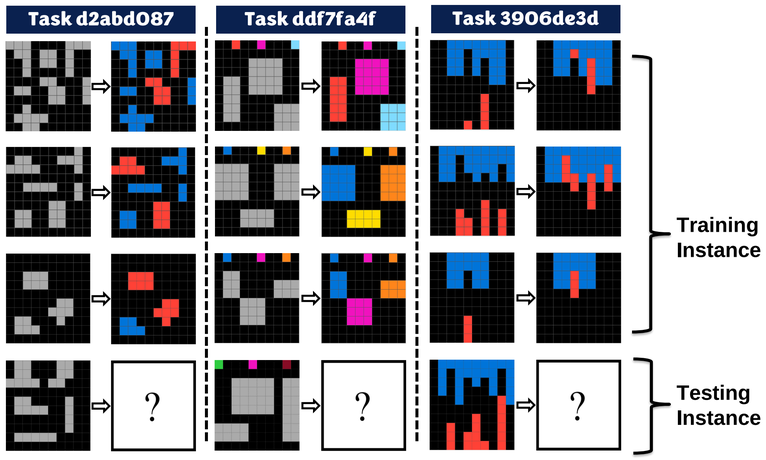
Graphs, Constraints and Search for the Abstraction and Reasoning Corpus

Mechanical & Industrial Engineering
UNIVERSITY OF TORONTO

Yudong Xu [1] *- Elias B. Khalil* [2] *- Scott Sanner* [3]

Introduction



The Abstraction and Reasoning Corpus (ARC) [1] is a collection of 1000 image-based reasoning tasks. We propose Abstract Reasoning with Graph Abstractions (ARGA): an object-centric framework for solving the ARC

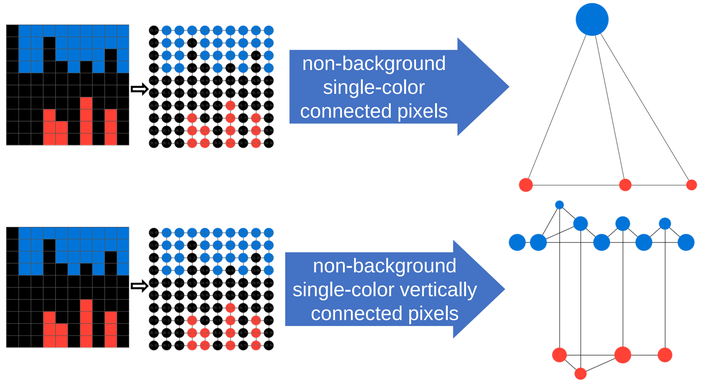
Methods Overview

Recognize objects through abstracted graph representations

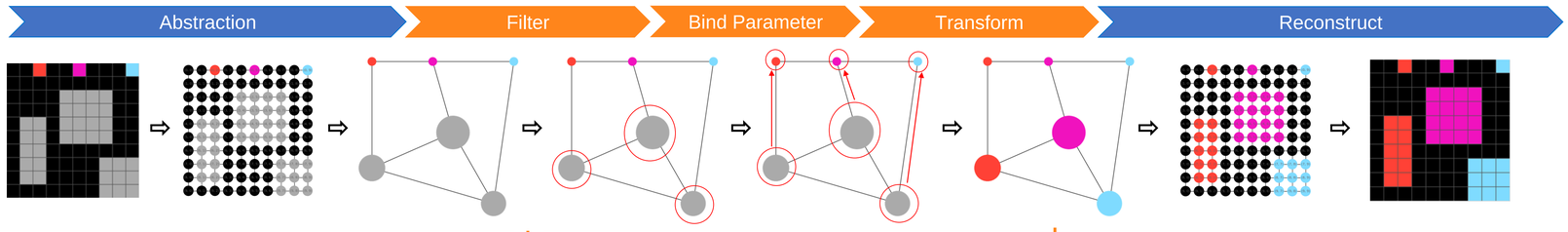
Define object-centric Domain Specific Language to represent the solution space

Synthesize solution through constraint-guided search

Recognize Objects: Graph Abstractions



Solution Space: Object-centric Domain Specific Language

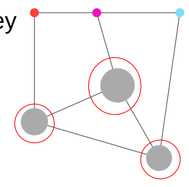


Filter identify nodes that satisfy some condition

$$
\begin{aligned}
&\text{Filter}(x)\\
&::= \text{Type}(x)\\
&::= \text{Filter}(x) \wedge \text{Filter}(x)\\
&::= \text{Filter}(x) \vee \text{Filter}(x)\\
&::= \neg\, \text{Filter}(x)\\
&::= \exists y\, \text{Rel}(x, y) \wedge \text{Filter}(y)\\
&::= \exists y\, \text{Rel}(y, x) \wedge \text{Filter}(y)\\
&::= \forall y\, \text{Rel}(x, y) \implies \text{Filter}(y)\\
&::= \forall y\, \text{Rel}(y, x) \implies \text{Filter}(y)\\
&::= \text{Rel}(x, c)[c \text{ is a constant}]\\
&::= \text{Rel}(c, x)[c \text{ is a constant}]
\end{aligned}
$$

Filter for nodes with color grey

$$filterByColorGrey(n) \equiv Node(n) \wedge color(n, grey)$$

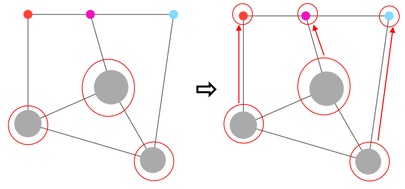


Bind Parameter retrieve values from the graph to use as transformation parameters

$$
\begin{aligned}
&\text{BindParam}(x, v)\\
&::= v = c\,[c \text{ is a constant}]\\
&::= \text{Rel}(x, v)\\
&::= \text{Rel}(v, x)\\
&::= \exists y\, \text{Rel}(x, y) \wedge \text{Filter}(y) \wedge \text{Param}(y, v)\\
&::= \exists y\, \text{Rel}(y, x) \wedge \text{Filter}(y) \wedge \text{Param}(y, v)
\end{aligned}
$$

Retrieve color of the size-1 neighbor

$$bindSize1NeighborColor(x, v) \equiv \exists y\, neighbor(x, y) \wedge size(y, 1) \wedge color(y, v)$$

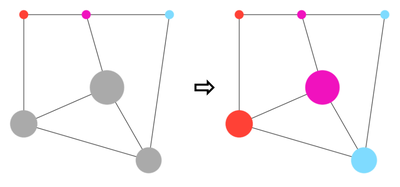


Transform updates object relations to reflect the desired changes

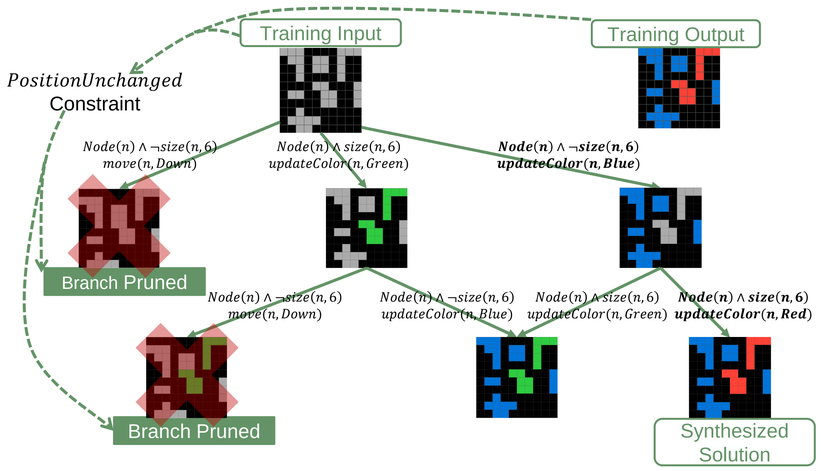
Transformation	Description
updateColor (Node, Color)	Update color of Node to Color
Move (Node, Direction)	Update pixels of Node to move in Direction
Rotate (Node)	Update pixels of Node to rotate it clockwise

Update the color of the selected nodes

$$updateColor(n : Node, c : Color) \longrightarrow color(n, c) \wedge \neg color(n, c') \quad \forall c' \in Color \text{ s.t. } c' \neq c$$



Synthesize Solution: Constraint-guided Search



Results and Conclusion

Model	Task Type	# Training Correct	# Testing Correct	Average Nodes	Average Time (sec.)
ARGA	movement	18/31 (58.06%)	17/31 (54.84%)	3830.35	89.75
	recolor	25/62 (40.32%)	23/62 (37.10%)	12316.87	326.83
	augmentation	20/67 (29.85%)	17/67 (25.37%)	4668.82	67.09
	all	63/160 (39.38%)	57/160 (35.62%)	7504.81	178.66
State of the Art [2]	movement	21/31 (67.74%)	15/31 (48.39%)	2176777.67	62.45
	recolor	23/62 (37.10%)	28/62 (45.16%)	2290441.32	93.19
	augmentation	35/67 (52.24%)	21/67 (31.34%)	2248151.1	66.07
	all	79/160 (49.38%)	64/160 (40.00%)	2249924.92	77.08

Our ARGA framework uses graph abstraction, domain-specific language, and constraint-guided search to solve ARC tasks. Though its accuracy lags state-of-the-art methods, it significantly outperforms in terms of efficiency, exploring an order of magnitude fewer nodes.

[1] Chollet, F. 2019. On the Measure of Intelligence. arXiv:1911.01547.
[2] top quarks. 2020. ARC-solution. https://github.com/top-quarks/ARC-solution.