Assignment 3: Hashi Game

Ayça Elif Aktaş 27802 - Pelinsu Saraç 28820

1 Introduction

In this document, we outline the design and modeling of a two-player Hashi game, implemented to explore artificial intelligence strategies in zero-sum game environments.

2 Game Model

2.1 Players

The game features two players:

- **Human Player**: Engages with the game via a keyboard, making strategic decisions based on the current state of the game board.
- **AI Player**: Operates based on minimax algorithm with alpha-beta pruning to analyze the board and make optimal moves.

2.2 States

The game state is represented by the game board configurations. The game board is represented by a matrix configuration, including:

- **Labeled Islands**: Numbered from 1 to 4, indicating the maximum number of connecting bridges.
- **Empty Islands**: Denoted by '0', potential to be labeled during the game with the label 3 or 4.
- **Bridges**: Either horizontal or vertical lines that connect two islands.
  - Single Horizontal bridge; -
  - Double Horizontal bridge; =
  - Single Vertical bridge; |
  - Double Vertical bridge; x
- **Empty Cells**: Denoted by '.', representing free spaces on the board.

2.3 Initial State

The initial state of the game board is loaded from the provided `HashiGame.py` file, depicting a predefined arrangement of labelled and unlabelled islands, and empty spaces in the matrix format described above. There is no bridges at the initial state.

2.4 Terminal States

Terminal states occur when no further moves are available, either due to full utilization (max two bridges between islands and labels of islands limiting the further placement of the bridges) of bridge and label possibilities or blockages preventing legal placements.

2.5 State Transition Function

State transitions occur through the application of moves by each player, executed via the `apply_move` and `auto_play` function in the implementation. This function updates the game state by adding bridges or labeling empty islands with 3 or 4 given the board state and the move by the current player, adhering to the game rules.

2.6 Payoff Function

The payoff function evaluates and adjusts players' scores based on the board's state after each move:

- When the number of bridges connected to an island matches the island's label, the current player scores points equivalent to the label number, and the opposing player loses the same amount, ensuring the game's zero-sum nature.