PYTHON FOR COMPUTATIONAL PROBLEM SOLVING

Introduction to the Course Title

Prof. Sindhu R Pai
PCPS Theory Anchor - 2024
Department of Computer Science and Engineering

PES UNIVERSITY
CELEBRATING 50 YEARS

Course Objectives

- Learn syntax and semantics of Python programming language
- Illustrate the process of structuring data using lists, tuples, sets and dictionaries
- Demonstrate use of built in functions to navigate the file system
- Learn various paradigms of programming and implement Object-Oriented Programming concepts in Python

Course Outcomes

At the end of this course, students will be able to:

- Program effectively using the Python language
- Identify the methods to create and manipulate lists, tuples, sets and dictionaries
- Discover commonly used operations using file system
- Think using different paradigms of programming and interpret the concepts of Object-Oriented Programming as used in Python

PES UNIVERSITY
CELEBRATING 50 YEARS

Textbooks

- "Think Python: How to Think Like a Computer Scientist" Allen B. Downey, 2nd Edition, Green Tea Press, 2015
https://greenteapress.com/thinkpython2/thinkpython2.pdf

- "Automate the Boring Stuff with Python" Al Sweigart, 1st Edition, No Starch Press, 2015
https://automatetheboringstuff.com

Reference Books

"Introduction to Computer Science Using Python: A Computational Problem-Solving Focus"
- Charles Dierbach, Wiley India Edition, John Wiley, 2015

"Learn Python Programming"
- Fabrizio Romano, 2nd Edition, Packet Publishing, 2018

"Fundamentals of Python: First Programs"
- Kenneth A. Lambert, Cengage, 2019

"Introduction to Computation and Programming Using Python: With Application to Understanding Data"
- John V. Guttag, MIT Press, 2016

What does a computer do?

Fundamentally performs calculations, remembers results

What types of calculations can a computer do?

- Basic built-in operations like simple arithmetic and logic
- More complex operations derived from the built-in operations

PES UNIVERSITY
CELEBRATING 50 YEARS

Can a computer perform / solve any task at all?

No, of course not!

This gives us a basic classification of the types of tasks:

- **Computational:** Those which can be solved by computers
  Eg: Sorting a list of numbers, weather forecasting etc.

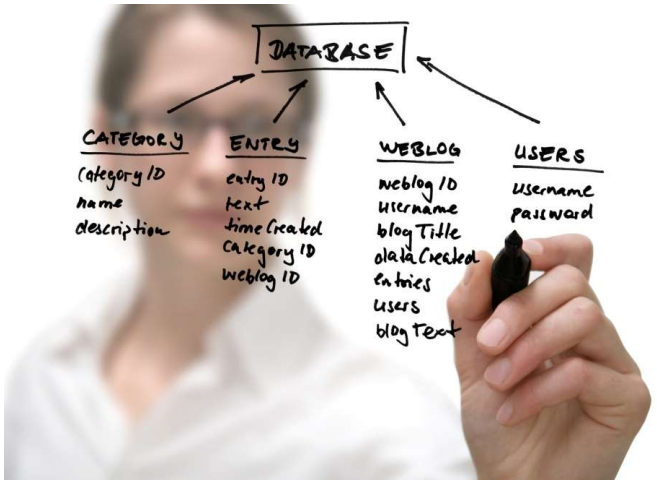
- **Non-computational:** Those which cannot be solved by computers
  Eg: Painting a picture, solving a moral dilemma etc.

PES UNIVERSITY
CELEBRATING 50 YEARS

So what exactly makes a problem computational?

Two main aspects to all computational problems:

- **A representation** that captures all the relevant aspects of the problem.
- **An algorithm** that solves the problem by use of the representation. A sequence of unambiguous instructions for solving a problem, i.e.,for obtaining a required output for any legitimate input in a finite amount of time. The word "algorithm" is derived from the name of a ninth century Arab mathematician, Al-Khwarizmi.



MCGW Problem

Problem Statement: Man, Cabbage, Goat, Wolf Problem

A man lives on the east side of a river. He wishes to bring a cabbage, a goat, and a wolf to a village on the west side of the river to sell. However, **his boat is only big enough to hold himself, and either the cabbage, goat, or wolf**. In addition, the man **cannot leave the goat alone with the cabbage** because the goat will eat the cabbage, and he **cannot leave the wolf alone with the goat** because the wolf will eat the goat. How does the man solve his problem?

MCGW Problem

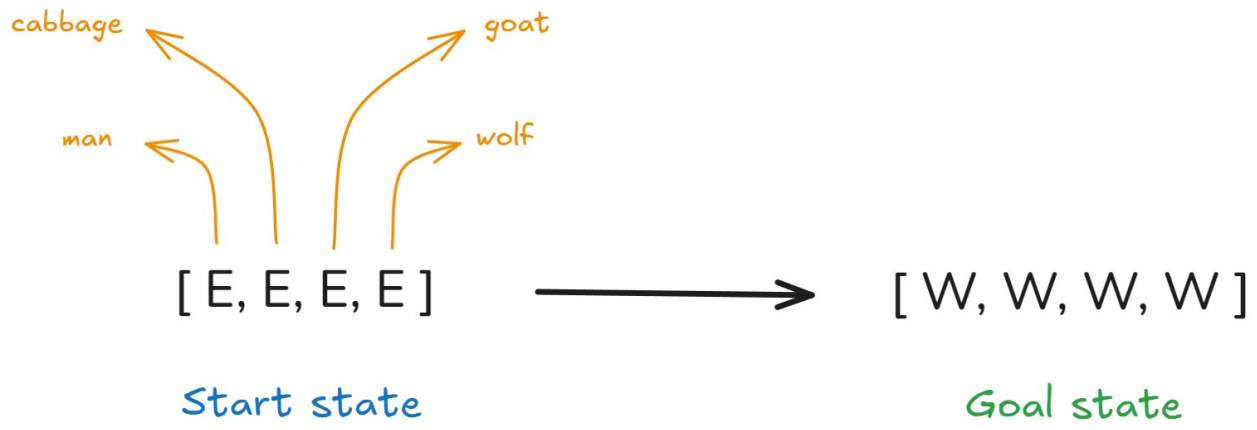
Solution:

There is a simple algorithmic approach for solving this problem by simply trying all Possible combinations of items that may be rowed back and forth across the river to find what works.

Trying all possible solutions is referred to as a brute force approach.

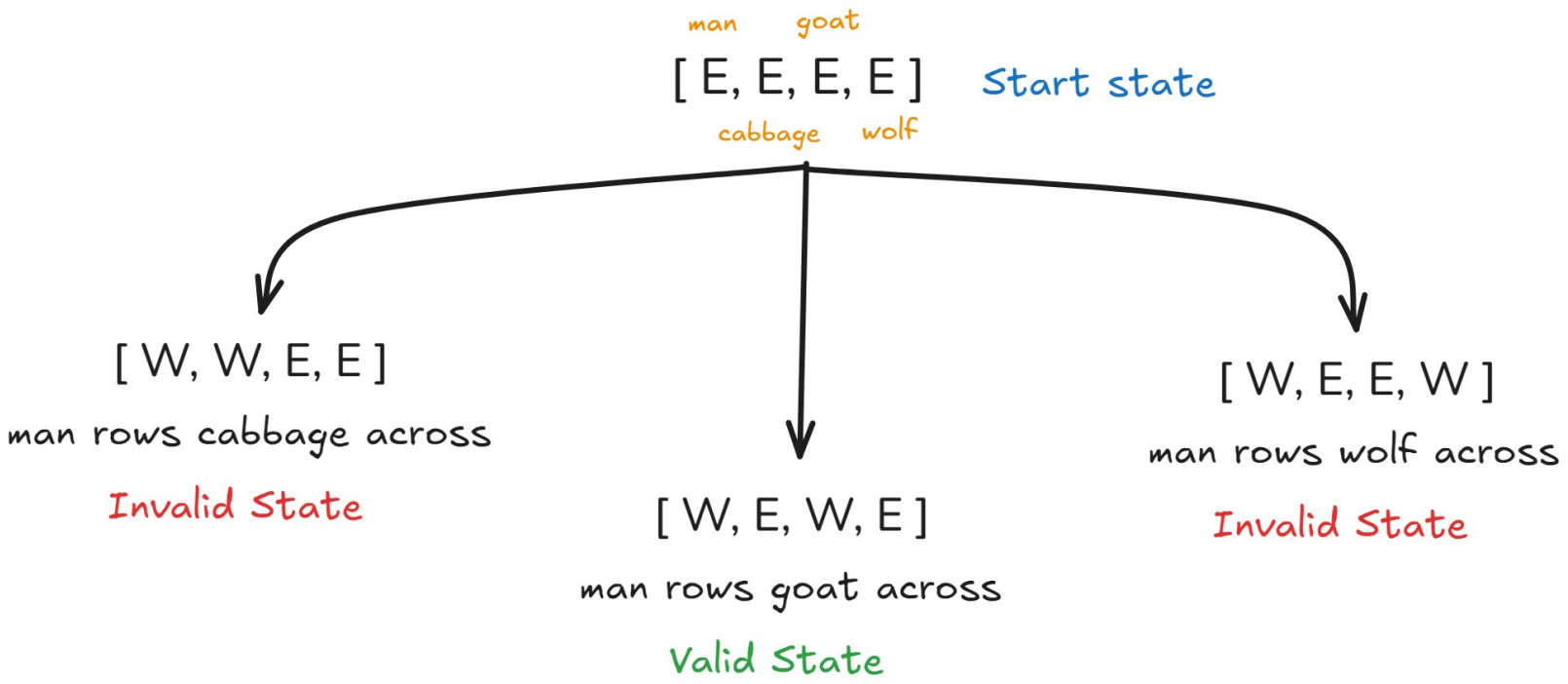
MCGW Problem

Computational Problem: Find a method to convert the **start state** (all objects on the east of the river) to the **goal state** (all objects on the west side of the river) with the **constraint** that certain invalid states should never be used.



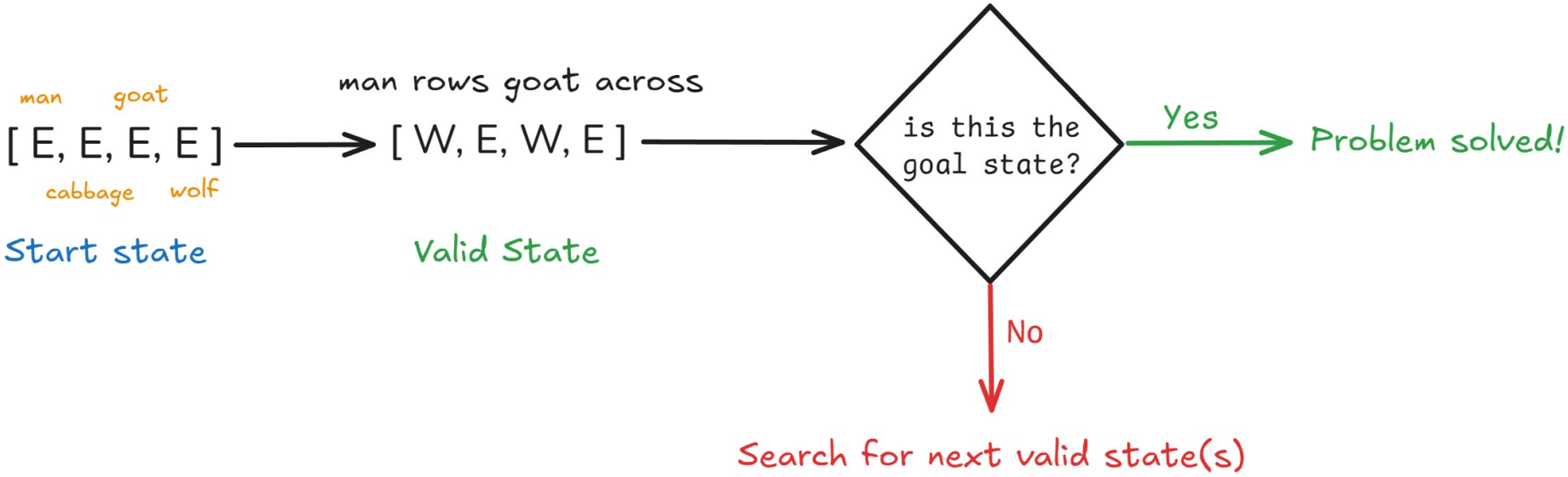
MCGW Problem

For example: From the start state, there are three possible moves that can be made, only one of which results in a valid state.



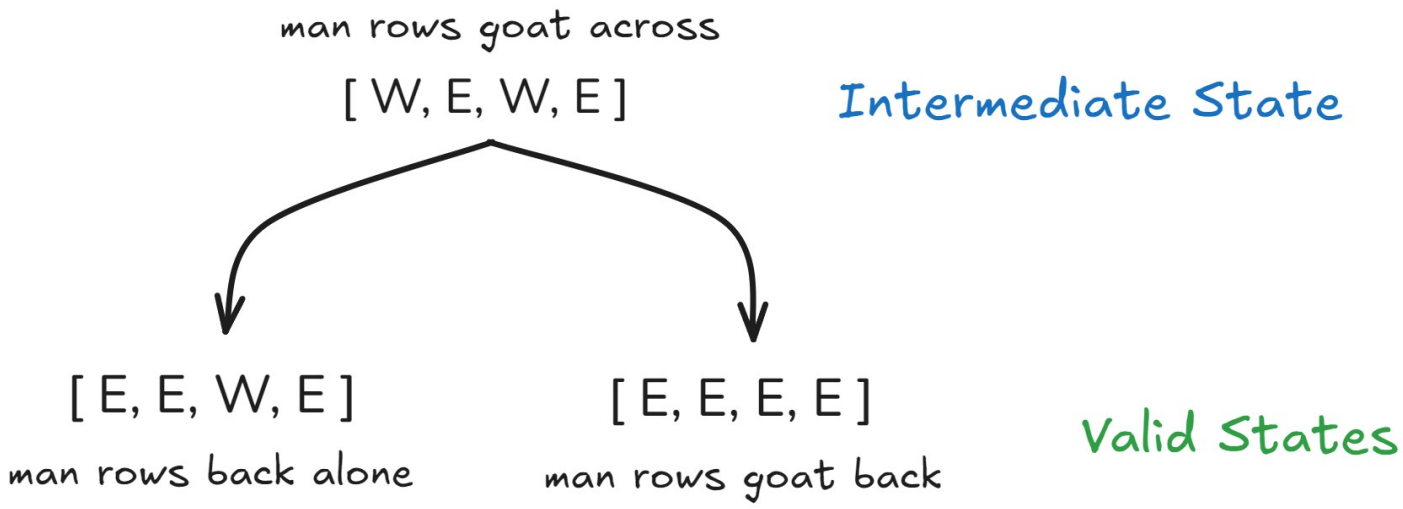
MCGW Problem

We check if the new problem state is the goal state. If true, then we've solved the problem in one step! (We know in our case that it isn't, but an algorithmic approach doesn't have the benefit of intuition like we do)



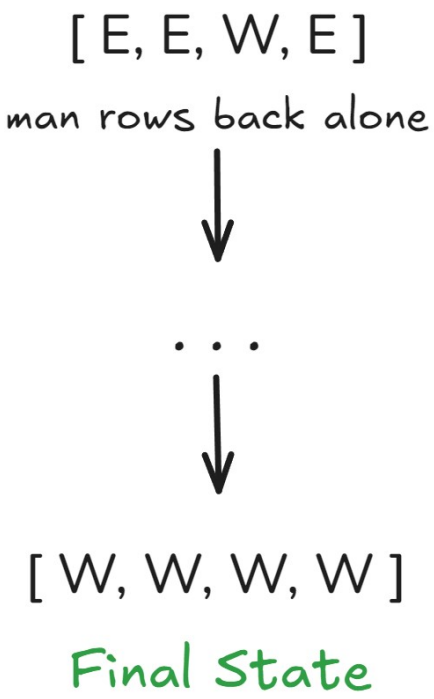
MCGW Problem

Since the man can only row across objects on the same side of the river, there are only two possible moves from here:



MCGW Problem

This would continue until the goal state is reached:



The computational problem of generating the goal state from the start state translates into a solution of the actual problem since each transition between states has a corresponding action in the actual problem.

Why Python?

- **Easy to Learn:** Simple readable syntax, looks similar to plain English

- **Versatile:** Wide range of applications - web development, data science, machine learning, automation, game development etc.

- **Extensive Libraries and Frameworks:** Huge number of external libraries and modules available for a variety of use cases

- **Community Support:** Large and active community of developers, extensive documentation and tutorials available

- **Cross-Platform:** Runs on various platforms (Windows, macOS, Linux etc.) without modification

Applications of Python

- **Data Science:** Libraries like NumPy, Pandas, Matplotlib etc. are used for predictive analysis, data processing and data visualisation
- **AI & Machine Learning:** Python is used to develop neural networks and NLP systems using libraries like Tensorflow and Pytorch
- **Web Development:** Frameworks like Django and Flask power large scale web applications like Instagram
- **Drug Discovery:** Python is used in molecular modelling and simulations using libraries like Open Babel and PyMOL

Applications of Python

- **IoT and Embedded Systems:** Integrations with platforms like Raspberry Pi and Arduino to control hardware devices
- **Circuit Design and Simulation:** Tools like PySpice, SKiDL are used for circuit simulation and PCB design
- **Computational Fluid Dynamics (CFD):** Libraries such as OpenFOAM are used for simulating fluid flow
- **Structural Analysis:** Frameworks like OpenSeesPy are used for structural modelling and earthquake engineering

THANK YOU

Department of Computer Science and Engineering

Dr. Shylaja S S, Director, CCBD & CDSAML, PESU
Prof. Sindhu R Pai – sindhurpai@pes.edu
Prof. Chitra G M

Ack: Teaching Assistant – Advaith Sanil Kumar