 2	4 vs 6	3 vs 4	1 vs 3

Why Bother?

- For **nxm** matrix with row and colum symmetry
  - there are **n!*m!** symmetries (super-exponentinal)
- It can be very expensive to search symmetric and failed branches of the search tree
- Eliminating all symmetries is not easy
  - Exact methods have to deal with very large number of symmetries
  - The effort required could easily be exponential
- Can we reduce many of the symmetries in a simplier way?

Our Contribution

- Identify an important class of symmetries that occur frequently in CSPs
  - A matrix of decision variables in which row and/or columns can be swapped.
- Show how simple constraints can be added to such matrix models to break these symmetries
- Extend our results to deal with
  - matrices with more than 2 dimensions
  - partial symmetries (ie, strict subsets of the rows/columns are symmetric)
  - symmetric values (e.g. teams in Sports Scheduling)

Overview of Rest of Talk

- Breaking Row and/or Column Symmetries
- Extensions
- Effectiveness via Experimental Results
- Breaking All Symmetries
- Conclusions and Future Work

Breaking Row (Column) Symmetry

- Lexicographic Ordering (used to order dictionaries)

  $[A,B,C] \leq_{lex} [D,E,F]$

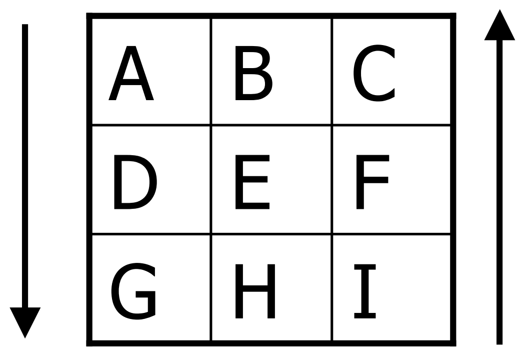
  - A<D or
  - (A=D and B<E) or
  - (A=D and B=E and C<F) or
  - (A=D and B=E and C=F)

Breaking Row (Column) Symmetry

- Lexicographic ordering is total
- Forcing the rows to be lexicographically ordered breaks all row symmetry

lexicographic ordering

$[A\ B\ C] \leq_{lex} [D\ E\ F] \leq_{lex} [G\ H\ I]$



anti-lexicographic ordering

$[G\ H\ I] \leq_{lex} [D\ E\ F] \leq_{lex} [A\ B\ C]$

Breaking Row and Column Symmetries

- Breaking both row and column symmetries is difficult
- Rows and columns intersect
- After constraining the rows to be lexicographically ordered
  - we distinguish the columns
  - the columns are not symmetric anymore!

Good News ☺

- A symmetry class is an equivalence class of assignments
  - two assignments are equivalent if there is a symmetry mapping one assignment into the other
- Each symmetry class of assignments has at least one element where both the rows and the columns are lexicographically ordered
  - But there may be no element with rows lex ordered and columns anti-lex ordered
- To break row and column symmetries, we can insist that the rows and columns are both lexicographically ordered (double-lex)

Bad news ☹

- A symmetry class of assignments may have more than one element where both the rows and the columns are lexicographically ordered
- Double-lex does not break all row and column symmetries

0	1
0	1
1	0

swap the columns
swap row 1 and row 3

0	1
1	0
1	0

How Effective is Double-lex?

- BIBD problem <7, 7, 3, 3, 1>

	Obj_1						Obj_7	
$Block_1$	0	1	1	0	0	1	0	$\Sigma = 3$
	1	0	1	0	1	0	0	
	0	0	1	1	0	0	1	.
	1	1	0	0	0	0	1	.
	0	0	0	0	1	1	1	.
	1	0	0	1	0	1	0	
$Block_7$	0	1	0	1	1	0	0	$\Sigma = 3$
	$\Sigma = 3$			. . .			$\Sigma = 3$	

BIBD

- Example: <7, 7, 3, 3, 1>

	Obj_1						Obj_7
$Block_1$	0	1	1	0	0	1	0
	1	0	1	0	1	0	0
	0	0	1	1	0	0	1
	1	1	0	0	0	0	1
	0	0	0	0	1	1	1
	1	0	0	1	0	1	0
$Block_7$	0	1	0	1	1	0	0

BIBD

- Example: <7, 7, 3, 3, 1>

	Obj_1						Obj_7
$Block_1$	0	1	1	0	0	1	0
	1	0	1	0	1	0	0
	0	0	1	1	0	0	1
	1	1	0	0	0	0	1
	0	0	0	0	1	1	1
	1	0	0	1	0	1	0
$Block_7$	0	1	0	1	1	0	0

BIBD

- Example: <7, 7, 3, 3, 1>

	Obj_1						Obj_7
$Block_1$	0	1	1	0	0	1	0
	1	0	1	0	1	0	0
	0	0	1	1	0	0	1
	1	1	0	0	0	0	1
	0	0	0	0	1	1	1
	1	0	0	1	0	1	0
$Block_7$	0	1	0	1	1	0	0

BIBD

- Example: <7, 7, 3, 3, 1>

	Obj_1						Obj_7
$Block_1$	0	1	1	0	0	1	0
	1	0	1	0	1	0	0
	0	0	1	1	0	0	1
	1	1	0	0	0	0	1
	0	0	0	0	1	1	1
	1	0	0	1	0	1	0
$Block_7$	0	1	0	1	1	0	0

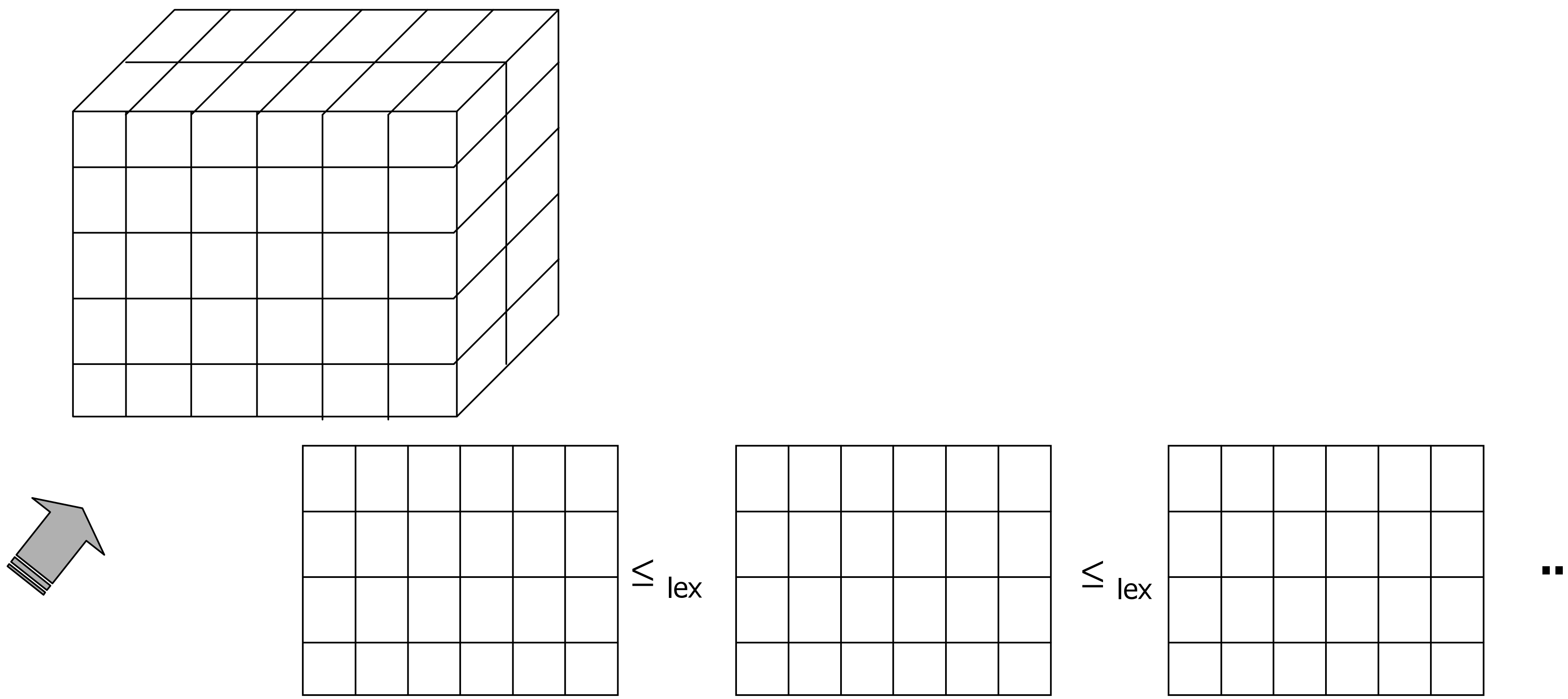
Experimental Results

- Double-lex reduces the total number of solutions from the orders of millions to the orders of tens

- Double-lex breaks much more symmetry than
  - imposing lexicographic ordering constraints only on the rows
  - imposing lexicographic ordering constraints only on the columns
  - setting the first row and column

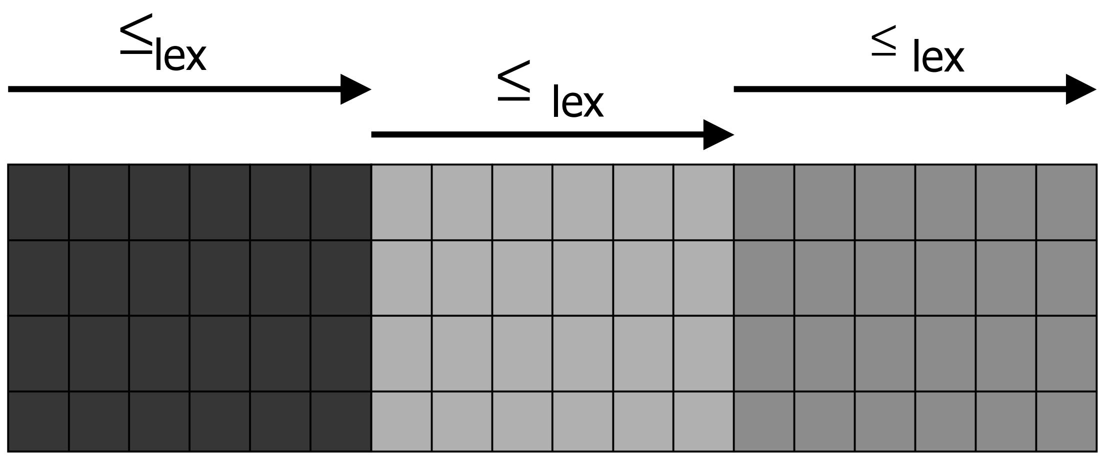
Extensions I: Higher Dimensions

- Given an n-dimensional matrix, at any dimension exhibiting symmetry, insist that the slices are lexicographically ordered



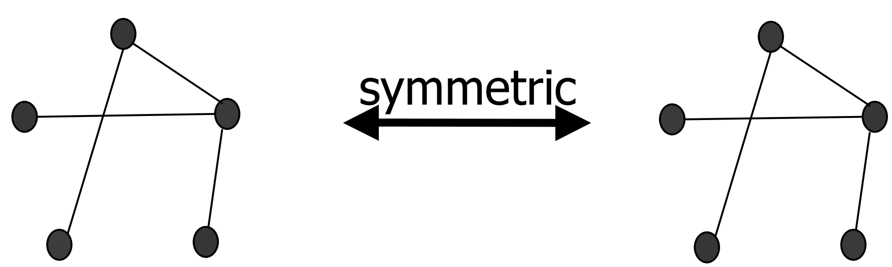
Extensions II: Partial Symmetry

- When strict subsets of the rows/columns are indistinguishable
- Impose lexicographic ordering constraints only on the rows/columns of the subsets



Extensions III: Value Symmetry

- The values are indistinguishable
- Values of variables can freely be permuted
- Example
  - Vertex Colouring
    - Variables: Vertices
    - Values: Colours
    - Assign a colour to every vertex such that neighbouring vertices are assigned different colours



Extensions III: Value Symmetry

indistingusihable values

X	Y	Z
{1,2}	{2,3}	{3}

indistingusihable rows

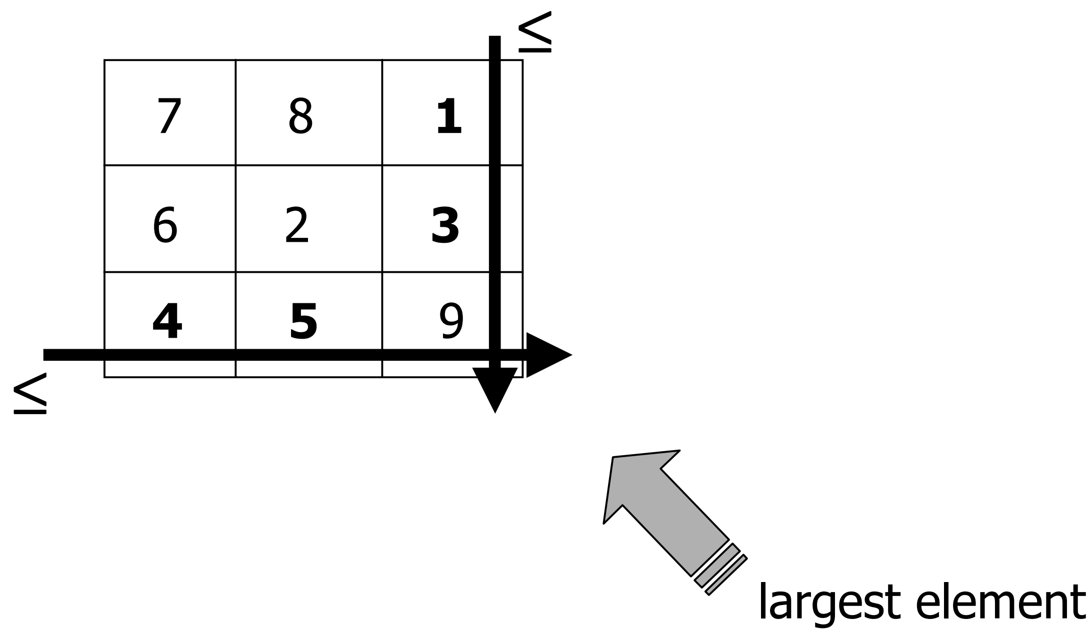
	X	Y	Z
V_1	0..1	0	0
V_2	0..1	0..1	0
V_3	0	0..1	1

$\Sigma=1$ $\Sigma=1$ $\Sigma=1$

$V_1 \leq_{lex} V_2 \leq_{lex} V_3$

Breaking All Symmetries

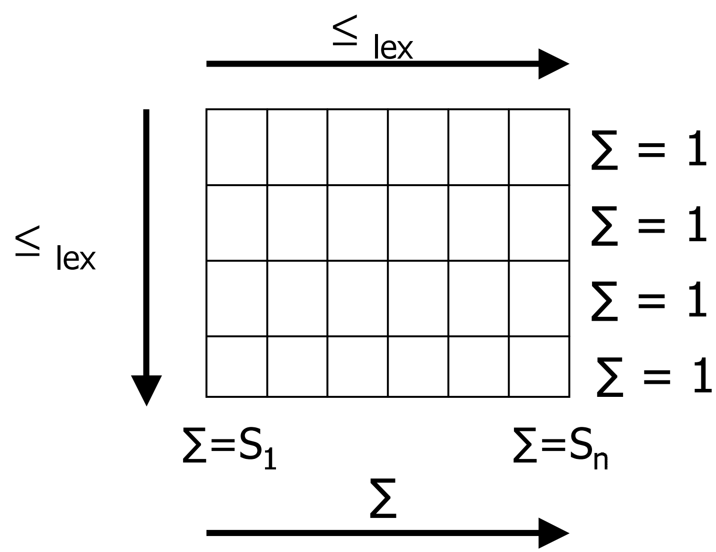
- It is possible to break all symmetry when
  - all values in the matrix are distinct



- Example application? Magic squares?

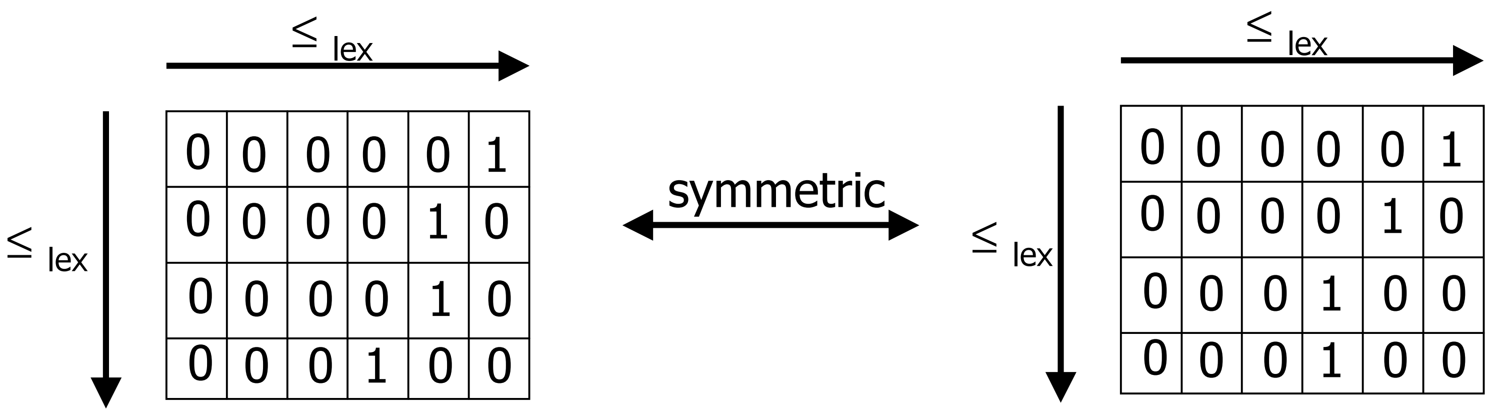
Breaking All Symmetries

- It is possible to break all symmetry when
  - every row of a 0/1 matrix must have a single 1
- Many Problems
  - Slab design, rack design, ...



$S_1 \leq S_2 \leq \ldots \leq S_n$

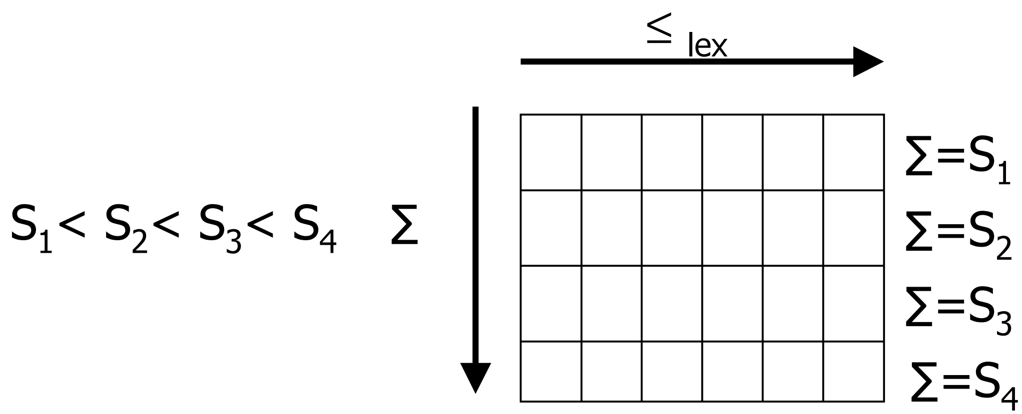
Why?



- The 1 in the next row must occur either directly below or one column to the left
- The only freedom is how many consecutive rows have 1s in the same column
- We break this symmetry by constraining the columns to be ordered by their sums

Breaking All Symmetries

- It is possible to break all symmetry when
  - every row sum is different, but we do not know the sums



alldifferent($[S_1,S_2,S_3,S_4]$)

- Ordering row sums breaks all row symmetry
  - The columns are still indistinguishable
- Lexicographic ordering columns then breaks all the symmetries

Conclusions

- Many CSPs can be naturally modelled by multi-dimensional matrices of decision variables.
- Row and column symmetries are very common in matrix models.
- An **n**x**m** matrix with row and column symmetries exhibits super-exponential number of symmetries
- Breaking all such symmetries is difficult
  - No one has an effective way of dealing with all row and column symmetries

Conclusions (cont'd)

- Constraining both the rows and columns to be lexicographically ordered breaks considerable amount of symmetries, if not all
- Posing lexicographic ordering constraint
  - GAC on $\leq_{lex}$ is O(n)
  - "Global Constraints for Lexicographic Orderings"
    by Ian Miguel on Thursday!

Conclusions (cont'd)

- Results can be extended cope with
  - symmetries in higher dimensions
  - partial symmetries
  - symmetric values
- It is sometimes possible to break all row and column symmetries

Future Work

- After imposing double-lex, how many symmetries remain?
- Is it worth trying to break the remaining symmetries?
- Can we devise DVOs to work well with (double-)lex?
- What other orderings can we impose?