Rozwiązywanie gier

Wybrane problemy algorytmiczne i technologiczne, seminarium

Michał Krakowiak

Gdańsk, 05.11.2019

Zawartość

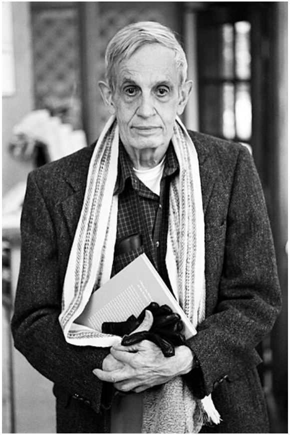
1. Wstęp
   - Zawartość
   - Tło historyczne
   - Definicja gry

Zawartość

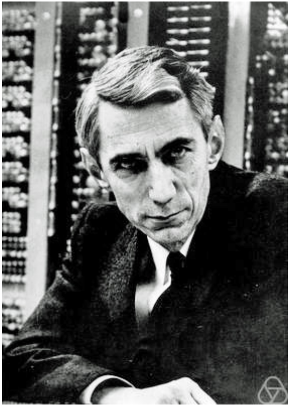
Zawartość

Zawartość

Tło historyczne



Rysunek: John Nash



Rysunek: Claude Shannon

Czym jest gra?

Definicja

Gra jest opisem strategicznej interakcji, która narzuca ograniczenia na udział graczy oraz akcje jakie mogą podjąć [2].

Gra w ujęciu kombinatorycznym

Właściwości rozważanych gier:

Gra w ujęciu kombinatorycznym

Właściwości rozważanych gier:

- Udział tylko dwóch graczy

Gra w ujęciu kombinatorycznym

Właściwości rozważanych gier:

- Udział tylko dwóch graczy
- Gracze wykonują ruchy naprzemiennie

Gra w ujęciu kombinatorycznym

Właściwości rozważanych gier:

- Udział tylko dwóch graczy
- Gracze wykonują ruchy naprzemiennie
- Przebieg nie zależy od czynników losowych

Gra w ujęciu kombinatorycznym

Właściwości rozważanych gier:

- Udział tylko dwóch graczy
- Gracze wykonują ruchy naprzemiennie
- Przebieg nie zależy od czynników losowych
- Gracze posiadają pełną wiedzę o stanie gry

Gra w ujęciu kombinatorycznym

Właściwości rozważanych gier:

- Udział tylko dwóch graczy
- Gracze wykonują ruchy naprzemiennie
- Przebieg nie zależy od czynników losowych
- Gracze posiadają pełną wiedzę o stanie gry
- Gra kończy się po skończonej liczbie ruchów

Gra w ujęciu kombinatorycznym

Właściwości rozważanych gier:

- Udział tylko dwóch graczy
- Gracze wykonują ruchy naprzemiennie
- Przebieg nie zależy od czynników losowych
- Gracze posiadają pełną wiedzę o stanie gry
- Gra kończy się po skończonej liczbie ruchów
- Wygrywa gracz, który wykona ostatni ruch

Klasyfikacja rozwiązań

- Mocne

Klasyfikacja rozwiązań

- Mocne
- Słabe

Klasyfikacja rozwiązań

- Mocne
- Słabe
- Ultra słabe

Rozwiązania mocne

- Jest znany algorytm pozwalający uzyskać optymalne ruchy z każdej pozycji (nawet jeżeli, któryś z graczy popełnił błąd)

Rozwiązania mocne

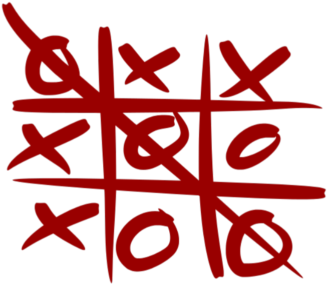
- Jest znany algorytm pozwalający uzyskać optymalne ruchy z każdej pozycji (nawet jeżeli, któryś z graczy popełnił błąd)
- Częste wykorzystanie metod siłowych

Rozwiązania mocne

- Jest znany algorytm pozwalający uzyskać optymalne ruchy z każdej pozycji (nawet jeżeli, któryś z graczy popełnił błąd)
- Częste wykorzystanie metod siłowych
- Dowód może nie być pomocny w zrozumieniu powodów dlaczego dana gra jest rozwiązywalna

Kółko i krzyżyk

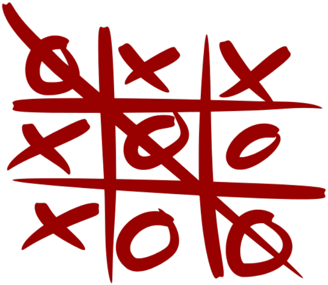
- Gra jest trywialna, ze względu na niewielkie drzewo gry



Rysunek: Kółko i krzyżyk

Kółko i krzyżyk

- Gra jest trywialna, ze względu na niewielkie drzewo gry
- Teoretyczna liczba stanów to $3^9 = 19683$



Rysunek: Kółko i krzyżyk

Kółko i krzyżyk

- Gra jest trywialna, ze względu na niewielkie drzewo gry
- Teoretyczna liczba stanów to $3^9 = 19683$
- Nie każdy stan jest możliwy

X	X	X
O	O	O

Kółko i krzyżyk

- Gra jest trywialna, ze względu na niewielkie drzewo gry
- Teoretyczna liczba stanów to $3^9 = 19683$
- Nie każdy stan jest możliwy
- Niektóre stany są tożsame

X	X	X
O	O	O

		O
X	X	O

Kółko i krzyżyk

- Gra jest trywialna, ze względu na niewielkie drzewo gry
- Teoretyczna liczba stanów to $3^9 = 19683$
- Nie każdy stan jest możliwy
- Niektóre stany są tożsame

X	X	X
O	O	O

		O
X	X	O

X	X	O
		O

Kółko i krzyżyk

- Gra jest trywialna, ze względu na niewielkie drzewo gry
- Teoretyczna liczba stanów to $3^9 = 19683$
- Nie każdy stan jest możliwy
- Niektóre stany są tożsame

X	X	X
O	O	O

		O
X	X	O

X	X	O
		O

O	X	X
O		

Kółko i krzyżyk

- Gra jest trywialna, ze względu na niewielkie drzewo gry
- Teoretyczna liczba stanów to $3^9 = 19683$
- Nie każdy stan jest możliwy
- Niektóre stany są tożsame
- W rezultacie jest tylko 765 różnych stanów [1]

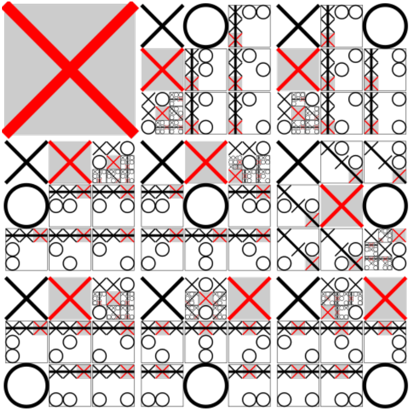
X	X	X
O	O	O

		O
X	X	O

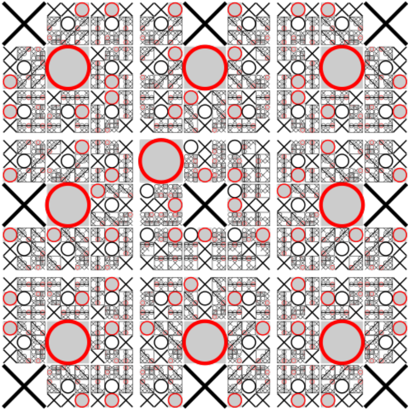
X	X	O
		O

O	X	X
O		

Kółko i krzyżyk



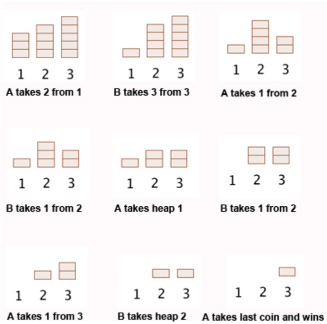
Rysunek: Optymalna gra dla X



Rysunek: Optymalna gra dla O

Nim

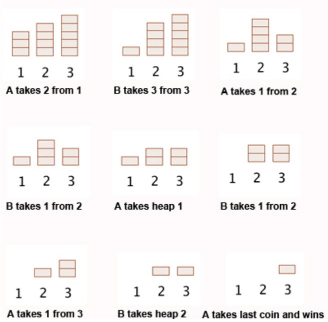
- Matematyczna gra strategiczna



Rysunek: Przykładowa rozgrywka w nim

Nim

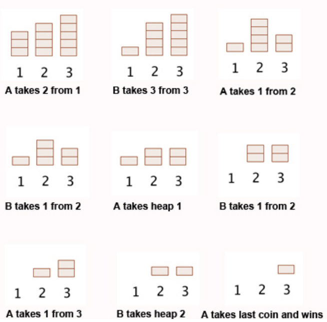
- Matematyczna gra strategiczna
- Gracze zabierają elementy ze stosów



Rysunek: Przykładowa rozgrywka w nim

Nim

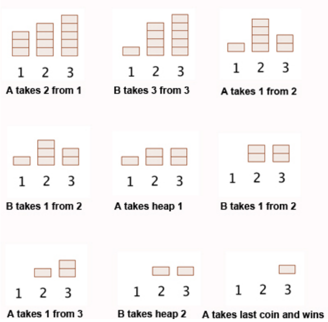
- Matematyczna gra strategiczna
- Gracze zabierają elementy ze stosów
- W czasie ruchu gracz bierze dowolną niezerową liczbę elementów



Rysunek: Przykładowa rozgrywka w nim

Nim

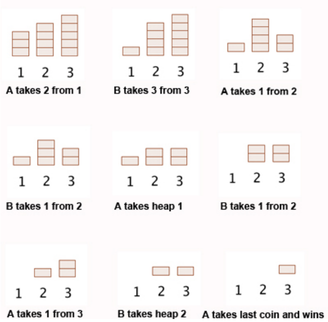
- Matematyczna gra strategiczna
- Gracze zabierają elementy ze stosów
- W czasie ruchu gracz bierze dowolną niezerową liczbę elementów
- W jednym ruchu można zabrać elementy z tylko jednego stosu



Rysunek: Przykładowa rozgrywka w nim

Nim

- Matematyczna gra strategiczna
- Gracze zabierają elementy ze stosów
- W czasie ruchu gracz bierze dowolną niezerową liczbę elementów
- W jednym ruchu można zabrać elementy z tylko jednego stosu
- W klasycznej wersji wygrywa gracz, który zabierze ostatni element



Rysunek: Przykładowa rozgrywka w nim

Nim - strategia

- Kombinatoryczna teoria gier definiuje działanie tzw. *nimsumy*: $x \oplus y$

Nim - strategia

- Kombinatoryczna teoria gier definiuje działanie tzw. *nimsumy*: $x \oplus y$
- Działanie jest także znane jako *alternatywa wykluczająca*, czyli xor

Nim - strategia

- Kombinatoryczna teoria gier definiuje działanie tzw. *nimsumy*: $x \oplus y$
- Działanie jest także znane jako *alternatywa wykluczająca*, czyli xor
- $(a \oplus b) \oplus c = a \oplus (b \oplus c)$

Nim - strategia

- Kombinatoryczna teoria gier definiuje działanie tzw. *nimsumy*: $x \oplus y$
- Działanie jest także znane jako *alternatywa wykluczająca*, czyli xor
- $(a \oplus b) \oplus c = a \oplus (b \oplus c)$
- $a \oplus b = b \oplus a$

Nim - strategia

- Kombinatoryczna teoria gier definiuje działanie tzw. *nimsumy*: $x \oplus y$
- Działanie jest także znane jako *alternatywa wykluczająca*, czyli xor
- $(a \oplus b) \oplus c = a \oplus (b \oplus c)$
- $a \oplus b = b \oplus a$
- $0 \oplus a = a$

Nim - strategia

- Kombinatoryczna teoria gier definiuje działanie tzw. *nimsumy*: $x \oplus y$
- Działanie jest także znane jako *alternatywa wykluczająca*, czyli xor
- $(a \oplus b) \oplus c = a \oplus (b \oplus c)$
- $a \oplus b = b \oplus a$
- $0 \oplus a = a$
- $a \oplus a = 0$

Nim - strategia

Twierdzenie

Gracz rozpoczynający ma wygrywającą strategię wtedy i tylko wtedy, gdy nimsuma rozmiarów stosów jest niezerowa. W przeciwnym razie drugi gracz posiada strategię wygrywającą.

Nim - strategia

- x_i - ilość elementów przed wykonaniem ruchu na *i*-tym stosie

Nim - strategia

- x_i - ilość elementów przed wykonaniem ruchu na i-tym stosie
- y_i - ilość elementów po wykonaniu ruchu na i-tym stosie

Nim - strategia

- x_i - ilość elementów przed wykonaniem ruchu na i-tym stosie
- y_i - ilość elementów po wykonaniu ruchu na i-tym stosie
- Niech $s = x_1 \oplus \ldots \oplus x_n$

Nim - strategia

- x_i - ilość elementów przed wykonaniem ruchu na i-tym stosie
- y_i - ilość elementów po wykonaniu ruchu na i-tym stosie
- Niech $s = x_1 \oplus \ldots \oplus x_n$
- Niech $t = y_1 \oplus \ldots \oplus y_n$

Nim - strategia

- x_i - ilość elementów przed wykonaniem ruchu na i-tym stosie
- y_i - ilość elementów po wykonaniu ruchu na i-tym stosie
- Niech $s = x_1 \oplus ... \oplus x_n$
- Niech $t = y_1 \oplus ... \oplus y_n$
- $t = 0 \oplus t$

Nim - strategia

- x_i - ilość elementów przed wykonaniem ruchu na i-tym stosie
- y_i - ilość elementów po wykonaniu ruchu na i-tym stosie
- Niech $s = x_1 \oplus ... \oplus x_n$
- Niech $t = y_1 \oplus ... \oplus y_n$
- $t = 0 \oplus t$
- $t = (s \oplus s) \oplus t$

Nim - strategia

- x_i - ilość elementów przed wykonaniem ruchu na i-tym stosie
- y_i - ilość elementów po wykonaniu ruchu na i-tym stosie
- Niech $s = x_1 \oplus \ldots \oplus x_n$
- Niech $t = y_1 \oplus \ldots \oplus y_n$
- $t = 0 \oplus t$
- $t = (s \oplus s) \oplus t$
- $t = s \oplus (s \oplus t)$

Nim - strategia

- x_i - ilość elementów przed wykonaniem ruchu na i-tym stosie
- y_i - ilość elementów po wykonaniu ruchu na i-tym stosie
- Niech $s = x_1 \oplus ... \oplus x_n$
- Niech $t = y_1 \oplus ... \oplus y_n$
- $t = 0 \oplus t$
- $t = (s \oplus s) \oplus t$
- $t = s \oplus (s \oplus t)$
- $t = s \oplus (x_1 \oplus ... \oplus x_n) \oplus (y_1 \oplus ... \oplus y_n)$

Nim - strategia

- x_i - ilość elementów przed wykonaniem ruchu na i-tym stosie
- y_i - ilość elementów po wykonaniu ruchu na i-tym stosie
- Niech $s = x_1 \oplus \ldots \oplus x_n$
- Niech $t = y_1 \oplus \ldots \oplus y_n$
- $t = 0 \oplus t$
- $t = (s \oplus s) \oplus t$
- $t = s \oplus (s \oplus t)$
- $t = s \oplus (x_1 \oplus \ldots \oplus x_n) \oplus (y_1 \oplus \ldots \oplus y_n)$
- $t = s \oplus (x_1 \oplus y_1) \oplus \ldots \oplus (x_n \oplus y_n)$

Nim - strategia

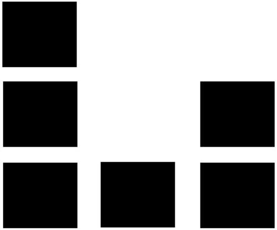
- x_i - ilość elementów przed wykonaniem ruchu na i-tym stosie
- y_i - ilość elementów po wykonaniu ruchu na i-tym stosie
- Niech $s = x_1 \oplus \ldots \oplus x_n$
- Niech $t = y_1 \oplus \ldots \oplus y_n$
- $t = 0 \oplus t$
- $t = (s \oplus s) \oplus t$
- $t = s \oplus (s \oplus t)$
- $t = s \oplus (x_1 \oplus \ldots \oplus x_n) \oplus (y_1 \oplus \ldots \oplus y_n)$
- $t = s \oplus (x_1 \oplus y_1) \oplus \ldots \oplus (x_n \oplus y_n)$
- $t = s \oplus 0 \oplus \ldots \oplus 0 \oplus (x_k \oplus y_n) \oplus 0 \oplus \ldots \oplus 0$

Nim - strategia

- x_i - ilość elementów przed wykonaniem ruchu na i-tym stosie
- y_i - ilość elementów po wykonaniu ruchu na i-tym stosie
- Niech $s = x_1 \oplus \ldots \oplus x_n$
- Niech $t = y_1 \oplus \ldots \oplus y_n$
- $t = 0 \oplus t$
- $t = (s \oplus s) \oplus t$
- $t = s \oplus (s \oplus t)$
- $t = s \oplus (x_1 \oplus \ldots \oplus x_n) \oplus (y_1 \oplus \ldots \oplus y_n)$
- $t = s \oplus (x_1 \oplus y_1) \oplus \ldots \oplus (x_n \oplus y_n)$
- $t = s \oplus 0 \oplus \ldots \oplus 0 \oplus (x_k \oplus y_n) \oplus 0 \oplus \ldots \oplus 0$
- $t = s \oplus x_k \oplus y_k$

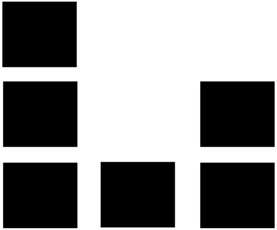
Nim - strategia

- x_i - ilość elementów przed wykonaniem ruchu na i-tym stosie
- y_i - ilość elementów po wykonaniu ruchu na i-tym stosie
- Niech $s = x_1 \oplus \ldots \oplus x_n$
- Niech $t = y_1 \oplus \ldots \oplus y_n$
- $t = s \oplus x_k \oplus y_k$



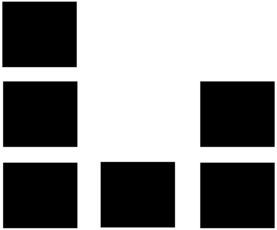
Nim - strategia

- x_i - ilość elementów przed wykonaniem ruchu na i-tym stosie
- y_i - ilość elementów po wykonaniu ruchu na i-tym stosie
- Niech $s = x_1 \oplus \ldots \oplus x_n$
- Niech $t = y_1 \oplus \ldots \oplus y_n$
- $t = s \oplus x_k \oplus y_k$
- Jeżeli $s = 0$, gracz wykonujący ruch przegrywa



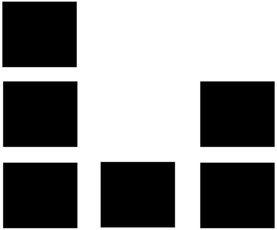
Nim - strategia

- x_i - ilość elementów przed wykonaniem ruchu na i-tym stosie
- y_i - ilość elementów po wykonaniu ruchu na i-tym stosie
- Niech $s = x_1 \oplus \ldots \oplus x_n$
- Niech $t = y_1 \oplus \ldots \oplus y_n$
- $t = s \oplus x_k \oplus y_k$
- Jeżeli $s = 0$, gracz wykonujący ruch przegrywa
- Jeżeli nie ma już możliwych ruchów, gra się skończyła i gracz już przegrał



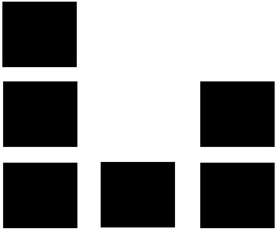
Nim - strategia

- x_i - ilość elementów przed wykonaniem ruchu na i-tym stosie
- y_i - ilość elementów po wykonaniu ruchu na i-tym stosie
- Niech $s = x_1 \oplus \ldots \oplus x_n$
- Niech $t = y_1 \oplus \ldots \oplus y_n$
- $t = s \oplus x_k \oplus y_k$
- Jeżeli $s = 0$, gracz wykonujący ruch przegrywa
- Jeżeli nie ma już możliwych ruchów, gra się skończyła i gracz już przegrał
- Każdy możliwy ruch sprawia, że $t = x_k \oplus y_k \neq 0$



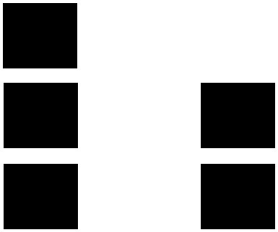
Nim - strategia

- x_i - ilość elementów przed wykonaniem ruchu na i-tym stosie
- y_i - ilość elementów po wykonaniu ruchu na i-tym stosie
- Niech $s = x_1 \oplus \ldots \oplus x_n$
- Niech $t = y_1 \oplus \ldots \oplus y_n$
- $t = s \oplus x_k \oplus y_k$
- Jeżeli $s = 0$, gracz wykonujący ruch przegrywa
- Jeżeli nie ma już możliwych ruchów, gra się skończyła i gracz już przegrał
- Każdy możliwy ruch sprawia, że $t = x_k \oplus y_k \neq 0$
- $x_k \oplus y_k \neq 0$, ponieważ gracz musi pobrać niezerową liczbę elementów



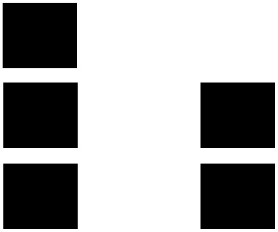
Nim - strategia

- x_i - ilość elementów przed wykonaniem ruchu na i-tym stosie
- y_i - ilość elementów po wykonaniu ruchu na i-tym stosie
- Niech $s = x_1 \oplus \ldots \oplus x_n$
- Niech $t = y_1 \oplus \ldots \oplus y_n$
- $t = s \oplus x_k \oplus y_k$



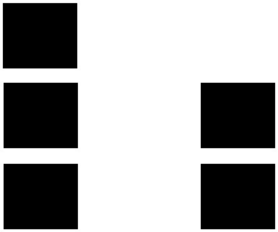
Nim - strategia

- x_i - ilość elementów przed wykonaniem ruchu na i-tym stosie
- y_i - ilość elementów po wykonaniu ruchu na i-tym stosie
- Niech $s = x_1 \oplus \ldots \oplus x_n$
- Niech $t = y_1 \oplus \ldots \oplus y_n$
- $t = s \oplus x_k \oplus y_k$
- Jeżeli $s \neq 0$, to możliwe jest wykonanie takiego ruchu, że $t = 0$, gracz wykonujący ruch wygrywa



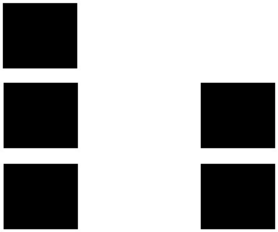
Nim - strategia

- x_i - ilość elementów przed wykonaniem ruchu na i-tym stosie
- y_i - ilość elementów po wykonaniu ruchu na i-tym stosie
- Niech $s = x_1 \oplus \ldots \oplus x_n$
- Niech $t = y_1 \oplus \ldots \oplus y_n$
- $t = s \oplus x_k \oplus y_k$
- Jeżeli $s \neq 0$, to możliwe jest wykonanie takiego ruchu, że $t = 0$, gracz wykonujący ruch wygrywa
- Żeby $t = 0$, to $y_k = s \oplus x_k$



Nim - strategia

- x_i - ilość elementów przed wykonaniem ruchu na i-tym stosie
- y_i - ilość elementów po wykonaniu ruchu na i-tym stosie
- Niech $s = x_1 \oplus \ldots \oplus x_n$
- Niech $t = y_1 \oplus \ldots \oplus y_n$
- $t = s \oplus x_k \oplus y_k$
- Jeżeli $s \neq 0$, to możliwe jest wykonanie takiego ruchu, że $t = 0$, gracz wykonujący ruch wygrywa
- Żeby $t = 0$, to $y_k = s \oplus x_k$
- $t = s \oplus x_k \oplus (s \oplus x_k) = 0$



Nim - strategia

- x_i - ilość elementów przed wykonaniem ruchu na i-tym stosie
- y_i - ilość elementów po wykonaniu ruchu na i-tym stosie
- Niech $s = x_1 \oplus \ldots \oplus x_n$
- Niech $t = y_1 \oplus \ldots \oplus y_n$

Nim - strategia

- x_i - ilość elementów przed wykonaniem ruchu na i-tym stosie
- y_i - ilość elementów po wykonaniu ruchu na i-tym stosie
- Niech $s = x_1 \oplus \ldots \oplus x_n$
- Niech $t = y_1 \oplus \ldots \oplus y_n$
- Strategia wygrywająca polega na zdejmowaniu elementów w taki sposób, aby zaszło $t = 0$

Nim - strategia

- x_i - ilość elementów przed wykonaniem ruchu na i-tym stosie
- y_i - ilość elementów po wykonaniu ruchu na i-tym stosie
- Niech $s = x_1 \oplus \ldots \oplus x_n$
- Niech $t = y_1 \oplus \ldots \oplus y_n$
- Strategia wygrywająca polega na zdejmowaniu elementów w taki sposób, aby zaszło $t = 0$
- Taki ruch jest możliwy wtedy i tylko wtedy, gdy $s \neq 0$ $\square$

Rozwiązania słabe

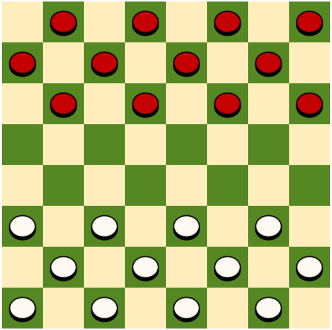
- Jest znany algorytm, który pozwala jednemu z graczy utrzymać zwycięstwo lub remis od początku gry, niezależnie od ruchów przeciwnika

Rozwiązania słabe

- Jest znany algorytm, który pozwala jednemu z graczy utrzymać zwycięstwo lub remis od początku gry, niezależnie od ruchów przeciwnika
- Dzięki podanemu algorytmowi uzyskano przynajmniej jedną *optymalną grę* oraz podano dowód, że każdy ruch jest optymalny dla gracza, który go wykonuje

Warcaby

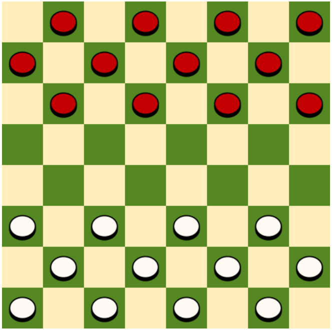
- Chinook - program komputerowy rozwijany w latach 1989 - 2007



Rysunek: Warcaby

Warcaby

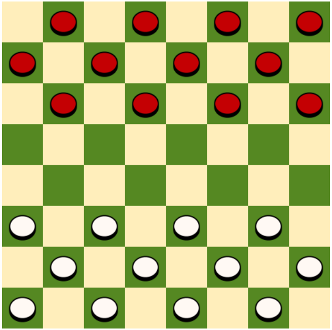
- Chinook - program komputerowy rozwijany w latach 1989 - 2007
- 19.07.2007 w magazynie *Science* twórcy opublikowali swój dowód, że w przypadku, gdy żaden z graczy nie popełni błędu partia kończy się remisem



Rysunek: Warcaby

Warcaby

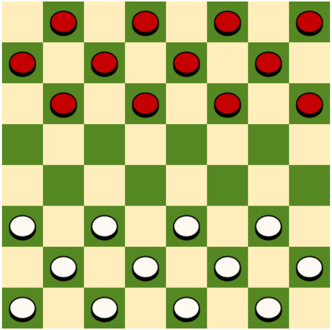
- Chinook - program komputerowy rozwijany w latach 1989 - 2007
- 19.07.2007 w magazynie *Science* twórcy opublikowali swój dowód, że w przypadku, gdy żaden z graczy nie popełni błędu partia kończy się remisem
- Baza danych końcówek o liczbie pionów ≤ 10, **237 GB**



Rysunek: Warcaby

Warcaby

- Chinook - program komputerowy rozwijany w latach 1989 - 2007
- 19.07.2007 w magazynie *Science* twórcy opublikowali swój dowód, że w przypadku, gdy żaden z graczy nie popełni błędu partia kończy się remisem
- Baza danych końcówek o liczbie pionów ≤ 10, **237 GB**
- Baza danych przechowuje wynik gry dla danej pozycji, jest wykorzystywana do analizy wstecz (retrograde)
- Drugi komponent: algorytm przeszukiwania wprzód (alphabeta)



Rysunek: Warcaby

Szachy Gardnera 5x5

- Udowodniono remis, przy optymalnej grze obu graczy [5]

Rysunek: Startowe ułożenie w szachach Gardnera 5x5

Szachy Gardnera 5x5

4.1 White moves b4

1 b4 cxb4 (**2 ♖b3 d4** ♯17♮◦, **2 ♖xb4 ♘xb4** ♯21♮◦, **2 c4 bxc4** ♯15♮◦ the b4 c4 pawn duo is too strong, **2 ♘d4 bxc3** ♯20♮◦, **2 e4 bxc3** ♯25♮◦)

- [1|1] **2 d4 bxc3** (**3 dxe5+** ♯17♮◦, **3 e4** ♯9♮◦, **3 f4** ♯10♮◦, **3 ♖b3** ♯17♮◦, **3 ♖b4** ♯12♮◦, **3 ♖xb5** ♯12♮◦, **3 ♘a4** ♯9♮◦, **3 ♕d3** ♯10♮◦, **3 ♕c4** ♯10♮◦, **3 ♕xb5** ♯7♮◦)

  3 ♗xc3 b4 (**4 e4 dxe4** ♯19♮◦, **4 f4 exf4** ♯30♮◦,**4 ♘xb4 ♘xb4** ♯26♮◦, **4 ♗xb4 ♘xb4** ♯24♮◦, **4 ♗d2 b3** ♯23♮◦, **4 ♕d2 bxc3** ♯18♮◦, **4 ♕d3 e4** ♯26♮◦, **4 ♕c4 dxc4** ♯10♮◦, **4 ♕b5 ♖xb5** ♯9♮◦)

  - [1|1.1] **4 dxe5+ ♗xe5** (**5 e4** ♯13♮◦, **5 f4** ♯15♮◦, **5 ♖b3** ♯15♮◦, **5 ♘xb4** ♯22♮◦, **5 ♘d4** ♯10♮◦, **5 ♕d2** ♯8♮◦, **5 ♕d3** ♯15♮◦, **5 ♕c4** ♯10♮◦, **5 ♕b5** ♯8♮◦, **5 ♗xb4** ♯17♮◦, **5 ♗d2** ♯11♮◦, **5 ♗d4** ♯29♮◦, **5 ♗xe5+ ♕xe5** ♯39♮◦)

    5 ♖xb4 ♖xb4

    (**6 e4** ♯9♮◦, **6 f4** ♯9♮◦, **6 ♗b2** ♯8♮◦, **6 ♗d2** ♯10♮◦, **6 ♗d4** ♯9♮◦, **6 ♗xe5+** ♯51?♮◦, **6 ♘xb4** ♯27♮◦, **6 ♘d4** ♯12♮◦, **6 ♕d2** ♯11♮◦, **6 ♕d3** ♯20♮◦, **6 ♕c4** ♯6♮◦, **6 ♕b5** ♯7♮◦)

    6 ♗xb4 ♘xb4 (**7 e4**♯7♮◦, **7 ♘d4**♯14♮◦, **7 ♕b5**♯9♮◦, **7 ♕c4**♯6♮◦, **7 ♕d3**♯6♮◦, **7 ♕d2**♯17♮◦)

    * [1|1.1.1] **7 f4 ♗c3** (**8 e4** ♯8♮◦, **8 ♘d4** ♯13♮◦, **8 ♕b5** ♯10♮◦, **8 ♕c4** ♯5♮◦, **8 ♕d3** ♯5♮◦, **8 ♕d2** ♯4♮◦, **8 ♕e2** ♯8♮◦, **8 ♔e2** ♯7♮◦) **8 ♘xb4 d4** (**9 exd4** ♯18♮◦, **9 ♘d5** ♯17♮◦, **9 ♕d3** ♯16♮◦, **9 ♕f3** ♯16♮◦, **9 ♘c6** ♯14♮◦, **9 ♔f3** ♯10♮◦, **9 e4** ♯7♮◦, **9 ♕d2** ♯5♮◦, **9 ♕b2 ♕xe3** checkmate, **9 ♕c2 ♕xe3** checkmate, **9 ♕c4 ♕xe3** checkmate, **9 ♕b5 ♕xe3** checkmate) = since on both ♘c2 and ♘d3 black exchanges Queen on e3 and the remaining position is draw △ **9 ♘c2 dxe3+ 10 ♘xe3 ♗d4 11 ♕e2 ♗xe3+ 12 ♕xe3 ♕xe3+ 13 ♔xe3**.
    * [1|1.1.2] **7 ♘xb4 d4** = the only move to avoid ...dxe3+ and the liquidation of all pawns is **8 e3 ♕b3 9 ♘d5+ ♔e6 10 exf5+ ♔xd5 11 f4 ♕e3+ 12 ♕xe3 dxe3+ 13 ♔xe3**.

  - [1|1.2] **4 ♖b3 f4** (**5 e4** ♯19♮◦, **5 ♖b2** ♯33♮◦, **5 ♖xb4** ♯27♮◦, **5 ♗xb4** ♯22♮◦, **5 ♘xb4** ♯27♮◦, **5 ♕b5** ♯10♮◦, **5 ♕c4** ♯8♮◦, **5 ♕d2** ♯20♮◦)

Rysunek: Fragment rozwiązania szachów 5x5

Rozwiązania ultra słabe

- Udowodniono, że gracz wygrywa/przegrywa/remisuje ze startowej pozycji, jeżeli wszyscy grają optymalnie

Rozwiązania ultra słabe

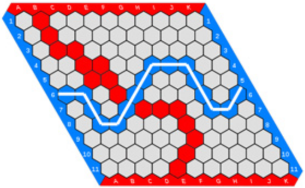
- Udowodniono, że gracz wygrywa/przegrywa/remisuje ze startowej pozycji, jeżeli wszyscy grają optymalnie
- Przeprowadzony dowód może być niekonstruktywny

Rozwiązania ultra słabe

- Udowodniono, że gracz wygrywa/przegrywa/remisuje ze startowej pozycji, jeżeli wszyscy grają optymalnie
- Przeprowadzony dowód może być niekonstruktywny
- **Nie jest wymagane określenie żadnego z ruchów optymalnej gry**

Hex

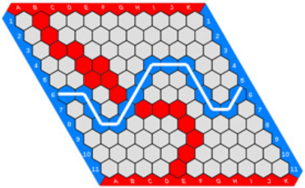
- Tradycyjnie rozgrywana na planszy w kształcie rombu 11x11



Rysunek: Rozgrywka w hex

Hex

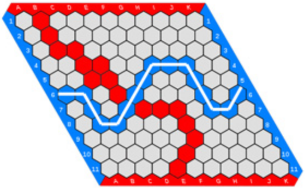
- Tradycyjnie rozgrywana na planszy w kształcie rombu 11x11
- Gracze dysponują kamieniami o odmiennych kolorach



Rysunek: Rozgrywka w hex

Hex

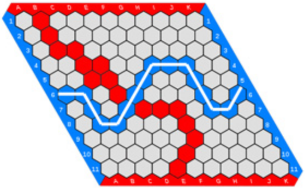
- Tradycyjnie rozgrywana na planszy w kształcie rombu 11x11
- Gracze dysponują kamieniami o odmiennych kolorach
- Gracze układają kamienie na wolnych polach



Rysunek: Rozgrywka w hex

Hex

- Tradycyjnie rozgrywana na planszy w kształcie rombu 11x11
- Gracze dysponują kamieniami o odmiennych kolorach
- Gracze układają kamienie na wolnych polach
- Wygrywa ten, który utworzy nieprzerwany ciąg łączący boki planszy własnego koloru



Rysunek: Rozgrywka w hex

Hex - Dowód Johna Nasha

Twierdzenie

Pierwszy gracz ma strategię wygrywającą

Lemat

Dodatkowy lub losowy element Twojego koloru nie może Ci zaszkodzić

Hex - Dowód Johna Nasha

Twierdzenie

Pierwszy gracz ma strategię wygrywającą

Lemat

Dodatkowy lub losowy element Twojego koloru nie może Ci zaszkodzić

Lemat

Hex nie może zakończyć się remisem

Hex - Dowód Johna Nasha

Twierdzenie

Pierwszy gracz ma strategię wygrywającą

Lemat

Dodatkowy lub losowy element Twojego koloru nie może Ci zaszkodzić

Lemat

Hex nie może zakończyć się remisem

Dowód.

Mamy dwóch graczy A i B, A zaczyna. Załóżmy, że B ma strategię wygrywającą. A może zagrać w losowe miejsce. Teraz A jest efektywnym drugim graczem i może grać strategią wygrywającą [6]. □

Hex - Dowód Johna Nasha

Przedstawiony dowód korzystający z kradzieży strategii, może być zastosowany do każdej symetrycznej gry, w której posiadanie dodatkowego elementu na planszy (lub dodatkowego ruchu) nie szkodzi danemu graczowi.

Go

- Tylko wersja gry bez komi

Rysunek: Goban

Go

- Tylko wersja gry bez komi
- Komi to wyrównanie punktowe dla drugiego gracza

Rysunek: Goban

Go

- Tylko wersja gry bez komi
- Komi to wyrównanie punktowe dla drugiego gracza
- Dowód korzysta z kradzieży strategii

Rysunek: Goban

Go

- Tylko wersja gry bez komi
- Komi to wyrównanie punktowe dla drugiego gracza
- Dowód korzysta z kradzieży strategii
- W Go można pasować

Rysunek: Goban

Go

- Tylko wersja gry bez komi
- Komi to wyrównanie punktowe dla drugiego gracza
- Dowód korzysta z kradzieży strategii
- W Go można pasować
- Jeżeli białe mają strategię wygrywającą, czarne mogą ją ukraść pasując już na początku gry

Rysunek: Goban

Go

- Tylko wersja gry bez komi
- Komi to wyrównanie punktowe dla drugiego gracza
- Dowód korzysta z kradzieży strategii
- W Go można pasować
- Jeżeli białe mają strategię wygrywającą, czarne mogą ją ukraść pasując już na początku gry
- W rezultacie czarne mogą wygrać lub zremisować przy optymalnej grze

Rysunek: Goban

Szachy

- 10^{120} możliwych wariantów

Rysunek: Szachy klasyczne

Szachy

- 10^{120} możliwych wariantów
- Rozwiązane częściowo

Rysunek: Szachy klasyczne

Szachy

- 10^{120} możliwych wariantów
- Rozwiązane częściowo
- Znane są rozwiązania mocne dla wszystkich końcówek liczących od 3 do 7 bierek (wliczając obu króli)

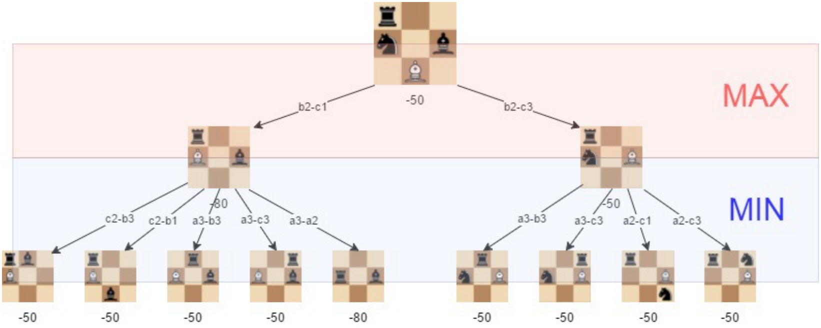
Rysunek: Szachy klasyczne

Szachy

- 10^{120} możliwych wariantów
- Rozwiązane częściowo
- Znane są rozwiązania mocne dla wszystkich końcówek liczących od 3 do 7 bierek (wliczając obu króli)
- Osiągnięto to dzięki bazie danych końcówek i analizie retrograde

Rysunek: Szachy klasyczne

Minimiax



Rysunek: Przykładowy minimax

Minimax

```
Function minimax(node, depth, maximizingPlayer):
    if depth = 0 or node is a terminal node then
        return heuristic value of node
    end
    if maximizingPlayer then
        value ← −∞
        foreach child in node do
            value ← max(value, minimax(child, depth −1, False))
        end
        return value
    else
        value ← ∞
        foreach child in node do
            value ← min(value, minimax(child, depth −1, True))
        end
        return value
    end
end
```

Heurystyki

Definicja

Heurystyka to metoda znajdowania rozwiązań, dla której nie ma gwarancji znalezienia rozwiązania optymalnego, a często nawet prawidłowego. Rozwiązań tych używa się np. wtedy, gdy pełny algorytm jest z przyczyn technicznych zbyt kosztowny lub gdy jest nieznany.

Negamax

- Pewne uproszczenie klasycznego minimaxa

Negamax

- Pewne uproszczenie klasycznego minimaxa
- Wykorzystuje własność gier o zerowej sumie

Negamax

- Pewne uproszczenie klasycznego minimaxa
- Wykorzystuje własność gier o zerowej sumie
- $\max(a, b) = -\min(-a, -b)$

Negamax

- Pewne uproszczenie klasycznego minimaxa
- Wykorzystuje własność gier o zerowej sumie
- $\max(a, b) = -\min(-a, -b)$

Function `negamax`(*node, depth, color*)**:**
    if *depth* $= 0$ *or node is a terminal node* **then**
        return color $\cdot$ heuristic value of node
    end
    value $\leftarrow -\infty$
    foreach *child in node* **do**
        value $\leftarrow$ max(value, minimax(child, depth -1, $-$color))
    end
    return value
end

Funkcja oceniająca

- Ocena stanu węzła w drzewie gry

Funkcja oceniająca

- Ocena stanu węzła w drzewie gry
- Ogólne podejście: kombinacja liniowa ważonych czynników

Funkcja oceniająca

- Ocena stanu węzła w drzewie gry
- Ogólne podejście: kombinacja liniowa ważonych czynników
- Nie ma analitycznego ani teoretycznego modelu dla nierozwiązanych gier

Funkcja oceniająca

- Ocena stanu węzła w drzewie gry
- Ogólne podejście: kombinacja liniowa ważonych czynników
- Nie ma analitycznego ani teoretycznego modelu dla nierozwiązanych gier
- Decyduje podejście empiryczne

Funkcja oceniająca

- Ocena stanu węzła w drzewie gry
- Ogólne podejście: kombinacja liniowa ważonych czynników
- Nie ma analitycznego ani teoretycznego modelu dla nierozwiązanych gier
- Decyduje podejście empiryczne
- Przykładowa funkcja dla szachów:

Funkcja oceniająca

- Ocena stanu węzła w drzewie gry
- Ogólne podejście: kombinacja liniowa ważonych czynników
- Nie ma analitycznego ani teoretycznego modelu dla nierozwiązanych gier
- Decyduje podejście empiryczne
- Przykładowa funkcja dla szachów:

$f(x) = c_1 \cdot$ wartość figur

Funkcja oceniająca

- Ocena stanu węzła w drzewie gry
- Ogólne podejście: kombinacja liniowa ważonych czynników
- Nie ma analitycznego ani teoretycznego modelu dla nierozwiązanych gier
- Decyduje podejście empiryczne
- Przykładowa funkcja dla szachów:

$f(x) = c_1 \cdot$ wartość figur $+ c_2 \cdot$ bezpieczeństwo króla

Funkcja oceniająca

- Ocena stanu węzła w drzewie gry
- Ogólne podejście: kombinacja liniowa ważonych czynników
- Nie ma analitycznego ani teoretycznego modelu dla nierozwiązanych gier
- Decyduje podejście empiryczne
- Przykładowa funkcja dla szachów:

$$f(x) = c_1 \cdot \text{ wartość figur } + c_2 \cdot \text{ bezpieczeństwo króla } + c_3 \cdot \text{ kontrola}$$

Funkcja oceniająca

- Ocena stanu węzła w drzewie gry
- Ogólne podejście: kombinacja liniowa ważonych czynników
- Nie ma analitycznego ani teoretycznego modelu dla nierozwiązanych gier
- Decyduje podejście empiryczne
- Przykładowa funkcja dla szachów:

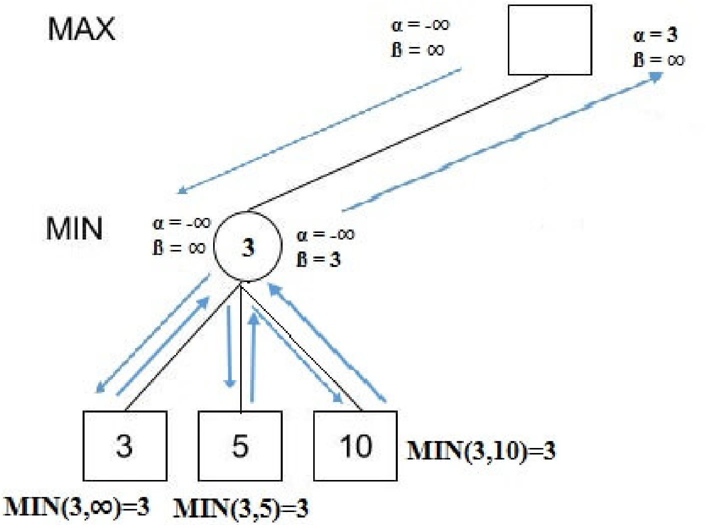
$$f(x) = c_1 \cdot \text{ wartość figur } + c_2 \cdot \text{ bezpieczeństwo króla } + c_3 \cdot \text{ kontrola } + \ldots$$

Funkcja oceniająca

- Ocena stanu węzła w drzewie gry
- Ogólne podejście: kombinacja liniowa ważonych czynników
- Nie ma analitycznego ani teoretycznego modelu dla nierozwiązanych gier
- Decyduje podejście empiryczne
- Przykładowa funkcja dla szachów:

$f(x) = c_1 \cdot$ wartość figur $+ c_2 \cdot$ bezpieczeństwo króla $+ c_3 \cdot$ kontrola $+ \ldots$
Współczynniki c_i są pewnymi wagami i mogą się zmieniać zależnie od fazy gry.

Alfa-Beta Pruning



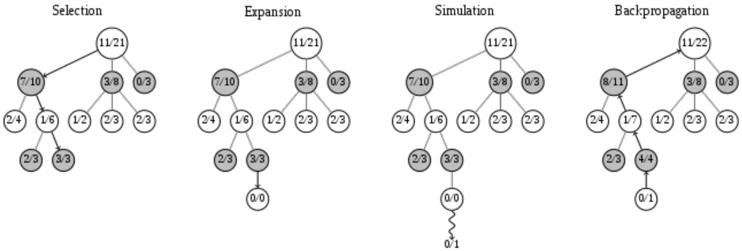
Alfa-Beta Pruning

Function `alphabeta`(*node, depth*, α, β, *maximizingPlayer*):
- **if** *depth* $= 0$ *or node is a terminal node* **then**
  - **return** heuristic value of node
- **end**
- **if** *maximizingPlayer* **then**
  - value $\leftarrow -\infty$
  - **foreach** *child in node* **do**
    - value $\leftarrow$ max(value, alphabeta(child, depth -1, α, β, False))
    - $\alpha \leftarrow$ max(α, value)
    - **if** $\alpha \geq \beta$ **then** break
  - **end**
  - **return** value
- **else**
  - value $\leftarrow +\infty$
  - **foreach** *child in node* **do**
    - value $\leftarrow$ min(value, alphabeta(child, depth -1, α, β, True))
    - $\beta \leftarrow$ min(β, value)
    - **if** $\alpha \geq \beta$ **then** break
  - **end**
  - **return** value
- **end**

end

Monte Carlo tree search

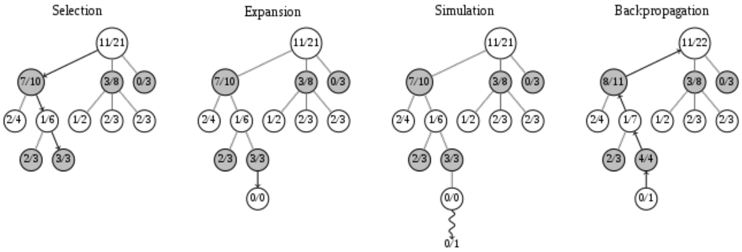
- Selekcja - Wybór liścia w drzewie gry



Rysunek: Monte Carlo tree search

Monte Carlo tree search

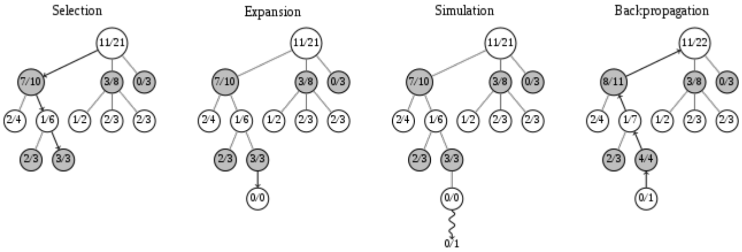
- Selekcja - Wybór liścia w drzewie gry
- Ekspansja - Utworzenie węzła potomnego - jeżeli liść nie kończy gry



Rysunek: Monte Carlo tree search

Monte Carlo tree search

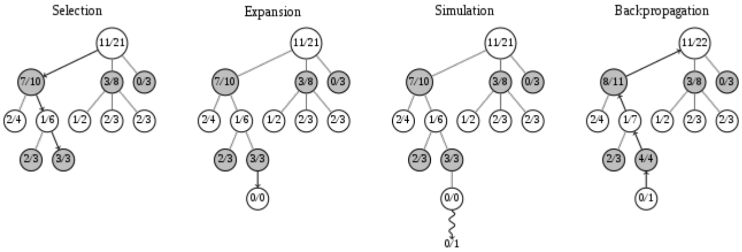
- Selekcja - Wybór liścia w drzewie gry
- Ekspansja - Utworzenie węzła potomnego - jeżeli liść nie kończy gry
- Symulacja - rozegranie losowej gry z wybranego węzła



Rysunek: Monte Carlo tree search

Monte Carlo tree search

- Selekcja - Wybór liścia w drzewie gry
- Ekspansja - Utworzenie węzła potomnego - jeżeli liść nie kończy gry
- Symulacja - rozegranie losowej gry z wybranego węzła
- Propagacja wsteczna - aktualizacja informacji w węzłach na podstawie wyniku rozegranej gry



Rysunek: Monte Carlo tree search

Null move

- Ruch zerowy, czyli pasowanie

Null move

- Ruch zerowy, czyli pasowanie
- W szachach jest niedozwolony

Null move

- Ruch zerowy, czyli pasowanie
- W szachach jest niedozwolony
- Założenie: zrzeczenie się ruchu jest gorsze niż wykonanie dowolnego legalnego ruchu

Null move

- Ruch zerowy, czyli pasowanie
- W szachach jest niedozwolony
- Założenie: zrzeczenie się ruchu jest gorsze niż wykonanie dowolnego legalnego ruchu
- Wykorzystywane w poszukiwaniu zagrożeń

Null move

- Ruch zerowy, czyli pasowanie
- W szachach jest niedozwolony
- Założenie: zrzeczenie się ruchu jest gorsze niż wykonanie dowolnego legalnego ruchu
- Wykorzystywane w poszukiwaniu zagrożeń
- Może powodować błędy np. będąc szachu

Killer heuristic

- Autorka: Barbara Liskov

Killer heuristic

- Autorka: Barbara Liskov
- Tylko niewielka liczba ruchów diametralnie zmienia sytuację

Killer heuristic

- Autorka: Barbara Liskov
- Tylko niewielka liczba ruchów diametralnie zmienia sytuację
- Założenie: jeżeli ruch tworzy odcięcie *(zabójczy ruch)*, to prawdopodobnie wytworzy je też w podobnej sytuacji

Killer heuristic

- Autorka: Barbara Liskov
- Tylko niewielka liczba ruchów diametralnie zmienia sytuację
- Założenie: jeżeli ruch tworzy odcięcie *(zabójczy ruch)*, to prawdopodobnie wytworzy je też w podobnej sytuacji
- Aby przyśpieszyć odcięcie algorytm rozpoczyna szukanie od zapisanych *zabójczych ruchów*

Literatura

- Anurag Bhatt, Pratul Varshney, Kalyanmoy Deb, Indian Institute of Technology Kanpur, *Evolution of No-loss Strategies for the Game of Tic-Tac-Toe*, `https://www.iitk.ac.in/kangal/papers/k2007002.pdf` (data dostępu: 28.10.2019)
- Martin J. Osborne, Ariel Rubinstein, The MIT Press, *A course in Game Theory*, 1994
- Andrey Kurenkov, *A 'Brief' History of Game AI Up To AlphaGo*, `https://www.andreykurenkov.com/writing/ai/a-brief-history-of-game-ai/` (data dostępu: 26.10.2019)
- Wikipedia, *Solved game*, `https://en.wikipedia.org/wiki/Solved_game` (data dostępu: 26.10.2019)
- Mehdi Mhalla, Frédéric Prost, *Gardner's Minichess Variant is solved*, `https://arxiv.org/pdf/1307.7118.pdf` (data dostępu: 27.10.2019)
- *Hex*, SP.269, Spring 2011 `http://web.mit.edu/sp.268/www/hex-notes.pdf` (data dostępu 03.11.2019)