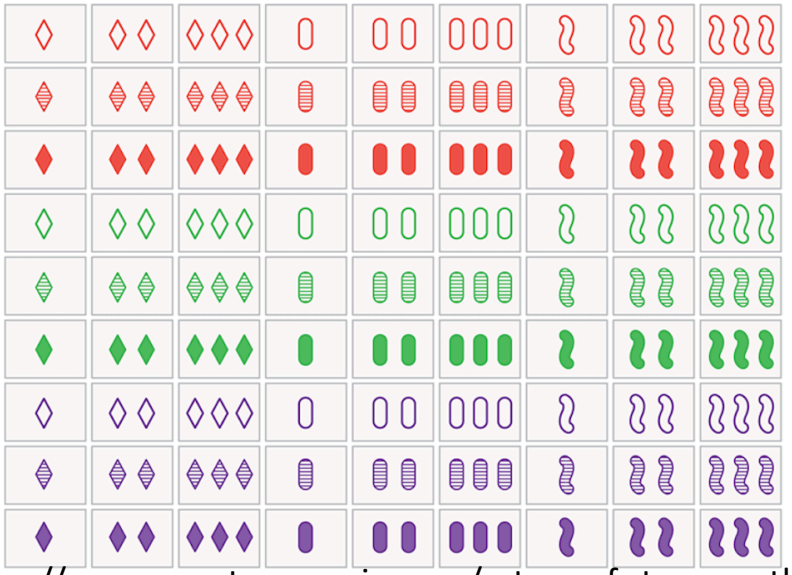
Solving the Generalized Form of the Game of Set Efficiently

STEVEN TAKESHITA'19, ADVISED BY PROFESSOR ZACHARY KINCAID

Motivation, Goal, and Problem Definition

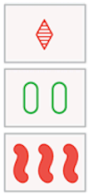
3 Values and 4 Properties

THE FULL DECK



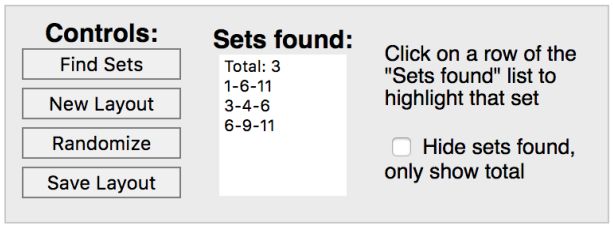
SET OR NO SET

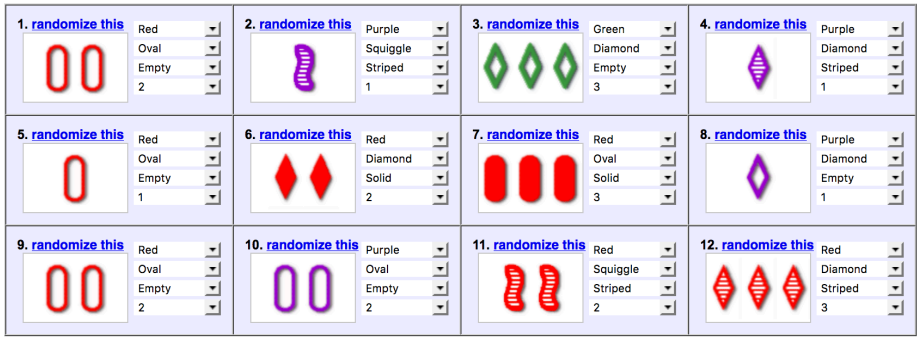
Some examples below:

Are the attributes all the same or all different?				
Color	✖	✖	all different	all the same
Shape	all different	all different	all different	all the same
Shading	all different	all different	all different	all the same
Number	✖	all different	all different	all different
	Not a set	Not a set	A set	A set

Source: https://www.quantamagazine.org/set-proof-stuns-mathematicians-20160531/

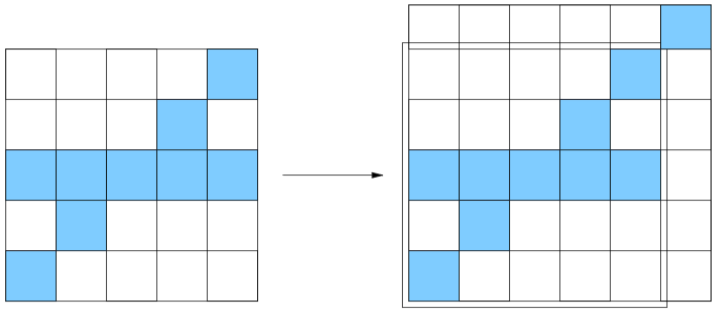
Related Work and Problem Background





Source: Nolte, *JavaScript Set Game Solver*

Perfect Dimensional Matching

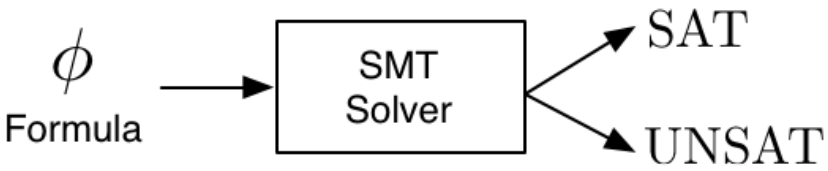


NP Complete

Source: Chadhuri et al (2003)

Approach

SMT Solver



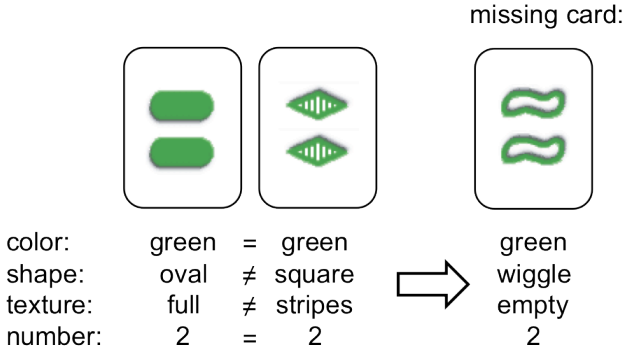
Ex.

Integer X. Constraint X + 1 = 2.

SAT => X = 1

Source: http://dreal.cs.cmu.edu/presentation/20130612/#/sec-1

Dynamic Algorithm

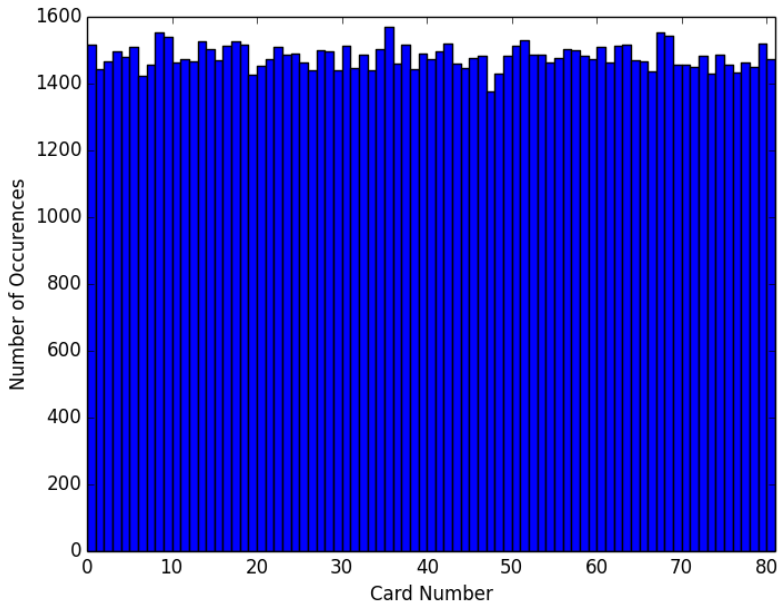


Norvig (2017) Probability of Set Decreases

Source: http://www.masterbaboon.com/2010/09/solving-the-game-set/

Implementation of Cards/Board

10000 Trials 3 Value, 4 Properties



Fisher Yates Shuffle with Rejection Sampling

Las Vegas Algorithm

Implementation and Reduction to SMT

$$K = \begin{bmatrix} k_{1,1} \\ k_{1,2} \\ \vdots \\ k_{1,p} \end{bmatrix} \begin{bmatrix} k_{2,1} \\ k_{2,2} \\ \vdots \\ k_{2,p} \end{bmatrix} \dots \begin{bmatrix} k_{v,1} \\ k_{v,2} \\ \vdots \\ k_{v,p} \end{bmatrix}$$

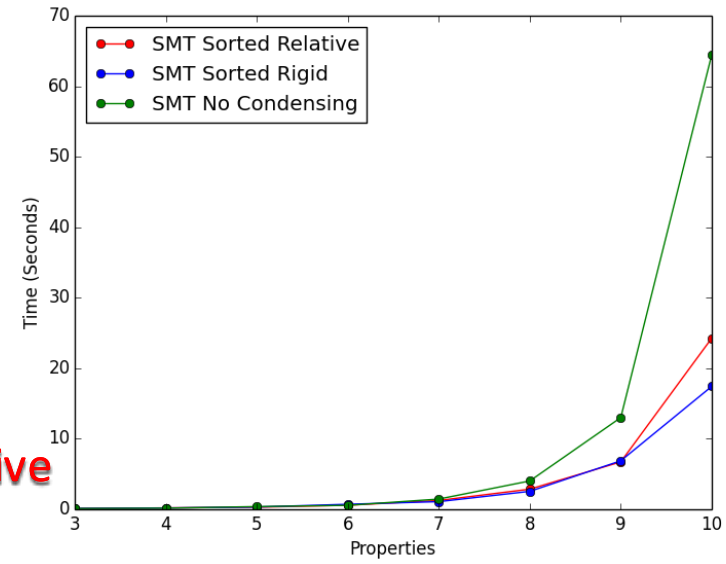
Constraints

1. All Different or All Same
2. Cards from the Board
3. Distinct Cards
4. **Symmetry Breaking**
5. Not Any Deleted Card

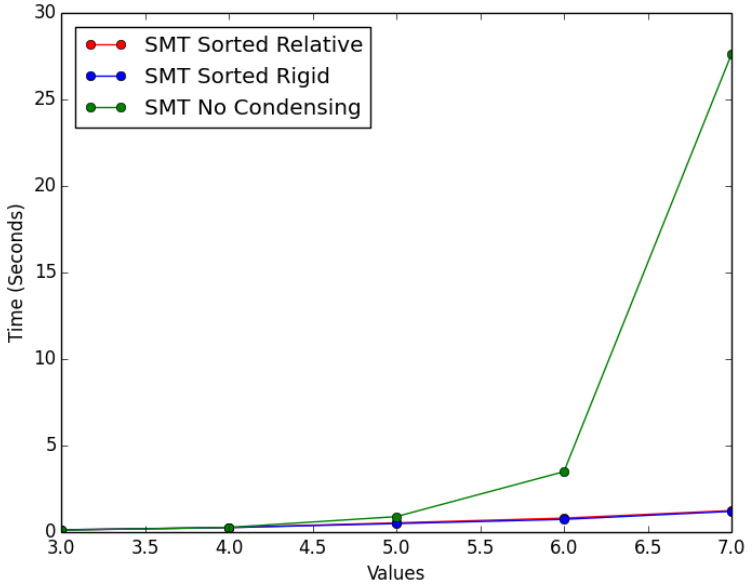
Z3

Results (SMT Symmetry Breaking)

3 Values, 5 Sets



3 Properties, 5 Sets



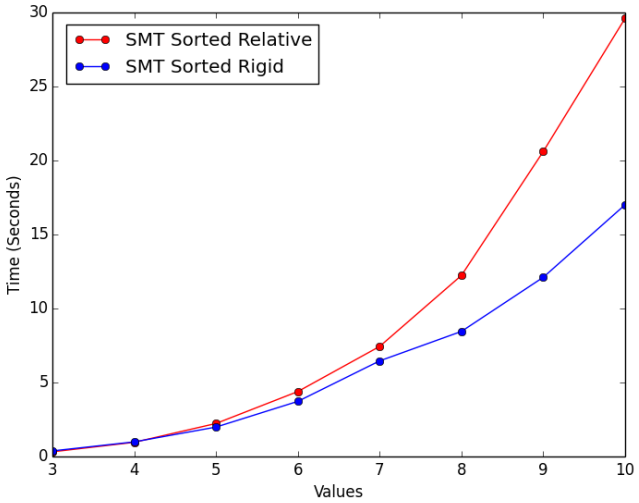
Sorted Relative

Sorted Rigid

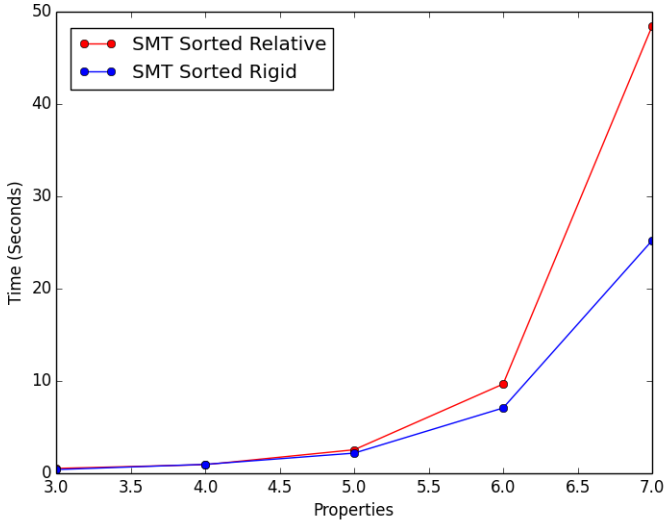
No Condensing

Results (SMT Runoff)

4 Properties, 10 Sets



4 Values, 10 Sets



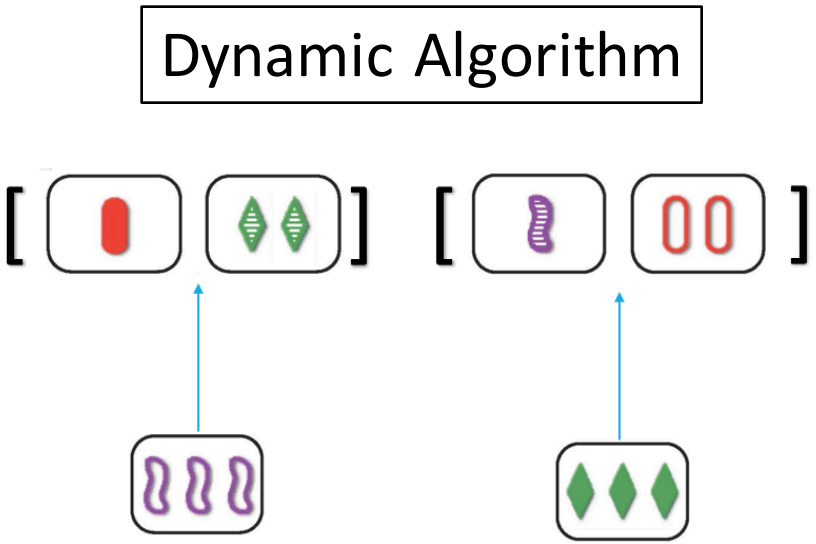
Sorted Relative

Sorted Rigid

Sorted Relative: Sorted by value

Sorted Rigid: 1st card = 0, 2nd = 1, … , vth = v-1

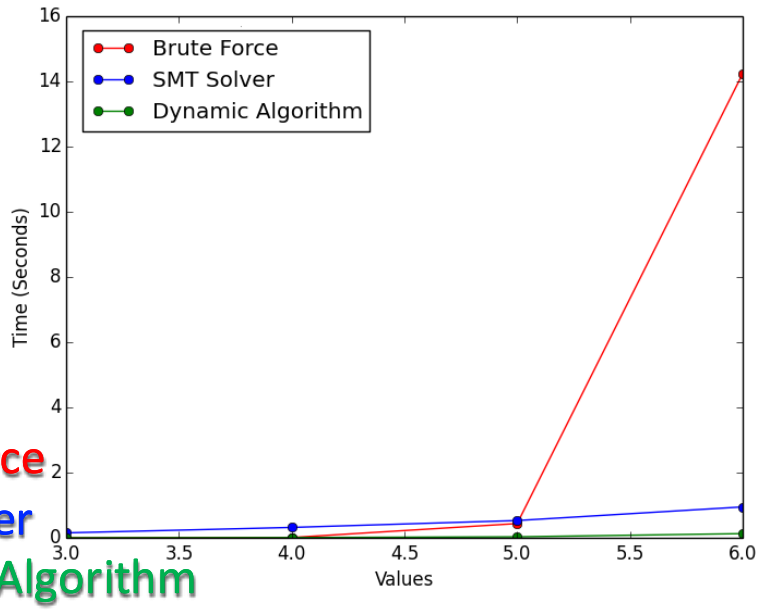
Implementation of Dynamic Algorithm



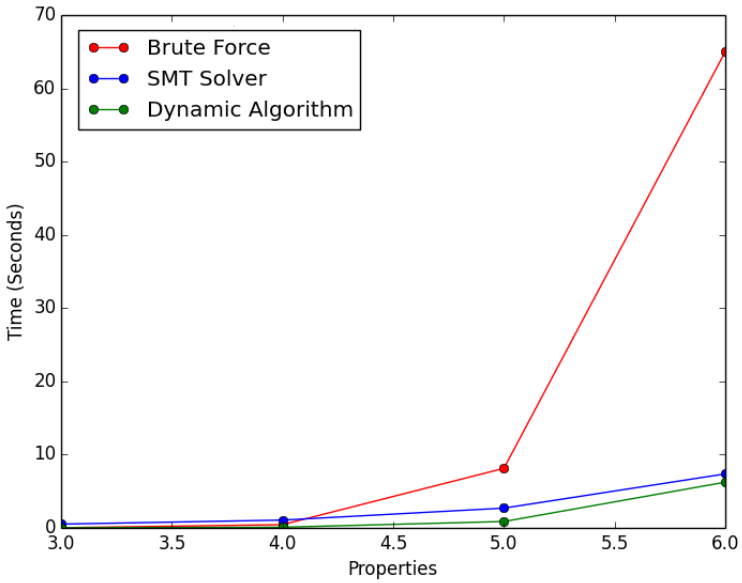
1. Build Partial Sets
2. Create Cards Searching For
3. Draw v New Cards
4. Quick Complete
5. Repeat

Results (with Brute Force)

3 Properties, 5 Sets



4 Values, 10 Sets



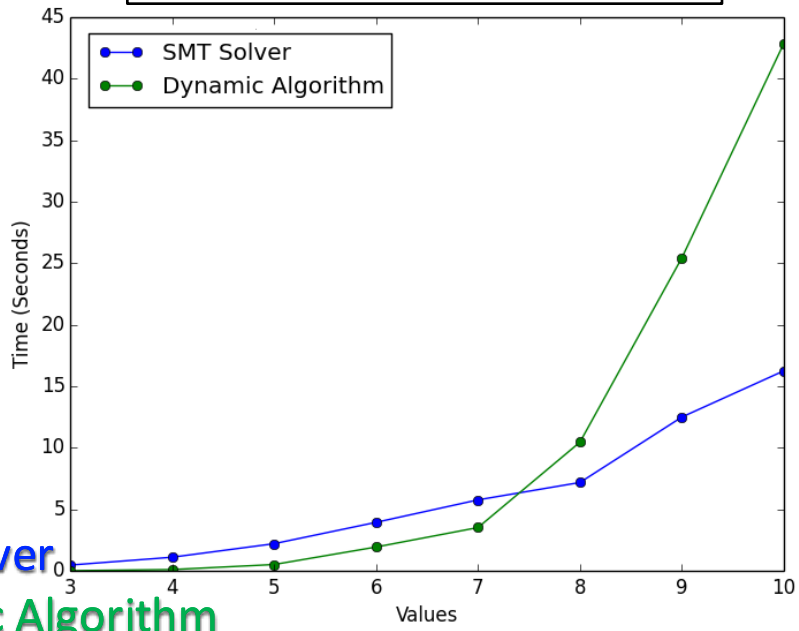
Brute Force

SMT Solver

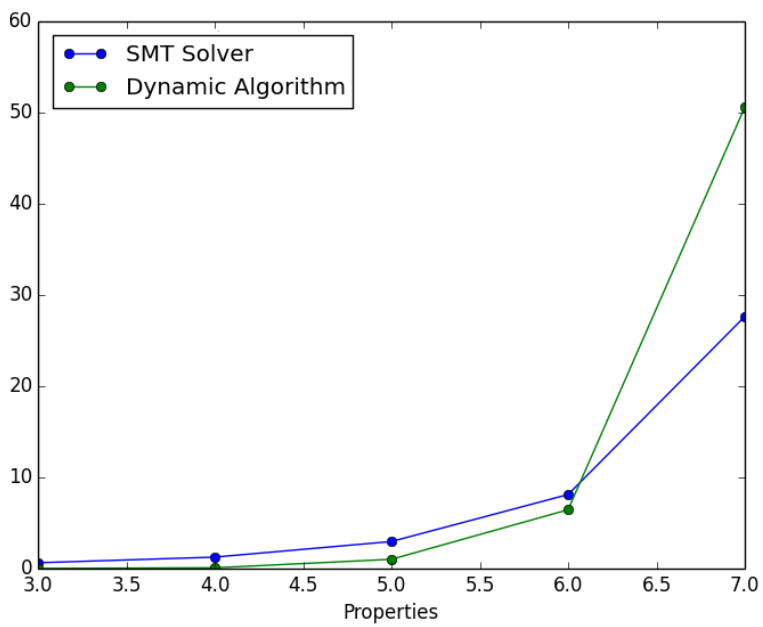
Dynamic Algorithm

Results (SMT vs Dynamic)

4 Properties, 10 Sets



4 Values, 10 Sets



SMT Solver
Dynamic Algorithm

Conclusion and Future Work

SMT Solver

Pros: Fast on Large Cases

Future: More Symmetry Breaking of Search Tree

Dynamic Algorithm

Pros: Fast on Medium Sized Cases

Future: Better Memory Management

Combine the two approaches?