Program Synthesis Meets Visual What-Comes-Next Puzzles

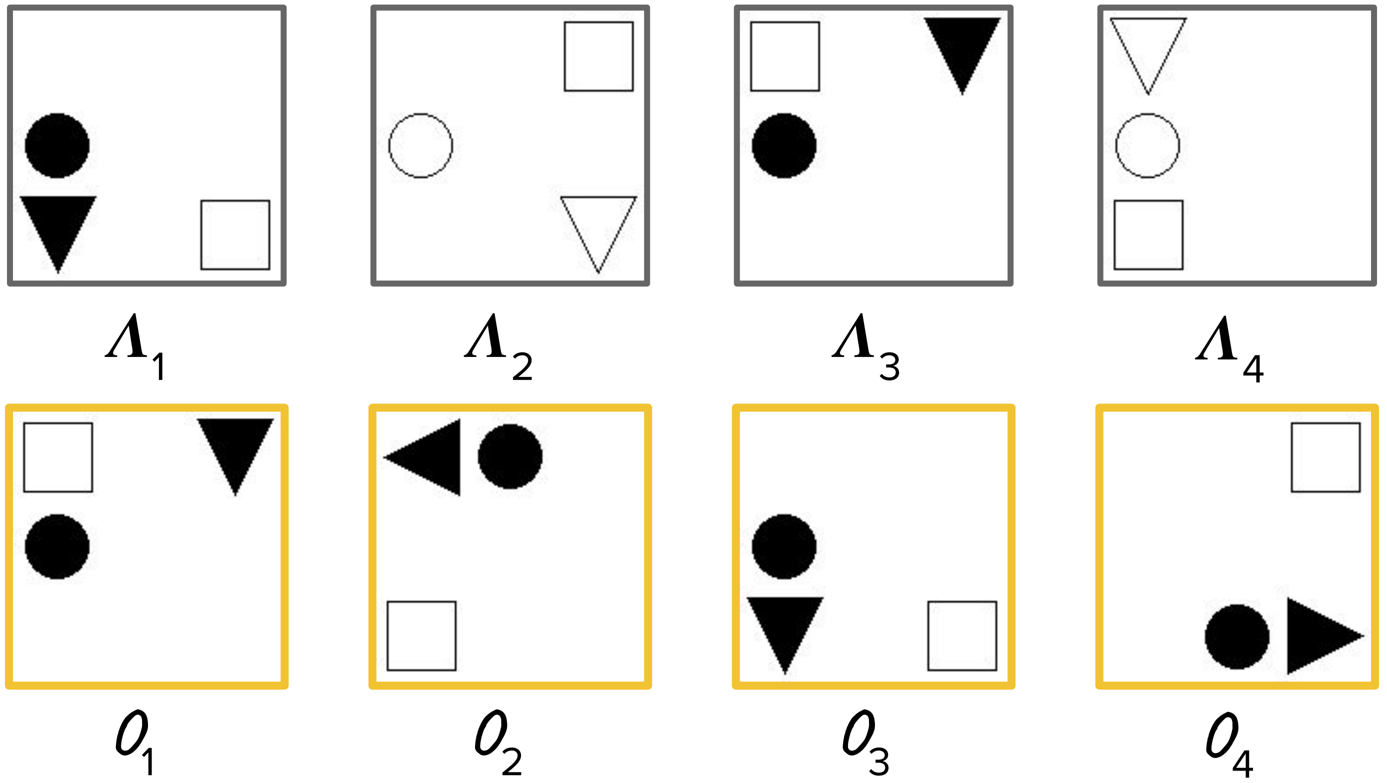
Sumit Lahiri, Pankaj Kumar Kalita, Akshay Kumar Chittora, Varun Vankudre, Subhajit Roy

Indian Institute of Technology Kanpur

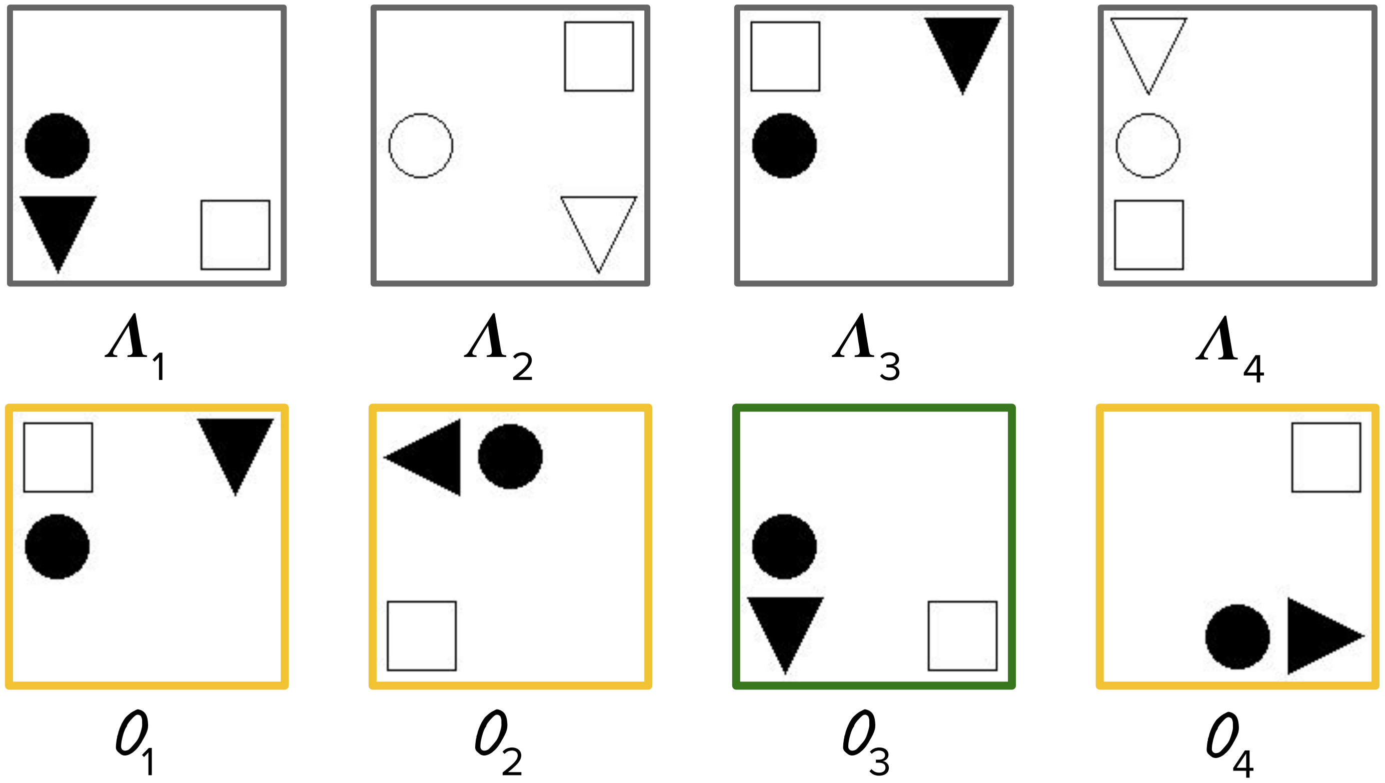
ASE 2024

What-Comes-Next Puzzles

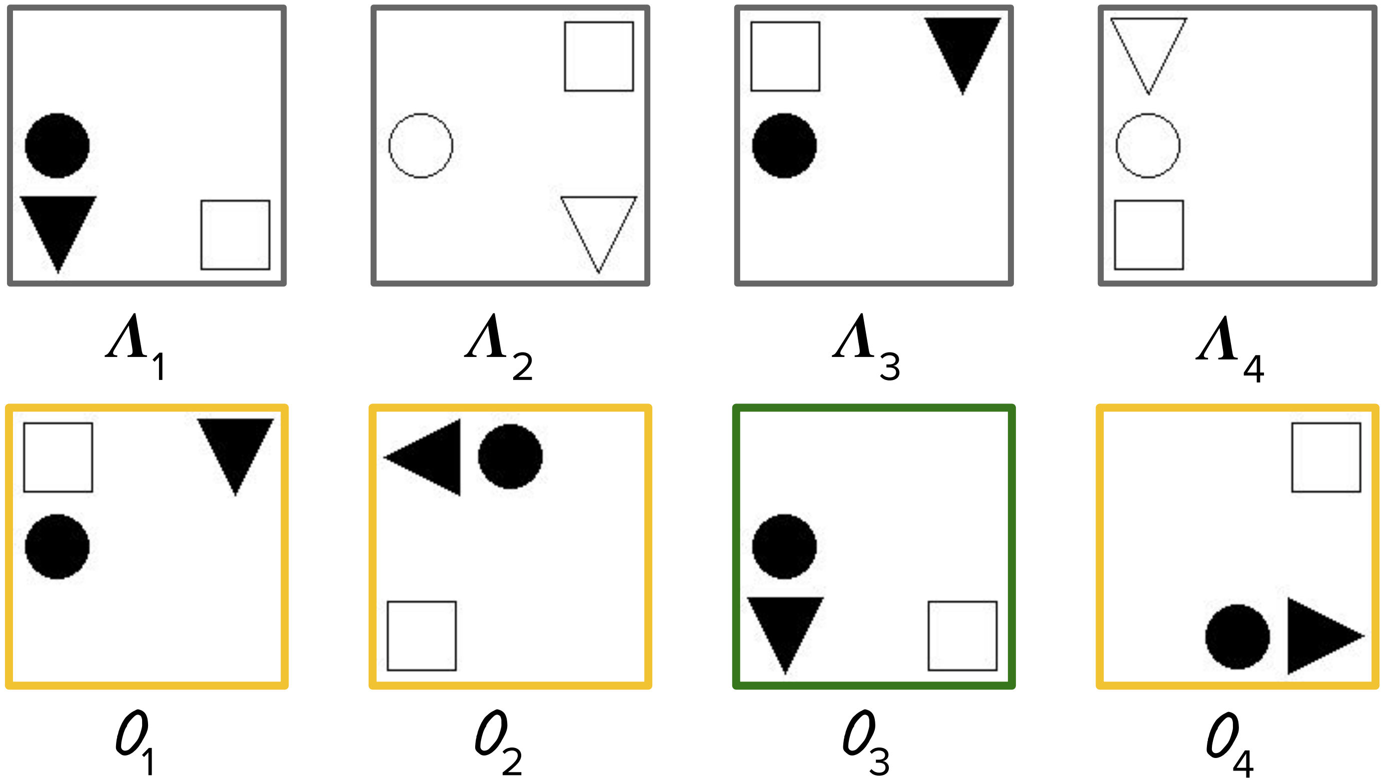
What-Comes-Next Puzzles



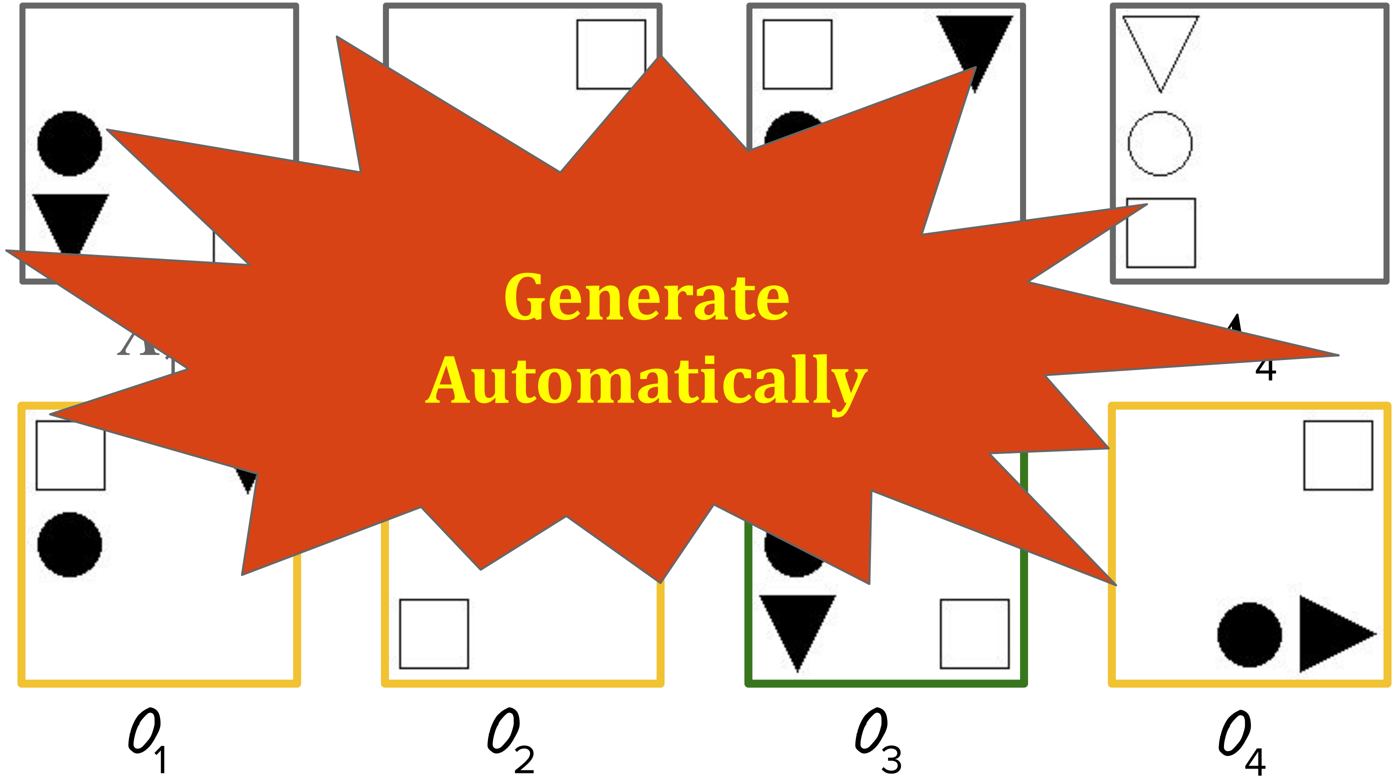
What-Comes-Next Puzzles



Our Goal

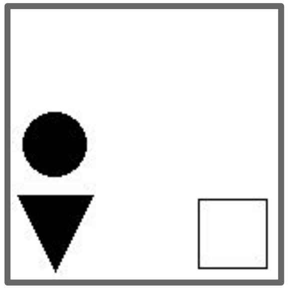


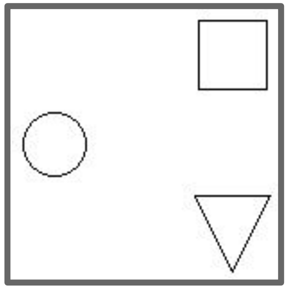
Our Goal

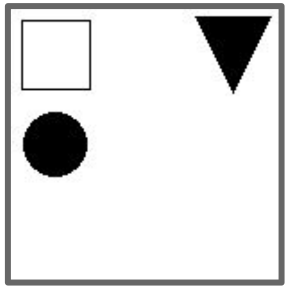


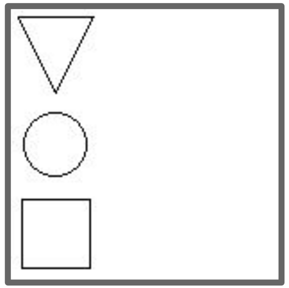
Challenges

Challenges


Λ_1


Λ_2

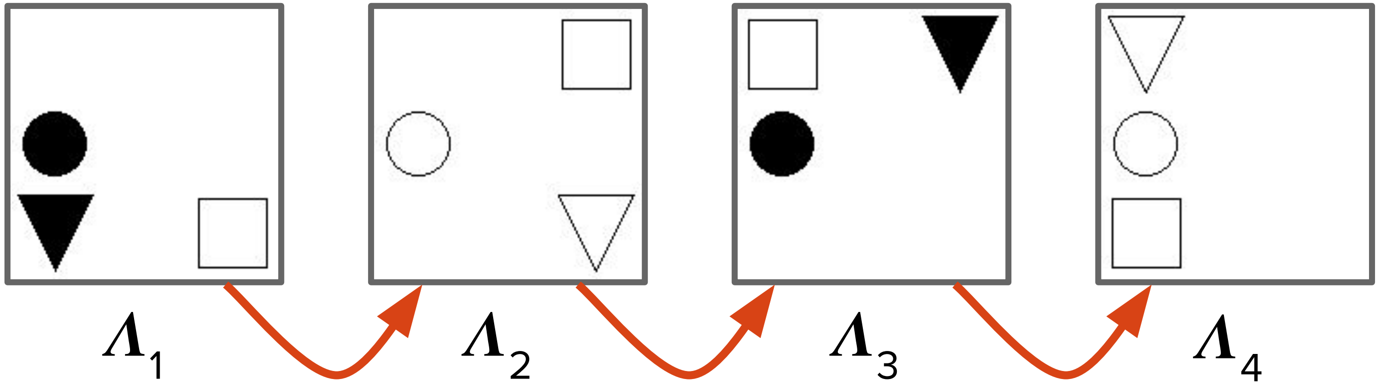

Λ_3


Λ_4

Logically Follows

Challenges

Logically Follows

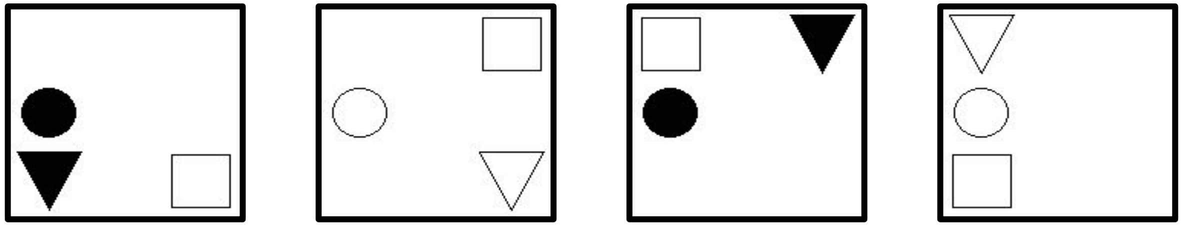


$$\exists \mathcal{P}.\ \mathcal{P}(\Lambda_1) = \Lambda_2;\ \ \mathcal{P}(\Lambda_2) = \Lambda_3;\ \ \mathcal{P}(\Lambda_3) = \Lambda_4$$

Challenges

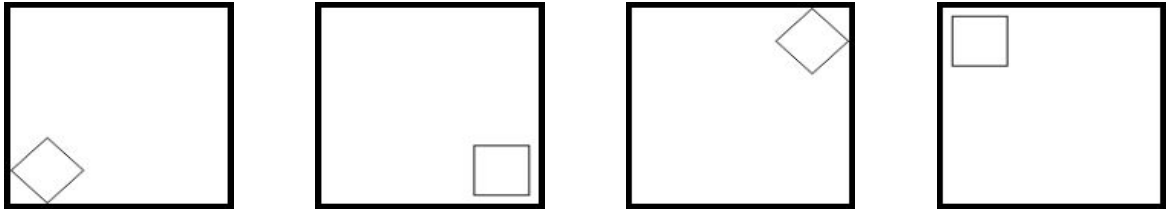
Puzzles should be appealing

(1)



Λ_1 Λ_2 Λ_3 Λ_4

(2)

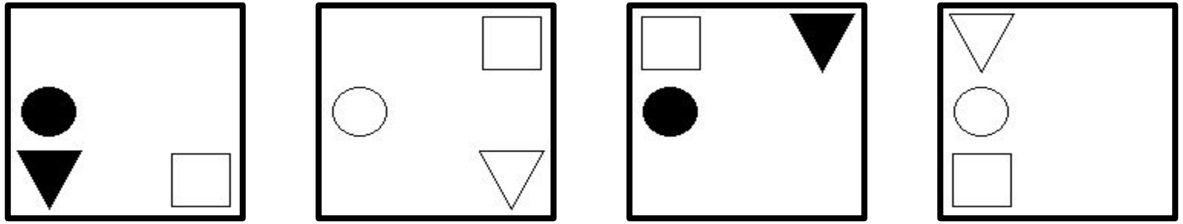


Λ_1 Λ_2 Λ_3 Λ_4

Challenges

Puzzles should be appealing

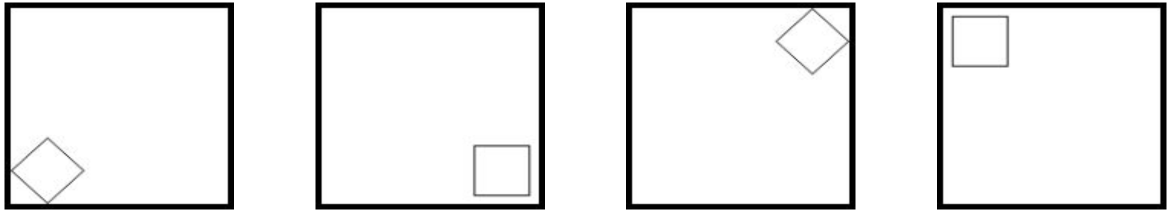
(1)



Λ_1 Λ_2 Λ_3 Λ_4

more

(2)



Λ_1 Λ_2 Λ_3 Λ_4

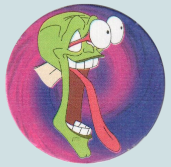
less

Methodology in a nutshell...

Logically Follows

$\exists \mathcal{P}.\ \mathcal{P}(\Lambda_1) = \Lambda_2;\ \ \mathcal{P}(\Lambda_2) = \Lambda_3;\ \ \mathcal{P}(\Lambda_3) = \Lambda_4$

Appealing



Methodology in a nutshell...

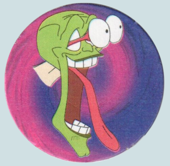
Logically Follows

$\exists \mathcal{P}.\ \mathcal{P}(\Lambda_1) = \Lambda_2;\ \mathcal{P}(\Lambda_2) = \Lambda_3;\ \mathcal{P}(\Lambda_3) = \Lambda_4$

↓

Second-Order Constraint Solving

Appealing



Methodology in a nutshell…

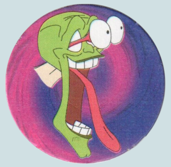
Logically Follows

$\exists \mathcal{P}.\ \mathcal{P}(\Lambda_1) = \Lambda_2;\ \mathcal{P}(\Lambda_2) = \Lambda_3;\ \mathcal{P}(\Lambda_3) = \Lambda_4$

↓

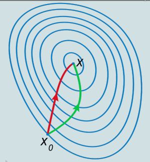
Second-Order Constraint Solving

Appealing



↓

Optimization

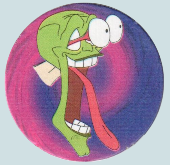


Methodology in a nutshell…

Logically Follows

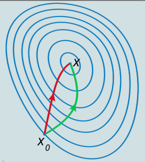
$\exists \mathcal{P}.\ \mathcal{P}(\Lambda_1) = \Lambda_2;\ \ \mathcal{P}(\Lambda_2) = \Lambda_3;\ \ \mathcal{P}(\Lambda_3) = \Lambda_4$

Appealing



Second-Order Constraint Solving

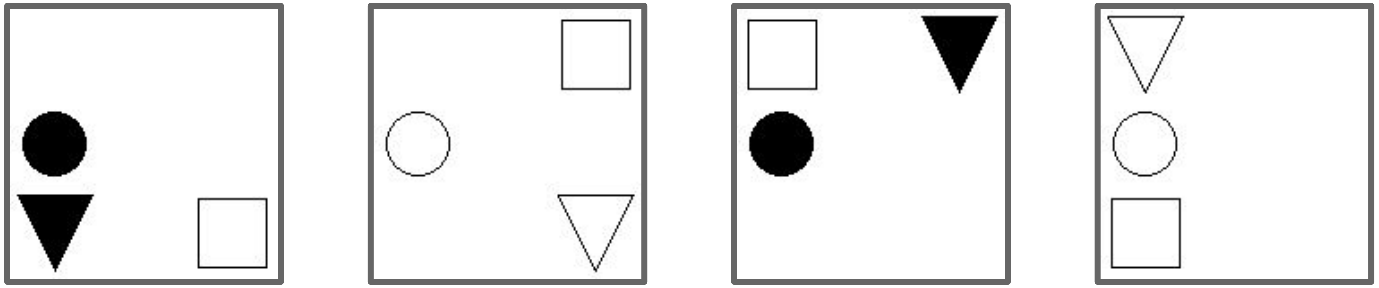
Optimization



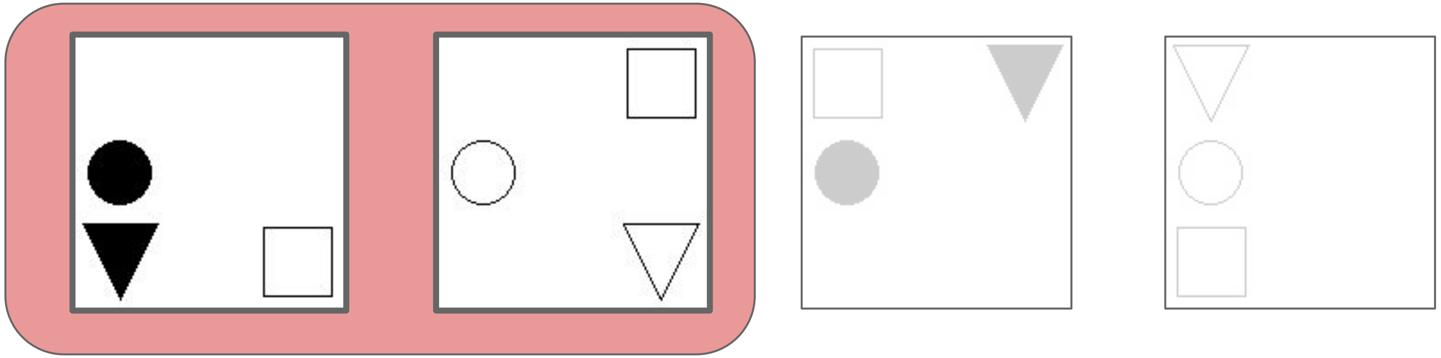
Constrained Optimization Problem (optimal synthesis)

Program Synthesis and Puzzle?

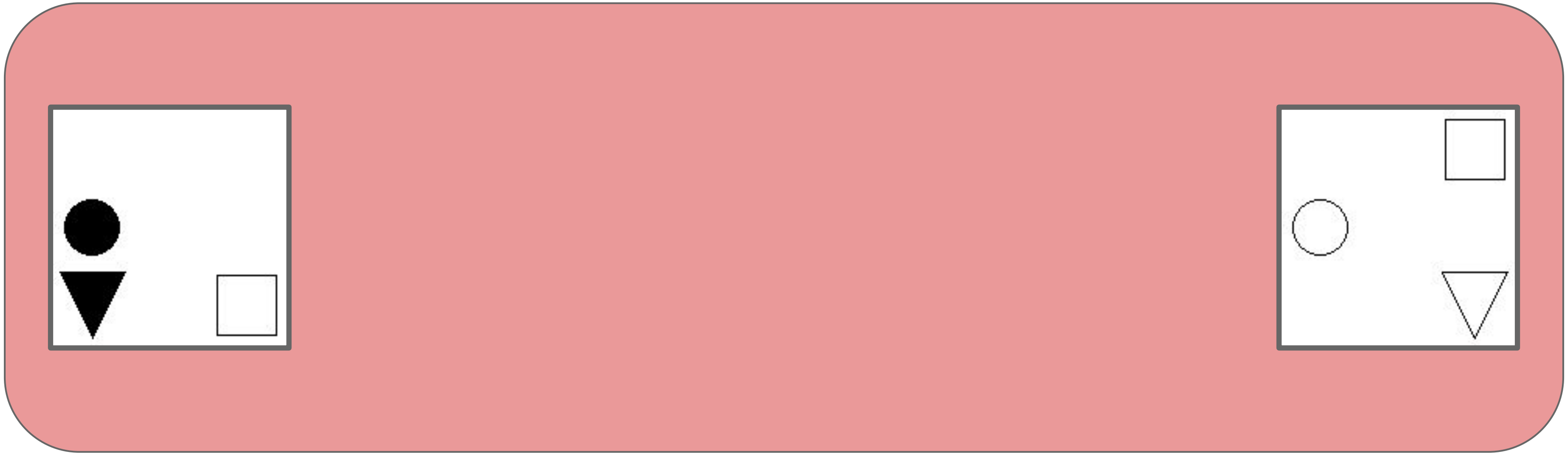
Program Synthesis and Puzzle?


Λ_1 Λ_2 Λ_3 Λ_4

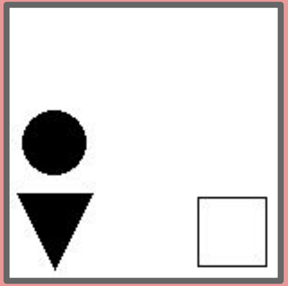
Program Synthesis and Puzzle?



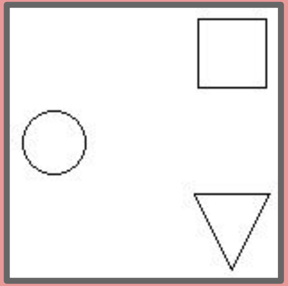
Program Synthesis and Puzzle?

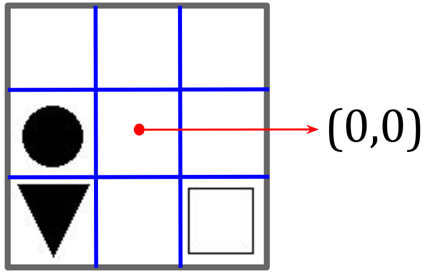


Program Synthesis and Puzzle?

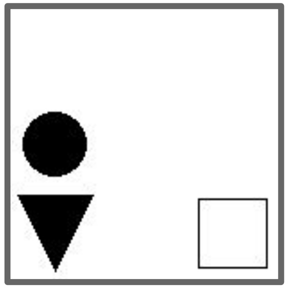


```
FlipFill(Circle)
FlipFill(Rotate(Triangle, 270, (0,0)))
Rotate(Square, 270, (0, 0))
```



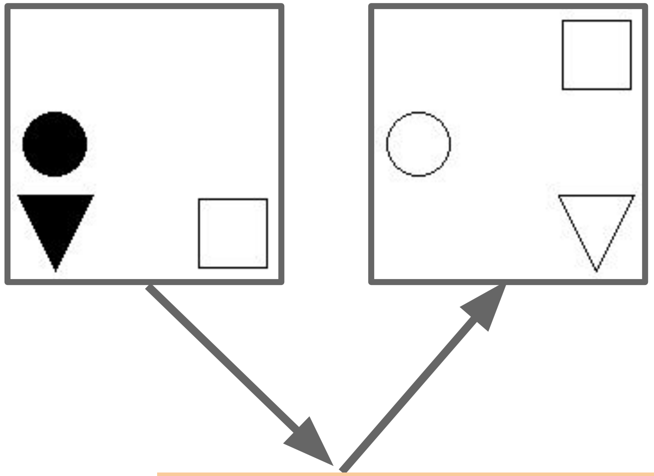


Program Synthesis and Puzzle?



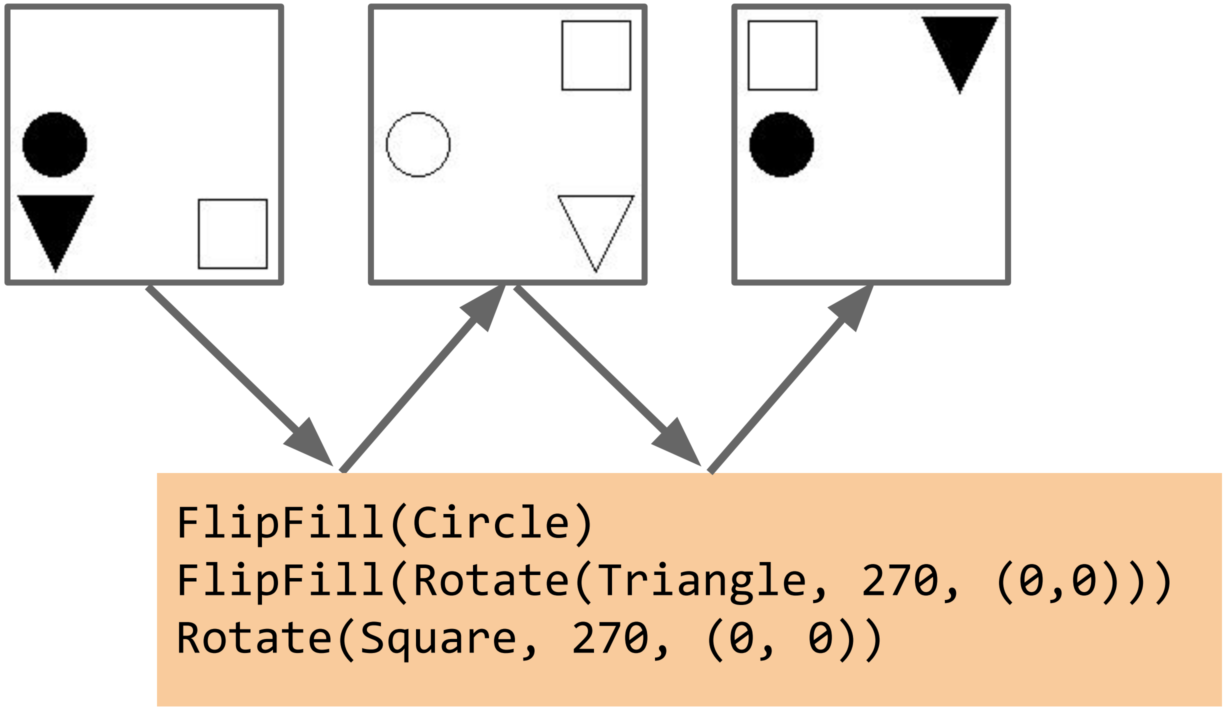
```
FlipFill(Circle)
FlipFill(Rotate(Triangle, 270, (0,0)))
Rotate(Square, 270, (0, 0))
```

Program Synthesis and Puzzle?



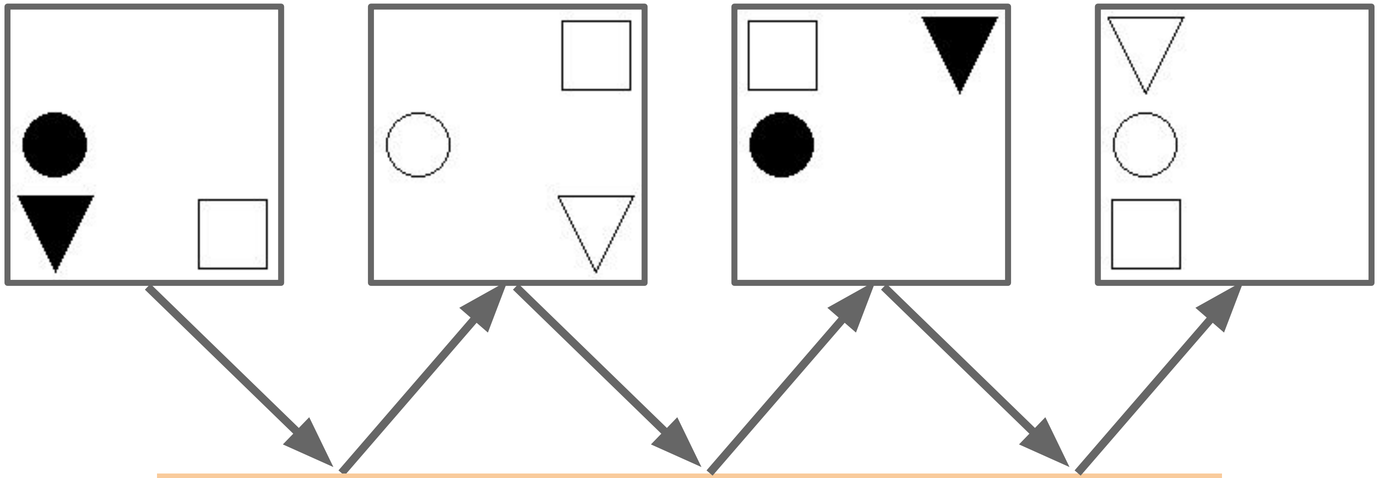
```
FlipFill(Circle)
FlipFill(Rotate(Triangle, 270, (0,0)))
Rotate(Square, 270, (0, 0))
```

Program Synthesis and Puzzle?



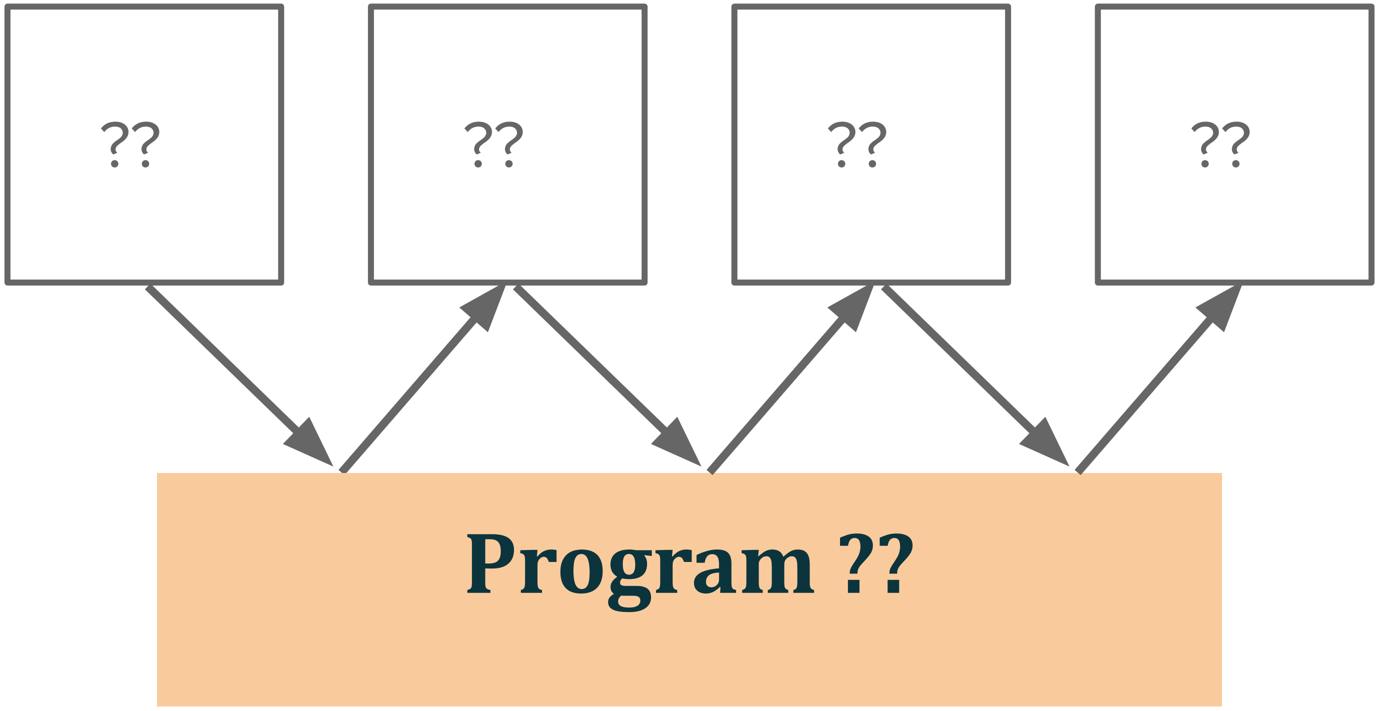
```
FlipFill(Circle)
FlipFill(Rotate(Triangle, 270, (0,0)))
Rotate(Square, 270, (0, 0))
```

Program Synthesis and Puzzle?



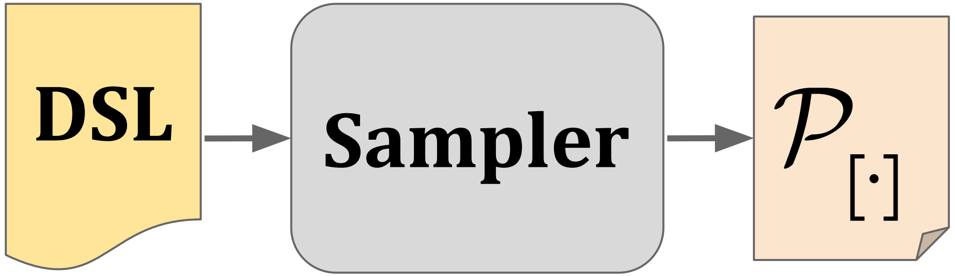
```
FlipFill(Circle)
FlipFill(Rotate(Triangle, 270, (0,0)))
Rotate(Square, 270, (0, 0))
```

Program Synthesis and Puzzle?

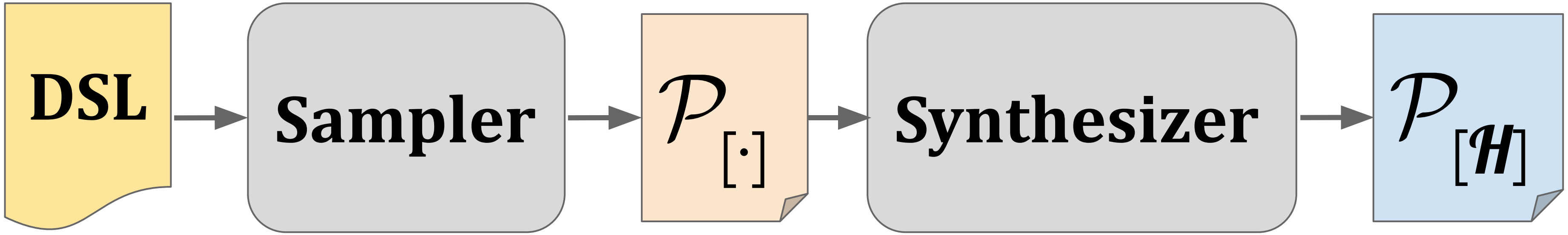


Puzzle Generation Using Program Synthesis

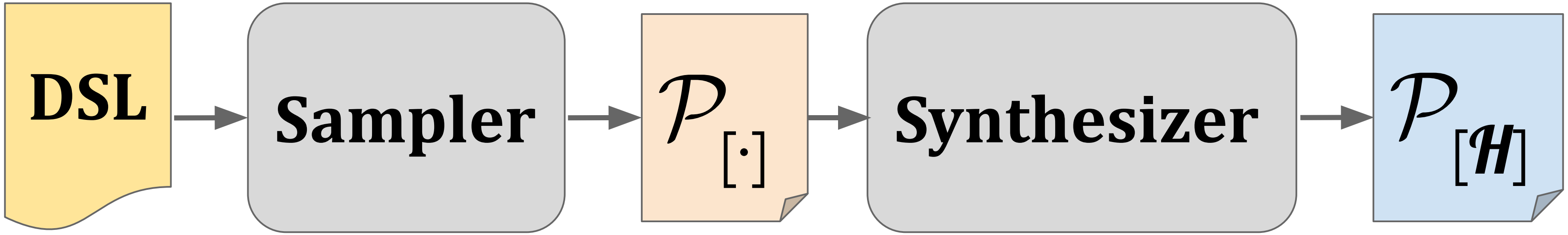
Puzzle Generation Using Program Synthesis



Puzzle Generation Using Program Synthesis

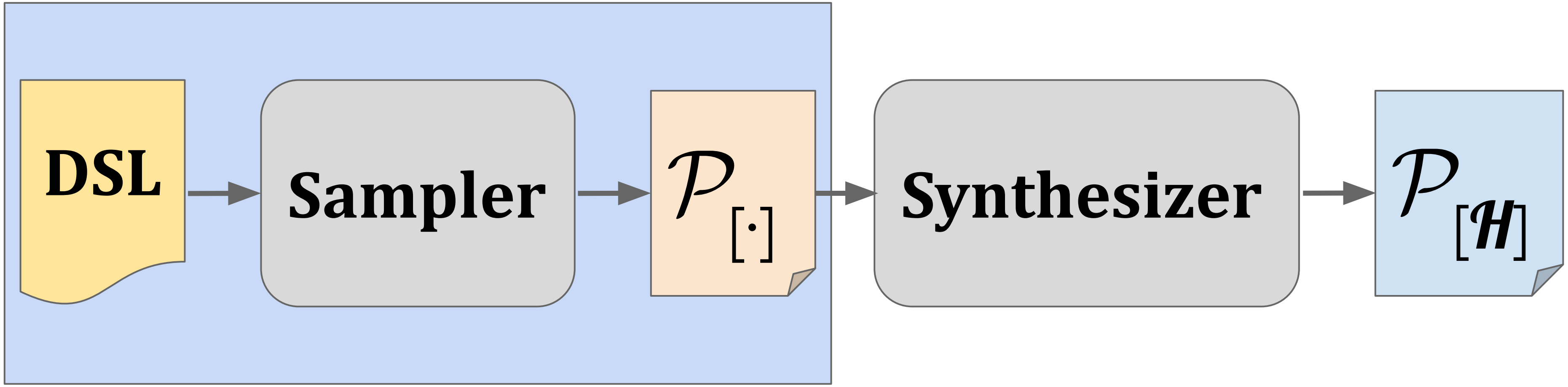


Puzzle Generation Using Program Synthesis



PuzzleGen

Puzzle Generation Using Program Synthesis



PuzzleGen

Abstract Program

$$
\begin{array}{rcl}
\langle\text{Program}\rangle & ::= & \lambda\Lambda.\langle\text{Entity}\rangle^{+}\ \ [\text{p}_1] \\
\langle\text{Entity}\rangle & ::= & \text{GetEntity}(\text{Num})\ \ [\text{p}_2]\ \ | \\
 & & \text{FlipFill}(\langle\text{Entity}\rangle)\ \ [\text{p}_3]\ \ | \\
 & & \text{Rotate}(\langle\text{Entity}\rangle, \boxed{\langle\text{Value}\rangle}, \boxed{\langle\text{Coord}\rangle})\ \ [\text{p}_4]\ \ | \\
 & & \text{SelfRotate}(\langle\text{Entity}\rangle, \boxed{\langle\text{Value}\rangle})\ \ [\text{p}_5]\ \ | \\
 & & \text{Translate}(\langle\text{Entity}\rangle, \boxed{\langle\text{Coord}\rangle})\ \ [\text{p}_6]\ \ | \\
 & & \text{Ite}(\boxed{\langle\text{Cond}\rangle}, \langle\text{Entity}\rangle, \langle\text{Entity}\rangle)\ \ [\text{p}_7] \\
\langle\text{Cond}\rangle & ::= & \Lambda.\text{sid mod } \boxed{\Lambda.id}\ \ [\text{p}_8] \\
\langle\text{Coord}\rangle & ::= & (\boxed{\langle\text{Value}\rangle}, \boxed{\langle\text{Value}\rangle})\ \ [\text{p}_9] \\
\langle\text{Value}\rangle & ::= & \text{Progress}(\boldsymbol{\alpha})\ \ [\text{p}_{10}]\ \ | \\
 & & \boxed{\text{Num}}\ \ [\text{p}_{11}]
\end{array}
$$

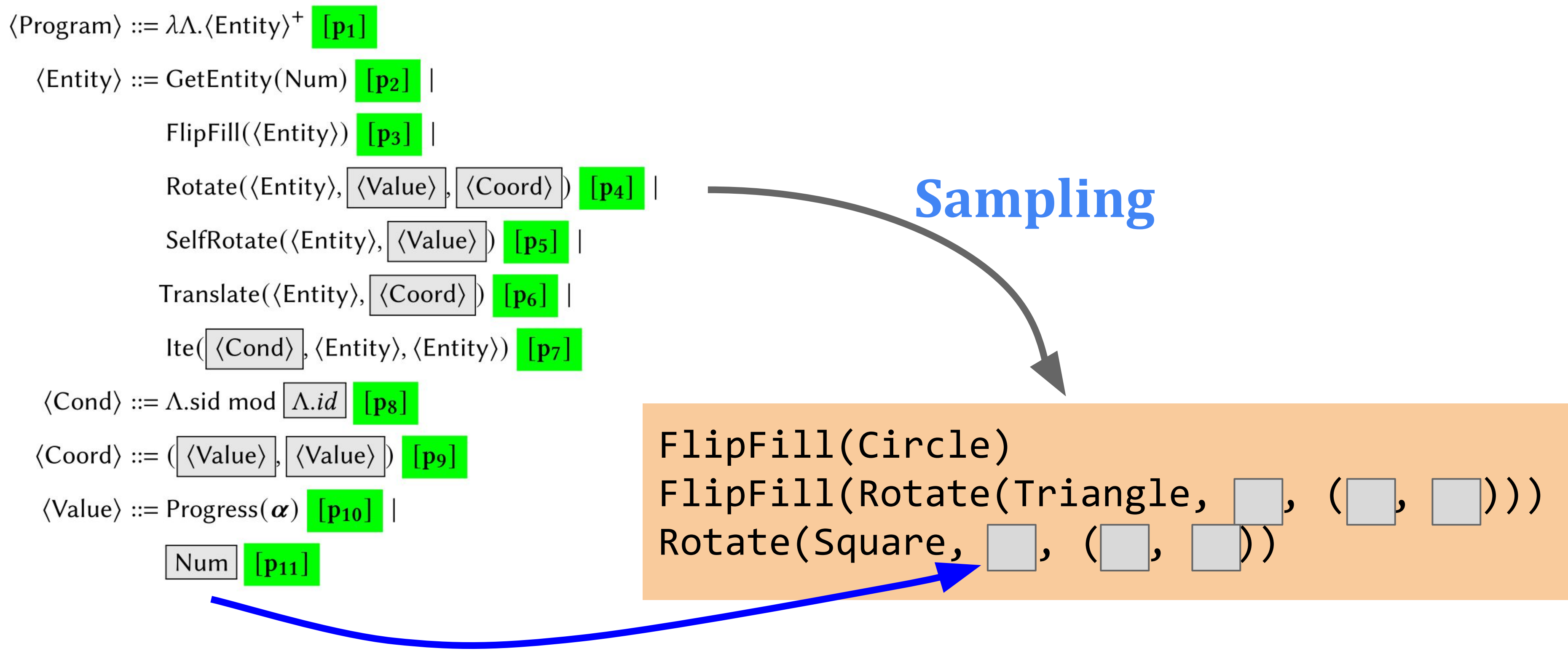
Abstract Program

$$
\begin{array}{rcl}
\langle\text{Program}\rangle & ::= & \lambda\Lambda.\langle\text{Entity}\rangle^{+} \;\; [p_1] \\
\langle\text{Entity}\rangle & ::= & \text{GetEntity}(\text{Num}) \;\; [p_2] \;\; | \\
 & & \text{FlipFill}(\langle\text{Entity}\rangle) \;\; [p_3] \;\; | \\
 & & \text{Rotate}(\langle\text{Entity}\rangle, \boxed{\langle\text{Value}\rangle}, \boxed{\langle\text{Coord}\rangle}) \;\; [p_4] \;\; | \\
 & & \text{SelfRotate}(\langle\text{Entity}\rangle, \boxed{\langle\text{Value}\rangle}) \;\; [p_5] \;\; | \\
 & & \text{Translate}(\langle\text{Entity}\rangle, \boxed{\langle\text{Coord}\rangle}) \;\; [p_6] \;\; | \\
 & & \text{Ite}(\boxed{\langle\text{Cond}\rangle}, \langle\text{Entity}\rangle, \langle\text{Entity}\rangle) \;\; [p_7] \\
\langle\text{Cond}\rangle & ::= & \Lambda.\text{sid mod } \boxed{\Lambda.id} \;\; [p_8] \\
\langle\text{Coord}\rangle & ::= & (\boxed{\langle\text{Value}\rangle}, \boxed{\langle\text{Value}\rangle}) \;\; [p_9] \\
\langle\text{Value}\rangle & ::= & \text{Progress}(\alpha) \;\; [p_{10}] \;\; | \\
 & & \boxed{\text{Num}} \;\; [p_{11}]
\end{array}
$$

Sampling

```
FlipFill(Circle)
FlipFill(Rotate(Triangle, □, (□, □)))
Rotate(Square, □, (□, □))
```

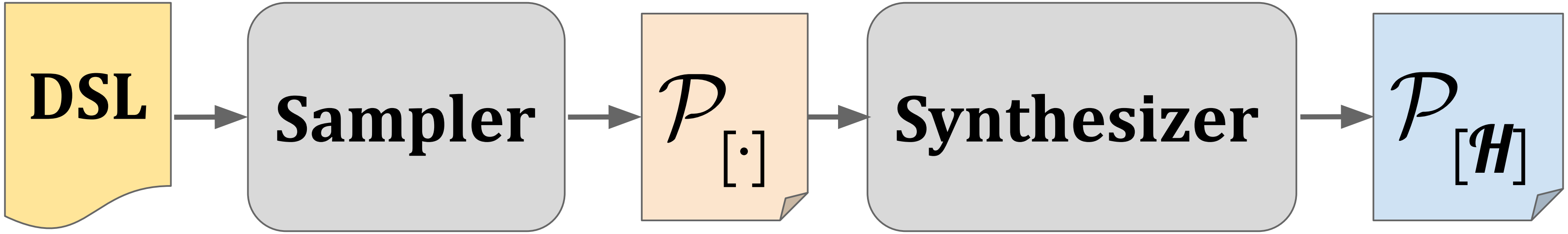
Abstract Program



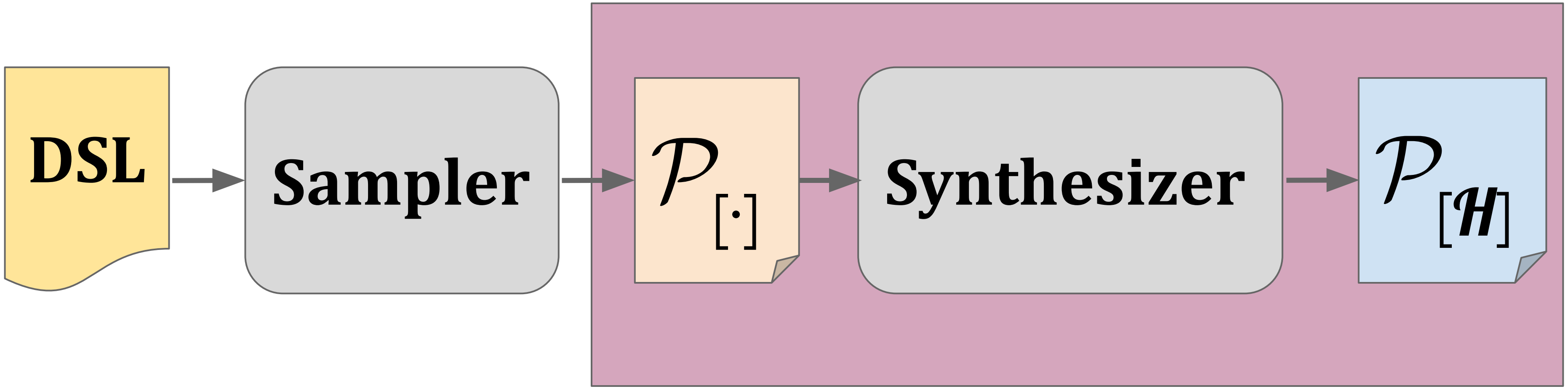
$$
\begin{aligned}
\langle\text{Program}\rangle &::= \lambda\Lambda.\langle\text{Entity}\rangle^{+} \; [p_1] \\
\langle\text{Entity}\rangle &::= \text{GetEntity}(\text{Num}) \; [p_2] \mid \\
&\quad \text{FlipFill}(\langle\text{Entity}\rangle) \; [p_3] \mid \\
&\quad \text{Rotate}(\langle\text{Entity}\rangle, \boxed{\langle\text{Value}\rangle}, \boxed{\langle\text{Coord}\rangle}) \; [p_4] \mid \\
&\quad \text{SelfRotate}(\langle\text{Entity}\rangle, \boxed{\langle\text{Value}\rangle}) \; [p_5] \mid \\
&\quad \text{Translate}(\langle\text{Entity}\rangle, \boxed{\langle\text{Coord}\rangle}) \; [p_6] \mid \\
&\quad \text{Ite}(\boxed{\langle\text{Cond}\rangle}, \langle\text{Entity}\rangle, \langle\text{Entity}\rangle) \; [p_7] \\
\langle\text{Cond}\rangle &::= \Lambda.\text{sid mod } \boxed{\Lambda.id} \; [p_8] \\
\langle\text{Coord}\rangle &::= (\boxed{\langle\text{Value}\rangle}, \boxed{\langle\text{Value}\rangle}) \; [p_9] \\
\langle\text{Value}\rangle &::= \text{Progress}(\alpha) \; [p_{10}] \mid \\
&\quad \boxed{\text{Num}} \; [p_{11}]
\end{aligned}
$$

```
FlipFill(Circle)
FlipFill(Rotate(Triangle, ☐, (☐, ☐)))
Rotate(Square, ☐, (☐, ☐))
```

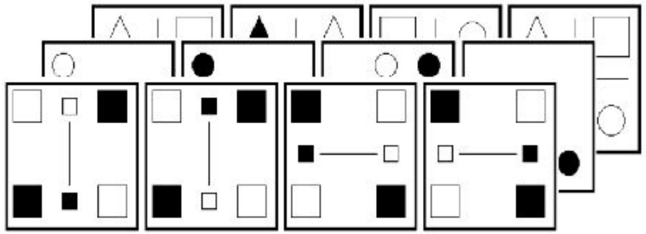
Puzzle Generation Using Program Synthesis



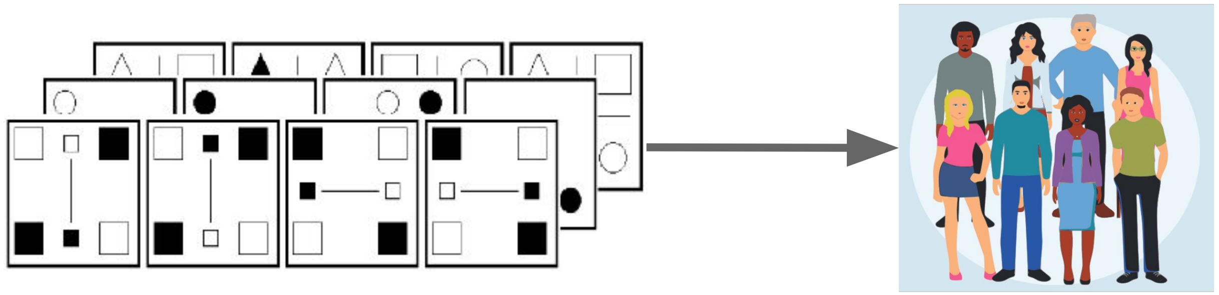
Puzzle Generation Using Program Synthesis



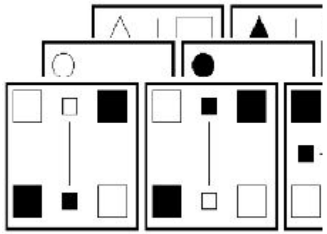
Make Appealing Puzzle

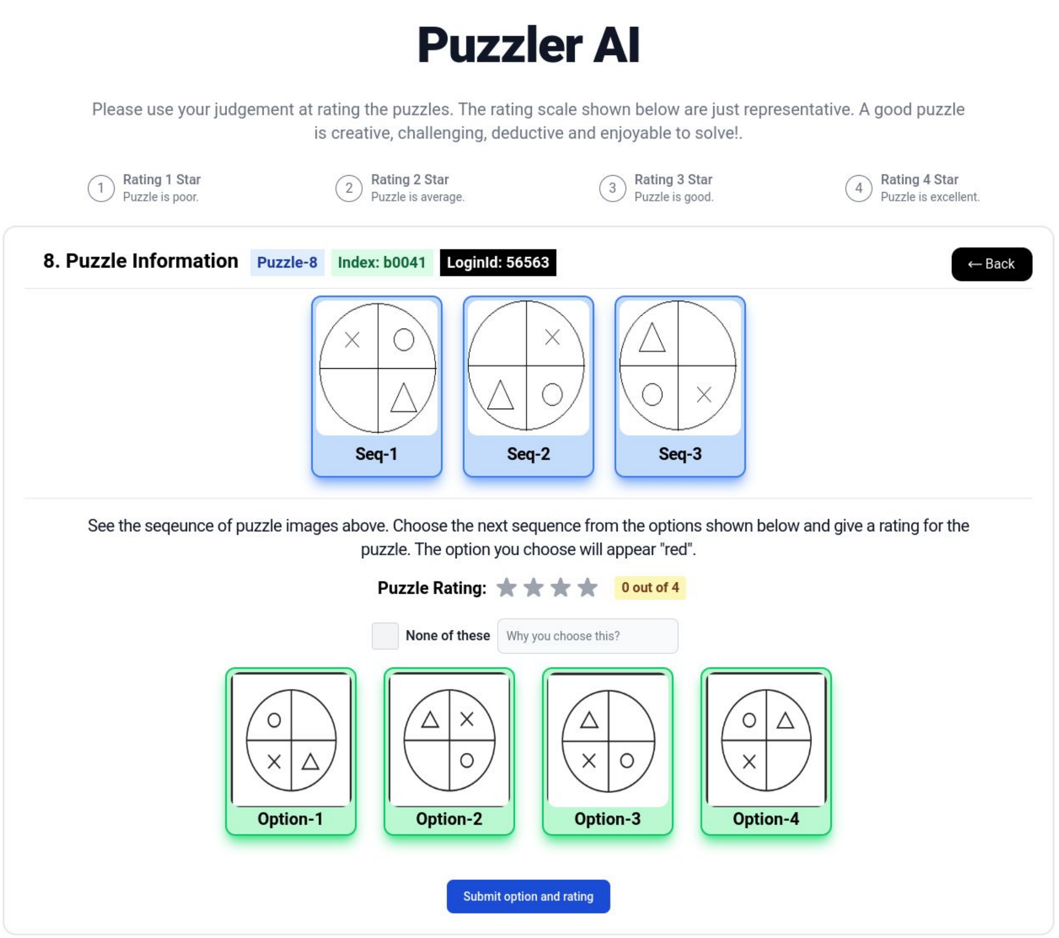


Make Appealing Puzzle



Make A





Puzzler AI

Please use your judgement at rating the puzzles. The rating scale shown below are just representative. A good puzzle is creative, challenging, deductive and enjoyable to solve!.

1 Rating 1 Star Puzzle is poor.
2 Rating 2 Star Puzzle is average.
3 Rating 3 Star Puzzle is good.
4 Rating 4 Star Puzzle is excellent.

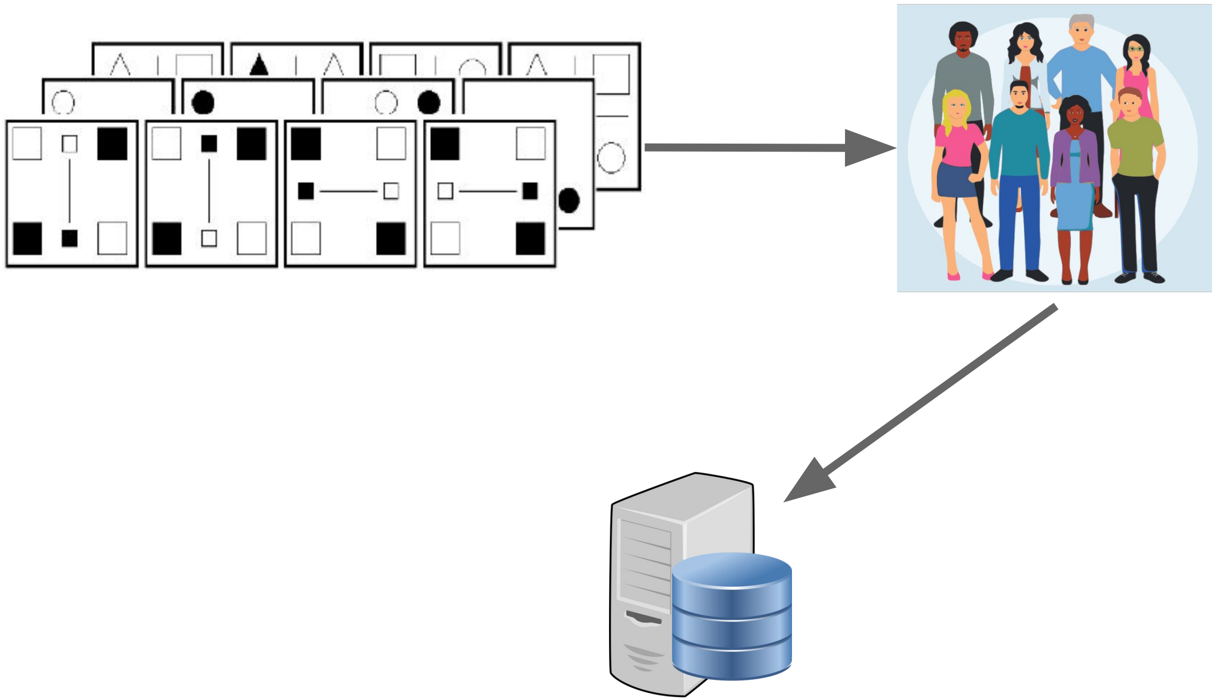
8. Puzzle Information Puzzle-8 Index: b0041 LoginId: 56563

See the seqeunce of puzzle images above. Choose the next sequence from the options shown below and give a rating for the puzzle. The option you choose will appear "red".

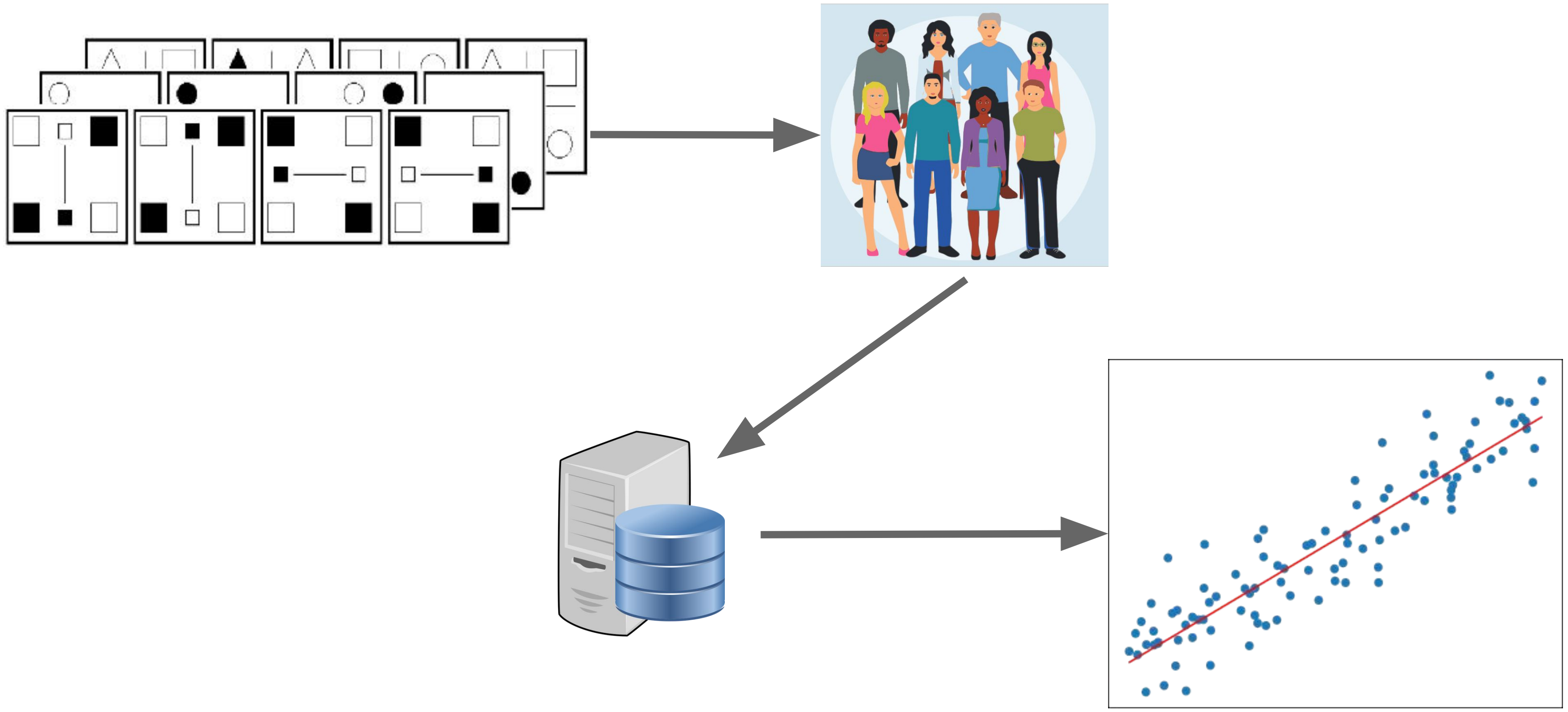
Puzzle Rating: ★★★★ 0 out of 4

None of these Why you choose this?

Make Appealing Puzzle



Make Appealing Puzzle

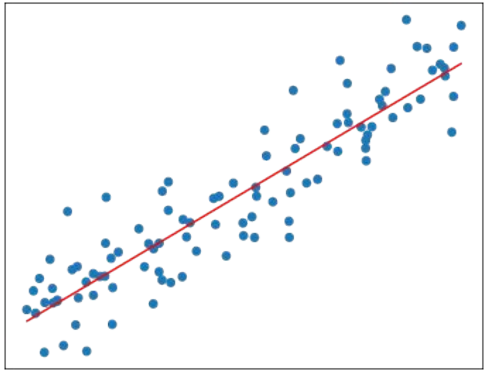


Make Appealing Puzzle

```
FlipFill(Circle)
FlipFill(Rotate(Triangle, ☐, (☐, ☐)))
Rotate(Square, ☐, (☐, ☐))
```

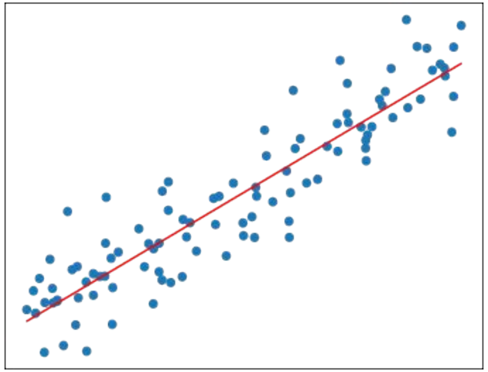
Make Appealing Puzzle

```
FlipFill(Circle)
FlipFill(Rotate(Triangle, ☐, (☐, ☐)))
Rotate(Square, ☐, (☐, ☐))
```



Make Appealing Puzzle

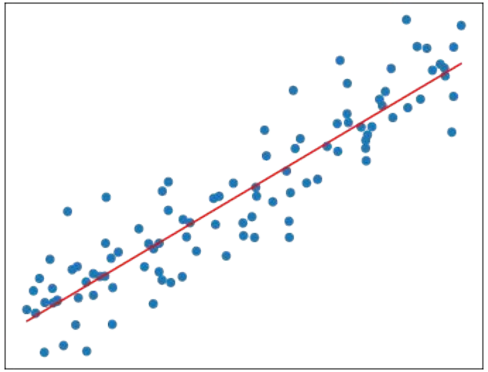
```
FlipFill(Circle)
FlipFill(Rotate(Triangle, □, (□, □)))
Rotate(Square, □, (□, □))
```



Synthesizer

Make Appealing Puzzle

```
FlipFill(Circle)
FlipFill(Rotate(Triangle, ☐, (☐, ☐)))
Rotate(Square, ☐, (☐, ☐))
```



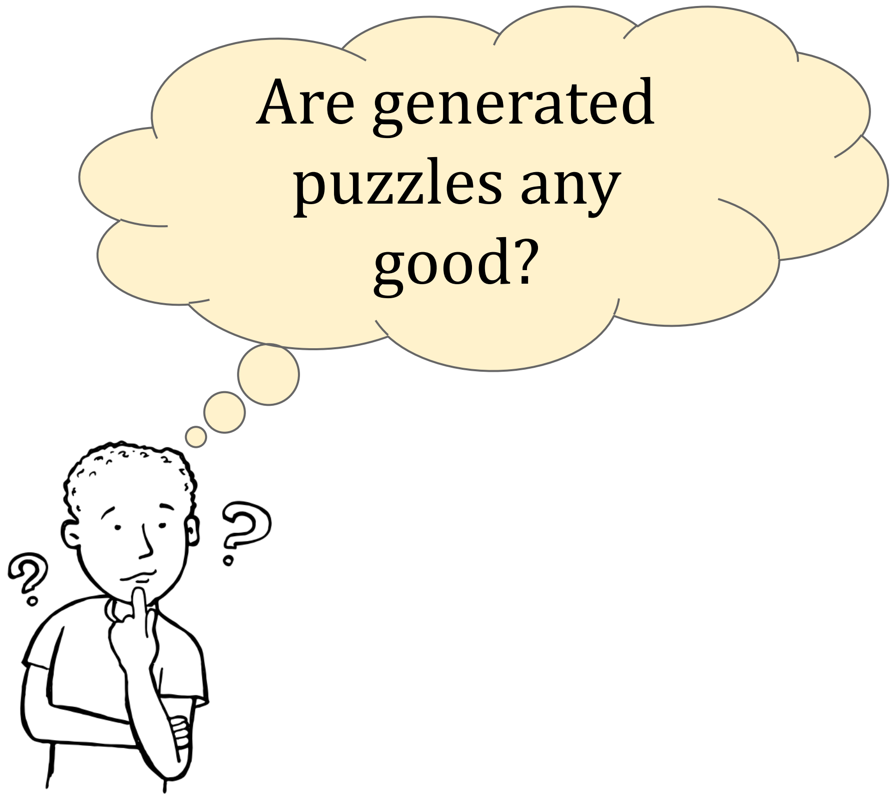
Synthesizer

```
FlipFill(Circle)
FlipFill(Rotate(Triangle, 270, (0, 0)))
Rotate(Square, 270, (0, 0))
```

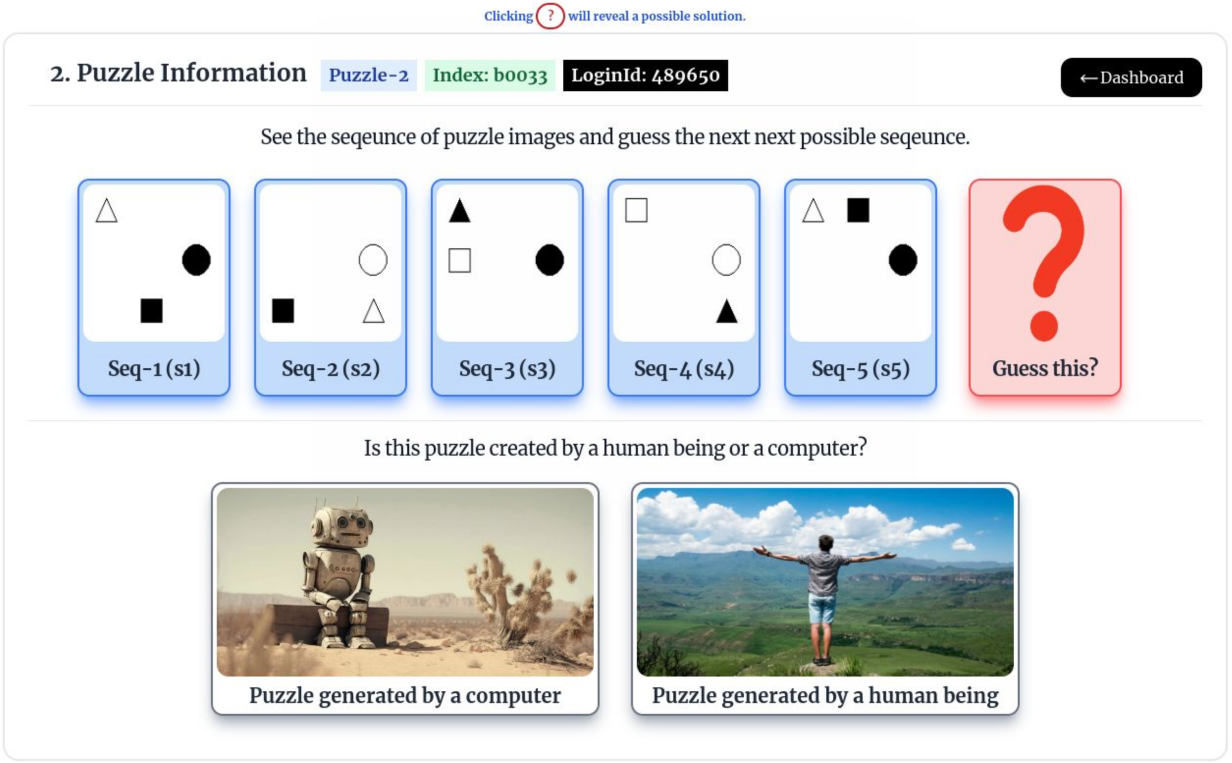
Formally...

$$\underset{\mathcal{H}}{\operatorname{argmax}}\Big\{\text{AppealScore}(\mathcal{P}_{[\mathcal{H}]}) \mid \mathcal{P}_{[\mathcal{H}]} \in \mathcal{L}(\mathcal{G})\ s.t.$$

$$\exists \Lambda_1, \ldots, \Lambda_{k+1}.\ \Big(\underbrace{\bigwedge_{i=1}^{k} (\mathcal{P}_{[\mathcal{H}]}(\Lambda_i) = \Lambda_{i+1})}_{\text{logically follows}}\Big) \bigwedge \underbrace{\forall_{i=1}^{k+1} \Omega(\Lambda_i)}_{\text{state constraints}}\Big\}$$


Are generated puzzles any good?

Humans as a classifier



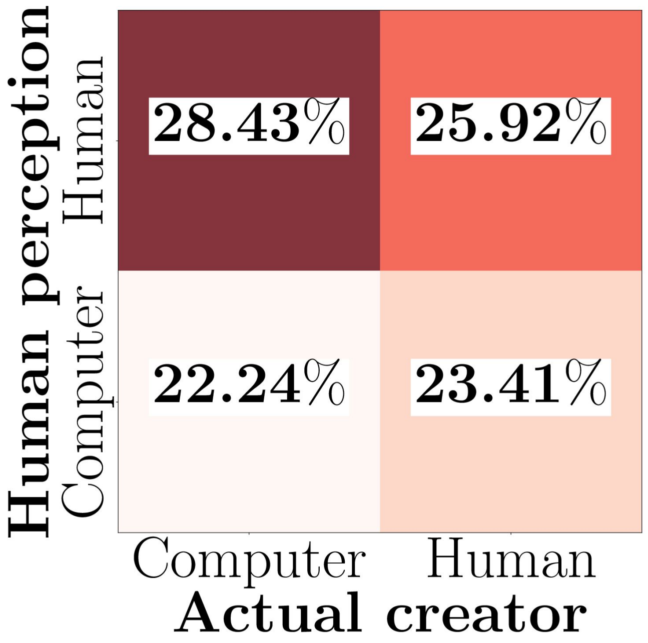
Clicking ? will reveal a possible solution.

2. Puzzle Information Puzzle-2 Index: b0033 LoginId: 489650

See the seqeunce of puzzle images and guess the next next possible seqeunce.

Is this puzzle created by a human being or a computer?

Humans as a classifier



The user responses were almost random.

Conclusion

- Synergistic combination of formal methods and machine learning to generate appealing puzzles
- PuzzleGen took 3.4 secs on average to generate puzzles
- Can be extended to generate puzzles of different categories

Thank You

Paper

Artifact