

Sztuczna inteligencja i inżynieria wiedzy,
Lista 2

Tymoteusz Trętowicz, 260451

1 Implementacja

Gra Reversi jest grą planszową, w której gracze wykonują ruchy na zmianę kładąc dyski swojego koloru na planszy w stylu siatki rozmiarów 8 na 8 dysków. Zwyczajowo zaczyna gracz grający dyskami koloru ciemnego. Gracze mogą kłaść dyski na planszy w taki sposób, żeby powstała linia zaczynała się i kończyła się od dysków gracza wykonującego ruch. Wówczas wszystkie dyski pomiędzy dwoma granicznymi dyskami są obracane, tak żeby były koloru gracza wykonującego ruch. Gra kończy się w momencie gdy żaden z graczy nie może wykonać legalnego ruchu. Wygrywa gracz, który ostatecznie ma na planszy więcej dysków swojego koloru.

Implementacja w języku Go opiera się na dwuwymiarowej tablicy liczb całkowitych. Puste pola są reprezentowane liczbą -1, pola białe liczbą 0, a pola czarne liczbą 1. Ruchy są reprezentowane jako para par liczb (wiersz i kolumna) oraz kolor dysków gracza wykonującego ruch.

W retrospektyce wygodniejszym rozwiązaniem byłoby przypisanie koloru białego do liczby -1, a pól pustych do liczby 0. Wówczas:

$$K_{\text{CZARNY}} = -K_{\text{BIAŁY}}$$

Funkcja

```
StartGame(whitePlayer moveGenerator, blackPlayer moveGenerator) → WinnerColor
```

akceptuje dwa parametry, które reprezentują graczy. Typ `moveGenerator` ma sygnaturę:

```
moveGenerator: f(*[8][8]int,int,int) → (int,int)
```

Czyli jest to funkcja, akceptująca jako paramter wskaźnik na planszę, kolor gracza wykonującego ruch i numer ruchu oraz zwracająca parę liczb, reprezentujących wygenerowany ruch. W implementacji dostarczone są 3 rodzaje graczy:

1. `StdinPlayer` - gracz wczytujący ruchy ze standardowego wejścia.
2. `RandomPlayer` - gracz wykonujący ruchy losowe.
3. `BestPlayer` - gracz szukający ruchów do podanej głębokości i wykonujący najlepsze wg. heurystyk ruchy.

Heurystyki

Do oceny danej pozycji na planszy wykorzystane są następujące heurystyki:

- Położonych dysków - preferowane są pozycje, w których dany gracz ma na planszy więcej dysków.
- Legalnych ruchów - preferowane są pozycje, w których gracz ma większy wybór ruchów.
- Kluczowych pól - preferowane są pozycje, w których gracz ma zajęte pola kluczowe: D4, E4, D5, E5.
- Rogów - preferowane są pozycje, w których gracz kontroluje rogi planszy, tj. pola: A1, A8, H1, H8.
- Krawędzi - preferowane są pozycje, w których gracz kontroluje krawędzie planszy, czyli pola z kolumn A i H oraz z rzędów 1 oraz 8.

Zatem ostatecznie ocena pozycji jest liczona jako:

$$E = H_1w_1 + H_2w_2 + H_3w_3 + H_4w_4 + H_5w_5$$

$$H_1 = \text{Liczba czarnych pól} - \text{Liczba białych pól}$$

$$H_2 = \text{Liczba legalnych ruchów gracza czarnego} - \text{Liczba legalnych ruchów gracza białego}$$

$$H_3 = \text{Liczba kluczowych pól gracza czarnego} - \text{Liczba kluczowych pól gracza białego}$$

$$H_4 = \text{Liczba rogów gracza czarnego} - \text{Liczba rogów gracza białego}$$

$$H_5 = \text{Liczba krawędzi gracza czarnego} - \text{Liczba krawędzi gracza białego}$$

gdzie w_i waga odpowiedniej heurystyki.

Ta implementacja nie uwzględnia, tego kto właśnie wykonał ruch. To powoduje, że może być oceniana pozycja w której gracz nie wykonali takiej samej liczby ruchów.

Wagi heurystyk są wyznaczone eksperymentalnie (odpowiednio $W = w_1 \cdots w_5 : 1, 4, 4, 5, 5$) lub poprzez algorytm genetyczny, który dla ustalonej liczby generacji oraz populacji w generacji ustala wagi. Wagi początkowo są ustalone na 1. Następnie dla każdego osobnika w populacji, każda waga jest zmieniana poprzez wymnożenie jej przez losową liczbę z przedziału $0.5 \cdots 1.5$. Następnie osobnika odbywa grę z przeciwnikiem (również uczniem algorytmu genetycznego). Jeżeli osobnik przegra, jego wagi są zapominane. W celu obliczenia wag następnej generacji jest brana średnia z wag zwycięzców oraz wagi na początku generacji, oraz jest normalizowana tak by: $\sum_{i=1}^5 w_i = 5$.