„A hierarchical approach to computer Hex“ (Vadim V. Anshelevich, 2002)

Präsentation von Robert Nitsch für das Seminar
„Knowledge Engineering und Lernen in Spielen“

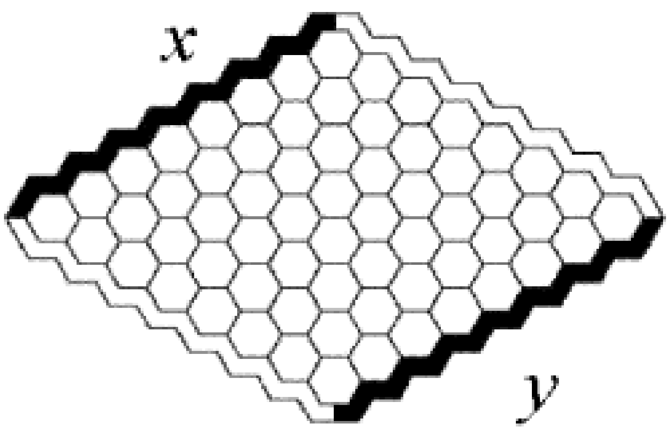
Gliederung

- Das Spiel Hex
- Konventionen
- **Sub-games** & ***virtual (semi)connections*** *(VCs)*
- Deduktionsregeln für VCs -> ***H-Search-Algorithmus***
- Evaluationsfunktion
- HEXY
- Quellen

Das Spiel Hex

- 2 Spieler: schwarz & weiß
- rautenförmiges Spielfeld aus Hexagons

- gegenüberliegende Seiten müssen verbunden werden
- dafür werden abwechselnd Steine auf die Hexagons gelegt



Das Spiel Hex - Besonderheiten

- Der erste Zug gibt einen großen Vorteil => oft mit Tauschregel

- Es kann kein Unentschieden geben
  - Nash zeigte: es existiert eine Gewinnstrategie für den *ersten* Spieler!
    - (für den Fall, dass kein Tausch erlaubt ist)
  - Gewinnen = Nicht verlieren bzw. „Angriff = Verteidigung“

- sehr hoher Branching-Faktor
  - vergleichbar mit Go; größer als bspw. bei Schach, Dame

- menschliche Spieler haben bei größeren Spielfeldern noch knapp die Oberhand
- Wikipedia: Hex *gelöst* bis zu 9x9

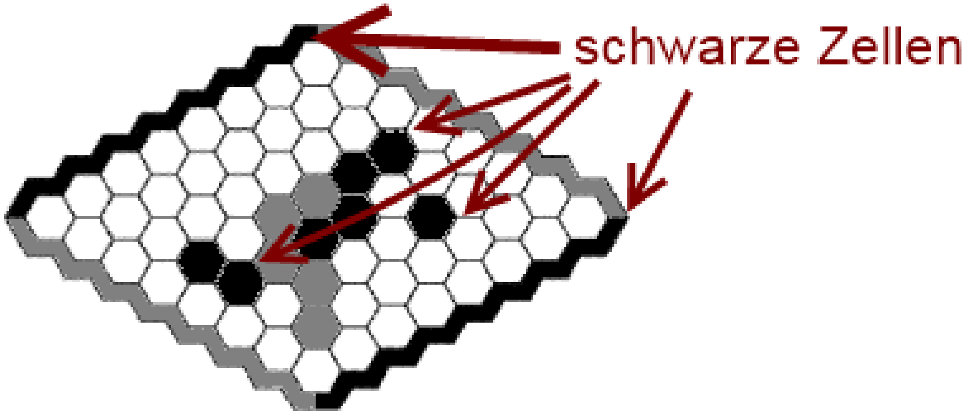
Konvention

- Wir betrachten das Spiel im Folgenden aus der Sicht von Schwarz.
  - (Das kann selbstverständlich stets umgedreht werden.)

- Spielfelder: „Zellen“ (engl. *cells*)
- Spielfeld mit Spielstein: schwarze bzw. weiße Zelle

- Gruppe von gleichfarbigen Zellen bildet ihrerseits eine schwarze bzw. weiße Zelle
- Die Spielfeld-Ränder gelten als weiße bzw. schwarze Zellen

Konvention - Beispiel

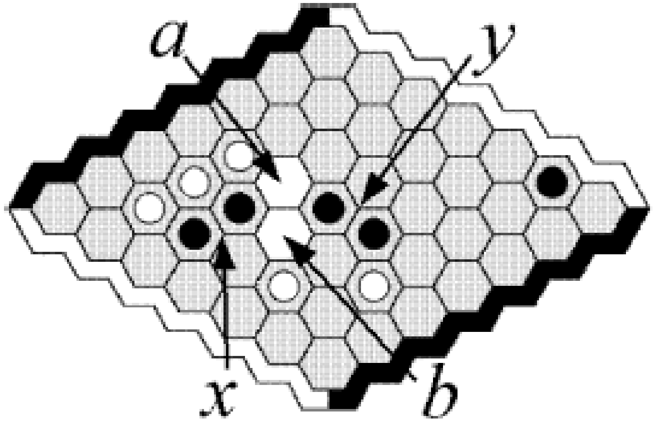


Two-Bridges: Sichere Verbindungen

- x und y formen zusammen eine *Two-Bridge*

- denn: **x** und **y** sind durch 2 versch. freie Zellen verbunden (**a** und **b**)

- Weiß kann nicht verhindern, dass Schwarz seine Zellen x und y verbindet

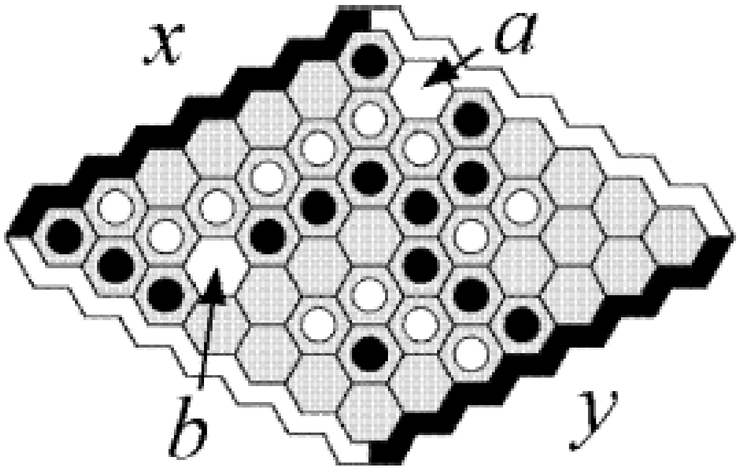


Two-Bridges: 2. Beispiel

- jetzt: **x** und **y** sind die schwarzen Randzellen

- wir wissen: Weiß kann die Verb. von **x** und **y** nicht verhindern

- also: Schwarz hat quasi bereits gewonnen



Verallgemeinerung des Brückenkonzepts –> Sub-games

- <u>Definition 1:</u>
- Ein Tripel (**x**, **A**, **y**) sei ein ***sub-game***, mit
  - **x** ≠ **y** den Zellen, die Schwarz verbinden soll
  - **A** die Menge der leeren Zellen, auf die Steine gelegt werden dürfen
  - **x** und **y** dürfen nicht mit A überlappen

- **A** sei definiert als ***carrier*** des sub-games („Träger“).
- **x** und **y** seien definiert als die ***ends*** des sub-games (die „Enden“).

- Also: Bei diesen sub-games hat Schwarz das Ziel die Enden **x** und **y** über den carrier **A** miteinander zu verbinden. Weiß versucht das zu verhindern.

Sub-games

- <u>Definition 2:</u>
  - Ein sub-game stellt eine **virtual connection** dar gdw. Schwarz eine Gewinnstrategie hat – sogar dann, wenn Weiß zuerst zieht.

- <u>Definition 3:</u>
  - Ein sub-game stellt eine **virtual semi-connection** dar gdw. Schwarz nur dann eine Gewinnstrategie hat, wenn Schwarz auch zuerst zieht (und sonst nicht).

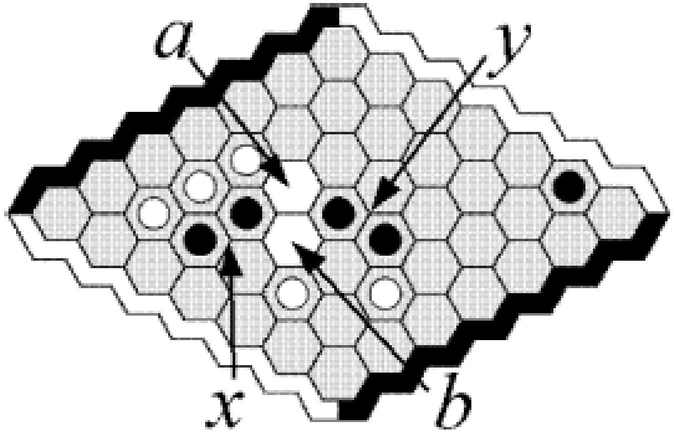
- „Gewinnstrategie“ bezieht sich auf das jeweilige sub-game.
- zu „ziehen“ heißt natürlich konkret: einen Stein auf eine der Zellen aus A legen

Sub-games vs. Two-Bridges

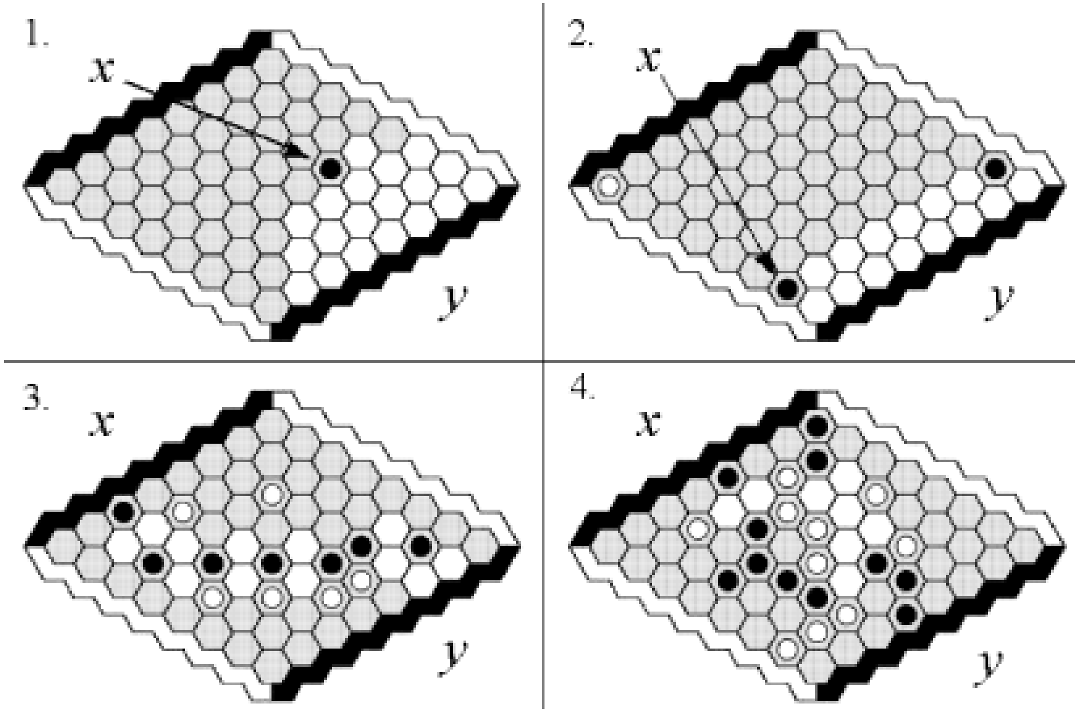
- Two-Bridges sind spezielle sub-games

- Genauer:
  - Eine Two-Bridge ist eine virtual connection, bei der |A|=2.

A = {a, b}



Sub-games: weitere Beispiele



Sub-games: + Rekursion

- Definition 2a:
  - Ein sub-game ist eine **virtual connection** (x, A, y) gdw. für jeden weißen Zug ein schwarzer Zug existiert sodass eine virtual connection (x, A‘, y) *daraus hervorgeht* (mit |A‘|=|A|-2).

- Definition 3a:
  - Ein sub-game ist eine **virtual semi-connection** gdw. das sub-game keine virtual connection ist und es einen schwarzen Zug gibt sodass eine virtual connection daraus hervorgeht.

Darstellung von sub-games

1. schwarz zu schwarz
2. schwarz zu leer
3. leer zu leer

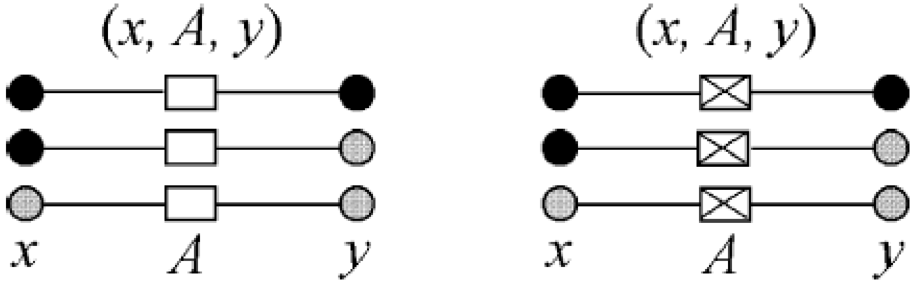


Fig. 3. Diagrams of virtual connections (left) and virtual semi-connections (right).

Tiefe von sub-games

- Die Anzahl der Züge die es braucht bis Schwarz ein sub-game gewonnen hat heißt ***Tiefe*** (engl. *depth*) des sub-games.
- Bedingung: Schwarz und Weiß spielen jeweils perfekt.

- Bemerke:
  - Eine virtual connection mit Tiefe d besitzt bereits Informationen über den (potenziellen) Spielzustand in d Zügen.

Bemerkungen

- Ein Paar benachbarter Zellen formt eine virtual connection mit leerem Träger und der Tiefe 0.

- Two-Bridges: Eine Two-Bridge ist eine virtual connection mit Tiefe d=2.

- Eindeutigkeit: Die Enden x und y können verschiedene virtual connections formen. (Der Träger A kann unterschiedlich gewählt werden.)
  - -> **Minimalität**: Eine virtual connection (x, A, y) ist minimal gdw. es keine virtual connection (x, B, y) gibt mit $B \subset A$

Bemerkungen (Fortsetzung)

- Gegeben sei eine minimale virtual connection (x, A, y).
  - Dann sind alle (x, B, y) mit $A \subset B$ ebenfalls virtual connections (jedoch: nicht mehr minimal).

- * (x, A, y) ist virtual (semi)connection gdw. (y, A, x) ist virtual (semi)connection
  - Vergleiche: **Symmetrie** von Relationen

AND-Deduktionsregel

- Gegeben: Virtual connections (x, A, u) und (u, B, y) mit einziger Überlappung bei u (formal: $x \neq y \wedge x \notin B \wedge y \notin A \wedge A \cap B = \emptyset$).

- Dann gilt:
  - (i) Wenn u schwarz ist, dann ist das sub-game (x, A ∪ B, y) eine virtual connection.
  - (ii) Wenn u leer ist, dann ist das sub-game (x, A ∪ u ∪ B, y) eine virtual semi-connection.

- * (i) - Vergleiche: **Transitivität** von Relationen. (Wenn wir x und u sowie u und y garantiert verbinden können, dann auch x und y.)

- * In Fall (ii) darf u eigentlich nur eine einzelne Zelle sein (!?).

OR-Deduktionsregel

- Gegeben: 2 bis n virtual <u>semi</u>-connections (x, A_i, y) mit disjunkten A_i. $(i = 1, ..., n.)$

- Dann gilt:
  - (x, A, y) ist eine virtual connection mit $$A = \bigcup_{i=1}^{n} A_i$$

- Anschaulich: Wenn Weiß bspw. in A_1 einen Stein legt, dann hat Schwarz den 1. Zug in allen anderen A_i.

Wdh.: Darstellung von sub-games

1. schwarz zu schwarz
2. schwarz zu leer
3. leer zu leer

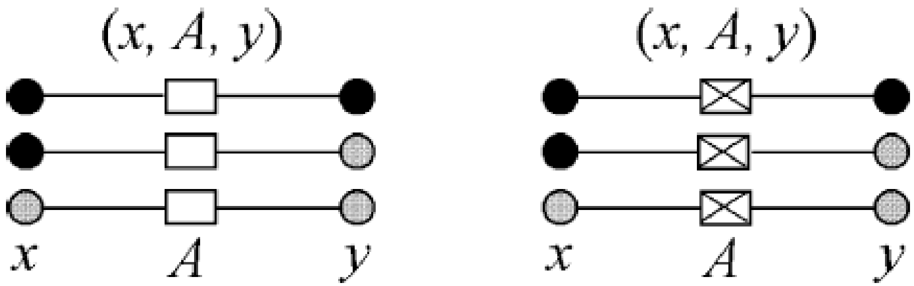


Fig. 3. Diagrams of virtual connections (left) and virtual semi-connections (right).

Beispiel: Deduktionsregeln

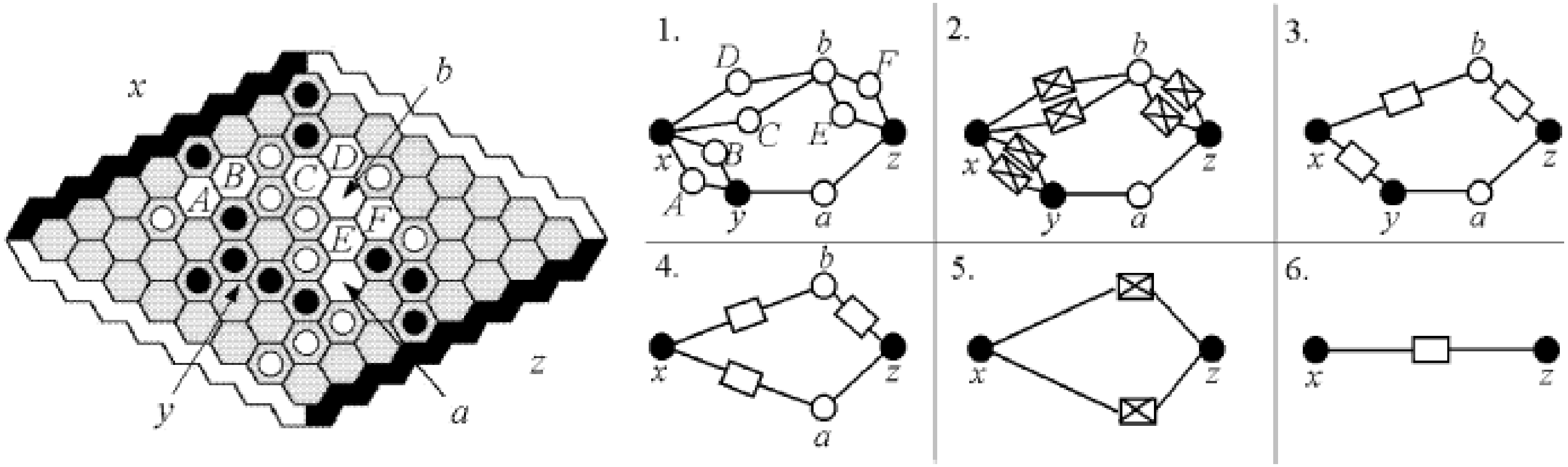


Fig. 8. The use of AND and OR deduction rules.

H-Search-Algorithmus

- Gegeben: Ein Spielzustand (node).

- Beginnend mit einer Grundmenge von virtual (semi)connections werden wiederholt die AND- bzw. OR-Regel angewandt, bis
  - keine weiteren Verbindungen erzeugt werden
  - oder eine gewinnende Verbindung erzeugt wurde

- Grundmenge: Im einfachsten Fall alle Paare zueinander benachbarter cells.

- Aber: genannte Deduktionsregeln sind nicht *vollständig*, können also i.A. nicht alle virtual connections erzeugen/finden
  - (Man könnte sie entsprechend erweitern, wurde hier aber nicht gemacht.)

Evaluationsfunktion

- Gegeben: Ein Spielzustand (node).
- Gesucht: Eine Funktion die schätzt wie „gut" diese Position für Schwarz/Weiß ist.

- Idee: Spielzustand als gegenüberliegende Seiten verbindender Graph. (Je 1 Graph für Schwarz und Weiß.)
  - Aus den Graphen lässt sich eine Distanz berechnen als Maß dafür, wie nah die Spieler an ihrem Ziel sind.

- 2 Fragen stellen sich:
  - Kanten <-> virtuelle Verbindungen? mit welchen Kosten?
  - Distanz <-> kürzester Pfad durch den jeweiligen Graphen?
  - => verschiedene Ansätze möglich!

Evaluationsfunktion

- Die Evaluationsfunktion ist allgemein $E = \log(D_{Schwarz}/D_{Weiß})$.
- Anshelevich bevorzugt einen Ansatz, der so ähnlich erstmals von Shannon angewendet wurde („Shannon switching game“).
  - Man fasst die Graphen als Schaltkreise auf.
  - An den Rändern wird eine elektrische Spannung angelegt.
- Also:
  - Kanten <-> elektrische Verb. mit ihren Kosten als Widerstand
  - Distanz <-> *totaler Widerstand* der jeweiligen Schaltung (physikalische Größe)

Spielfeld als Schaltkreis

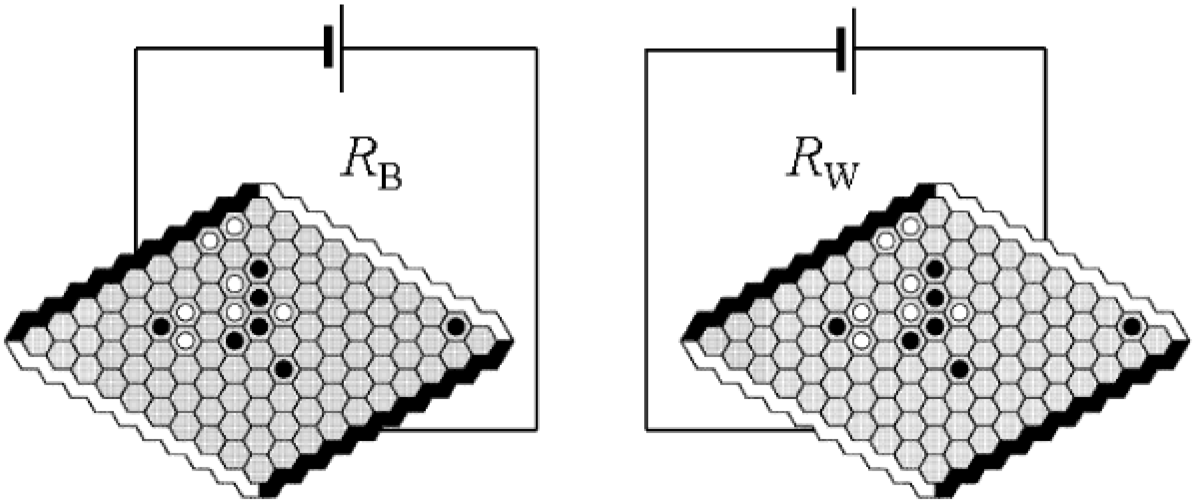


Fig. 10. Black's and White's circuits.

Die Schaltkreise im Detail

- Also: Zu jeder Spielposition werden 2 „Schaltkreise" modelliert, jeweils für Schwarz bzw. Weiß.
- Jeder Zelle c wird ein Widerstand r (resistance) zugewiesen
  - In Schwarz' Schaltkreis:
    - (analog für Weiß)

$$r_B(c) = \begin{cases} 1 & \text{if } c \text{ is empty,} \\ 0 & \text{if } c \text{ is occupied by a black piece,} \\ +\infty & \text{if } c \text{ is occupied by a white piece.} \end{cases}$$

- Benachbarte Zellen c1 und c2 werden verbunden mit Widerstand $r_B(c1) + r_B(c2)$ bzw. $r_W(c1) + r_W(c2)$.
- Weitere virtuell verbundene Zellen x und y werden verbunden mit Widerstand 3+ (um direkte Verbindungen aufzuwerten).

- (Verallgemeinert entsprechen diese Verbindungen den Kanten des Graphen, der die Ränder miteinander verbindet.)

Die Schaltkreise im Detail 2

- Berechnet werden können dann die totalen Widerstände des Schaltkreises von Schwarz (R_B) und von Weiß (R_W).
  - (Totaler Widerstand kann berechnet werden durch Lösung eines linearen Gleichungssystems.)

- Warum eigentlich so kompliziert?
  - Aus der Physik weiß man (Kirchhoffsche Regeln):
    - der *totale Widerstand* berücksichtigt nicht nur die Länge des kürzesten Pfades sondern auch alle anderen Pfade, ihre Längen und ihre Kreuzungen/Überschneidungen.
  - Also berücksichtigt diese Evaluationsfunktion nicht nur die einfachste zu vervollst. Kette, sondern auch alle weiteren möglichen Ketten.
  - Außerdem: Dank der virtual connections blickt die Evaluationsfunktion sehr weit voraus! Nicht selten VCs mit Tiefe 20!

Evaluationsfunktion im Überblick

- Zu jedem Spielzustand modelliert man also wie gezeigt 2 Graphen bzw. „Schaltkreise“ mit totalen Widerständen R_B und R_W als Distanzen.

- Die bevorzugte Evaluationsfunkt. von Anshelevich ergibt sich dann insgesamt als
  - $E = \log(R_B/R_W)$

    - * Wenn $R_B > R_W$ <-> E positiv (schlecht für Schwarz)
    - * Wenn $R_B < R_W$ <-> E negativ (gut für Schwarz)
    - * $E = 0$ <-> ausgeglichen
    - * $E \to +\infty$ <-> Weiß wird gewinnen
    - * $E \to -\infty$ <-> Schwarz wird gewinnen

HEXY

- HEXY ist Anshelevichs Hex-Software (nur 192KB!).
- HEXY gewann bei der „5th Computer Olympiad" (Hex) in London, im August 2000.
  - **I**nternational **C**omputer **G**ames **A**ssociation: http://www.icga.org/
    - neben Hex viele weitere Spiele!
- Bemerkung:
  - nach London 2000 hat HEXY an keiner Computer Olympiad mehr teilgenommen

HEXY - Vorgehen

TECHNISCHE UNIVERSITÄT DARMSTADT

- Verwendet eine Kombination aus game tree search & H-Search.
  - alpha-beta Such-Algorithmus (quasi Minimax-Algorithmus).
  - Benutzt KEIN opening book o.ä.!

- Alle bei der alpha-beta-Suche betrachteten nodes werden wie gezeigt **evaluiert**.
- Bei diesen Evaluationen kommt H-Search zum Einsatz um die virtual connections zu jeder node zu finden.

HEXY – Vorgehen 2

- In der Praxis behält HEXY von node zu node so viele virtual connections wie möglich bei.
  - Es wird analysiert, wie sich die Menge der virtual connections verändert, wenn ein Spielstein hinzukommt.
  - Insbesondere berechnet HEXY nur *minimale* virtual connections.

- * Zugreihenfolge: Es werden diejenigen Zellen bevorzugt belegt, die in dem Evaluationsgraphen am schlechtesten „vernetzt“ sind.
  - (Es werden quasi zuerst die Schwachstellen ausgemerzt.)

HEXY - Optimierung

- Wichtige Parameter wurden experimentell optimiert.
- Am einflussreichsten sind die Parameter D, M und K.
  - D: Tiefe der game tree search.
  - M: Limitiert die Anzahl verschiedener minimaler virtual connections zwischen gleichen Enden x und y, die von HEXY erzeugt/mitgeführt werden.
  - K: schränkt die Anzahl der virtual semi-connections ein die als Eingabe für die OR-Regel verwendet werden.
    - Typische Werte: K=4 oder K=5

- Bspw. für 10x10 Hex beste Wahl: D=3 und M=20.

Hex Computer Olympiad
-> Nach London 2000

Edition	Event (Teilnehmer)	Gewinner	Gewinner basiert auf HEXYs Konzepten
15	Kanazawa 2010 Steht noch aus!	-	-
14	Pamplona 2009 (4)	MoHex 2009	?
13	Beijing 2008 (4)	Wolve 2008	Ja
11	Turin 2006 (3)	Six	Ja
9	Ramat-Gan 2004 (2)	Six	Ja
8	Graz 2003 (2)	Six	Ja
5	London 2000 (3)	HEXY	(natürlich)

Quellen

- **„A hierarchical approach to computer Hex“**
  - Vadim V. Anshelevich, 2002
  - Download: http://home.earthlink.net/~vanshel/

- “The Game of Hex: An Automatic Theorem Proving Approach to Game Programming”
  - Vadim V. Anshelevich, 2000
  - Download: http://home.earthlink.net/~vanshel/

- ICGA Hex Tournaments
  - http://www.grappa.univ-lille3.fr/icga/game.php?id=7

Fragen?

- ???
- ???
- ???
- ???
- ???
- ???