ICSP Final Report

Winning Strategy on ICG

X. Yang & J. T. Xue & J. H. Liu

December 31, 2025

7th Group

Contents

Introduction about ICG

Introduction-What is ICG?

Definition

We call a game **impartial combinatorial games** (ICG), if the following conditions hold:

- 1. It is a game with two players.
  2. Two players make a movement (must act) in turn.
  3. When someone can not move, the game ends, and we say that this player loses.
- 1. A state can be reached only once.
  2. The game ends with finite movements.
  3. Both players have the same strategy to win under a same state.

Example

There are initially n cards on the table. A and B take turns drawing cards, each time drawing a number of cards in range of $1 \sim k$. The person who draws the last card wins.

Our Work

Our Work

Our work is to give a winning strategy for a particular interesting ICG.
That means we will help you win the game if possible.

Our Work

Our Expectation: To create an ICG that is simply packaged and is interactive for players so that they may get a glimpse of principles of this kind of games.

Our Work Division: J. H. Liu and X. Yang will be responsible for the game and UI. J. T. Xue will design the winning algorithm.

Possible Difficulties

- No UI design experience.

Possible Difficulties

- No UI design experience.
- Hard to formalize the problem.

Possible Difficulties

- No UI design experience.
- Hard to formalize the problem.
- Design an interesting and interactive game.

Our Solutions

UI Design: Study the pygame module, or turn to other languages to solve this problem.

Algorithm Design: Let J. T. Xue be more hard working and get inspiration from the similar problem.

The Significance of Our Work

To Ourselves

- Familiar with the problem with concrete examples.

To Others

The Significance of Our Work

To Ourselves

- Familiar with the problem with concrete examples.
- Gain an experience of UI design and improve our programming proficiency.

To Others

The Significance of Our Work

To Ourselves

- Familiar with the problem with concrete examples.
- Gain an experience of UI design and improve our programming proficiency.

To Others

- Promote recognition of this math model.

The Significance of Our Work

To Ourselves

- Familiar with the problem with concrete examples.
- Gain an experience of UI design and improve our programming proficiency.

To Others

- Promote recognition of this math model.
- Inspire a new pathway to tackling simple dual player games.

The Significance of Our Work

To Ourselves

- Familiar with the problem with concrete examples.
- Gain an experience of UI design and improve our programming proficiency.

To Others

- Promote recognition of this math model.
- Inspire a new pathway to tackling simple dual player games.
- Promote the study of combinatorial games.

Future Progress

- Apply more intricate UI design to enhance players experience.

Future Progress

- Apply more intricate UI design to enhance players experience.
- Develop a game with more operations.

Future Progress

- Apply more intricate UI design to enhance players experience.
- Develop a game with more operations.
- Integrate the theorem with machine learning and AI.

Analysis

The question is how to design the algorithm properly.

- Let's enumerate all possibilities!

Analysis

The question is how to design the algorithm properly.

- Let's enumerate all possibilities!
- WTF?

Analysis

The question is how to design the algorithm properly.

- Let's enumerate all possibilities!
- WTF?
- If there are n choices with each steps and the game ends with a large k steps, then your computer will see the god with n^k on his head!

Analysis

To analyze the problem systematically, we need a mathematical model.

Analysis i

Consider you are playing the game. There are some choices for you, and you need to pick one of them. It is just like there are many one-way streets in front of you. You go through the street, and then make decision again at next corner.

We abstract it as a graph. Each vertex is a state, and each edge is a move. The rule of the ICG implies that the graph should be finite, without loops, and unidirectional.

A graph G is a triple $(V(G), E(G), \phi_G)$, contains a non-empty set $V(G)$ as **vertices**, set $E(G)$ as **edges**, and a **association function**:

$$\phi_G : E(G) \longrightarrow V(G) \times V(G). \tag{1}$$

Analysis ii

In a directed graph, the association function maps the elements $e \in E(G)$ to an ordered pair (u, v), consisting of the two vertex, where u is the starting point and v is the ending point of e.

In particular, the graph we are studying is tree-shaped (there is a unique path between any two vertices and there are no loops). There exists a unique $r \in V(G)$, such that

$$\forall (e, v) \in E(G) \times V(G),\ \phi_G(e) \neq (v, r). \tag{2}$$

We call it the **root** of the graph. Similarly, we call $l \in V(G)$ the **leaf** of the graph if

$$\forall (e, v) \in E(G) \times V(G),\ \phi_G(e) \neq (l, v). \tag{3}$$

We denote L_G as

$$L_G := \{l \in V(G) \mid \forall (e, v) \in E(G) \times V(G),\ \phi_G(e) \neq (l, v)\}. \tag{4}$$

Analysis iii

Then, we can know that $r(G)$ is the beginning of the game, and L_G contains all the possible ends of the game.

An important fact is that we win if and only the opponent loses. Since the graph is certain under any state, so we want to construct a function to evaluate the state, and indicate the best move by return a special value.

The processing detail is quite complicated. If you are interested in this problem, please scan the QR code.[1]

[1] There are some problems in the article, we will fix them when going through the project.

Conclusion

Summary

We aim to create simple game with winning strategies based on the ICG model. By incorporating UI and algorithm design, we aim to promote recognition of this model in strategizing for combinatorial games. We may further study UI design to cope with problems along the way. The harvest of this project maybe applied to machine learning and AI.

Questions?

Thanks!