

You can compare the outputs with the family tree to ensure that the answers are indeed correct.

## Analyzing geography

Let's use logic programming to build a solver to analyze geography. In this problem, we will specify information about the location of various states in the US and then query our program to answer various questions based on those facts and rules. The following is a map of the US:



You have been provided with two text files named `adjacent_states.txt` and `coastal_states.txt`. These files contain the details about which states are adjacent to each other and which states are coastal. Based on this, we can get interesting information like What states are adjacent to both Oklahoma and Texas? or Which coastal state is adjacent to both New Mexico and Louisiana?

Create a new Python file and import the following:

```
from logpy import run, fact, eq, Relation, var
```

Initialize the relations:

```
adjacent = Relation()
coastal = Relation()
```

Define the input files to load the data from:

```
file_coastal = 'coastal_states.txt'
file_adjacent = 'adjacent_states.txt'
```

Load the data:

```
# Read the file containing the coastal states
with open(file_coastal, 'r') as f:
    line = f.read()
    coastal_states = line.split(',')
```

Add the information to the fact base:

```
# Add the info to the fact base
for state in coastal_states:
    fact(coastal, state)
```

Read the adjacency data:

```
# Read the file containing the coastal states
with open(file_adjacent, 'r') as f:
    adjlist = [line.strip().split(',') for line in f if line and
               line[0].isalpha()]
```

Add the adjacency information to the fact base:

```
# Add the info to the fact base
for L in adjlist:
    head, tail = L[0], L[1:]
    for state in tail:
        fact(adjacent, head, state)
```

Initialize the variables x and y:

```
# Initialize the variables
x = var()
y = var()
```

We are now ready to ask some questions. Check if Nevada is adjacent to Louisiana:

```
# Is Nevada adjacent to Louisiana?
output = run(0, x, adjacent('Nevada', 'Louisiana'))
print('\nIs Nevada adjacent to Louisiana?:')
print('Yes' if len(output) else 'No')
```

Print out all the states that are adjacent to Oregon:

```
# States adjacent to Oregon
output = run(0, x, adjacent('Oregon', x))
print('\nList of states adjacent to Oregon:')
for item in output:
    print(item)
```

List all the coastal states that are adjacent to Mississippi:

```
# States adjacent to Mississippi that are coastal
output = run(0, x, adjacent('Mississippi', x), coastal(x))
print('\nList of coastal states adjacent to Mississippi:')
for item in output:
    print(item)
```

List seven states that border a coastal state:

```
# List of 'n' states that border a coastal state
n = 7
output = run(n, x, coastal(y), adjacent(x, y))
print('\nList of ' + str(n) + ' states that border a coastal state:')
for item in output:
    print(item)
```

List states that are adjacent to both Arkansas and Kentucky:

```
# List of states that adjacent to the two given states
output = run(0, x, adjacent('Arkansas', x), adjacent('Kentucky', x))
print('\nList of states that are adjacent to Arkansas and Kentucky:')
for item in output:
    print(item)
```

The full code is given in `states.py`. If you run the code, you will see the following output:

```
Is Nevada adjacent to Louisiana?:  
No  
  
List of states adjacent to Oregon:  
Washington  
California  
Nevada  
Idaho  
  
List of coastal states adjacent to Mississippi:  
Alabama  
Louisiana  
  
List of 7 states that border a coastal state:  
Georgia  
Pennsylvania  
Massachusetts  
Wisconsin  
Maine  
Oregon  
Ohio  
  
List of states that are adjacent to Arkansas and Kentucky:  
Missouri  
Tennessee
```

You can cross-check the output with the US map to verify if the answers are right. You can also add more questions to the program to see if it can answer them.

## Building a puzzle solver

Another interesting application of logic programming is in solving puzzles. We can specify the conditions of a puzzle and the program will come up with a solution. In this section, we will specify various bits and pieces of information about four people and ask for the missing piece of information.

In the logic program, we specify the puzzle as follows:

- Steve has a blue car
- The person who owns the cat lives in Canada
- Matthew lives in USA
- The person with the black car lives in Australia
- Jack has a cat
- Alfred lives in Australia
- The person who has a dog lives in France
- Who has a rabbit?