Lesson: Combinatorial Game Theory

Bookmark this page

By the end of this lesson, you will be able to **define the key ingredients of combinatorial game theory: arenas and strategies**.

Video - Combinatorial Game Theory

0:25 / 5:12 1.50x

It's me again, welcome!

I'm so glad to see you again.

Let's start with some background about game theory.

Game theory is the domain of science interested in studying games, strategies, and winning

conditions.

Context

Games play an important role in many domains of science.

Games are widely used in economics and mathematics, but also have applications in cryptography

where, typically, the goal of a game is to protect pieces of information, as well as

in operating systems where the goal of the game is typically to prevent errors, or in

Introduction

Game theory is the domain of science interested in studying games, strategies, and winning conditions.

Games play an important role in many domains of science. Games are widely used in economics and mathematics, but also have applications in cryptography where, typically, the goal of a game is to protect pieces of information, as well as in operating systems where the goal of the game is typically to prevent errors, or in artificial intelligence.

Games are a very adapted framework for our game in the maze, and a great excuse to investigate advanced algorithms. If you manage to code a program that computes winning positions in a game, you'll be ready for just about any programming challenge!

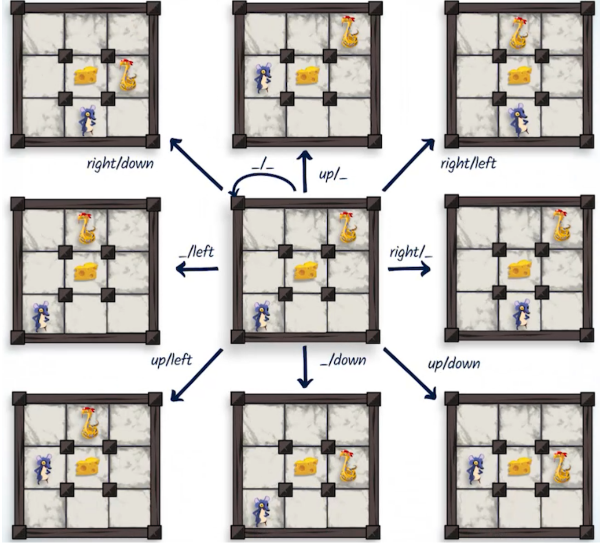
Basic definitions

Recall that we would like to play a game in a maze where two players, a rat and a python, are trying to grab all the pieces of cheese before their opponent. During the game, these players can make simultaneous decisions. For example, the python can decide to move left while the rat decides to move up.

To describe a game, we use a very specific graph that we call the arena. The arena is a graph where each vertex summarizes the actual state of game. By state of the game, we mean the positions of the players, the positions of remaining pieces of cheese, and the score for each player. For example, there's an initial vertex that corresponds to the initial configuration where both the rat and the python have score 0 and are at their starting positions, and all pieces of cheese are spread throughout the maze. Then there's also vertices corresponding to all possible configurations that might occur during a game of PyRat.

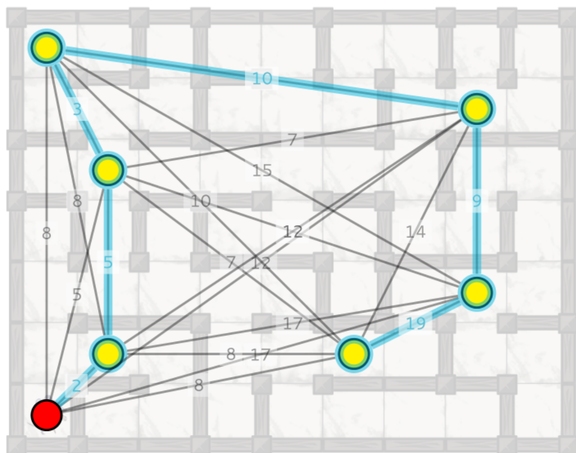
Vertices are connected by edges that are labelled according to two variables.
The first variable corresponds to the decision made by the rat and the second variable to the decision made by the python.

Are you ready for an example? Let's consider a simple game where a 3 by 3 maze contains a single piece of cheese at the center and the rat and python are both competing for it. From the initial configuration several things could happen. The rat can move right, up, or not at all, and the python can move left, down, or not at all. We thus have 3 times 3, so 9, configurations that can be reached from the initial configuration.



There are several vertices in the arena corresponding to the end of the game, when the rat or the python reaches the central position that contains the cheese. The first player to reach this position will be the winner.

A *strategy* of a player is a function that associates each vertex in the arena, which can also be described as each state of the game, with a move for that player.
One example of a strategy is a greedy algorithm which involves systematically going to the nearest piece of cheese in the maze:



Once the two players have chosen their strategies, the game can be entirely known and the winner can be identified by letting the game of PyRat play out. As such, a play can be seen as a walk in the arena, depending on the two chosen strategies.

In the next lesson, we will see how to compute good strategies

Previous   Next

edX

About
Affiliates
edX for Business
Open edX
Careers
News

Legal

Terms of Service & Honor Code
Privacy Policy

Accessibility Policy
Trademark Policy
Sitemap

Connect

Blog
Contact Us
Help Center
Media Kit
Donate