

Tetris ist NP-vollständig

Alexander Wiese

12. Mai 2006

Zusammenfassung. In dieser Arbeit wird die NP-vollständigkeit von Tetris bewiesen. Dies wird mit einer Reduktion des Problems 3-PARTITION auf das Tetris-Problem realisiert. Hierbei handelt es sich um eine „offline“ Version von Tetris, bei der das initiale Spielbrett und die Folge der Spielsteine bekannt sind.

1 Einführung

Der in dieser Arbeit beschriebene Beweis stammt von Breukelaar, Hoogeboom und Kusters [Breukelaar]. Wenn nicht anders gekennzeichnet, beziehe ich mich auf dieses Paper.

Zunächst werden in den folgenden beiden Kapiteln die beiden Probleme erläutert, mit denen sich die Reduktion beschäftigt, nämlich das Tetris-Problem (Kapitel 2) und 3-PARTITION (Kapitel 3). Das hier betrachtete Tetris-Problem ähnelt sehr stark dem bekannten Computerspiel, bei dem die Spielsteine herunterfallen und es Aufgabe des Spielers ist, diese so zu platzieren, dass keine Lücken entstehen. Es unterscheidet sich vom Computerspiel vor allem dadurch, dass man vollständige Information hat, d.h. alle Steine sind im voraus bekannt. 3-PARTITION ist ein bekanntes NP-vollständiges Problem, bei dem es darum geht, eine Sequenz von natürlichen Zahlen in Teilmengen einzuteilen, so dass gewisse Eigenschaften erfüllt sind.

Im Kapitel 4 über die Reduktion wird zunächst die Funktion beschrieben, die eine Instanz des 3-PARTITION-Problems auf eine Instanz des Tetris-Problems abbildet. Dann wird gezeigt, dass die Funktion korrekt ist.

2 Offline Tetris

Das Spiel Tetris wird mit folgenden Steinen, auch Tetrominos genannt, auf einem Spielbrett fester Größe gespielt:

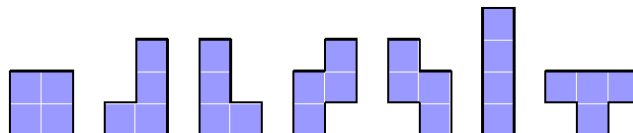


Abbildung 1: Tetrominos

Aus [Breukelaar, S. 1]

Die bekannte Online-Version von Tetris hat wahrscheinlich jeder schon gespielt. Bei der hier betrachteten Offline-Version hat man vollständige Information, d.h. das Spielbrett sowie die komplette Folge von Tetrominos sind im voraus bekannt.

Die einzelnen Tetrominos können durch Translationen und Rotationen bewegt werden. Zu den Translationen zählen Verschiebungen um eine Spalte nach rechts oder links oder um eine Zeile nach unten. Die Steine können um 90° gegen oder mit dem Uhrzeigersinn rotiert werden. Berührt ein Stein einen anderen oder den „Boden“, ist er fixiert und kann nicht mehr bewegt werden. Wie bei der bekannten Online-Version werden dann vollständig besetzte Zeilen entfernt. Das Tetris-Problem lässt sich also folgendermaßen formulieren:

Gegeben Ein initiales Spielbrett und eine endliche Folge von Tetrominos.

Frage Kann das Spielbrett durch Rotation und Translation der Tetrominos (in der gegebenen Reihenfolge) vollständig gelöscht werden?

Eine sehr detaillierte und formale Definition des Tetris-Problems ist in [Demaine, Kapitel 2] gegeben. Offensichtlich ist Tetris in NP. Die Translationen und Rotationen können einfach geraten werden (guess and check).

3 Das 3-Partition-Problem

Um die Reduktion von 3-PARTITION auf Tetris verstehen zu können, sollte man sich zunächst etwas genauer mit 3-PARTITION beschäftigen.

3.1 Definition

Formal ist 3-PARTITION folgendermaßen definiert:

Gegeben Eine Sequenz A von positiven natürlichen Zahlen a_1, \dots, a_{3s} und eine positive natürliche Zahl T , so dass

- $T/4 < a_i < T/2$ für alle $1 \leq i \leq 3s$ und
- $\sum_{i=1}^{3s} a_i = sT$.

Frage 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 $1 \leq j \leq s$.

3.2 Beispiele

Betrachte folgende 3-PARTITION-Instanz: $T = 20$ und $A = \{6,6,6,6,7,7,7,8\}$. s ist hier also 3 und die Summe der a_i beträgt $3 * T = 60$. Wir müssen nun die Werte aus A in drei disjunkte Teilmengen unterteilen, so dass die Summe der Elemente jeder Teilmenge 20 ist. Eine Möglichkeit ist $B_1 = \{6,7,7\}$, $B_2 = \{6,6,8\}$ und $B_3 = \{6,7,7\}$.

Die Instanz $T = 20$ und $A = \{6,6,6,6,6,7,7,9\}$ ist dagegen unlösbar, weil die Teilmenge, die die 9 beinhaltet noch Elemente mit Summe 11 aufnehmen muss, was aber mit 6ern und 7ern unmöglich ist.

Eine allgemeine Besonderheit von 3-PARTITION ist, dass $|B_j| = 3$ für alle j gelten muss, um eine Instanz zu lösen. Dies muss so sein, weil bei $|B_j| < 3$ gilt, dass $\sum_{a_i \in B_j} a_i < 2 * T/2 = T$.

Falls $|B_j| > 3$ folgt, dass $\sum_{a_i \in B_j} a_i > 4 * T/4 = T$ gilt, was in beiden Fällen der Forderung $\sum_{a_i \in B_j} a_i = T$ widerspricht.

4 Die Reduktion

Sei P das Problem 3-PARTITION und Q das Tetris-Problem. Es wird $P \leq_p Q$ gezeigt. Seien Σ und Γ Alphabete. Allgemein gilt für zwei Probleme $A \subseteq \Sigma^*$ und $B \subseteq \Gamma^*$ $A \leq_p B$ genau dann, wenn es eine totale und in polynomieller Zeit berechenbare Funktion $f : \Sigma^* \rightarrow \Gamma^*$ gibt, so dass $a \in A \Leftrightarrow f(a) \in B$. [Schoening, S. 156]

Die Funktion f wird informal im folgenden Abschnitt 4.1 beschrieben.

4.1 Die Reduktionsabbildung

Die Reduktionsabbildung bildet eine 3-PARTITION-Instanz (A, T) auf ein Spielbrett und eine Folge von Tetrominos ab. Das Spielbrett sieht dabei wie in Abbildung 2 aus.

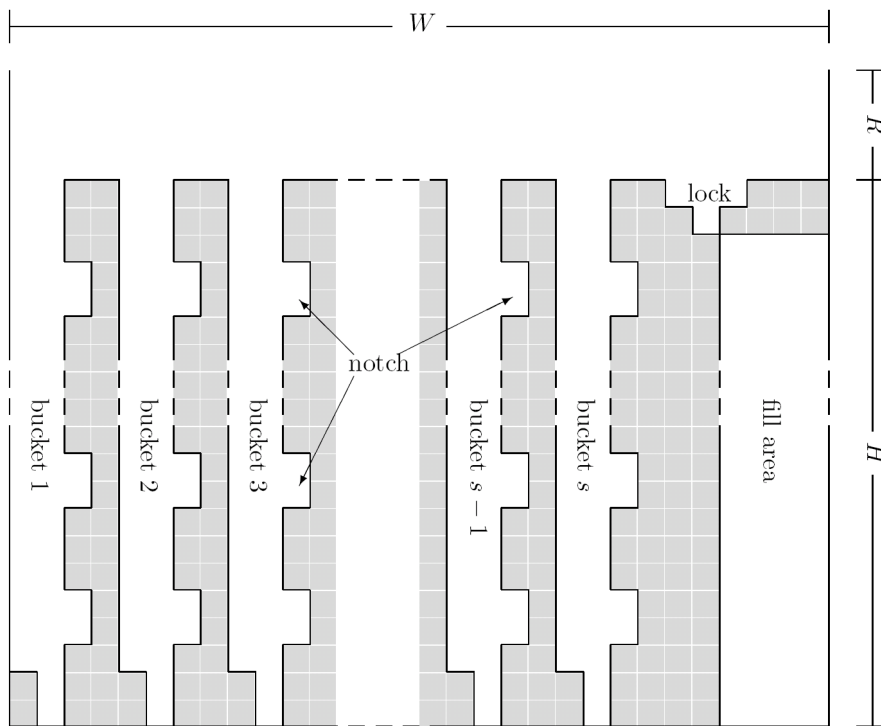


Abbildung 2: Spielbrett

Aus [Breukelaar, S. 3]

Dabei repräsentieren die s buckets die s Teilmengen der 3-PARTITION-Instanz, wie diese mit Steinen gefüllt werden, werden wir später noch sehen.







R ist der Platz der zum Rotieren und Verschieben nach links bzw. rechts benötigt wird und somit von konstanter Größe. Die Breite W des Spielbretts hängt natürlich davon ab, wie viele Elemente die Sequenz A hat. W beträgt $4s+6$, weil ein bucket 4 Spalten einnimmt und 6 Spalten für *fill area* und *lock* benötigt werden. Warum die Höhe H eines buckets $5T + 18$ beträgt bzw. $T + 3$ notches in einem bucket sind, wird später noch deutlich.


Fill area und *lock* sorgen dafür, dass erst Zeilen gelöscht werden, wenn die zwei Zeilen von *lock* gelöscht sind. Die *notches* sind dazu da um sicherzustellen, dass die buckets nur auf eine


Weise gefüllt werden können ohne Lücken zu hinterlassen. Man beachte, dass das Spielbrett in polynomieller Zeit berechenbar ist.


Die Sequenz der Tetrominos sieht folgendermaßen aus:

1. 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

4.2 Eine lösbare Instanz

In diesem Abschnitt wird gezeigt, dass die Reduktionsfunktion eine lösbare 3-PARTITION-Instanz auf eine Tetris-Instanz abbildet, die ebenfalls lösbar ist, d.h. bei der das Spielbrett gelöscht werden kann.

Dazu müssen die *buckets* gefüllt werden. Das Füllen des *bucket*s j mit dem Wert a_i korrespondiert dabei mit dem Aufnehmen des Wertes a_i in B_j . Abbildung 3 zeigt, wie der Wert 3 in einen *bucket* aufgenommen wird.

Auf diese Weise können mehrere Werte aufgenommen werden. Man beachte, dass dabei $a_i + 1$ *notches* gefüllt werden. Jetzt sieht man auch, warum die Höhe eines *bucket*s $5T + 18$ sein muss: Bei einer lösbaren 3-PARTITION-Instanz gilt $\sum_{a_i \in B_j} a_i = T$. Da beim Füllen eines *bucket*s mit der Zahl a immer $a + 1$ *notches* belegt werden, werden $T + 3$ *notches* benötigt, was eine Höhe von $5 * (T + 3)$ ergibt. Es muss noch 3 für den Stein vom ersten „Anfang“ addiert werden, was dann $5T + 18$ ergibt.

Nachdem alle a_i untergebracht sind, kommt auf jeden *bucket* ein  (s. Abbildung 4). Nun ist fast das komplette Spielbrett gefüllt, es fehlen noch *lock* und *fill area*, die leicht mit den noch ausstehenden Steinen gefüllt werden können.

4.3 Eine unlösbare Instanz

Zu zeigen, dass eine unlösbare 3-PARTITION-Instanz auf ein Spielbrett und eine Sequenz von Tetrominos abgebildet wird, so dass das Spielbrett nicht gelöscht werden kann, ist etwas schwieriger. Dazu wird eine ganze Reihe von Lemmas bewiesen. Dabei ist das erste Ziel, zu zeigen,

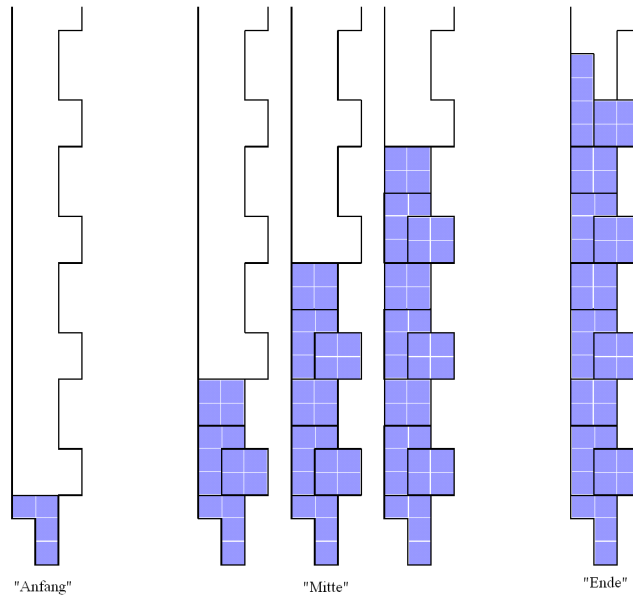


Abbildung 3: Füllen der buckets

Aus [Breukelaar, S. 5]

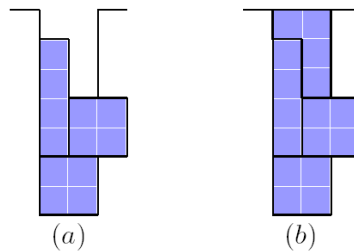


Abbildung 4: Abschließen der *buckets*

Aus [Breukelaar, S. 6]

dass die Steine für einen Wert a_i in genau einem *bucket* untergebracht werden müssen und zwar so, wie in 4.2 beschrieben, sonst kann das Spielbrett nicht gelöscht werden. Darauf aufbauend wird gezeigt, dass ein *bucket* genau drei Werte a_i enthalten muss und die Summe dieser Werte T betragen muss, um das Spielbrett zu löschen. Wenn das gezeigt ist, folgt leicht, dass bei einer unlösbaren Instanz das Spielbrett nicht gelöscht werden kann.

Lemma 1 *Wenn ein Stein oberhalb der $5T + 18$ Zeilen platziert wird, kann das Spielbrett nicht gelöscht werden.*

Wenn ein Stein oberhalb der $5T + 18$ Zeilen platziert wird, müssen mehr als $5T + 18$ Zeilen gelöscht werden. Denn wenn x Zeilen gelöscht werden, fällt alles darüber auch nur x Zeilen nach unten, auch wenn manche Teile weiter fallen könnten. Da alle Steine das gleiche Volumen haben und nur genauso viele Steine in der Sequenz vorhanden sind, um die $5T + 18$ Zeilen zu löschen, kann es nicht sein, dass mehr als $5T + 18$ Zeilen gelöscht werden. Man beachte, dass die Anzahl der Steine unabhängig davon ist, ob eine lösbare oder unlösbare Instanz vorliegt.

Lemma 2 *Um das Spielbrett zu löschen, darf kein anderer Stein als der dafür vorgesehene den Platz bei lock füllen.*

Es gibt keinen anderen Stein, welcher den Platz ausfüllen könnte ohne über die $5T + 18$ Zeilen zu kommen. Also folgt nach Lemma 1, dass dann das Spielbrett nicht gelöscht werden kann.

Lemma 3 *Wenn das Platzieren eines Steins außer dem für lock eine Lücke hinterlässt, die kein anderer Stein durch Translation und Rotation erreichen kann, kann das Spielbrett nicht gelöscht werden.*

Es können erst Zeilen gelöscht werden, wenn der Platz bei lock ausgefüllt wird und dies muss mit dem entsprechenden Stein geschehen (Lemma 2). Wenn aber vorher eine Lücke entsteht, die nicht durch Translation und Rotation eines Steins geschlossen werden kann, führt das dazu, dass irgendwann ein Stein über die $5T + 18$ Zeilen hinausragen muss. Aufgrund von Lemma 1 kann das Spielbrett dann nicht gelöscht werden.

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.*

Wenn der Stein für „Anfang“ in einem bucket platziert ist, haben alle anderen buckets die gleiche Gestalt. In Abbildung 5 werden alle Möglichkeiten gezeigt, die Steine von „Mitte“ und „Ende“ in einem falschen bucket unterzubringen.

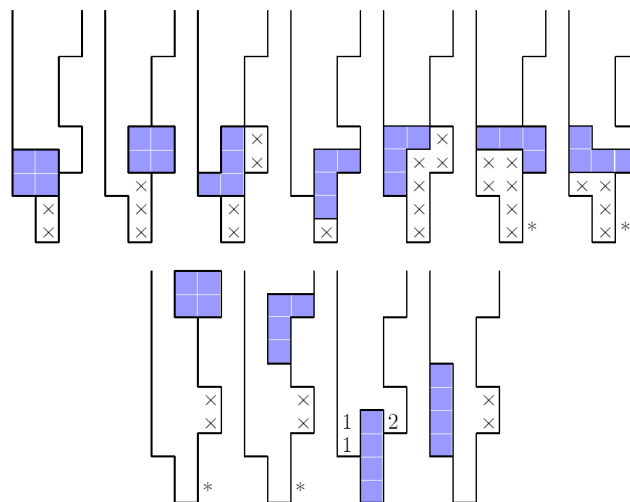
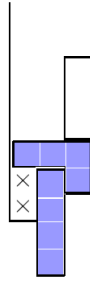


Abbildung 5: Alle Möglichkeiten

Aus [Breukelaar, S. 7]

Dabei wird immer eine Lücke hinterlassen, die nicht durch Translation und Rotation eines Steines erreicht werden kann. Ein x markiert eine solche Lücke. Ein Stern kennzeichnet einen Fall, bei dem der Stein auch höher hätte platziert werden können, dies aber nur zu noch mehr unerreichbaren Lücken führen würde. Der Fall mit den Nummern sollte genauer betrachtet werden. Die einzige Möglichkeit den Platz bei 2 zu erreichen, ist folgende (wobei sich darüber streiten lässt, ob das wirklich möglich ist):



Dann entsteht aber eine Lücke, durch x gekennzeichnet. Wenn aber im ersten Schritt die 1en gefüllt werden, kann die 2 auf obige Weise nicht mehr erreicht werden, wodurch wieder eine Lücke entsteht. Wenn man sich auf den Standpunkt stellt, dass das nicht möglich ist, was in der obigen Abbildung dargestellt ist, folgt auch, dass eine Lücke entsteht, nämlich bei der 2.

Also folgt nach Lemma 3, dass die Steine einer Sequenz von „Anfang“, „Mitte“, „Ende“ für einen Wert a_i in einem bucket untergebracht werden müssen.

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 4.2 beschrieben wurde.*

Wiederum müssen alle Möglichkeiten durchgegangen werden, was in den Abbildungen 6, 7 und 8 dargestellt ist.

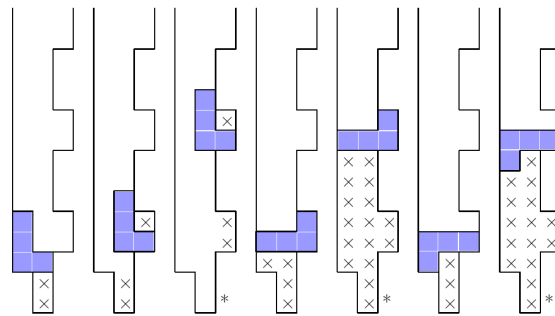


Abbildung 6: Alle Möglichkeiten für „Anfang“

Aus [Breukelaar, S. 8]

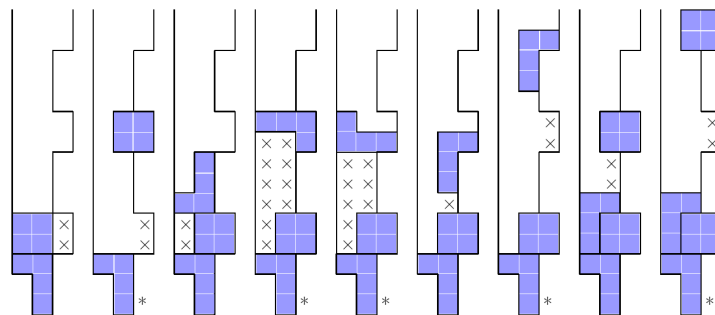


Abbildung 7: Alle Möglichkeiten für „Mitte“

Aus [Breukelaar, S. 8]

Die Bedeutung von x und den Sternen ist die gleiche wie beim vorigen Lemma. Für den Fall mit den Nummern gilt dasselbe wie im vorigen Lemma.

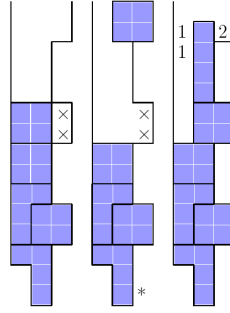


Abbildung 8: Alle Möglichkeiten für „Ende“

Aus [Breukelaar, S. 8]

Die Steine für ein a_i dürfen nur in einen *bucket* kommen (Lemma 4) und alle Möglichkeiten, Steine auf andere Weise als in 4.2 beschrieben in einem *bucket* zu platzieren, führen dazu, dass Lücken hinterlassen werden. Um das Spielbrett zu löschen, muss also so wie in 4.2 vorgegangen werden.

Lemma 6 *Um das Spielbrett zu löschen, muss ein bucket genau drei Werte a_i enthalten und die Summe dieser Werte muss T betragen.*

Um ein a_i in einem *bucket* unterzubringen, müssen $a_i + 1$ *notches* gefüllt werden. Dies folgt aus den Lemmas 4 und 5. Also werden in einem *bucket* genau $\sum_{a_i \in B} a_i + |B|$ *notches* belegt. Um das Spielbrett zu löschen, müssen pro *bucket* $T + 3$ *notches* belegt werden, also gilt $\sum_{a_i \in B} a_i + |B| = T + 3$. Falls $|B| < 3$, folgt dass $\sum_{a_i \in B} a_i > T$ gelten muss. Wenn $|B| > 3$, muss $\sum_{a_i \in B} a_i < T$ gelten. Beides ist aber aufgrund der Einschränkung $T/4 < a_i < T/2$ für alle $1 \leq i \leq 3s$ nicht möglich. Also muss $|B| = 3$ und $\sum_{a_i \in B} a_i = T$ gelten.

Wir haben also in Lemma 6 gesehen, dass jeder der s *buckets* drei Werte aufnehmen muss und die Summer dieser Werte T betragen muss. Dies ist aber nur in einer lösbaren Instanz möglich und deshalb kann bei einer unlösbaren Instanz das Spielbrett nicht gelöscht werden.

5 Schlussfolgerungen

Sei P das 3-PARTITION-Problem, Q das Tetris-Problem und f die Reduktionsabbildung, die in 4.1 informal beschrieben wurde. Es wurde in 4.2 gezeigt, dass $(A, T) \in P \Rightarrow f(A, T) \in Q$ und in 4.3, dass $(A, T) \notin P \Rightarrow f(A, T) \notin Q$ gilt. Daraus folgt, dass $(A, T) \in P \Leftrightarrow f(A, T) \in Q$ gilt und Tetris somit NP-hart ist. Da Tetris auch in NP ist, ist Tetris damit NP-vollständig.

Literatur

- [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
- [Demaine] Demaine, E.D., Hohenberger, S. und Liben-Novell D. (2002) Tetris is Hard, Even to Approximate, Laboratory of Computer Science, Massachusetts Institute of Technology 2002
- [Schoening] Schöning, U. (2001) Theoretische Informatik kurzgefasst, Heidelberg, Berlin 2001