Optimized Concept Learning in BEN:
Better identifying when transformations apply

Author: Jonas Duifs **Supervisor:** Dekel Zak **Professor:** Dr. Sebastijan Dumančić

1. Background

BEN: a Program Syntesis agent, unique because of its **Divide, Align, and Conquer architecture**

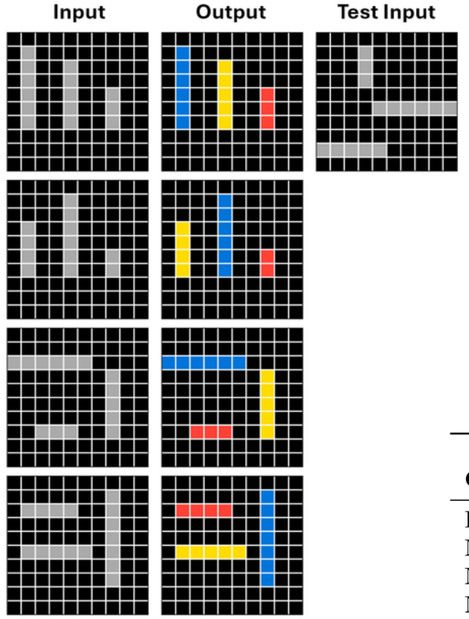
- **DIVIDE:** segment examples into input and output objects
- **ALIGN:** align objects to input-output correspondences
- **CONQUER:** Find transformations for corresspondences, and determine **concepts** when to apply them

ARC:

- Visual reasoning **benchmark** testing fluid intelligence with different tasks.
- Extreme data scarcity, **very few examples**

Concept Learning: Synthesize boolean rules to determine when transformations apply.

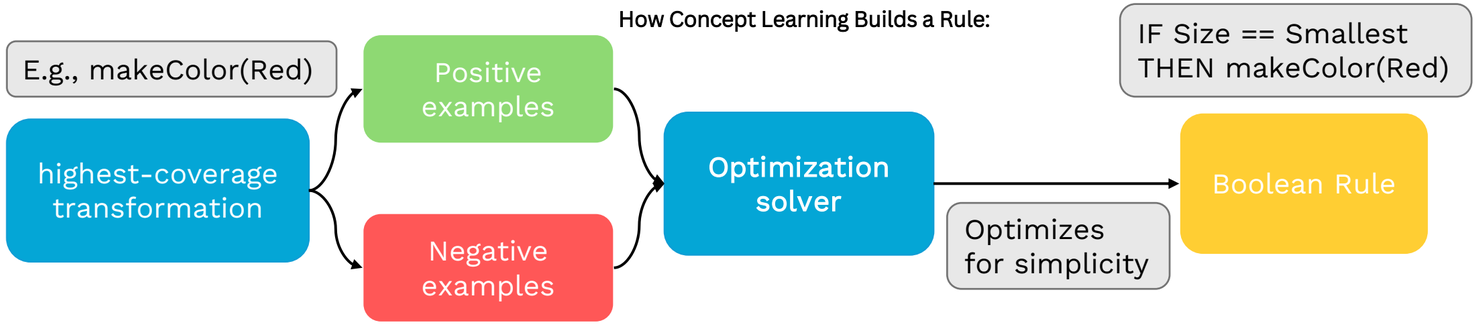
1. Select the transformation that correctly reconstructs most output objects (highest coverage).
2. Generate boolean rule (based on object properties) that separates positive from negative examples.
3. Repeat until all output objects are reconstructed.



Example of ARC task

2. Research Question

How can we more efficiently find when transformations apply?



3. Methods

Three improvements made, giving BEN better nuance in complexity:

- **Simple Minimal Transformation Coverage:** Choose transformations, not only on coverage, but also transformation complexity and ambiguity (Many-to-one vs one-to-one correspondences)
- **Rule Generality:** Prefer using properties multiple times across rules
- **Concept score as Complexity Heuristic:** Generated concept is given score from solver (coverage of objects, properties), use score to compare rules

4. Experiments

Ablation study, remove one improvement to isolate effect:

Configuration	Task Accuracy	Exec. Time (s)	Concept Learn. Time (s)	Unique Prop./Task	Clauses /Task	Clauses /Rule
Baseline	32/400	15.71	0.26	3.05	3.05	0.99
No Rule Generality	35/400	**15.07**	**0.25**	3.00	3.09	1.00
No Concept Score	**36/400**	15.47	0.38	3.00	3.09	1.00
No Simple Transf.	33/400	15.69	0.39	3.05	3.05	0.99
Optimized BEN	**36/400**	15.57	0.39	3.00	3.09	1.00

Note: all measurements are averages, with the exception of accuracy

5. Conclusions

- No regression in performance: baseline tasks always succeeding
- **Increase in accuracy**, at **negligible computational overhead**
- Choosing simple transformations and more general concepts allow BEN to better generalize
- Consistently one clause per rule, likely inherent in ARC, potentially good metric for backtracking
- Improvements likely also useful in other applications, as simpler and more general solutions also preferable in real-world applications