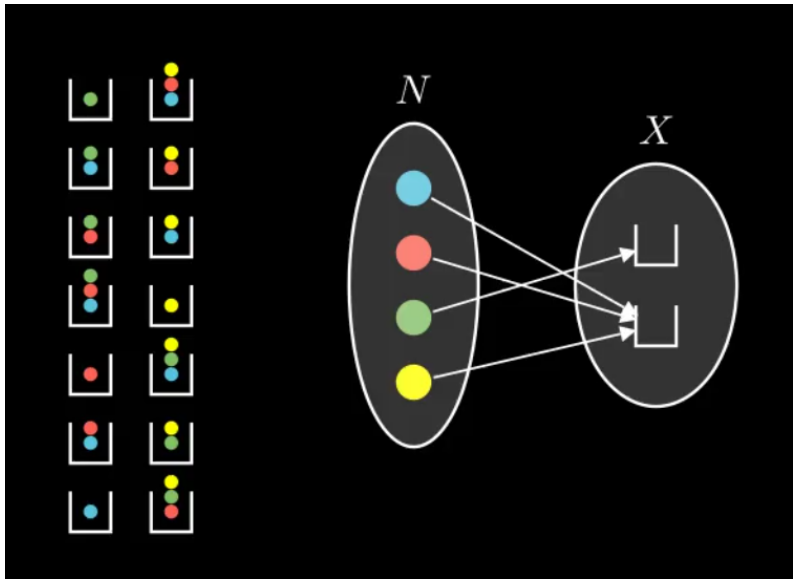
Visualizing Combinatorial Problem Resolution

Why is it Relevant?

- Cfr. Symbolab
  - Mathematical problems
  - Step-by-step solving
  - Help Students
- Our Visualization
  - Didactic tool
  - Focus = Combinatorics
  - Lifted Reasoning

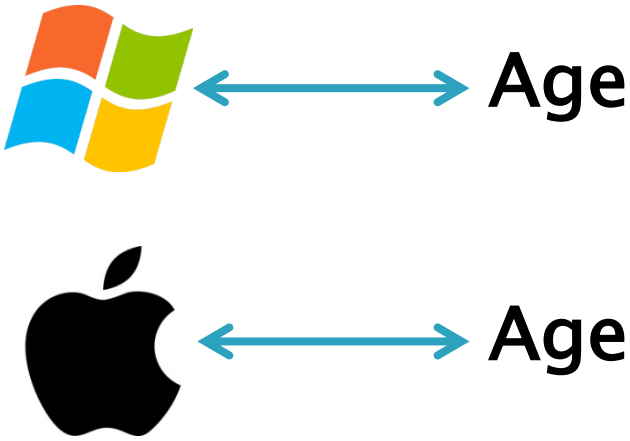


Bibliography

1. Lifted Reasoning for Combinatorial Counting
2. Towards High-Level Probabilistic Reasoning with Lifted Inference
3. Constructing Area-Proportional Venn and Euler

Lifted Reasoning

- (= Lifted Inference)
- Conditional Dependencies
- Traditional Model
  - 2 Seperate functions
  - No use of symmetries
  - Grounded Inference
- Lifted Model
  - Generalization
  - Faster processing



Lifted Reasoning

- Initial Proposal (David Poole)
  - High-level Reasoning
- Shift of Focus
  - Machine learning
- Now: Combinatorics

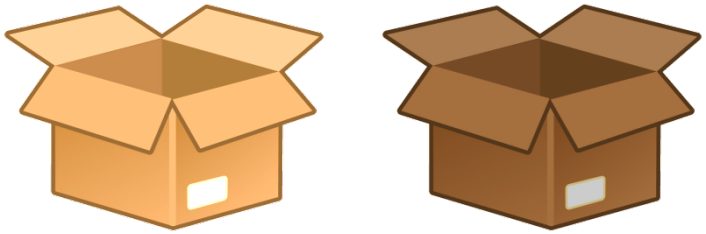
Combinatorics

- Counting possible arrangements
  - (E.g. shuffled deck)
- Constraints
  - (E.g. first = ace of spades)
- Human solving

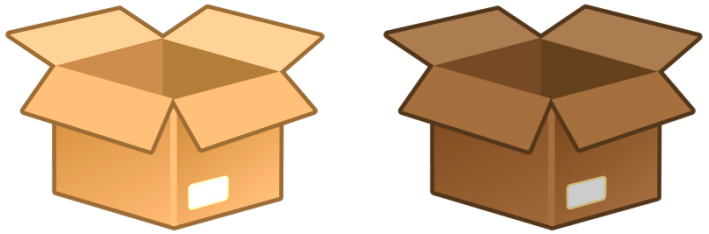
Twelvefold Way

- 12 Common Counting Problems
  - Mathematical Formulas
- 3 Function Types
  - Any, Injective, Surjective
- 4 Distinguishibilities
  - $\neq$, $=x$, $=y$, $=xy$

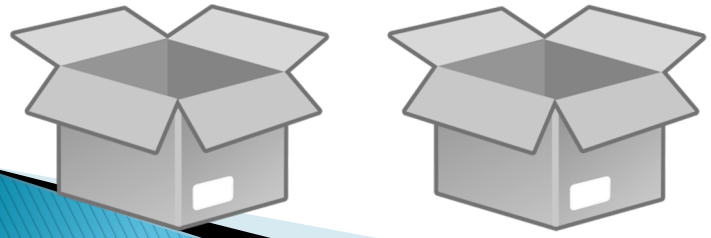
≠



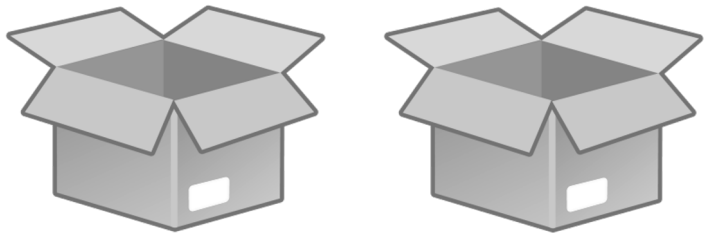
=X



=Y



=X
Y



CoLa

- Elements
  - Atomic objects
- Domains
  - Sets of elements
- Structures
  - Twelvefold way problem
  - (D, F)

CoLa (Constraints)

- Domain Formulas
  - Set operations
- Choice Constraints
  - Fixed position
- Counting Constraints
  - Limit on domains
  - Boolean operators (>, <, >=, <=, =)

#CSP

- Constraint Satisfaction Problem $\langle V, D, C \rangle$
  - Set of variables V
  - Domain D
  - Set of constraints C
  - Goal: whether a satisfying assignment exists
    - Assignment f satisfies all the constraints in C
- *Counting* Constraint Satisfaction Problem (#CSP)
  - Goal: Find the number of satisfying assignments $MC(V, D, C)$
- Combinatorics problems can be expressed as #CSPs
  - Set X → set of variables
  - Set Y → domain

Lifted Reasoning Strategies

- Exchangeability
  - Reason over groups → exponential improvements
  - A tuple of variables $(X_1, \ldots, X_n)$ are defined exchangeable:
    - If for all satisfying assignments $(X_1 = d_1, \ldots, X_n = d_n)$ and all permutations π of (1, … , n), $\{X_1 = d_{\pi(1)}, \ldots, X_n = d_{\pi(n)}\}$ is a satisfying assignment as well.

Lifted Reasoning Strategies

Example: coin flipping game

- $(X_1, X_2, X_3, X_4, X_5)$ represent the tosses
- With constraint #heads $>$ #tails

- The model (H, H, T, H, T) is a solution
- But so is
  - (H, T, H, T, H)
  - (H, H, H, T, T)
  - (T, H, T, H, H)
  - … → order of tosses don't matter!

Lifted Reasoning Strategies

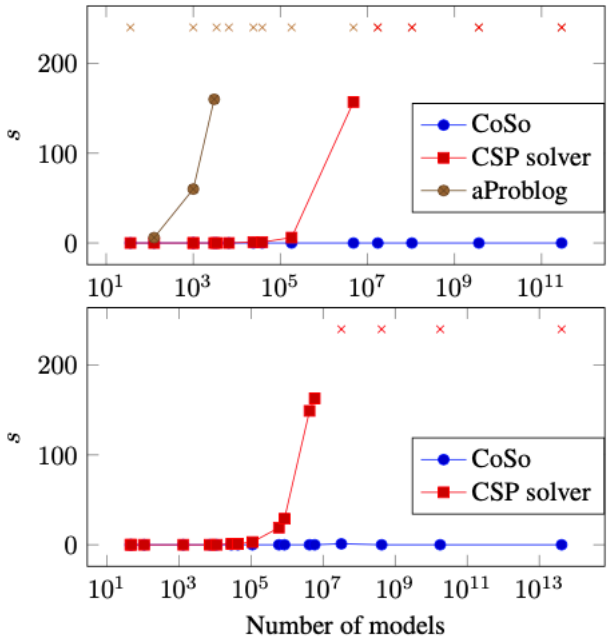
- Split
  - Multiplication rule: $MC(V_1 \cup V_2, D, C_1 \cup C_2) = MC(V_1, D, C_1) \cdot MC(V_2, D, C_2), V_1 \cap V_2 \neq \emptyset$
  - Splits a #CSP into two #CSPs
  - Product = original #CSP
- Shattering
  - Generalizes a split
  - Repeated use of a split
  - The multiplication rule only applies if the subproblems are independent

Lifted Reasoning Strategies

- Constraint Split
  - Split a constraint
  - *Example:* A constraint split for $\{V_1 \neq V_2\}$ could be $(\{V_1 = 1\}, \{V_2 \neq 1\})$.
- Constraint Shattering
  - Series of constraint splits over the entire domain
  - *Example:* constraint shattering for the previous constraint would be $(\{V_1 = i\}, \{V_2 \neq i\} \mid i \in D)$.

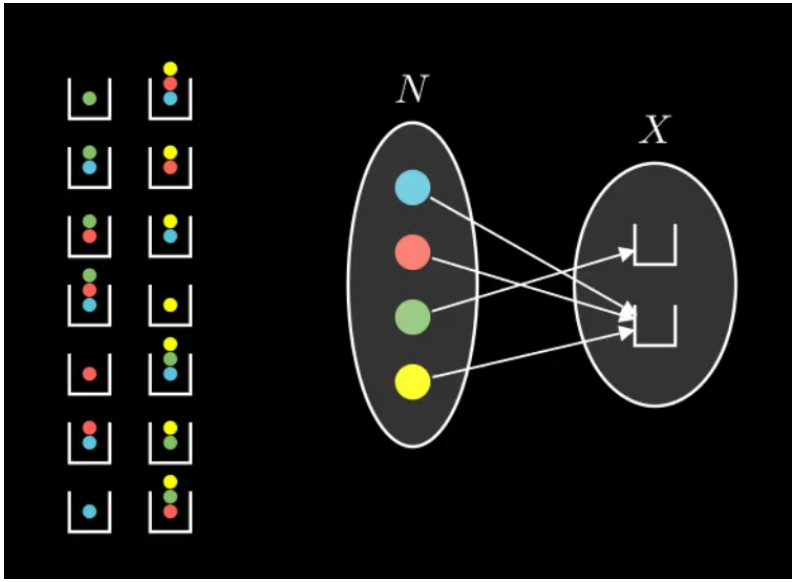
CoSo

- Solver for combinatorial problems
  - Based on exchangeability and constraint shattering
- Outperforms existing approaches based on enumeration



Visualization

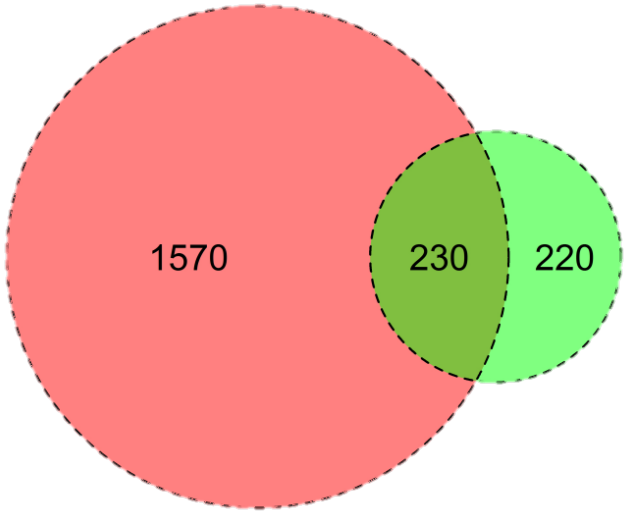
- Extend CoSo
  - Graphical Interface
- Show solving steps
  - Breaking problem down
  - Human thinking
  - Help Students



Visualization (Implementation)

- Godot Engine
  - GDscript
  - External files
- Steps
  - Venn diagrams
  - Basic problems
  - Constraints
- Area Proportional Venn
  - Circle area ~ domain size
  - Python library

GODOT



Questions?