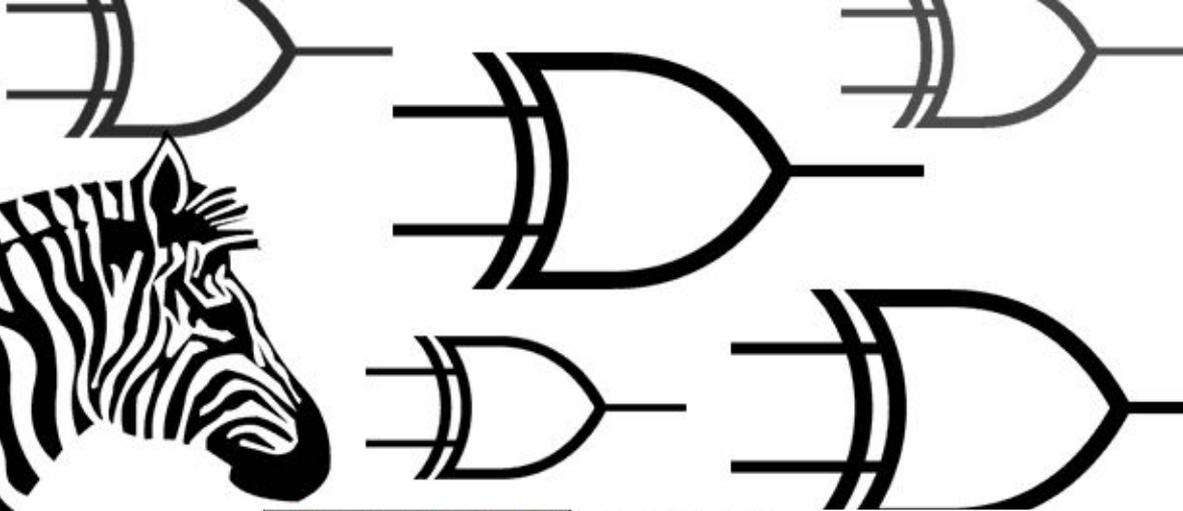


Logiczzle

Gabby Wright, Michaela Murray, Yifei He



2			5		
3			6		
4			7		

1. There are five houses.
2. The white-haired girl lives in the red house.
3. The black-haired girl worships the Fire god.
4. The person fighting with a bow lives in the green house.
5. The redhead fights with scimitars.
6. The green house is immediately to the right of the ivory house.
7. The person of house Bear worships the Moon god.
8. The person of house Dragon lives in the yellow house.
9. Daggers are used by the person living the middle house.
10. The blonde lives in the first house.
11. The person of house Badger lives in the house next to the house of the person who worships the Rising Sun god.
12. The person of House dragon lives next to the house of the person who worships the Star god.
13. The person of house Lion fights with an axe.
14. The brunette belongs to house Stag.
15. The blonde lives next to the blue house.

8			10			12			14		
9			11			13			15		



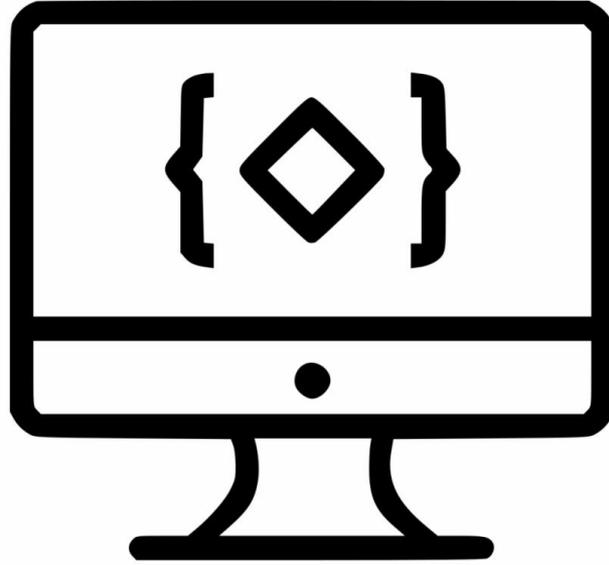
Game Generator

Automatically generate
a logic puzzle



Game Solver

Automatically solve
a logic puzzle



(Almost) Natural Language Generator & Parser

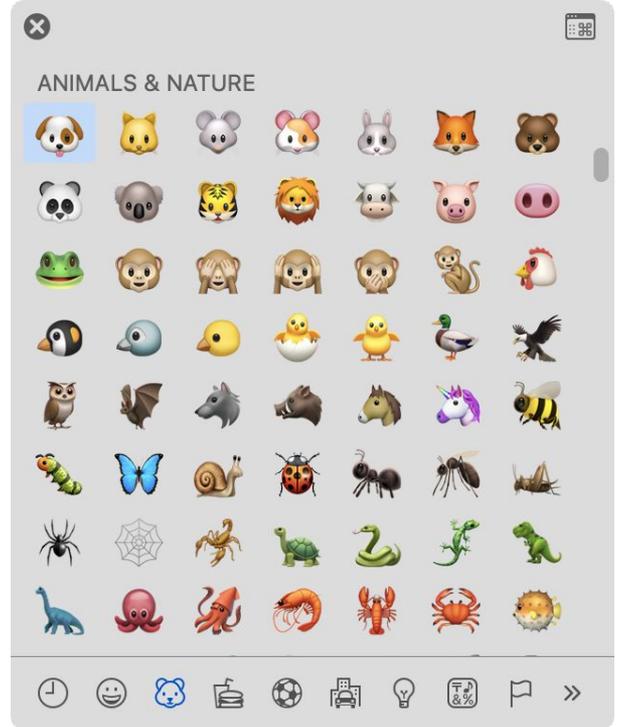
Parse strict language input
Output language

Why logic programming?

1. **Logic** programming is **logical** → it is the most **logical** choice to solve/generate a **logic** puzzle!
2. Relations work **two-way** → can **solve** ↔ can **generate**

Items (emojis) with categories

```
...
item(animal-bird, "turkey", "🦃")
item(animal-bird, "chicken", "🐔")
item(animal-bird, "duck", "🦆")
...
item(animal-mammal, "wolf face", "🐺")
item(animal-mammal, "fox face", "🦊")
item(animal-mammal, "panda face", "🐼")
...
item(book-paper, "green book", "📖")
item(book-paper, "blue book", "📕")
item(book-paper, "orange book", "📗")
...
item(country-flag, "Canada", "🇨🇦")
item(country-flag, "China", "🇨🇳")
item(country-flag, "United States", "🇺🇸")
...
item(food-fruit, "strawberry", "🍓")
item(food-fruit, "kiwi fruit", "🥝")
item(food-fruit, "tomato", "🍅")
...
```



Natural Language Generator & Parser

- The natural language generator is able to create a series of rules based on the set of emojis in the lambda.
- The natural language parser is able to take a series of inputted rules and convert them into the clue text format.
- Example for converting between clue constructor and “natural” language:
 - “ is ” \Leftrightarrow `clue(["", ""], is)`
 - “ is ” \Leftrightarrow `clue(["", ""], is)`
 - “ is to the left of ” \Leftrightarrow `clue(["", ""], is_to_the_left_of)`
 - “ is to the right of ” \Leftrightarrow `clue(["", ""], is_to_the_right_of)`
 - “ is to either side of C” \Leftrightarrow `clue(["", "C"], is_to_either_side_of)`
 - “ is at position 3” \Leftrightarrow `clue(["", 3], is_at_certain_location]`
 - ...

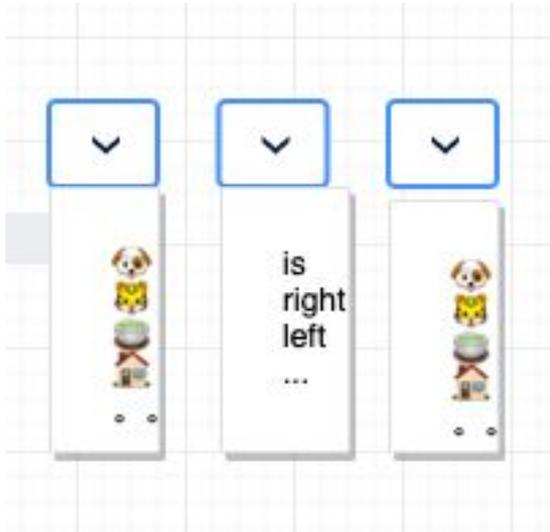
Natural Language Generator & Parser

- We also aim to parse more natural input with verbs.
- Based on the verb used in the particular rule, we represent the statement in an easy to read/represent format.
- Example for converting between natural language statement and clue constructor:
 - “🇨🇦 has 🐻” ⇒ `clue(["🇨🇦", "🐻"], is)`
 - “🦊 loves 🚗” ⇒ `clue(["🦊", "🚗"], is)`
 - “✈️ is owned by 🧑” ⇒ `clue(["✈️", "🧑"], is)`
 - “🐎 lives to the right of 🦊” ⇒ `clue(["🐎", "🦊"], is_to_the_right_of)`
 - “🇬🇧 lives to the left of 🇨🇦” ⇒ `clue(["🇬🇧", "🇨🇦"], is_to_the_left_of)`
 - “☕ is drunk next to 💜” ⇒ `clue(["☕", "💜"], is_to_either_side_of)`
 - “🇬🇧 lives at #5” ⇒ `clue(["🇬🇧", 5], is_at_certain_location)`
 - ...

```
...  
verb("has")  
verb("have")  
verb("had")  
verb("having")  
verb("love")  
verb("loves")  
...
```

Logic Puzzle Solver - First Step

GUI interface:



Text interface:

```
There is a row of five houses.  
The Englishman lives in the red house.  
The Spaniard owns the dog.  
Coffee is drunk in the green house.  
The Ukrainian drinks tea.  
The green house is immediately to the right of the  
ivory house.  
The Old Gold smoker owns snails.  
Kools are smoked in the yellow house.  
Milk is drunk in the middle house.  
The Norwegian lives in the first house.  
The man who smokes Chesterfields lives in the house  
next to the man with the fox.  
Kools are smoked in the house next to the house where  
the horse is kept.  
The Lucky Strike smoker drinks orange juice.  
The Japanese smokes Parliaments.  
The Norwegian lives next to the blue house.
```

Logic Puzzle Solver - Second Step

Example: For the Zebra Puzzle, a sample output would look like this:

Nationality					
Color					
Drink					
Pet					
Smoke	K	C	O	L	P

Logic Puzzle Solver

```
solve_puzzle(Ans) :- createMNMMatrix(Ans,5,5) & rules(5,5,
[
  clue(["Englishman","Red"],is),
  clue(["Spaniard","Dog"],is),
  ...
  clue(["Ivory","Green"],is_to_left_of),
  clue(["Snails","Old Gold"],is),
  clue(["Fox","Chesterfield"],is_to_either_side_of),
  clue(["Norwegian","Blue"],is_to_either_side_of),
  clue(["Horse","Kools"],is_to_either_side_of),
  clue(["Water"],is_type),
  clue(["Norwegian",1],is_at_certain_location),
  clue(["Milk",3],is_at_certain_location),
  clue(["Zebra"],is_type)
], [
  listpair(0,nationality),
  listpair(1,color),
  listpair(2,drink),
  listpair(3,pet),
  listpair(4,smoke)
], Ans
)
```

```
item(nationality,"Norwegian",)
item(nationality,"Ukrainian",)
item(nationality,"Englishman",)
")
item(nationality,"Spaniard",)
item(nationality,"Japanese",)

...

item(drink,"Water",)
item(drink,"Tea",)
item(drink,"Milk",)
item(drink,"Orange juice",)
item(drink,"Coffee",)

...

item(pet,"Fox",)
item(pet,"Horse",)
item(pet,"Snails",)
item(pet,"Dog",)
item(pet,"Zebra",)
```

Logic Puzzle Solver

```
[  
  ["Norwegian", "Ukrainian", "Englishman", "Spaniard", "Japanese"],  
  ["Yellow", "Blue", "Red", "Ivory", "Green"],  
  ["Water", "Tea", "Milk", "Orange juice", "Coffee"],  
  ["Fox", "Horse", "Snails", "Dog", "Zebra"],  
  ["Kools", "Chesterfield", "Old Gold", "Lucky Strike", "Parliament"]  
]
```

```
solve_puzzle(Ans) :- createMMatrix(Ans,5,5) & rules(5,5,  
  [  
    clue(["Englishman","Red"],is),  
    ...  
    clue(["Zebra"],is_type)  
  ], [  
    listpair(0,nationality),  
    listpair(1,color),  
    listpair(2,drink),  
    listpair(3,pet),  
    listpair(4,smoke)  
  ], Ans  
)
```


Logic Puzzle Generator

2. Generate all possible clues

```
[clue(["🐺", "🍰"], is),  
clue(["🐺", "👩"], is),  
...  
clue(["🐰", "🇨🇳"], is),  
clue(["🐺", "🦊"], is_to_left_of),  
clue(["🐺", "🍬"], is_to_left_of),  
...  
clue(["🇷🇺", "👩"], is_to_left_of),  
clue(["🇷🇺", "🇨🇳"], is_to_left_of),  
clue(["🦊", "🐺"], is_to_right_of),  
clue(["🍬", "🐺"], is_to_right_of),  
...  
clue(["👩", "🇷🇺"], is_to_right_of),  
clue(["🇨🇳", "🇷🇺"], is_to_right_of),  
...  
% (CONTINUED TO NEXT PAGE)
```



Logic Puzzle Generator

2. Generate all possible clues

% (CONTINUED FROM LAST PAGE)

```
clue(["🐺", "🦊"], is_to_either_side_of),  
clue(["🦊", "🐺"], is_to_either_side_of),  
...  
clue(["🇷🇺", "🇨🇳"], is_to_either_side_of),  
clue(["🇨🇳", "🇷🇺"], is_to_either_side_of),  
clue(["🐺", 1], is_at_certain_location),  
clue(["🍰", 1], is_at_certain_location),  
...  
clue(["👱", 5], is_at_certain_location),  
clue(["🇨🇳", 5], is_at_certain_location)  
]
```



3. Randomly choose some clues from all the possible clues.

Aim to choose minimum number that makes the puzzle solvable.

Problem: might be inefficient - have to generate thousands of clues. Alternative: generate clues then try solve it?

Logic Puzzle Generator - First Step

Please choose the dimensions of your game:

The image shows two identical vertical dropdown menus. Each menu has a blue square button at the top containing a downward-pointing chevron. Below the button is a list of numbers: 1, 2, 3, 4, and 5. A large black 'X' is positioned between the two dropdown menus. A light gray horizontal bar is partially visible behind the left dropdown menu.

#	1	2	3	4	5
Animal Mammal					
Food Sweet					
Person Role					
Country Flag					

Clues:

- 🐎 is 🍰
- 🐰 is 🇨🇳
- 🧑 is at position 5
- 🐎 is to the left of 🍬
- 🇨🇳 is to the right of 🇷🇺
- WIP...

Add one more hints!

Green background when the item in the cell is compliant to the clues related to this item (but green cell does not necessarily mean that is the correct answer).

Red background when the item in the cell raise a contradiction to the clues related to this item.

#	1	2	3	4	5
Animal Mammal					
Food Sweet					
Person Role					
Country Flag					

Clues:

- 🐎 is 🍰
- 🐰 is 🇨🇳
- 🇨🇳 is at position 5
- 🐎 is to the left of 🍬
- 🇨🇳 is to the right of 🇷🇺
- WIP...

Add one more hints!

The clues will be automatically checked/unchecked based on whether selection above satisfied this clue.

#	1	2	3	4	5
Animal Mammal					
Food Sweet					
Person Role					
Country Flag					

Clues:

- 🐎 is 🍰
- 🐰 is 🇨🇳
- 🧑 is at position 5
- 🐎 is to the left of 🍬
- 🇨🇳 is to the right of 🇷🇺
- WIP...

Add one more hints!

A "Hint" button automatically generates a new hint.

#	1	2	3	4	5
Animal Mammal					
Food Sweet					
Person Role					
Country Flag					

Clues:

- 🐎 is 🍰
- 🐰 is 🇨🇳
- 🇨🇳 is at position 5
- 🐎 is to the left of 🍬
- 🇨🇳 is to the right of 🇷🇺
- WIP...

Add one more hints!

Player can rate a puzzle as easy, medium or hard after they have solved the puzzle. User-aggregated data determines the difficulties of a particular (random) puzzle. They can also share or save the puzzle.

Related statistics: user rating, numbers of hints needed, time used to solve it

Logiczzle

Gabby Wright, Michaela Murray, Yifei He