



Seminar

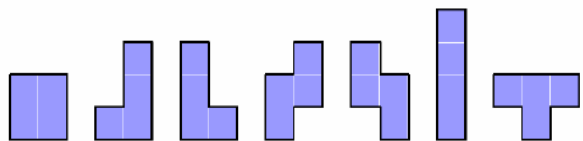
Knowledge Engineering und Lernen in Spielen

„Reinforcement Learning to Play Tetris“

Überblick

- Allgemeines zu Tetris
- Tetris ist NP-vollständig
- Reinforcement Learning
 - Anwendung auf Tetris
 - Repräsentationen des Zustandsraumes
- Relational Reinforcement Learning

Allgemeines zu Tetris

Tetrominos: 

Funktionen: Rotation, Translation

Spielbrett: $m \times n$, opt. gefüllt

Allgemeines zu Tetris

- „Offline“ – Version:
 - Spielbrett, sowie komplette Folge von Tetrominos sind bekannt.
- Geg.: Initiales Spielbrett und endliche Folge von Tetrominos
Frage: Kann das Spielbrett gelöscht werden?
(=> Tetris Problem)

Tetris ist NP - Vollständig

- Grundlagen
 - P: Komplexitätsklasse der Probleme mit „effizienten“ Algorithmen.
 - NP: zusätzlich die Probleme für die noch kein „effizienter“ Algorithmus gefunden wurde.
 - $P \subseteq NP$
 - Seien A und B zwei Probleme. Dann heißt A auf B „*polynomial reduzierbar*“ ($A \leq_p B$), falls es eine totale und mit polynomialer Komplexität berechenbare Funktion gibt, mit $x \in A \Leftrightarrow f(x) \in B$

Tetris ist NP - Vollständig

- Grundlagen
 - A heißt NP-*hart* (*schwer*), falls für alle Probleme $L \in \text{NP}$ gilt:
 $L \leq_p A$.
 - A heißt NP-*vollständig*, falls A NP-*hart* ist und $A \in \text{NP}$ ist.
 - \leq_p ist transitiv. $\Rightarrow L \leq_p A$ und $A \leq_p B$ folgt
 $L \leq_p B$
- Falls A NP-*hart*, genügt $A \leq_p B$ und $B \in \text{NP}$ zu zeigen.

Tetris ist NP - Vollständig

- Das 3 – Partitions – Problem

- Geg.: Sequenz A von positiven natürlichen Zahlen a_1, \dots, a_{3s} und eine positive Zahl T, so dass

- 1.) $T/4 < a_i < T/2$ für alle $1 \leq i \leq 3s$ und

- 2.) $\sum_{i=1}^{3s} a_i = sT$

Kann A in s disjunkte Teilmengen B_1, \dots, B_s unterteilt werden, so dass $\sum_{a_i \in B_j} a_i = T$ für alle $1 \leq j \leq s$?

$\Rightarrow |B_j| = 3$, da falls $|B_j| < 3$ gilt: $\sum_{a_i \in B_j} a_i < 2 * T/2 = T$ wegen 1.) und falls $|B_j| > 3$ gilt: $\sum_{a_i \in B_j} a_i > 4 * T/4 = T$

Tetris ist NP - Vollständig

Bsp.: $T = 20$ $A = \{6,6,6,6,7,7,7,7,8\}$. s ist hier also 3 und die Summe der a_i beträgt $3 * T = 60$. s disjunkte Teilmengen:
 $B_1 = \{6,7,7\}$ $B_2 = \{6,6,8\}$ $B_3 = \{6,7,7\}$

Sei P das 3-Partitionen-Problem und Q das Tetris Problem:
Zu zeigen: $P \leq_p Q$

Ges.: Abbildung f von (A,T) auf ein Spielbrett und auf eine Folge von Tetrominos.

Tetris ist NP - Vollständig

Spielbrett:

S buckets == s
Teilmengen

$W=4s+6$

$H=5T+18$




Abbildung 2: Spielbrett



Aus [Breukelaar, S. 3]


Tetris ist NP - Vollständig


- Unsere gesuchte Funktion informell beschrieben :
 - Für jedes $a_i \in A$ erzeuge folgende Sequenz:


- „Anfang“: 

- gefolgt von a_i -mal „Mitte“:  ,  , 

- gefolgt von „Ende“:  , 

2. s -mal  , um die *buckets* abzuschließen

3.  , für *lock*

4. $5T + 16$ -mal  , um den Rest zu löschen

Tetris ist NP - Vollständig

Füllen eines Buckets mit Wert 3

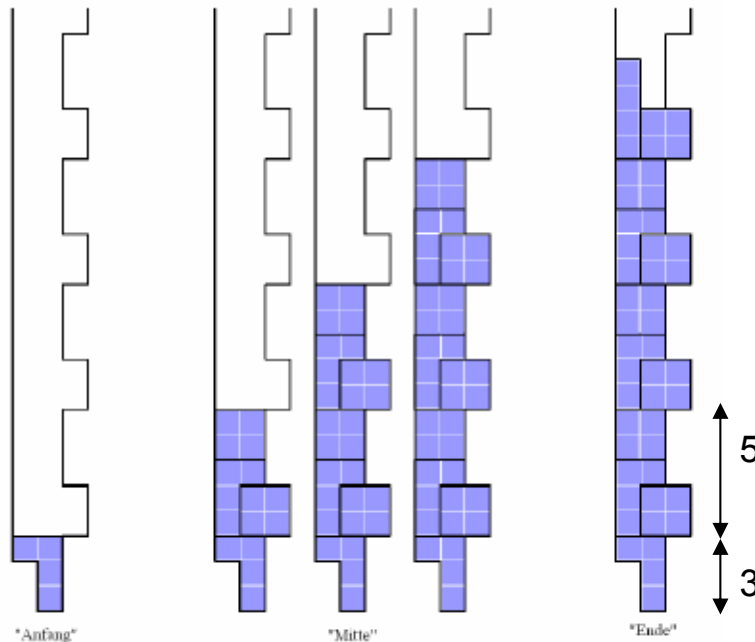


Abbildung 3: Füllen der buckets

Aus [Breukelaar, S. 5]

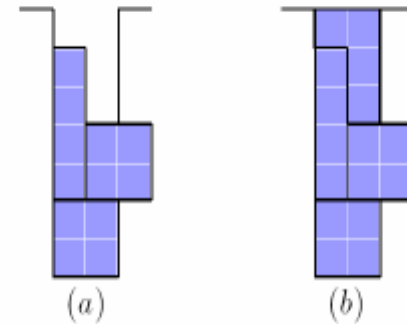


Abbildung 4: Abschließen der *buckets*

Aus [Breukelaar, S. 6]

Tetris ist NP - Vollständig

- Eine lösbare Instanz
 - Füllen von bucket j mit Wert a_i korrespondiert mit Aufnehmen des Wertes a_i in B_j .
 - Es werden $a_i + 1$ notches gefüllt.
 - Höhe $5T+18$ wegen $\sum_{a_i \text{ aus } B_j} a_i = T \Rightarrow T+3$ notches und somit $5*(T+3) = 5T+15$, $+3$ (Anfang) = $5T+18$

Tetris ist NP - Vollständig

- Eine unlösbare Instanz
 - Lemma 1
 - *Wenn ein Stein oberhalb der $5T + 18$ Zeilen platziert wird, kann das Spielbrett nicht gelöscht werden. (jedenfalls nicht mit unserer Funktion)*
 - Lemma 2
 - *Um das Spielbrett zu löschen, darf kein anderer Stein als der dafür vorgesehene den Platz bei lock füllen. (alle anderen hinterlassen lücken-L1)*
 - Lemma 3
 - *Wenn das Platzieren eines Steins eine Lücke hinterlässt, die kein anderer Stein durch Translation und Rotation erreichen kann, kann das Spielbrett nicht gelöscht werden. (L1)*

Tetris ist NP - Vollständig

- Eine unlösbare Instanz
 - Lemma 4
 - Wenn zwei Steine einer Sequenz von "Anfang", "Mitte", "Ende" für einen Wert a_i in verschiedene buckets platziert werden, kann das Spielbrett nicht gelöscht werden. (Stein für Anfang wurde schon platziert)

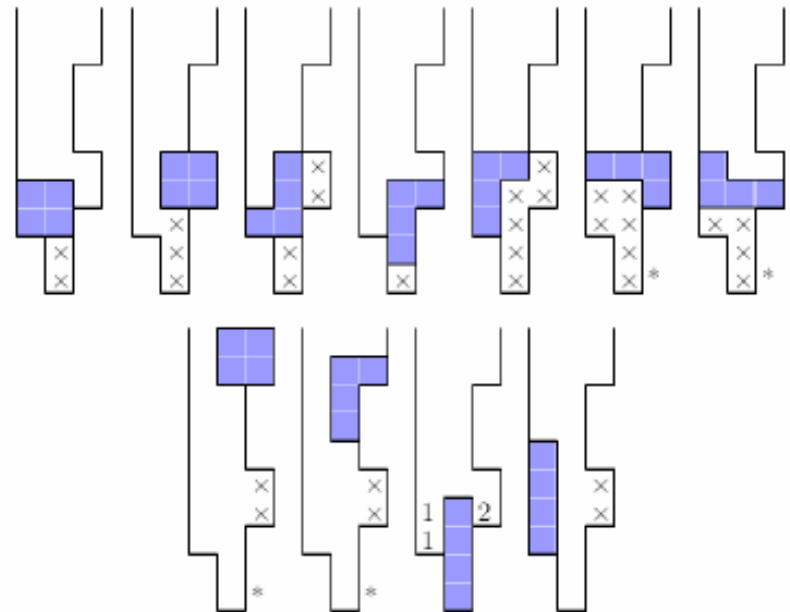


Abbildung 5: Alle Möglichkeiten

Aus [Breukelaar, S. 7]

Tetris ist NP - Vollständig

- Eine unlösbare Instanz
 - Lemma 5
 - *Um das Spielbrett zu löschen, müssen die Steine der Sequenz für ein a_i genau so in einem bucket untergebracht werden, wie es in der lösbaren Instanz beschrieben wurde.*

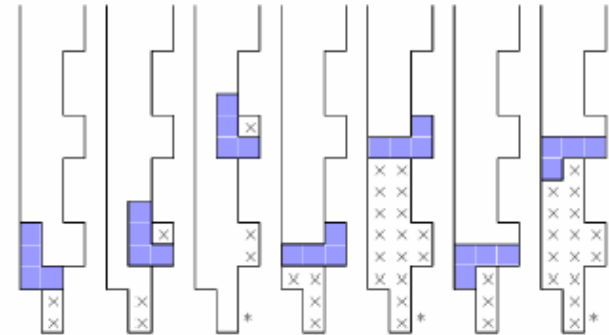


Abbildung 6: Alle Möglichkeiten für „Anfang“

Aus [Breukelaar, S. 8]

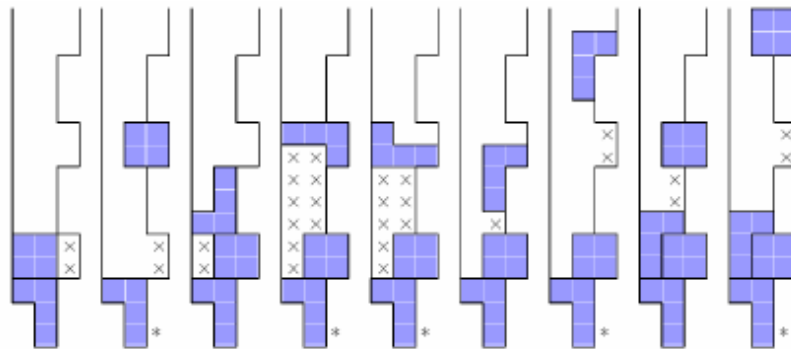


Abbildung 7: Alle Möglichkeiten für „Mitte“

Aus [Breukelaar, S. 8]

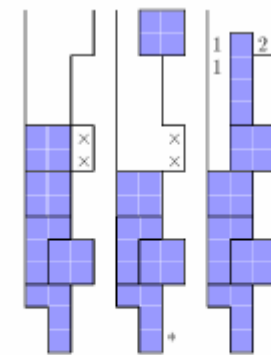


Abbildung 8: Alle Möglichkeiten für „Ende“

Aus [Breukelaar, S. 8]

Tetris ist NP - Vollständig

- Eine unlösbare Instanz
 - Lemma 6
 - *Um das Spielbrett zu löschen, muss ein bucket genau drei Werte a_i enthalten und die Summe dieser Werte muss T betragen.*
 - Das Spielbrett kann aufgrund der Höhe von $5T+18$ aber nur gelöscht werden, wenn $T+3$ notches pro Bucket gefüllt werden. Dies ist mit einer lösbaren Instanz möglich, denn $T+3 = \sum_{a_i \in B} a_i + |B|$. In einer unlösbaren Instanz ist es nicht möglich.
 - Damit ist gezeigt, dass es eine Funktion gibt mit:
 - $(A,T) \in P \Rightarrow f(A,T) \in Q$ und $(A,T) \notin P \Rightarrow f(A,T) \notin Q$
 - $\Rightarrow \underline{(A,T) \in P \Leftrightarrow f(A,T) \in Q}$. D.h. Tetris ist NP-hart; da Tetris auch n NP ist, folgt daraus dass Tetris NP-Vollständig ist.

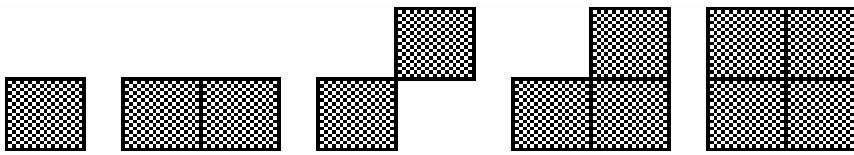
Reinforcement Learning

- Variante des Maschinellen Lernens
- Agent lernt durch Belohnung & Strafe
- Zustand → Aktion
- Nutzen maximieren : Nutzenfunktion

Anwendung auf Tetris

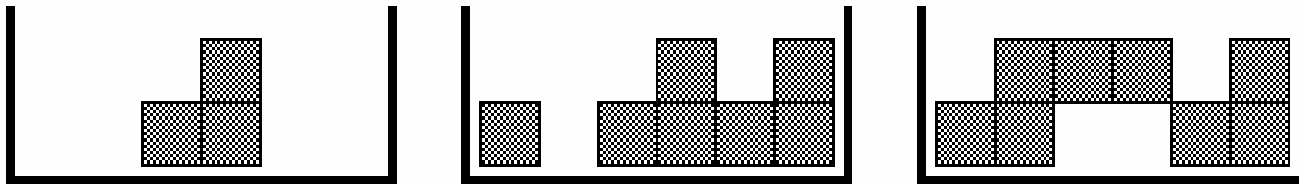
- Ziel: Elimination möglichst vieler Reihen
- Problem: Zufällige Blöcke → Nichtdeterministisch
- Nutzen maximieren mit Nutzentabelle
- Echtzeitproblem

Vereinfachte Version

- Blöcke: 
Quelle: [1]

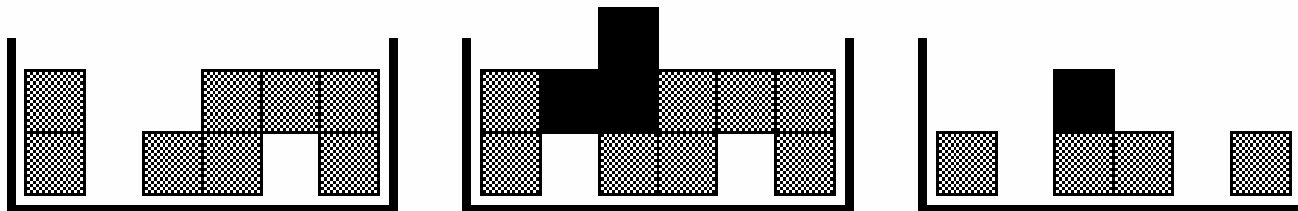
- Spielfeldgröße: 6x2 Einheiten

- Mögliche Zustände:



Quelle: [1]

- Zeileneliminierung:



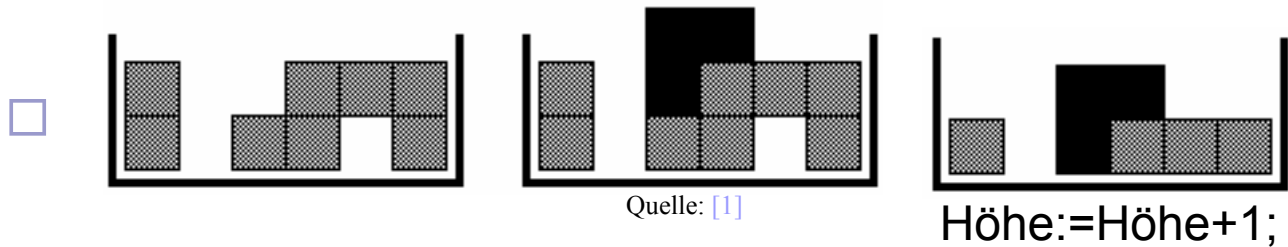
Quelle: [1]

TU - Darmstadt

Mustafa Gökhan Sögüt, Harald Matussek

Vereinfachte Version

- Sonderfall:



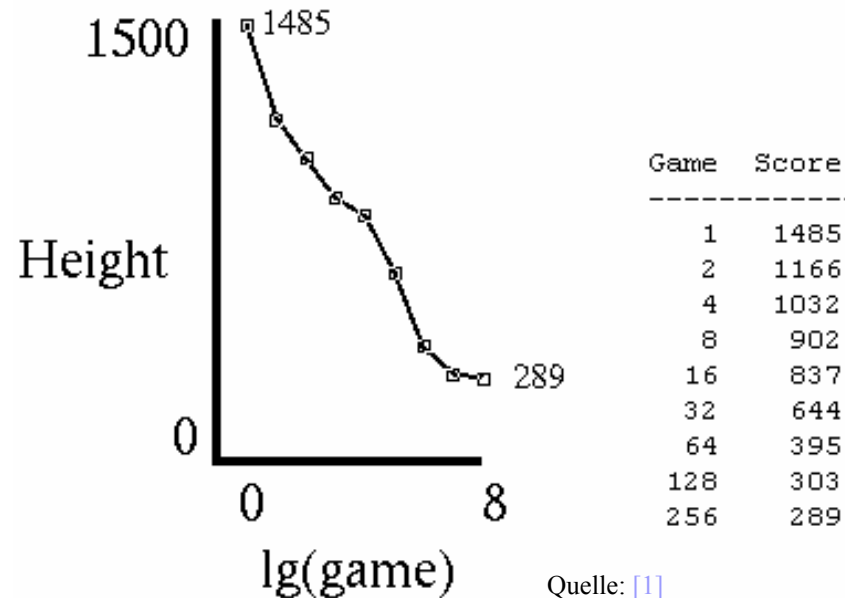
- Spielrundendauer: 10.000 Blöcke

- Größe der Nutzentabelle: 4096



Vereinfachte Version

■ Performanz



■ Update:

$$\square U(\text{Zust.}) = U(\text{Zust.}) * (1 - \alpha) + (\text{reward} + \gamma * U(\text{nächst. Zust.})) * \alpha$$

Vereinfachte Version

■ Verbesserungsvorschlag

- $\alpha = 1/n$
- $n \rightarrow$ Spielrunde

($\gamma=0.8$) value of Alpha

Game	0.002	0.02	0.2
1	1451	1485	1404
2	1204	1166	1043
4	1043	1032	752
8	971	902	525
16	938	837	420
32	912	644	370
64	955	395	342
128	848	303	339
256	679	289	351

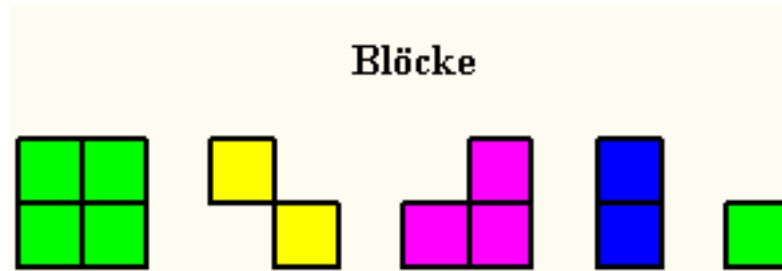
Quelle: [1]

■ Probleme

- Exploitation \longleftrightarrow Exploration
- Spielfeldgröße \rightarrow Größe der Nutzentabelle

Repräsentation des Zustandsraumes

- Konturbeschreibung (Skyline)
- TOP TWO LEVEL



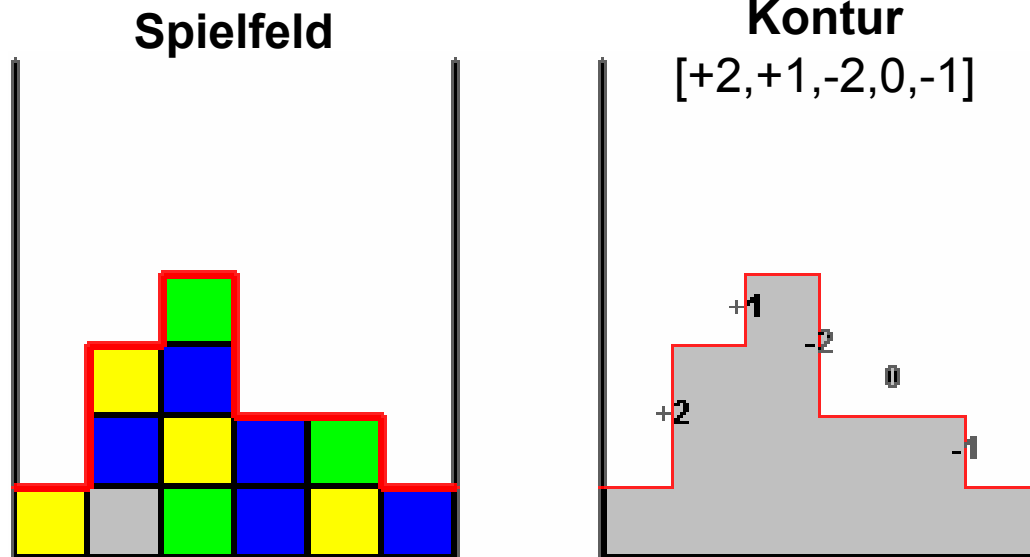
Quelle: [2]

Konturbeschreibung

- Höhenunterschiede benachbarter Spalten
 - Werte: [-2,-1,0,1,2]
- Informationsverlust: Löcher
- Speicherreduzierung: 3125 Zustände

Konturbeschreibung

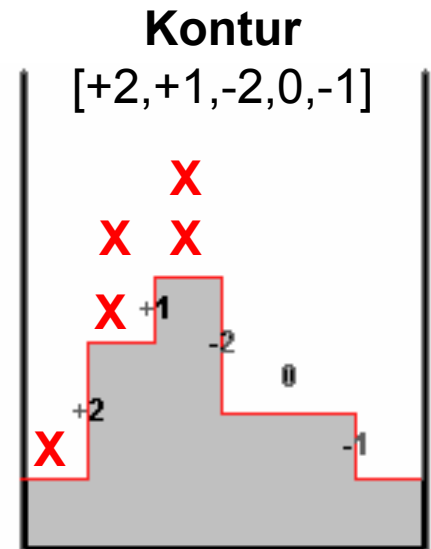
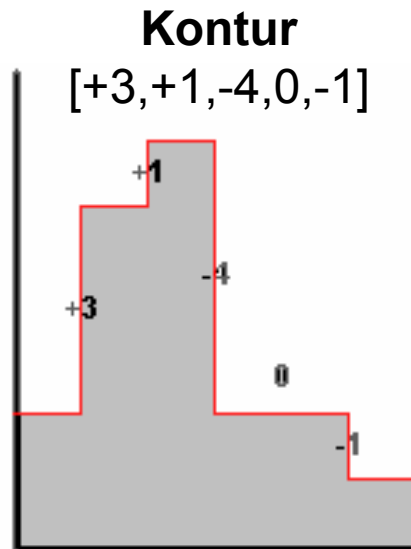
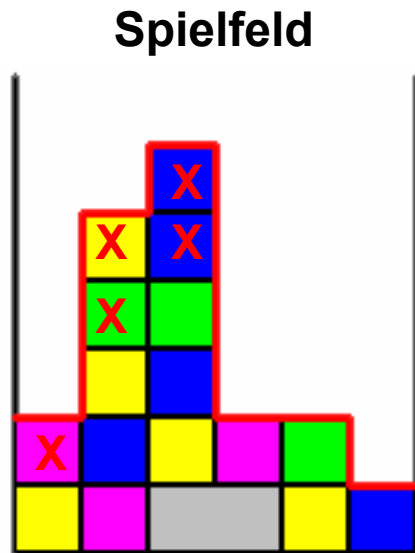
■ Beispiel



Quelle: [2]

Konturbeschreibung

■ Sonderfall

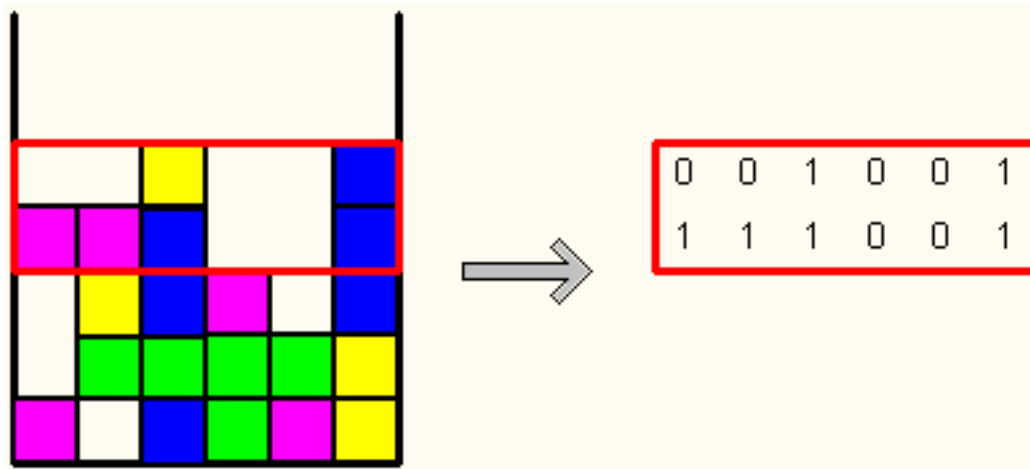


Quelle: [2]

Begrenzung der Konturen

TOP TWO LEVEL

- M: höchste Spalte
- Informationen der Reihen (M-1) und M



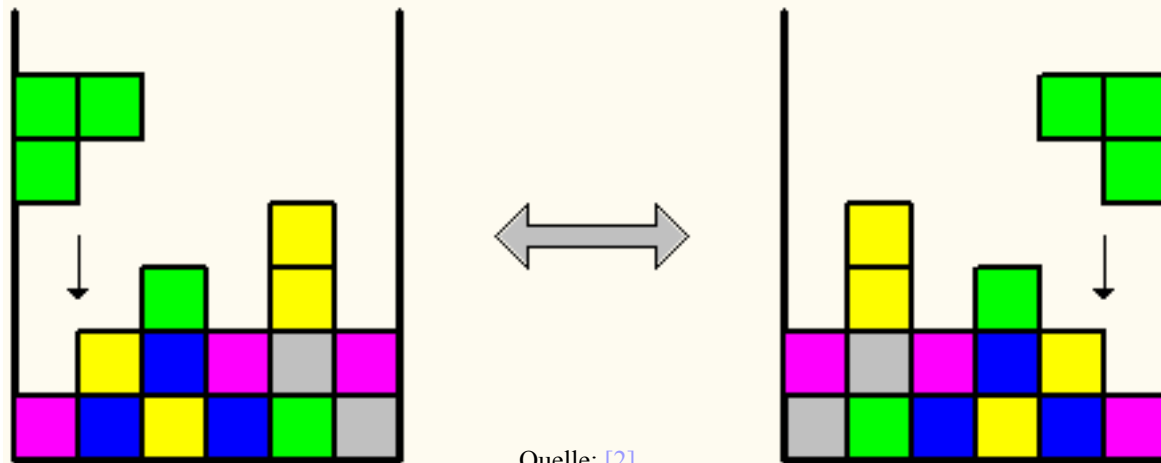
Quelle: [2]

- 4096 Zustände
- Informationen über Löcher

TOP TWO LEVEL

■ Verbesserung

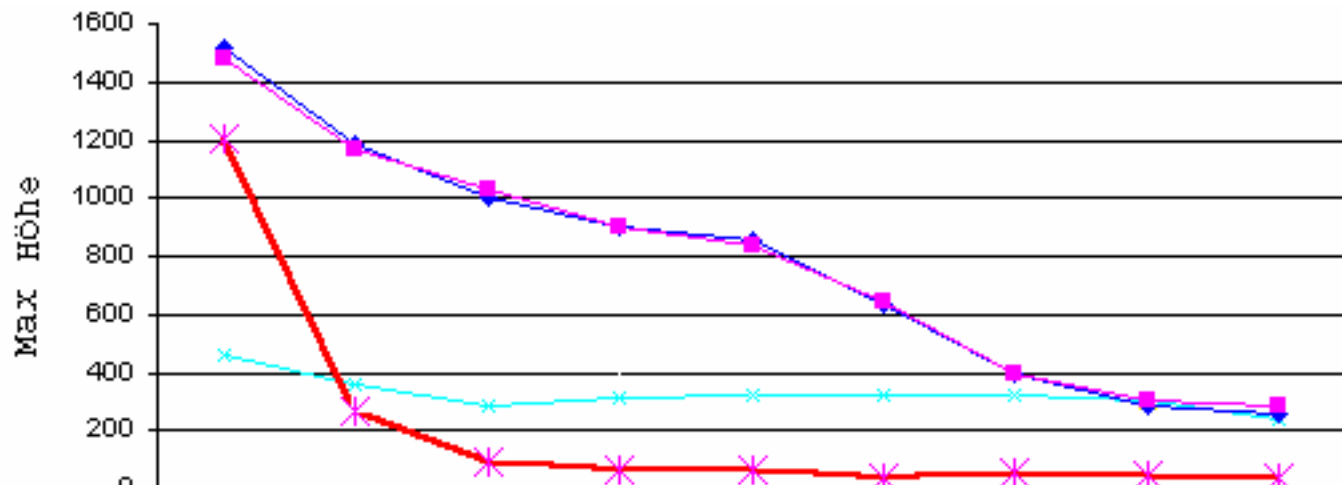
- Zustandsraumverkleinerung: Symmetrie
- 2080 Zustände



■ Ziel: Schnelleres Lernen

Benchmark

■ 1024 Spiele, 10.000 Blöcke



	1	2	4	8	16	32	64	128	256
Contour	456	359	287	317	322	325	325	307	242
Symmetry	1209	268	91	68	61	41	51	42	39
Melax (TD)	1520	1182	1000	903	855	635	394	289	256
Melax (Q)	1485	1166	1032	902	837	644	395	303	289

Quelle: [2]

TU - Darmstadt

Lg(Spiel)

Repräsentation des Zustandsraumes

- Zustandsraum \leftrightarrow Lernen
- Relevante Informationen: Oberfläche
- $\alpha=1/n \rightarrow$ Konvergenz

Relational Reinforcement Learning

■ RL-Problem

- Speicherproblem
- Konvergenz der Q-Funktion

■ Lösung RRL

- Q-Learning + Relationale Repräsentation

Relational Reinforcement Learning

- Exploration des Zustandsraumes
- Q-Funktion Generalisierung
 - Regression Tree
 - Schätzungen für Zustands Aktionspaare
- Anwendung auf andere/ähnliche Situationen

Fazit

- RL-Technik
 - Effizient trainieren & spielen
- Zustandsraumrepräsentation
 - Detaillierungsgrad: Anwendung auf Vollversion
- RRL-Technik
 - Bandbreite der Anwendungsmöglichkeiten

Quellen

- Stan Melax. Reinforcement Learning Tetris Example, 1998. [1]
 - <http://www.melax.com/tetris/>
- Yael Bdolah and Dror Livnat. Reinforcement Learning Playing Tetris, Course Project, Tel Aviv University 2000. [2]
 - http://www.tau.ac.il/~mansour/rl-course/student_proj/livnat/tetris.html
- K. Driessens, and S. Dzeroski, *Integrating guidance into relational reinforcement learning*, Machine Learning **57**, pp. 271-304, 2004. [3]
 - http://www.cs.kuleuven.ac.be/~kurtd/papers/2004_mlj_driessens.pdf
 - http://www.cs.waikato.ac.nz/~kurtd/papers/2001_acai_driessens_chapter.ps.gz
 - http://www.cs.waikato.ac.nz/~kurtd/papers/2005_aic_driessens.pdf
 - http://www.cs.waikato.ac.nz/~kurtd/papers/2004_phd_driessens.pdf
- www.wikipedia.org
- Tetris ist NP-vollständig, Alexander Wiese Uni Stuttgart
- [Breukelaar] Breukelaar, R., Hoogeboom H.J. und Kusters W.A. (2003), Tetris is Hard, Made Easy, Leiden Institute of Advanced Computer Science, Universität Leiden 2003

Danke

? ? ? FRAGEN ? ? ?