Agenda

- General Game Playing – z czym to się je?
- Najważniejsi gracze:
  - ClunePlayer
  - FluxPlayer
  - CADIAPlayer
  - nnrg.hazel
- Podsumowanie

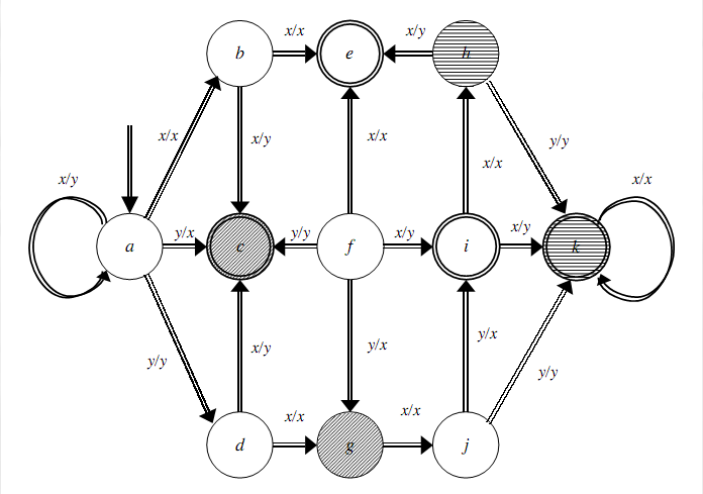
General Game Playing?

- Cel:
  - stworzenia systemu umiejącego grać/nauczyć się grać we „wszystkie” gry
- Turniej w ramach AAAI National Conference
  - corocznie od 2005 roku
- Przebieg rozgrywki w ramach turnieju:
  - prezentacja zasad i czas na ich analizę (np. 90s)
  - rozgrywka z ograniczeniem czasowym na wykonanie pojedynczego ruchu

Model gry

- Model gry:
  - skończona
  - skończona liczba graczy
  - synchroniczna
    - wszyscy gracze ruszają się równocześnie (ale dopuszczalne ruchy typu *noop*)
  - skończona liczba legalnych ruchów w każdym ze skończonej liczby stanów
  - zmiany stanu tylko w wyniku ruchów
  - ➔ maszyna stanowa

Model gry



M. Genesereth, N. Love, and B. Pell. General Game Playing: Overview of the AAAI Competition. AI Magazine, 26(2):62-72, 2005.

Game Definition Language

- Opis gry jako maszyny stanowej: zbyt rozwlekły
- Krótszy sposób opisu: Game Definition Language (GDL)
  - opis gry za pomocą fomuł logicznych
  - język bazujący na zmodyfikowanym Datalogu
    - a Datalog to podzbiór Prologa
  - podstawowe relacje:
    - *role*, *true*, *init*, *next*, *legal*, *does*, *goal*, *terminal*

GDL: Tic-Tac-Toe

```
 1. (role xplayer)
 2. (role oplayer)
 3. (init (cell 1 1 b))
 4. (init (cell 1 2 b))
 ...
 5. (init (cell 3 3 b))
 6. (init (control xplayer))
 7. (<= (next (cell ?m ?n x))
 8.     (does xplayer (mark ?m ?n))
 9.     (true (cell ?m ?n b)))
10. (<= (next (control xplayer))
11.     (true (control oplayer)))
 ...
13. (<= (row ?m ?x)
14.     (true (cell ?m 1 ?x))
15.     (true (cell ?m 2 ?x))
16.     (true (cell ?m 3 ?x)))
17. (<= (line ?x)
18.     (row ?m ?x))
 ...
19. (<= (legal ?w (mark ?x ?y))
20.     (true (cell ?x ?y b))
21.     (true (control ?w)))
22. (<= (legal xplayer noop)
23.     (true (control oplayer)))
 ...
24. (<= (goal xplayer 0)   (line o))
25. (<= (goal oplayer 100) (line o))
 ...
26. (<= terminal (line x))
```

J. Reisinger, E. Bahceci, I. Karpov, and R. Miikkulainen. Coevolving strategies for general game playing. In Proceedings of the IEEE Symposium on Computational Intelligence and Games (CIG 2007), 320-327, Honolulu, Hawaii, 2007. IEEE Press.

ClunePlayer

- Zwycięzca turnieju GGP w 2005
- Pomysł:
  - stworzenie heurystycznej funkcji ewaluacyjnej jako funkcji działającej na uproszczonym modelu gry
  - uproszczony model bazujący na 3 koncepcjach:
    - wypłata (*payoff*)
    - kontrola/mobilność (*control*)
    - terminalność/czas (*termination*)

Generowanie komponentów funkcji ewaluacyjnej

- **Krok 1** – identyfikacja wyrażeń logicznych:
  - wydobycie wyrażeń pojawiających się w opisie reguł
  - generowanie nowych wyrażeń:
    - podstawienie wszystkich możliwych stałych w miejsce zmiennych
    - identyfikacja wyróżnionych stałych wśród zidentyfikowanych

Generowanie komponentów funkcji ewaluacyjnej

- **Krok 2** – generowanie komponentów z wyrażeń:
  - 3 interpretacje wyrażeń:
    - *liczność*
      - poprzez znalezienie liczby różnych rozwiązań wyrażenia
    - *odległość*
      - wyznaczana z wykorzystaniem grafu zależności między symbolami w opisie gry
    - *częściowe rozwiązanie*
      - stosowane dla wyrażeń złożonych, które częściowo pokrywają się z wyrażeniami celu
  - stabilność komponentów:
    - stosunek zmienności całkowitej do zmienności między kolejnymi stanami (obliczane na bazie symulacji)

Model abstrakcyjny

Table 1: Abstract Model Parameters

Parameter	Meaning
$P : \Omega \rightarrow [0, 100]$	Approximates payoff function.
$C : \Omega \rightarrow [-1, 1]$	Degree of control player has.
$T : \Omega \rightarrow [0, 1]$	Probability that state is terminal.
$S_P : (1, \infty)$	Stability of P.
$S_C : (1, \infty)$	Stability of C.

- Komponenty nie wpływające na funkcję celu są usuwane (na podstawie analizy zależności)
- Dla pozostałych wyznaczany jest kierunek korelacji z cechami modelu abstrakcyjnego (na bazie symulacji)
  - wartości cech dla stanów wyznaczane w sposób przybliżony, np. wypłata jako potencjalna wartość funkcji celu, gdyby dany stan był terminalny
- Komponenty w funkcji ewaluacyjnej ważone są stabilnością

Przykładowe rezultaty

- Reversi
  - tylko wypłata zidentyfikowana:

Payoff =
3.361 (lower right corner: # white - # black)
+2.015 (upper right corner: # white - # black)
+1.379 (upper edge: # white - # black)
+1.221 (lower left corner: # white - # black)
+1.114 (lower edge: # white - # black)
+0.789 (upper left corner: # white - # black)
+0.646 (right edge: # white - # black)
+0.592 (left edge: # white - # black)
+50.000

Przykładowe rezultaty

- Chess
  - tylko funkcja kontroli zidentyfikowana:

Control =
0.097 (# white queens - # black queens)
+0.092 (# white rooks - # black rooks)
+0.042 (# white bishops - # black bishops)
+0.039 (# white knights - # black knights)
+0.022 (# empty squares: rank 1 - rank 8)
+0.004 (# white pawns - # black pawns)

FluxPlayer

FluxPlayer

- Zwycięzca turnieju GGP w 2005
- Pomysł:
  - stworzenie heurystycznej funkcji ewaluacyjnej na podstawie oceny stopnia spełnienia celów gry
    - wykorzystanie elementów logiki rozmytej

Algorytm przeszukiwania

- Algorytm przeszukiwania:
  - depth-first z niejednorodną głębokością przeszukiwania
  - korzystający z iteracyjnego pogłębiania
  - usprawnienia:
    - tablice transpozycji
    - history heuristic
    - w przypadku gier 2-sobowych: alfa-beta obcięcia
  - dla gier symultanicznych:
    - założenie paranoiczne: modelowany gracz decyduje pierwszy

Funkcja ewaluacyjna

- Idea:
  - wyznaczenie stopnia spełnienia celu
  - unikanie stanów terminalnych przy niskich (ujemnych) wartościach funkcji celu
  - wykorzystanie logiki rozmytej
- Ale:
  - każda t-norma zwróci 0 dla koniunkcji, w której jedno z wyrażeń jest fałszywe
- Rozwiązanie:
  - stosowanie wartości p i 1-p zamiast 0 i 1
    - ale wtedy złożona koniunkcja uzyskuje zbyt małe wartości
    - rozwiązanie:
      - zastąpienie t-normy wyrażeniem z dolnym ograniczeniem

CADIAPlayer

CADIAPlayer

- Zwycięzca turnieju GGP w 2007 i 2008
- Koncepcja:
  - rozwiązanie oparte na symulacjach Monte-Carlo
    - konkretniej: UCT
      - (Upper Confidence-bounds applied to Trees)
  - logika gry reprezentowana przez automatycznie generowany podprogram w Prologu

Algorytm przeszukiwania

- Dal gier jednoosobowych:
  - Memory Enhanced IDA*
    - (Iterative Deepening A*)
    - aplikowane już na etapie analizy reguł
    - jeśli nie znajdzie na tym etapie przynajmniej częściowego rozwiązania, na etapie gry będzie wykorzystywany znów UCT
- Dla pozostałych gier:
  - UCT

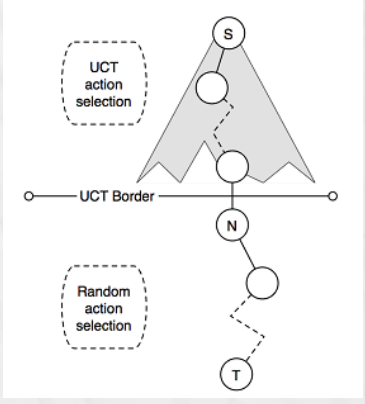
Algorytm przeszukiwania

- Sposób wybierania kolejnego ruchu podczas symulacji (UCT):

$$a^* = argmax_{a \in A(s)} \left\{ Q(s,a) + C\sqrt{\frac{\ln N(s)}{N(s,a)}} \right\}$$

  - Q(s,a) - średni dotychczasowy wynik pary stan-ruch
  - N – liczba wizyt w danym stanie/wykonań danej akcji
  - akcje nigdy nie wykonywane wybierane są w pierwszej kolejności

Algorytm przeszukiwania



Algorytm przeszukiwania

- Ograniczanie zużycia pamięci:
  - usuwanie danych o węzłach powyżej aktualnego po każdym (realnie) wykonanym ruchu
  - dodawanie tylko jednego nowego węzła per symulacja
- Modelowanie przeciwnika
  - każdy przeciwnik ma przydzielony swój własny model
- Dodatkowe usprawnienie:
  - *history heuristic*
    - modyfikuje prawdopodobieństwo wybrania niewypróbowanej akcji (zarówno w fazie UCT, jak i MC)

nnrg.hazel

- 6. miejsce (z 12) w turnieju GGP w 2006 roku
- Idea:
  - funkcja ewaluacyjna reprezentowana przez sieć neuronową
  - nauka z wykorzystaniem ko-ewolucji

nnrg.hazel

- Najważniejsze wyróżniki rozwiązania:
  - NEAT
    - (NeuroEvolution of Augmenting Topologies)
    - jednoczesna ewolucja wag i architektury sieci neuronowej
    - rozpoczęcie procesu od bardzo prostej struktury sieci
  - specjacja
  - losowe mapowanie opisów stanu na wejście sieci

Podsumowanie i wnioski

- Rozwiązania inteligencji obliczeniowej jak na razie się nie sprawdzają
  - prawdopodobna przyczyna: ograniczony czas na naukę
- Rozwiązania dominujące:
  - podejście logiczne
    - analiza i przekształcenia modelu gry
  - podejście symulacyjne
    - *de facto* siłowe (zrównoleglane na 8-12 CPU)

Podsumowanie i wnioski

- Cel pracy:
  - start w GGP?
  - wykorzystanie jedynie GDL?
    - zagadnienia:
      - modelowanie przeciwnika?
      - przenoszenie wiedzy między grami?
      - wysoki poziom gry?
- Kwestie techniczne:
  - wydajna reprezentacja gry?

Najważniejsza bibliografia

- M. Genesereth, N. Love, and B. Pell. General Game Playing: Overview of the AAAI Competition. AI Magazine, 26(2):62-72, 2005
- J. Reisinger, E. Bahceci, I. Karpov, and R. Miikkulainen. Coevolving strategies for general game playing. In Proceedings of the IEEE Symposium on Computational Intelligence and Games (CIG 2007), 320-327, Honolulu, Hawaii, 2007. IEEE Press
- Hilmar Finnsson, Yngvi Björnsson: Simulation-Based Approach to General Game Playing. AAAI 2008: 259-264
- Stephan Schiffel, Michael Thielscher: Fluxplayer: A Successful General Game Player. AAAI 2007: 1191-1196
- James Clune: Heuristic Evaluation Functions for General Game Playing. AAAI 2007: 1134-1139