



# Nauka wzorców w Go

Stanisław Kaźmierczak

# Agenda

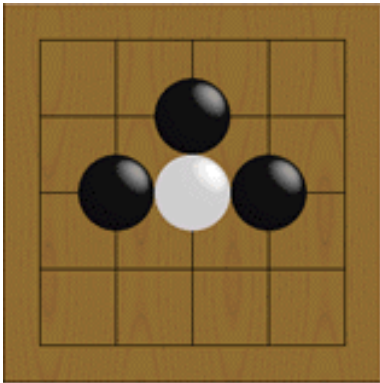
- 
- Zasady gry
  - Wprowadzenie
  - Oznaczanie *łączności*
  - Poziomy *łączności*
  - Oznaczanie rodzaju ruchu
  - Pozostałe oznaczenia
  - Ekstrakcja cech:
    - cechy łańcucha (*Chain Features*)
    - *Common Fate Graph* (CFG)
    - *Relative Subgraph Features* (RSF)
    - *Relative Subgraph Path Features* (RSP)
  - Klasyfikator



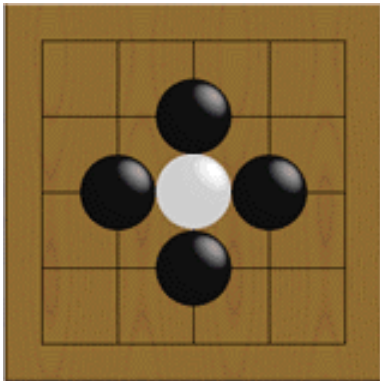
# Zasady gry

- Dwaj gracze (czarny i biały) wykonują ruchy kładąc naprzemiennie po jednym kamieniu na planszy
- Kamienie muszą być kładzione na przecięciu pionowych i poziomych linii
- Gdy kamień jest położony, nie można go przemieścić, jedynie w określonych warunkach – usunąć
- Celem w Go jest utworzenie większego niż przeciwnik "terytorium"
- Kamienie, które są otoczone, zostają usuwane z planszy i zabierane przez przeciwnika jako jeńcy
- Każdy jeniec będzie miał wartość jednego punktu

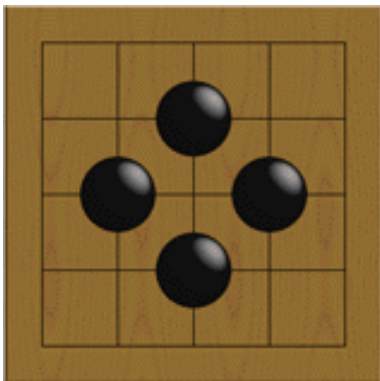
# Zdobywanie kamieni



Po jeszcze jednym ruchu biały będzie całkowicie otoczony i zostanie usunięty.

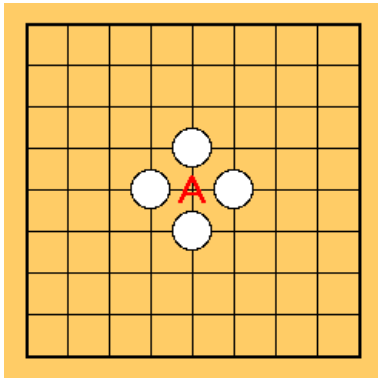


Biały kamień jest teraz otoczony i dlatego...

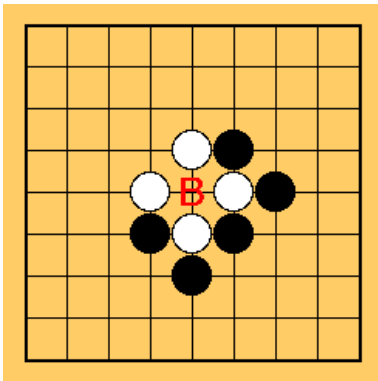


... zostaje usunięty z planszy.

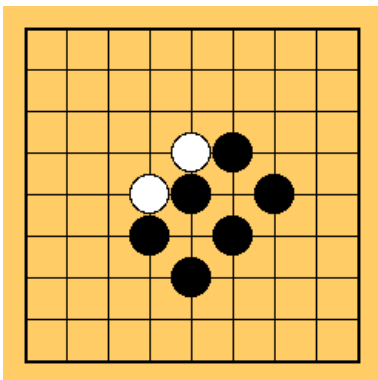
# Ruchy niedozwolone



Czarny nie może wykonać ruchu w A.  
Gdy czarny tam zagra, jego kamień będzie całkowicie otoczony.



Czarny może jednak zagrać w B.  
Wraz z położeniem czarnego kamienia w B, dwa sąsiednie białe kamienie będą otoczone i usunięte.





# Wprowadzenie

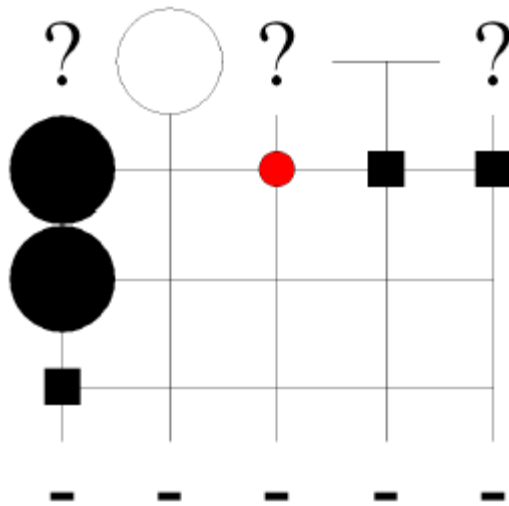
- Historia *komputerowego Go* liczy ponad 25 lat
- Wiele kierunków poszukiwań, które nie przyniosły zadowalających korzyści
- Jedną z przyczyn jest brak dostępnej cyfrowej wiedzy eksperckiej (w konsekwencji szansę na powodzenie mają jedynie mistrzowie Go z umiejętnościami programowania)
- Różnice względem szachów:
  - *branching factor*
  - efekt ruchu często widoczny po kiludziesięciu ruchach
  - efekty nie są lokalne
  - trudna do zdefiniowania funkcja ewaluacyjna
  - niemożliwe napisanie sensownej książki otwarć
- ... a jednak w 2012 Zen19D osiągnął poziom 6 dan



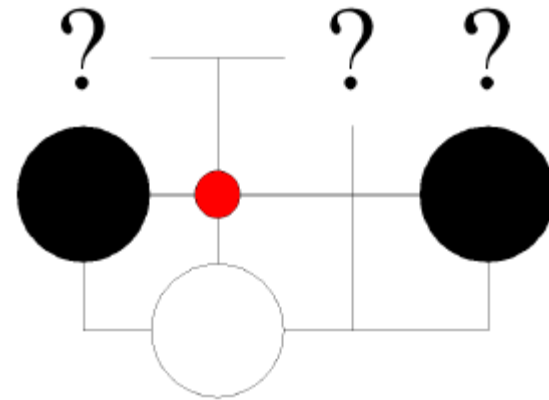
# Podejście wysokopoziomowe

- Np. wpływ, obrona terytorium
- Czyni proces wykonywania konkretnego ruchu dokładniejszy
- Pojęcie abstrakcyjne, posiada odniesienie do ludzkiej intuicji
- Źródła dostępnych danych eksperckich:
  - zapisy partii zawodowych graczy
  - zbiór problemów *life and death*
  - ręcznie tworzona baza abstrakcyjnych wzorców (z rodzajem ruchu potrzebnym do osiągnięcia zamierzonego rezultatu), np. taka jak w GnuGo
  - *Joseki* – optymalne sekwencje ruchów, na ogół w rejonie rogu planszy

# Przykładowe wzorce



Wzorzec kategorii  
*edge block/expand*



Wzorzec kategorii  
*cut/connection*

- Czerwone kropki – kolejny ruch czarnych
- Czarny kwadrat – pole jest puste albo zawiera czarny kamień
- Kreski – odniesienie wzorca do krawędzi
- Znak zapytania – cokolwiek, wewnątrz planszy

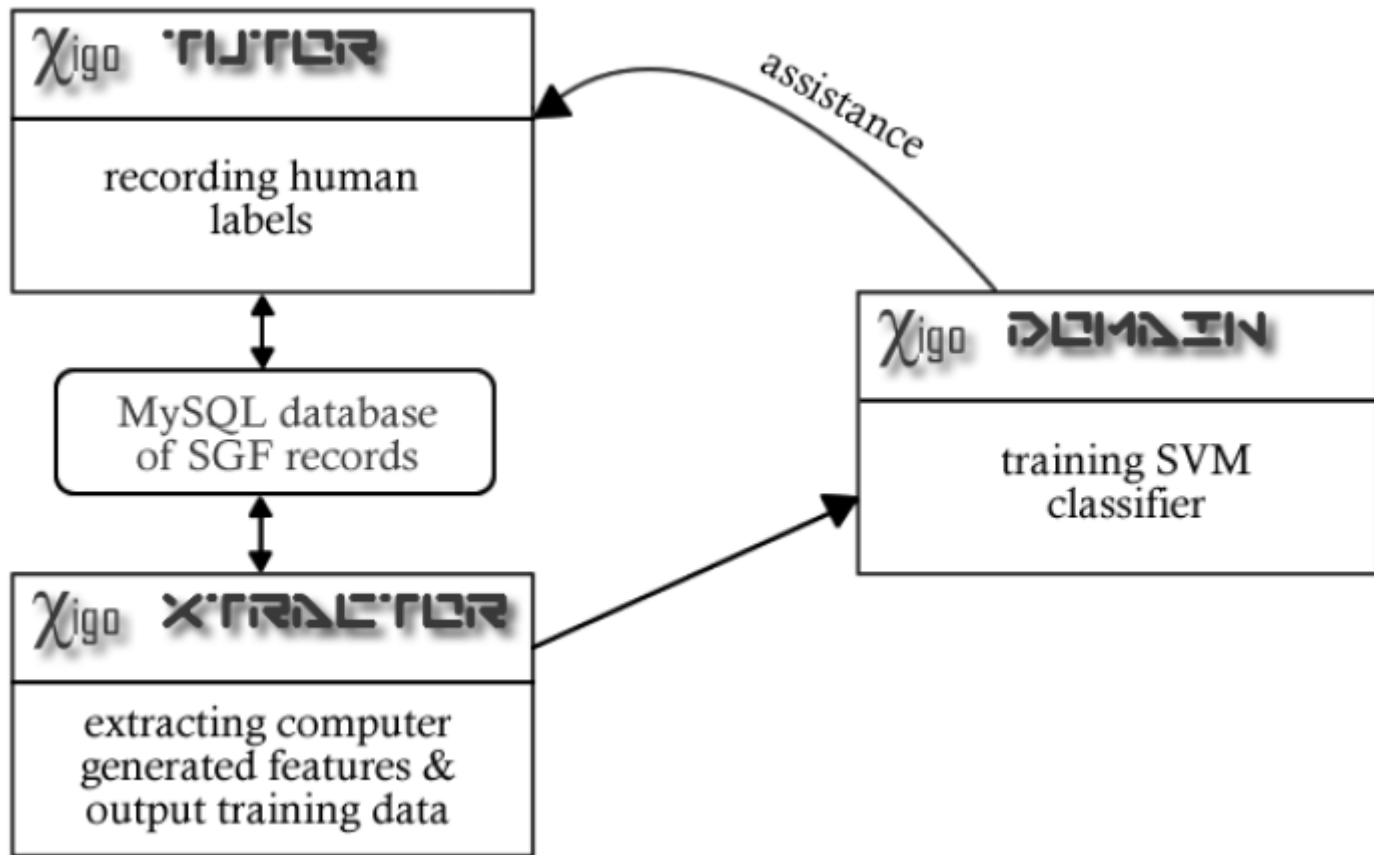




# Idea uczenia

- Dość ciężko jest stworzyć skuteczną metodą nauki Go
- Dużo łatwiej jest przygotować lepszy materiał do uczenia
- Należy dodawać dodatkowe informacje do zapisu partii, np. ocena pozycji
  - *czarna grupa w lewym górnym rogu jest bezpieczna, a ponadto wywiera nacisk na białą grupę po prawo*
  - *terytorium białych nie wygląda jeszcze solidnie, czarne powinny szukać sposobu wykorzystania słabości białych*
- Bez powyższych informacji wiedza o ruchu jest bezużyteczna
- Wiedza ekspercka powinna być możliwa do zapisania w sformalizowany sposób (XGF, rozszerzony format SGF bazujący na XMLu)
- Zależności opisane są w logice pierwszego rzędu

# XiGo





# Ekstrakcja cech i klasyfikacja

Kluczowe aspekty:

- Rodzaj i sposób zapisu wiedzy eksperckiej w programie
- Reprezentacja; jak cechy mogą być ekstrahowane przy danej reprezentacji
- Metoda uczenia

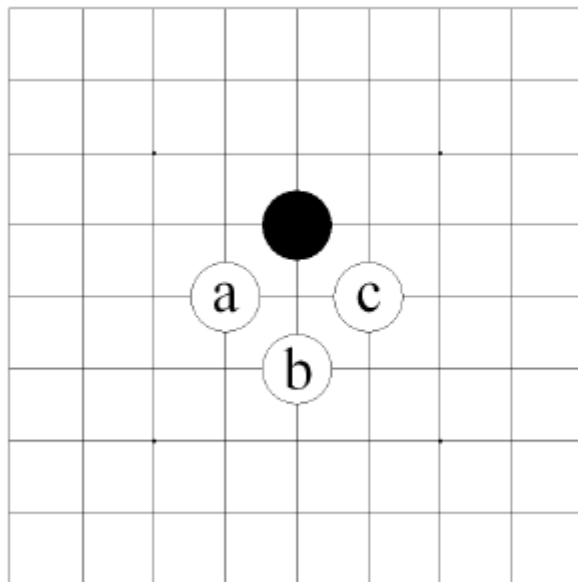


# Oznaczanie łączności

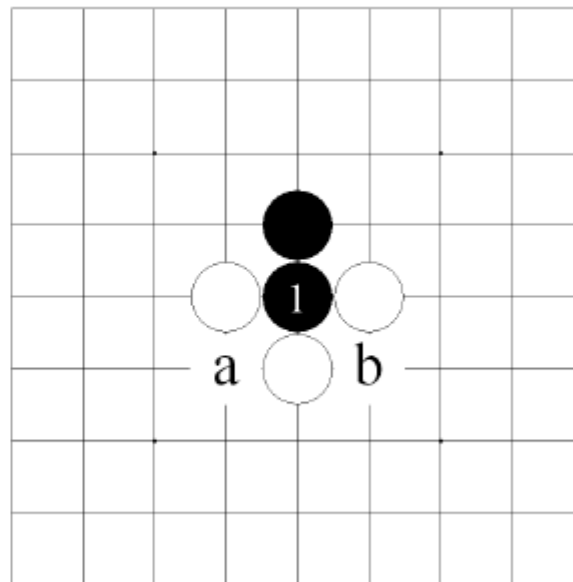
- *First cut, then think*
- Rozdzielenie dwóch dużych grup często decyduje o zwycięstwie
- Wiedza na temat miejsca, gdzie może wystąpić rozcięcie pozwala zastosować środki prewencyjne
- Znajdowanie cięcia przez profesjonalnego gracza często składa się z dwóch etapów:
  - rozpoznanie kształtu cięcia
  - zbadanie sytuacji i potwierdzenie wyniku (analiza)
- Im bardziej doświadczony gracz, tym trafniejszy wynik etapu rozpoznania (intuicja oparta na doświadczeniu)
- Nauka łączności w oparciu o rozpoznawanie wzorców jest dobrym obszarem do zastosowania AI → jest dużo łatwiejsza niż znalezienie najlepszego następnego ruchu

# Oznaczanie łączności

- Łączność nie jest przechodnia!
- Stwierdzenie łączności przy użyciu wnioskowania logicznego staje się trudnym zadaniem



*a* i *b* są połączone,  
*b* i *c* są połączone



Po ruchu czarnych w 1,  
białe nie mogą połączyć  
w obu punktach *a* i *b*

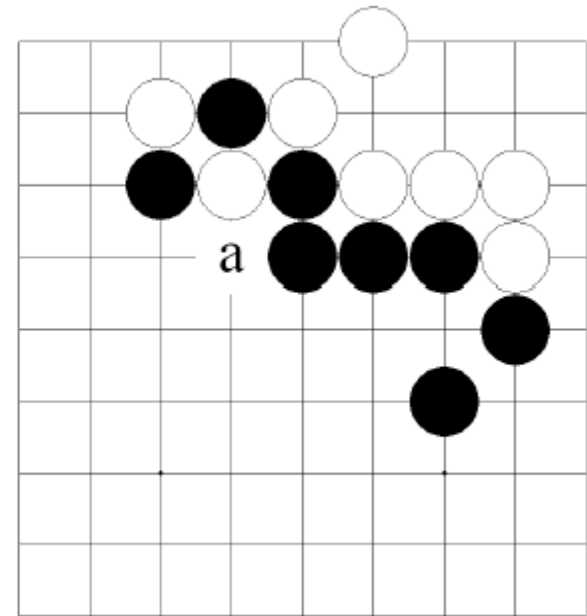
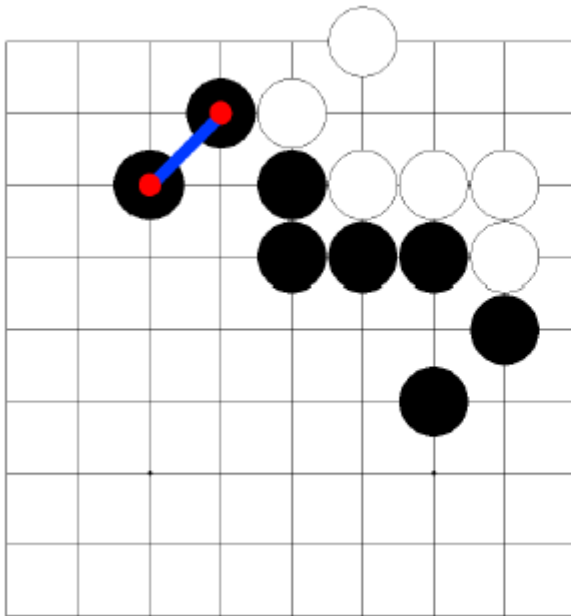


# Oznaczanie *łączności*

- Mając daną pozycję ekspert (człowiek) wskazuje łańcuchy, które w jego przekonaniu będą użyteczne w procesie uczenia
- Następnie oznacza ich *łączność*
- Poziomy *łączności* określające siłę połączenia między dwoma łańcuchami tego samego koloru:
  - *mocno połączone*
  - *połączone*
  - *połączone warunkowo*
  - *rozdzielone*
  - *silnie rozdzielone*
  - *brak połączenia*

# Poziomy łączności

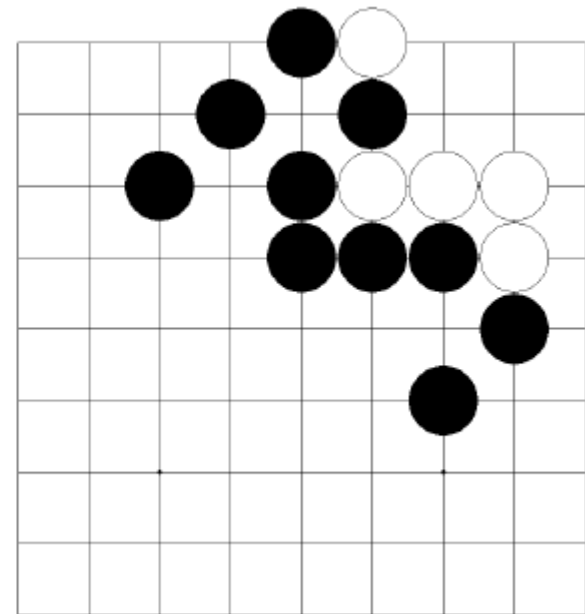
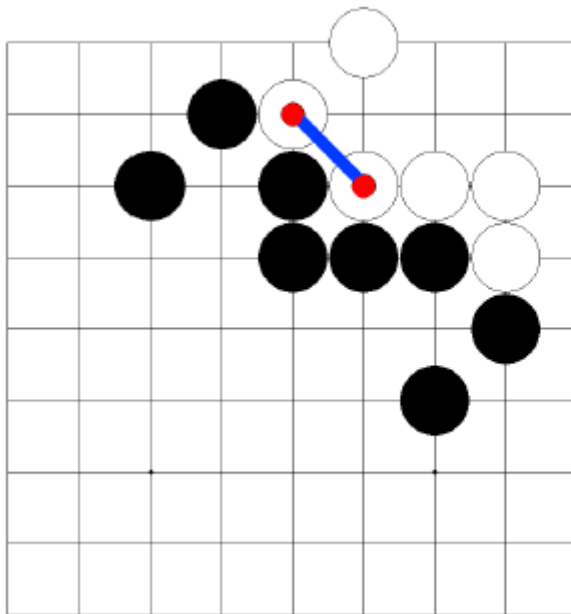
Łańcuchy są *silnie połączone* – nie mogą być rozdzielone nawet, gdyby przeciwnik mógł zagrać dwa ruchy jeden po drugim.



Nawet gdyby biały mógł zagrać dwa ruchy próbując rozdzielić zaznaczone kamienie, czarny może zabić jeden z białych kamieni grając na *a*

# Poziomy łączności

Łańcuchy są *połączone* – mogą być rozdzielone tylko wtedy, gdyby przeciwnik mógł zagrać dwa ruchy jeden po drugim.

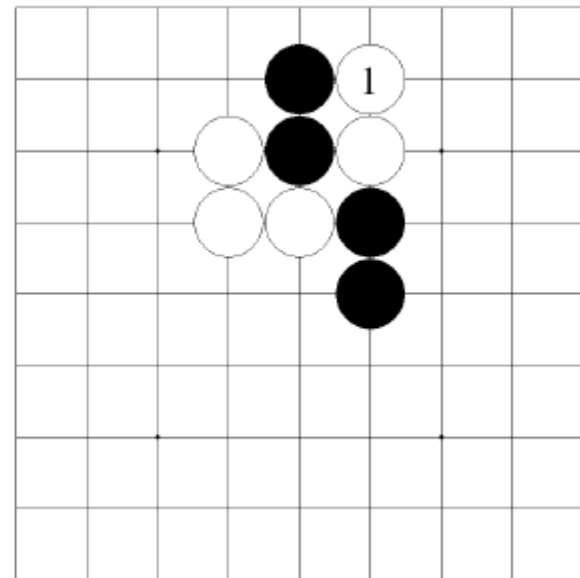
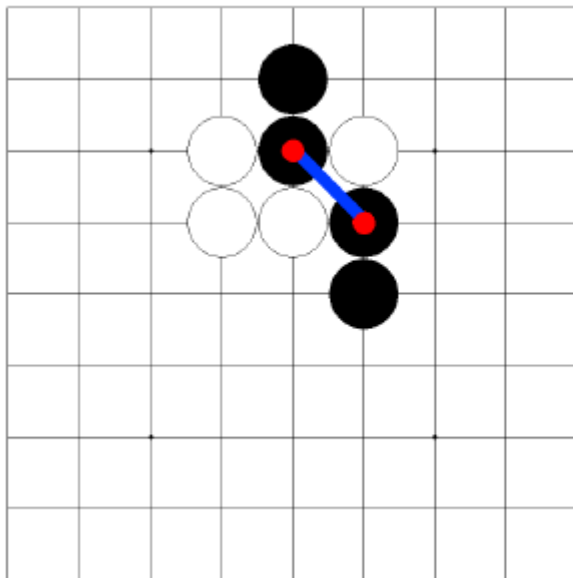


Zaznaczone łańcuchy nie są *silnie połączone*, ponieważ jeden z kamieni może być zabity po wykonaniu dwóch ruchów czarnych



# Poziomy łączności

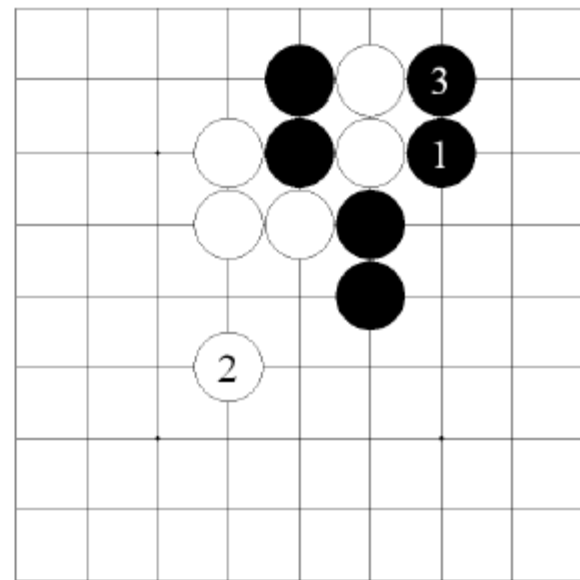
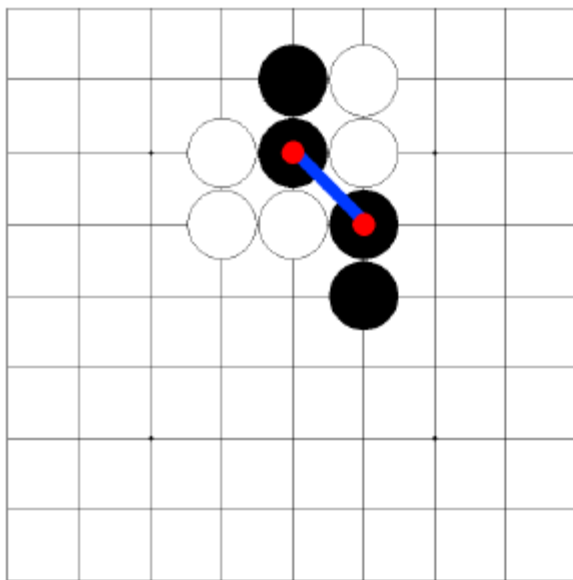
Łańcuchy są *połączone warunkowo* – mogą być rozdzielone wtedy, gdy kolej ruchu przypada na przeciwnika.



Jeżeli ruch przypada na białe, mogą one rozdzielić czarne kamienie grając na pole oznaczone 1

# Poziomy łączności

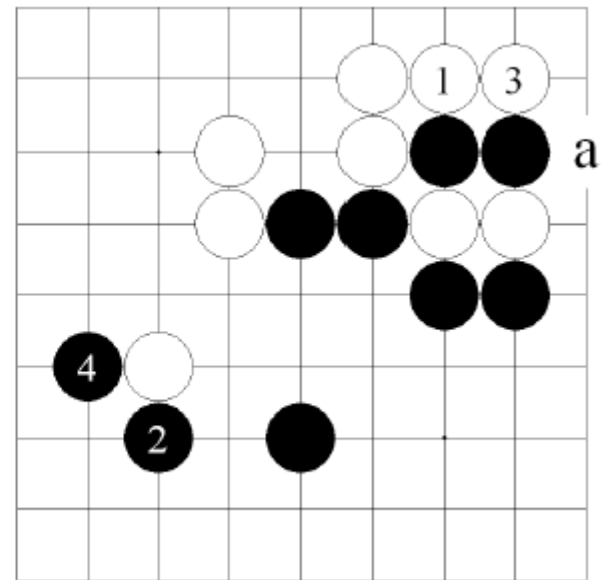
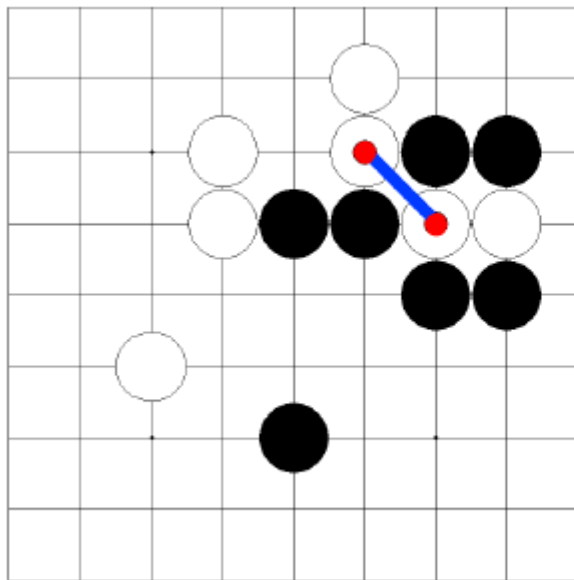
Łańcuchy są *rozdzielone* – mogą być połączone tylko wtedy, gdy gracz posiadający łańcuchy może wykonać dwa ruchy jeden po drugim na danym obszarze.



Zaznaczone łańcuchy mogą być połączone tylko wtedy, gdy białe mogą wykonać dwa ruchy pod rząd w rozpatrywanym obszarze

# Poziomy łączności

Łańcuchy są *silnie rozdzielone* – mogą być połączone tylko wtedy, gdy gracz posiadający łańcuchy może wykonać trzy ruchy pod rząd na danym obszarze.



Zaznaczone łańcuchy mogą być połączone tylko wtedy, gdy białe mogą wykonać trzy ruchy pod rząd w rozpatrywanym obszarze, zdobywając czarne kamienie grając ostatecznie na *a*

# Oznaczanie łączności

- Między łańcuchami występuje *brak połączenia*, jeśli nie występuje żaden wyższy poziom *łączności*
- Ustalenie łączności między wszystkimi łańcuchami jest często niewykonalne ze względu na dużą liczbę łańcuchów  
→ dla 20 łańcuchów ekspert musiałby rozpatrzyć  $\frac{20 \cdot 19}{2} = 190$  różnych relacji *łączności*
- W związku z tym ekspert wybiera łańcuchy według własnego uznania
- Procedura ta prowadzi do średnio 11 połączeń na problem
- Etykietowane są również nietrywialne połączenia, np. łańcuchy rozdzielone łańcuchem przeciwnika, który może zostać zdobyty po zagraniu odpowiedniej sekwencji ruchów!  
→ dobry trening dla klasyfikatora połączeń
- $XC[\textit{chainA}|\textit{chainB}|\textit{category}]$



# Oznaczanie rodzaju ruchu i wyniku

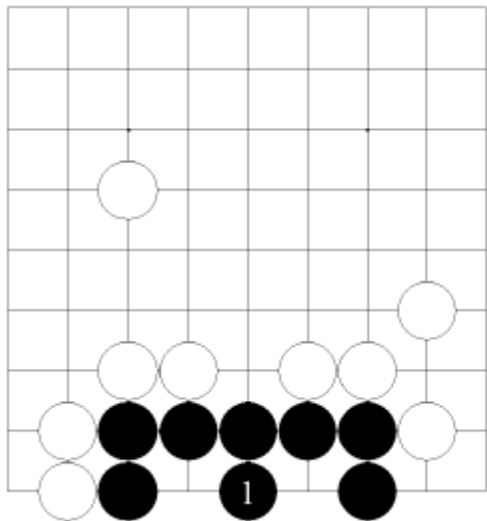
- Wprowadzane pozycje są typu *tesuji* (istnieje zdecydowanie najlepszy ruch, którego zagranie da zupełnie inny rezultat niż niezaganie)
- Z każdym problemem (pozycją) skojarzony jest ruch *tesuji* oraz rodzaj i rezultat tego ruchu
- 43 rodzaje ruchów i 45 rodzajów rezultatu
- $GN[moveType|outcomeType]$



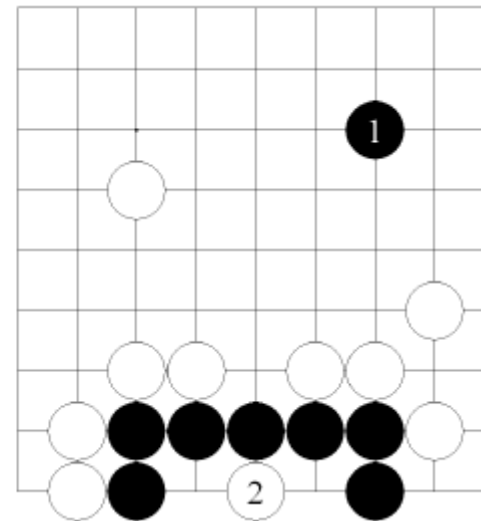
# Oznaczanie *oczu* i *punktów witalnych*

- Oczy odgrywają kluczową rolę w Go
- Dwoje *oczu* → żywa grupa
- Ustalenie czy grupa jest żywa może okazać się trudne  
→ ekspert patrzy na obszar z potencjalną możliwością utworzenia *oczu*
- *Punkty witalne*:
  - kluczowe punkty, na którym zależy obu graczom
  - jeżeli gracz nie postawi tam kamienia, wtedy kamień postawi tam przeciwnik (likwidując przy tym jedno oko i czyniąc tym samym grupę martwą)

# Oznaczanie *oczu* i *punktów witalnych*



Czarne tworzą dwoje *oczu*  
(grupa żywa)



Czarne grają gdzie indziej, białe  
zdobywają czarną grupę



# Oznaczanie oczu i punktów witalnych

- *Obszar oczny zawiera:*
  - *Silny kształt żywy* – jeżeli przeciwnik mógłby wykonać dwa ruchy pod rząd
  - *Kształt żywy* – jeżeli przeciwnik mógłby zagrać pierwszy
  - *Dwa oka* – jeżeli gracz potrzebowałby jednego ruchu
  - *Solidne oko* – jeżeli przeciwnik potrzebowałby dwóch ruchów
  - *Silne jedno oko* – jeżeli przeciwnik mógłby zagrać pierwszy
  - *Jedno oko* – jeżeli gracz potrzebowałby jednego ruchu
  - *Nieprawdopodobne oko* – jeżeli gracz mógłby wykonać dwa ruchy pod rząd w danym obszarze bez przeszkód
- $XE[a_1, \dots, a_n | \text{chain}_1, \dots, \text{chain}_n | \text{category}]$





# Oznaczanie *oczu* i *punktów witalnych*

- Dwie kategorie *punktów witalnych*:
  - punkt decydujący o *życiu i śmierci*
  - punkt *poprawiający kształt lub zaczepny*
- $XV[emptyIntersections|category]$
- Obszary oczne i punkty witalne zostały wprowadzone na próbę, ale nie zostały ekstrahowane, ponieważ nie stworzono żadnej metody uczącej dla tych rodzajów wzorców

# Reprezentacja i ekstrakcja cech

- Dobra reprezentacja problemu → proces uczenia przyniesie większą dokładność wyników
- Kluczowy problem: jakie są istotne (użyteczne) cechy pozycji dla procesu uczenia?
- Wydaje się niemożliwe znalezienie takiego opisu sytuacji, który pasowałby do procesu uczenia każdego abstrakcyjnego pojęcia
- Każde zadanie uczenia należy rozpatrzyć oddzielnie i wybrać odpowiednie cechy stosownie do problemu
- Przykładowe (przekonywujące) reprezentacje cech:
  - cechy łańcucha
  - graf *Common Fate* (CFG)
  - cechy *pokrewnego* podgrafu (RSP)
  - cechy ścieżki *pokrewnego* podgrafu

# Cechy łańcucha

- Niektóre cechy łańcuchów mogą być łatwo otrzymane algorytmicznie:
  - liczba kamieni
  - liczba wolnych miejsc przylegających do łańcucha
  - liczba sąsiadujących łańcuchów przeciwnika
  - liczba sąsiadujących kamieni przeciwnika (suma kamieni sąsiadujących łańcuchów przeciwnika)
  - liczba łańcuchów (własnych), które mogą być dołączone przy użyciu jednego ruchu
  - maksymalna liczba wolnych pól, które mogą być uzyskane przez dodanie kolejnego kamienia do łańcucha (od 2 do -1)
- Opisują sytuację łańcucha zbyt abstrakcyjnie
- Niektóre z cech zawierają się w CFG



# Graf *Common Fate* (CFG)

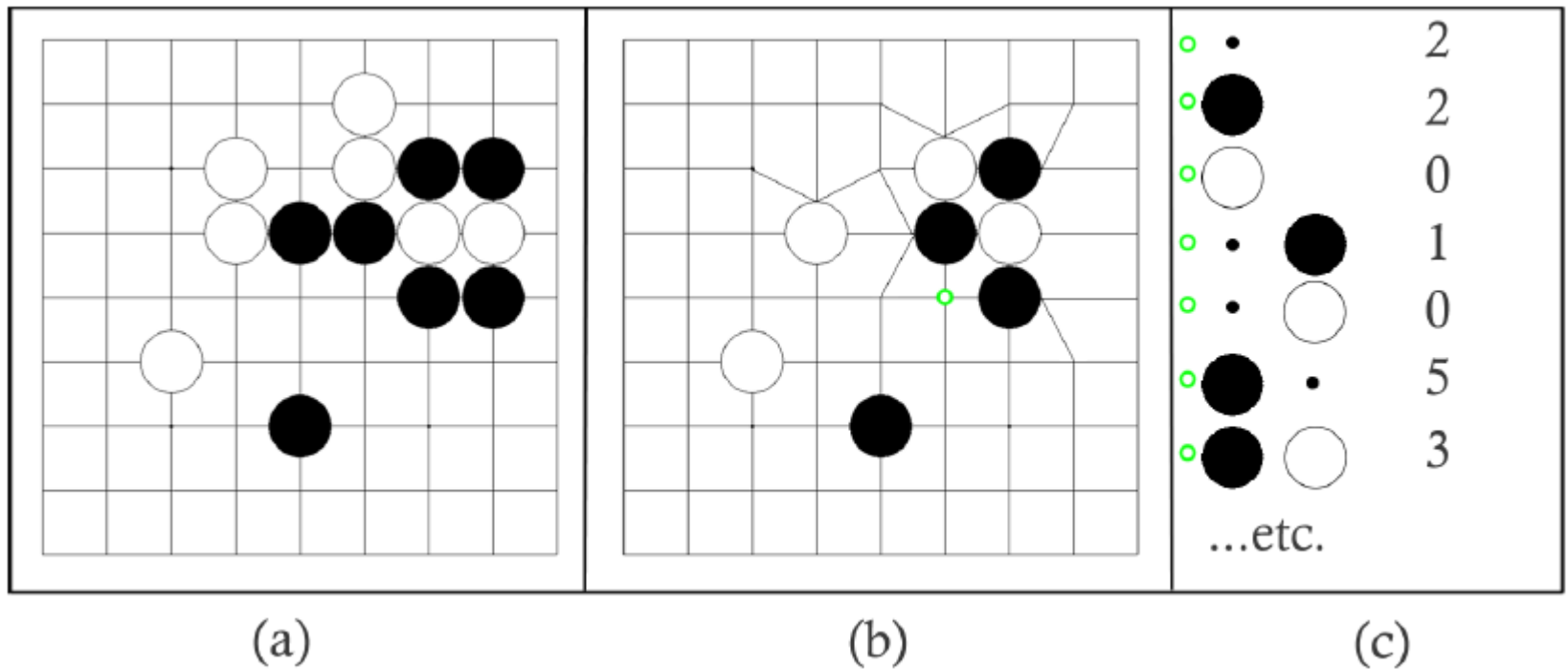
- Obserwacja:
  - łańcuchy dzielą ten sam los
  - kamienie łańcucha mogą być *streszczone* do pojedynczego wężła w grafie
- Reprezentacja planszy przez graf:
  - Full Graph Representation* (FGR)
  - nieskierowany graf  $G_{FGR} = (P, E)$
  - $P = \{p_1, \dots, p_{N_p}\}$  – zbiór reprezentujący punkty na planszy
  - każdy  $p_i$  posiada etykietę  $l : P \rightarrow \{black, white, empty\}$
  - $E = \{e_1, \dots, e_{N_e}\}$ ,  $e_i \in \{\{p, p'\} : p, p' \in P\}$  opisuje relacje między połączonymi punktami na planszy



# Graf *Common Fate* (CFG)

- CFG powstaje z FGR poprzez przekształcenie:
  - łączenie węzłów, które mają ten sam kolor
  - usunięcie zdublowanych krawędzi
- Formalna definicja:
  - Niech  $p, p' \in P, \{p, p'\} \in E, l(p) = l(p') \neq \text{empty}$
  - $P \rightarrow P \setminus \{p'\}$  (łączenie węzłów)
  - $E \rightarrow E \setminus \{\{p', p''\} \in E\} \cup \{\{p, p''\} : \{p', p''\} \in E\}$   
(usuwa zduplikowane krawędzie)
- Stosowanie powyższej transformacji na FGR do momentu, gdy nie będą istniały sąsiadujące węzły w tym samym kolorze, prowadzi do CFG
- Odniesienie do człowieka: Gracze Go również upraszczają sąsiadujące kamienie w jedną całość

# Graf *Common Fate* (CFG)



Proces ekstrakcji cechy. W (a) dana jest reprezentacja RFG, (b) przedstawia CFG otrzymany z (a), (c) prezentuje RSF zielonych węzłów wewnątrz CFG.

# Cechy pokrewnego podgrafu

- Używane do nauki pojęcia rodzaju ruchu i najlepszego lokalnie ruchu
- Wektory cechy  $\vec{x} \in \mathbb{R}^d$
- Jak wyekstrahować wektory cechy z CFG dla problemu uczenia?
- RSF opisuje otoczenie wężła w grafie
- RSF wężła powstaje przy użyciu odwzorowania  $\phi : CFG \times P \longrightarrow \mathbb{R}^d$  :
  - $d$  – liczba możliwych połączonych podgrafów
  - $\tilde{G}_i = (\tilde{P}_i, \tilde{E}_i) \in CFG, i = 1, \dots, d, p \in \tilde{P}_i$
  - wybierany są tylko  $\tilde{G}_i$  takie, że podgraf nie ma gałęzi i pętli i jeden z punktów końcowych podgrafu jest równy  $p$ .
- Wektor cechy  $x_i = \phi_i(p)$  jest liczbą podgrafów  $\tilde{G}_i$ , które mogą być znalezione w CFG



# Cechy pokrewnego podgrafu

- RSF tworzony tylko dla pustych węzłów
- Długość podgrafu ograniczona przez stałą  $s \geq |\tilde{P}_i|$  (rozmiar wektora cechy ma rozsądy rozmiar)
- Intuicyjnie RSF jest opisem wzorca wokół węzła (podobnie wygląda obraz przetwarzany przez żywego gracza Go)
- Słabości:
  - ocena cechy *empty-black-empty* może wprowadzać w błąd, ponieważ nie wiadomo, jak silny lub słaby jest czarny węzeł w podgrafie
  - nie można wykluczyć, że dwie skrajnie różne pozycje utworzą identyczny wektor



# Cechy ścieżki pokrewnego podgrafu

- Używane do nauki pojęcia *łączności*
- Opisuje *łączność* pomiędzy dwoma łańcuchami przy użyciu CFG
- Patrzenie na połączenie pomiędzy dwoma węzłami w grafie polega na analizie ścieżek prowadzących z łańcucha A do łańcucha B
- Jeżeli rozpatrywanie najkrótszych ścieżek z łańcucha A do B nie jest wystarczające, żeby opisać złożoność sytuacji, wtedy analizowane są wszystkie możliwe ścieżki z A do B
- Rodzaje ścieżek mogą być wyekstrahowane przy pomocy RSF
- Cel: wektory RSF powinny opisywać relację *łączności* pomiędzy łańcuchami

# Cechy ścieżki *pokrewnego* podgrafu

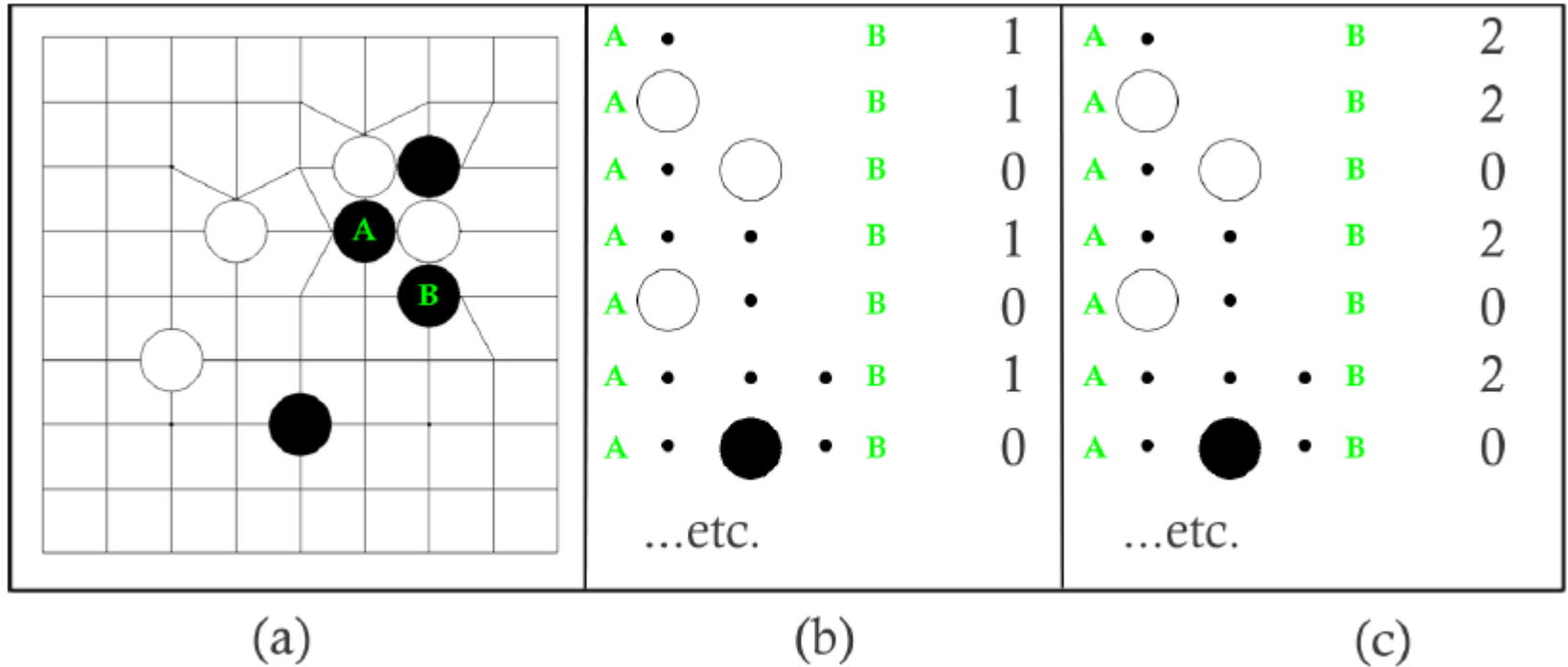
Procedura ekstrakcji cech RSP:

- Rozważ dwa niepołączone łańcuchy tego samego koloru
- Wystartuj w węźle A grafu i odwiedź wszystkie sąsiadujące węzły
- Z każdego węzła przeszukaj wszystkie nieodwiedzone sąsiadujące węzły dopóki łańcuch B nie zostanie odwiedzony
- Ścieżka lub podgraf został znaleziony, gdy osiągnięty zostanie łańcuch B. Początkowy węzeł A i końcowy B w uzyskanym podgrafie są nieistotne, ponieważ są zawsze tego samego koloru. Zatem tylko węzły leżące pomiędzy nimi tworzą podgraf. Dla każdej ścieżki zostaje utworzony wektor cechy w taki sam sposób, jak w RSF

# Cechy ścieżki *pokrewnego* podgrafu

- 3 przypadki reprezentacji ścieżek (w różnych sytuacjach różne reprezentacje dają najlepszy wynik):
  - Typ A: używane są tylko cechy ze ścieżek z łańcucha A do łańcucha B
  - Typ B: dodawane są cechy ze ścieżek z łańcucha A do łańcucha B i vice versa (jeden wektor)
  - Typ C: używane są cechy ze ścieżek z łańcucha A do B łańcucha B i vice versa (dwa wektory cech generujące dwukrotną liczbę danych do nauki)
- Maksymalna długość ścieżki użyta w eksperymencie wynosiła 8 (odpowiednia długość do opisu relacji łączności dla planszy Go)

# Cechy ścieżki pokrewnego podgrafu



Proces ekstrakcji cechy RSP. W (a) dana jest reprezentacja CFG, (b) przedstawia wektor cech RSP dla węzłów A i B typu A, a (c) RSP typu B.



# Klasyfikator

- Mając dane oznaczenia nałożone przez eksperta oraz cechy wygenerowane przez system możliwy jest trening klasyfikatora (*Support Vector Machine classifier* – SVM), który będzie oznaczał nowe pozycje
- Definicja problemu:
  - oznaczone przez eksperta dane treningowe  $(x_i, y_i)$ ,  
 $i = 1, \dots, l$
  - $x_i$  – opis instancji w  $n$ -wymiarowej przestrzeni (komputerowo wygenerowane cechy wyekstrahowane z konkretnej próbki)
  - $y_i$  – oznaczenie instancji dokonane przez eksperta,  
 $y \in \{1, -1\}^l$

# Klasyfikator

Definicja problemu:

- Klasyfikator rozwiązuje problem:

$$\min_{w,b,\xi} \left[ \frac{1}{2} w^T w + C \sum_{i=1}^l \xi_i \right]$$

przy założeniach:

$$y_i (w^T \phi(x_i) + b) \geq 1 - \xi_i, \xi_i \geq 0$$

- Punkt  $x$  jest klasyfikowany zgodnie z funkcją decyzyjną:

$$\hat{y}(x) = \operatorname{sgn} \left( \sum_{i=1}^l y_i \alpha_i K(x_i, x) + b \right)$$

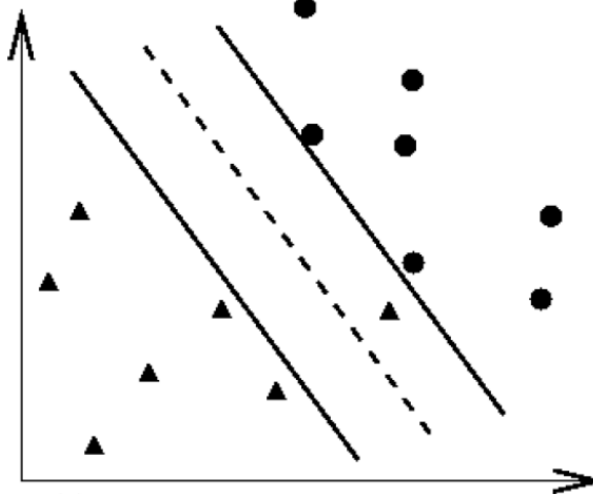
gdzie  $K(x_i, x)$  jest funkcją jądra dla wierzchołków

# Klasyfikator

- Przykład funkcji jądra – radialna funkcja bazowa:

$$K(x, x') = \exp(-\gamma \|x - x'\|^2), \quad \gamma > 0$$

- Przykład zadania dla klasyfikatora (2-wymiarowa przestrzeń cech):



Zadanie uczenia polega na znalezieniu płaszczyzny oddzielającej dwie różne oznaczone grupy obiektów

# Bibliografia

- Emil H.J. Nijhuis, *Learning Patterns in the Game of Go*, MSc Thesis at Universiteit van Amsterdam, 2006
- <http://playgo.to>
- <http://go.art.pl/kurs>





Dziękuję za uwagę