



universität
wien

BACHELORARBEIT

Die NP-Vollständigkeit von Tetris

Verfasser

Michael Tobias König

angestrebter akademischer Grad

Bachelor of Science (BSc.)

Wien, im Monat Dezember 2017

Studienkennzahl lt. Studienblatt: A 033 621

Studienrichtung lt. Studienblatt: Mathematik

Matrikelnummer: 01026100

Betreuerin: Dr. Sandra Uhlenbrock, MSc MSc

Zusammenfassung

In dieser Arbeit wird die NP-Vollständigkeit von Tetris bewiesen. Hierfür werden zuerst die grundlegenden Begriffe zu der Thematik geklärt. Im Nachfolgenden wird das Tetris Problem vorgestellt, welches im Anschluss bewiesen wird. Der Beweis wird mittels einer Reduktion des 3-Partitionen Problems auf Tetris realisiert. Es werde sowohl die Reduktionsfunktion beschrieben, als auch die detaillierten Beweise, einer lösbaren und unlösbaren Instanz, durchgeführt. Neben dem ausführlichen Beweis, werden Beispiele sowie Bilder zum besseren Verständnis der Arbeit verwendet.

Inhaltsverzeichnis

1	Einleitung	1
2	Grundlagen	2
2.1	Das Spiel Tetris	2
2.2	Komplexitätsklassen	4
2.2.1	Deterministische Turingmaschinen	4
2.2.2	Nichtdeterministische Turingmaschinen	6
2.2.3	Die Klassen P, NP und NP-vollständig	7
3	Das Tetris Problem	9
4	Beweis	10
4.1	Beweisidee	10
4.2	Tetris ist in NP	10
4.3	Das 3-Partitionen Problem	11
4.4	Reduktionsfunktion	13
4.5	Lösbare Instanz	15
4.6	Unlösbare Instanz	18
5	Schlussfolgerung	24
	Literatur	25
A	Anhang	26

1 Einleitung

Die Faszination für das Spiel Tetris entwickelte sich bei mir schon in jungen Jahren. Als kleiner Junge habe ich immer gerne Tetris auf meinem Gameboy gespielt. Ziel war es, einen möglichst hohen Highscore zu erreichen und möglichst viele Tetrizen, das sind Sprengungen von vier Reihen gleichzeitig, zu bekommen. Damals wusste ich noch nicht, dass die Faszination für dieses Spiel so lange anhalten würde. Viele Jahre später entdeckte ich, im Laufe meines Studiums, in dem Buch *Das Mathebuch*[5] von Clifford A. Pickover, das Spiel wieder. In diesem Buch sind 250 Meilensteine der Geschichte der Mathematik, von Pythagoras bis in die 57. Dimension, mit bunten Illustrationen, kurz beschrieben. Dies animierte mich, mich mit der Thematik hinter dem Spiel Tetris auseinander zu setzen. Schnell erkannte ich, dass sich dieses Thema gut als Bachelorarbeit eignet. Mittels Recherche war schnell die grundlegende Literatur für die Arbeit gefunden. Die Artikel *Tetris is Hard, Even to Approximate*[2], *Tetris is Hard, Made Easy*[1] und *Tetris ist NP-vollständig*[7] galt es zu verstehen. Da die Thematik auch Informatikkenntnisse benötigt, wurden zusätzlich die Bücher *Einführung in die Theoretische Informatik: Formale Sprachen und Automatentheorie*[4] und *Computers and intractability: a guide to the theory of NP-completeness*[3] verwendet und studiert. Die Informationen zum Spiel Tetris stammen aus dem Buch *Game Over: Nintendo's Battle to Dominate an Industry*[6] sowie persönlichen Erfahrungen. Die folgende Arbeit ist in 5 Abschnitte unterteilt. Im ersten Abschnitt werden einführende Worte sowie das Entstehen der Arbeit beschrieben. Abschnitt zwei befasst sich mit den Grundlagen dieser Arbeit, welche für das Verständnis der nachstehenden Abschnitte benötigt werden. Abschnitt drei stellt das Tetris Problem vor, welches im darauffolgenden vierten Abschnitt bewiesen wird. Diese beiden Abschnitte bilden den Kern der Arbeit. Im letzten Abschnitt werden die Ergebnisse zu einem Großen und Ganzen zusammengeführt und damit die Thematik abgerundet.

2 Grundlagen

Die Bachelorarbeit trägt den Titel „Die NP-Vollständigkeit von Tetris“. Im folgenden Abschnitt werden die beiden Begriffe *Tetris* und *NP-vollständig* sowie einige Grundlagen zu diesen Begrifflichkeiten beschrieben. Dieser Teil der Arbeit bildet die Basis für das grundlegende Verständnis der Thematik. Die entsprechende Literatur ist der Artikel *Tetris is Hard, Even to Approximate*[2] sowie das Buch *Einführung in die Theoretische Informatik: Formale Sprachen und Automatentheorie*[4].

2.1 Das Spiel Tetris

Tetris[6] ist ein Klassiker unter den Computerspielen. Es wurde vom russischen Programmierer Alexey Pajitnov entwickelt. Die erste spielbare Version wurde im Jahr 1984 fertiggestellt. Es handelt sich bei Tetris um ein puzzleartiges Computerspiel, welches nie gewonnen werden kann. Seit seiner Entwicklung wurde Tetris über 100 Millionen Mal verkauft und erfreut sich auch heute noch großer Beliebtheit.

Das Spielprinzip: Im Spiel Tetris geht es darum, mit sogenannten Tetrominos, das sind Spielsteine, die aus 4 Quadraten bestehen, ein rechteckiges Gitter, das Spielfeld \mathbb{S} , zu füllen. Wenn eine Reihe im Spielfeld lückenlos gefüllt ist, wird diese gelöscht und alle Spielsteine, die sich oberhalb dieser Reihe befinden, fallen um die gelöschte Reihe hinab. Ziel von Tetris ist es, möglichst viele Reihen zu löschen und dafür Punkte zu erhalten, um einen Highscore aufzustellen. Sobald ein Tetromino platziert wurde, erscheint ein neuer Spielstein oben in der Mitte des Spielfeldes. Das Spiel kann daher nie gewonnen werden. Abbildung 2.1 zeigt die sieben Tetrominos, welche von links nach rechts mit Q (Quadrat), LL (linkes L), RL (rechtes L), RS (rechte Schlange), LS (linke Schlange), I und T bezeichnet werden.

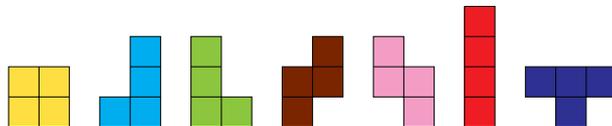


Abbildung 2.1: Tetrominos

Während des Spiels ist es der Spielerin beziehungsweise dem Spieler erlaubt, einige Aktionen auszuführen. Spielsteine dürfen, solange sie noch nicht fixiert sind, in Schritten von $\pm 90^\circ$ rotiert sowie innerhalb des Spielfeldes nach links und rechts bewegt werden. Ein Tetromino wird fixiert, wenn es auf dem Boden des

Spielfeldes oder auf einem schon fixierten Spielstein zu liegen kommt. Hieraus ergibt sich direkt die Notwendigkeit einer Definition bezüglich des Zustandes eines Tetrominos:

Definition: Der Zustand Z eines Tetrominos lässt sich durch ein Quadrupel $Z = \langle t, o, \langle i, j \rangle, w \rangle$ beschreiben, wobei:

- t ist der Typ: $t \in \{Q, LL, RL, RS, LS, I, T\}$ (Quadrat, linkes L, rechtes L, rechte Schlange, linke Schlange, I, T)
- o beschreibt die Orientierung: $o \in \{0^\circ, 90^\circ, 180^\circ, 270^\circ\}$
- $\langle i, j \rangle$ kennzeichnet die Position $\langle i, j \rangle \in \{1, \dots, m\} \times \{1, \dots, n\}$ für ein $m \times n$ Spielfeld
- w heißt der Wert: $w \in \{fixiert, unfixiert\}$

In einem einzelnen Spielzug können unfixierte Tetrominos ihren Zustand $Z = \langle t, o, \langle i, j \rangle, unfixiert \rangle$ ändern. Dafür gibt es folgende Möglichkeiten:

- (i) eine Rotation um $\pm 90^\circ$: $Z \mapsto Z' = \langle t, o \pm 90^\circ, \langle i, j \rangle, unfixiert \rangle$
- (ii) eine Translation um ± 1 : $Z \mapsto Z' = \langle t, o, \langle i, j \pm 1 \rangle, unfixiert \rangle$
- (iii) hinunter fallen um eine Zeile: $Z \mapsto Z' = \langle t, o, \langle i - 1, j \rangle, unfixiert \rangle$
- (iv) fixiert werden: $Z \mapsto Z' = \langle t, o, \langle i, j \rangle, fixiert \rangle$

Fixierte Tetrominos können ihren Zustand $Z = \langle t, o, \langle i, j \rangle, fixiert \rangle$ nur ändern, falls unter ihnen eine Reihe gelöscht wird: $Z \mapsto Z' = \langle t, o, \langle i - 1, j \rangle, fixiert \rangle$. Beim Löschen von Reihen kommt es vor, dass fixierte Tetrominos ihre zusammenhängende Struktur auflösen. Das bedeutet, dass beispielsweise von einem Quadrat Q nur noch zwei Kästchen über bleiben. Befindet sich beim Löschen einer Reihe unter einem solchen Rest eines Tetrominos ein Loch, so fällt der Rest dieses Tetrominos dennoch nur um die gelöschte Reihe herunter. In so einem Fall schwebt dann ein solches Rest-Tetromino quasi in der Luft.

Eine Abfolge mehrerer gültiger Spielzüge Z_1, Z_2, \dots, Z_k für ein Tetromino ergibt eine Trajektorie $T = \langle Z_1, Z_2, \dots, Z_k \rangle$ auf dem Spielfeld \mathbb{S} . Jede solche Trajektorie T beginnt an der Startposition und endet mit dem Zug, bei welchem das Tetromino fixiert wird. Ein komplettes Spiel S entsteht durch eine Abfolge von Trajektorien auf einem gegebenen Spielfeld \mathbb{S} , also $S = \langle \mathbb{S}, T_1, T_2, \dots, T_p \rangle$.

2.2 Komplexitätsklassen

Der Begriff *NP-vollständig* entstammt der theoretischen Informatik. Um diesen Begriff zu verstehen, sind einige weitere Begriffe des Fachbereiches notwendig. NP-vollständig beschreibt eine Komplexitätsklasse der Informatik, in welcher gewisse Probleme liegen. Um Komplexitätsklassen definieren zu können, werden weitere Begriffe, unter anderem die deterministische Turingmaschine und die nichtdeterministische Turingmaschine, benötigt.

2.2.1 Deterministische Turingmaschinen

Eine deterministische Turingmaschine ist ein theoretisches Modell der Informatik, welches aus drei einfachen Dingen besteht, allerdings dennoch sehr mächtig ist. Diese drei Dinge sind:

- Speicher/Ausgabeband: ein unendlich langes Band
- Lese-/Schreibkopf: zum Lesen beziehungsweise Schreiben von Zeichen
- Steuergerät: ein Gerät um den Lese-/Schreibkopf auf dem Band zu bewegen, welches verschiedene Zustände einnehmen kann, insbesondere Anfangs- und Endzustand

Abbildung 2.2¹ zeigt den schematischen Aufbau einer deterministischen Turingmaschine.

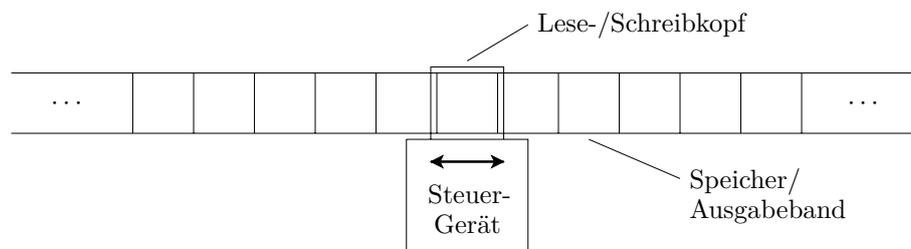


Abbildung 2.2: deterministische Turingmaschine

Die Arbeitsweise einer deterministischen Turingmaschine lässt sich wie folgt beschreiben:

Auf dem, an sonst leeren, Speicher/Ausgabeband steht das zu bearbeitende Problem geschrieben. Die Turingmaschine beginnt mit dem Lesen des Problems auf dem ersten Zeichen des Problems, hierbei befindet sich das Steuergerät im

¹Die Grundlage für diese Abbildung ist ein Bild aus dem Buch *Einführung in die Theoretische Informatik: Formale Sprachen und Automatentheorie*, S.146 Abschnitt 5.1 Arbeitsweise von Turingmaschinen.

Anfangszustand. Anschließend wird eine Reihe von Arbeitsschritten ausgeführt. Die Schritte hängen vom jeweiligen gelesenen Zeichen sowie vom aktuellen Zustand des Steuergerätes ab. In jedem Arbeitsschritt wird eine der drei folgenden Aktionen durchgeführt:

- (i) der Lese-/Schreibkopf bewegt sich am Speicher/Ausgabeband um ein Feld nach rechts
- (ii) der Lese-/Schreibkopf bewegt sich am Speicher/Ausgabeband um ein Feld nach links
- (iii) der Lese-/Schreibkopf überschreibt das gelesene Zeichen mit einem neuen Zeichen

Die Arbeitsschritte werden so lange wiederholt, bis die Turingmaschine den Endzustand erreicht und in diesem stoppt. Zusammengefasst ergibt dies die formale Definition einer deterministischen Turingmaschine:

Definition (deterministische Turingmaschine): Eine deterministische Turingmaschine lässt sich durch ein Quintupel $M = \langle X, Y, h, q_A, q_E \rangle$ beschreiben, wobei:

- X die Zustandsmenge ist, für die gilt $X \neq \emptyset$ und $|X| \in \mathbb{N}$,
- Y das Maschinenalphabet ist, für das gilt $Y \neq \emptyset$ und $|Y| \in \mathbb{N}$, es besteht aus unterscheidbaren Symbolen,
- h Überföhrungsfunktion heißt, für die gilt

$$h : (X \setminus \{q_E\}) \times (Y \cup \{b\}) \rightarrow X \times (Y \cup \{b\} \cup \{L, R\}),$$
 b steht für ein Leerzeichen auf dem Speicher/Ausgabeband,
 L, R stehen für *Links* und *Rechts*,
- q_A ist der Anfangszustand, $q_A \in X$,
- q_E ist der Endzustand, für den gilt $q_E \in X$ und $q_A \neq q_E$.

2.2.2 Nichtdeterministische Turingmaschinen

Da die Arbeitsweise einer nichtdeterministischen Turingmaschine sehr ähnlich ist wie die einer deterministischen Turingmaschine, lässt sich der Unterschied am besten in der Definition sehen.

Definition (nichtdeterministische Turingmaschine): Eine nichtdeterministische Turingmaschine lässt sich durch ein Quintupel $M = \langle X, Y, g, q_A, q_E \rangle$ beschreiben, wobei:

- X, Y, q_A, q_E sind definiert wie bei der deterministischen Turingmaschine,
- g eine Überführungsrelation ist, für die gilt:
 $g \subseteq (X \setminus \{q_E\}) \times (Y \cup \{b\}) \times X \times (Y \cup \{b\} \cup \{L, R\})$.

Der Unterschied in der Arbeitsweise der Turingmaschinen liegt darin, dass es bei einer nichtdeterministischen Turingmaschine für einen Arbeitsschritt mehrere Möglichkeiten gibt, welche gleichzeitig bearbeitet werden können. Am einfachsten kann die Abarbeitung aller möglichen Verläufe eines Problems, welches durch eine nichtdeterministische Turingmaschine gelöst wird, mittels eines Baumes charakterisiert werden. Abbildung 2.3² zeigt eine solche Charakterisierung eines Problems.

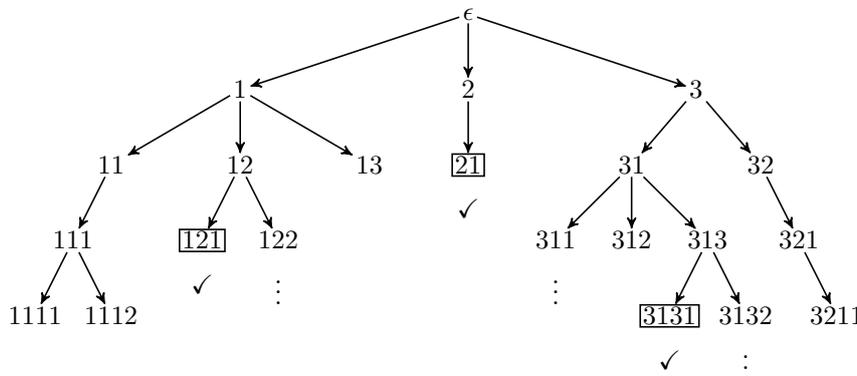


Abbildung 2.3: nichtdeterministische Turingmaschine

Zahlen in diesem Baum beschreiben die verschiedenen Möglichkeiten der Arbeitsschritte einer nichtdeterministischen Turingmaschine. Verzweigungen bedeuten, dass es für das gelesene Zeichen mehrere mögliche Arbeitsschritte gibt, die gleichzeitig ausgeführt werden. Die Wurzel ϵ des Baumes entspricht dem Anfangszustand, die umrahmten Ziffern 121, 21 und 3131 dem einzigen Endzustand der Turingmaschine. Wie in diesem Beispiel zu sehen ist, gibt es drei mögliche

²Ein ähnliches Beispiel wird im Buch *Einführung in die Theoretische Informatik: Formale Sprachen und Automatentheorie*, S.158 Abschnitt 5.4.2 Die Klassen P und NP beschrieben.

Verläufe, wodurch die nichtdeterministische Turingmaschine ihren Endzustand erreicht. Die einzelnen Pfade führen, von der Wurzel ϵ aus, entweder in den einzigen Endzustand, in eine Sackgasse oder sie sind unendlich lang. In Abbildung 2.3 führen die Möglichkeiten 1111, 1112, 13, 312 sowie 3211 in eine Sackgasse. Die vertikalen Punkte bei den Pfaden 122, 311 sowie 3132 deuten an, dass diese unendlich lang sind. Die Möglichkeiten 121, 21 und 3131 führen, wie oben bereits erwähnt, in den einzigen Endzustand der Turingmaschine.

2.2.3 Die Klassen P, NP und NP-vollständig

Man spricht davon, dass *ein Problem* von der Turingmaschine *gelöst wird*, wenn diese nach endlich vielen Schritten im Endzustand stehen bleibt.

Die Zeit ist, bei einer Turingmaschine, mit der Anzahl der Arbeitsschritte gleich zu setzen. Somit können die Laufzeiten der beiden Turingmaschinen wie folgt definiert werden:

Definition: Die Anzahl der Arbeitsschritte einer deterministischen Turingmaschine, die zum Lösen eines Problems benötigt wird, nennt sich die Laufzeit einer deterministischen Turingmaschine.

Bei einer nichtdeterministischen Turingmaschine wird als Laufzeit jene Zeit definiert, die zum Lösen eines minimalen akzeptierten Pfades benötigt wird.

Zur Definition der Komplexitätsklassen der theoretischen Informatik sind folgende Begriffsdefinitionen notwendig:

Definition (Polynom): Ein Polynom ist eine Funktion $p : \mathbb{N} \rightarrow \mathbb{N}$ der Form

$$p(n) = a_k n^k + a_{k-1} n^{k-1} + \dots + a_1 n^1 + a_0$$

mit $k \in \mathbb{N}$, $a_i \in \mathbb{N}$ für $i = 0, 1, 2, \dots, k$. Die Zahlen a_i bezeichnen die Koeffizienten des Polynoms, wobei man das größte i mit $a_i \neq 0$ den Grad des Polynoms nennt.

Ein Beispiel für ein Problem der Länge n ist das Sortieren von n Zahlen. Die Länge eines Problems wird wie folgt definiert:

Definition (Länge eines Problems): Ein Problem hat die Länge n genau dann, wenn es aus n Eingabezeichen auf dem Speicherband besteht.

Ein Problem kann in unterschiedlichen Zeiten gelöst werden. Für die Definition der Komplexitätsklassen P und NP ist folgende Zeit wichtig.

Definition (in polynomialer Zeit lösbar): Eine Turingmaschine löst ein Problem L in *polynomialer Zeit*, wenn es ein Polynom p gibt, so dass das Problem L der Länge n in der Zeit t mit $t \leq p(n)$ gelöst wird.

Mit diesen Voraussetzungen lassen sich die Komplexitätsklassen P und NP definieren.

Definition (Klasse P): Die Komplexitätsklasse P besteht aus all jenen Problemen, die von einer deterministischen Turingmaschine in polynomialer Zeit gelöst werden können.

Definition (Klasse NP): Die Komplexitätsklasse NP umfasst alle Probleme, die von einer nichtdeterministischen Turingmaschine in polynomialer Zeit gelöst werden können.

In weiterer Folge soll der Begriff *NP-vollständig* definiert werden. Um dies zu bewerkstelligen, fehlt lediglich ein letzter Begriff: *Reduktion* beziehungsweise *reduzierbar*.

Definition (polyzeit Reduktion): Seien L_1, L_2 Probleme und $f : L_1 \rightarrow L_2$ eine Funktion für die gilt:

1. f ist berechenbar und total,
2. $\forall \omega : \omega \in L_1 \Leftrightarrow f(\omega) \in L_2$,
3. f ist durch eine deterministische Turingmaschine berechenbar, deren Laufzeit ein Polynom in der Eingabelänge ist.

Wenn so ein f existiert, dann heißt f polyzeit Reduktion. Man sagt, dass L_1 in polynomialer Zeit auf L_2 reduzierbar ist, in Zeichen $L_1 \leq_p L_2$.

Damit kann der Begriff *NP-vollständig* definiert werden:

Definition (NP-vollständig): Ein Problem L nennt sich NP-vollständig, wenn gilt:

- $L \in \text{NP}$,
- $\forall L' \in \text{NP}$ gilt $L' \leq_p L$.

Das heißt, dass jedes Problem L' aus NP in polynomialer Zeit auf das Problem L reduzierbar ist.

3 Das Tetris Problem

Im folgenden Abschnitt wird die Problemstellung vorgestellt mit der sich diese Arbeit auseinandersetzt. Es gibt verschiedene Probleme, die bei Tetris betrachtet werden können. Diese Problemstellungen befassen sich mit der offline Variante des Spiels, bei welcher, im Gegensatz zur online Version von Tetris, vollständige Information vorliegt. Dies bedeutet, dass die endliche Folge von n Stück Tetrominos und das Spielfeld S vorab bekannt sind.

Ein paar klassische Fragestellungen zu offline Tetris, mit einer vorgegebenen Folge von Spielsteinen, sind die folgenden:

- Wie groß ist die maximale Anzahl an gelöschten Reihen, die erreicht werden kann?
- Welches ist die maximale Anzahl an Tetrissen (entspricht vier Reihen auf einmal löschen), die erzielt werden kann?
- Welche maximale Höhe kann während des Spieles, mit einem fixierten Tetromino, erreicht werden?
- Befindet sich nach dem Einsetzen des letzten Tetrominos kein Tetromino mehr auf dem Spielfeld?

Im nächsten Abschnitt wird allerdings keine Lösung einer solchen konkreten Problemstellung vorgestellt, sondern die Einteilung des Tetris Problems in die Komplexitätstheorie der theoretischen Informatik.

Das Tetris Problem, welches untersucht wird, lautet: Ist mit einer vorgegebenen endlichen Folge von Tetrominos ein vorgegebenes endliches Spielfeld lösbar? Das bedeutet, nachdem alle Tetrominos durch Translation und Rotation in dem Spielfeld platziert wurden, soll es am Ende leer sein (siehe oben 4. Fragestellung). Dieses und weitere Tetris Probleme wurden erstmals von Erik D. Demaine, Susan Hohenberger und David Liben-Nowell in ihrer Arbeit *Tetris is Hard, Even to Approximate*[2] im Jahr 2002 als NP-vollständig klassifiziert. Im darauffolgenden Jahr wurde der Beweis zu dem hier beschriebenen Problem von Ron Breukelaar, Hendrik Jan Hoogeboom und Walter A. Kosters in ihrer Arbeit *Tetris is Hard, Made Easy*[1] vereinfacht. 2006 brachte Alexander Wiese mit seiner Arbeit *Tetris ist NP-vollständig*[7] auch eine deutschsprachige Version der Problemstellung heraus.

Im nachstehenden Abschnitt wird ein ausführlicher und leicht verständlicher Beweis geführt, dass das oben angeführte Tetris Problem NP-vollständig ist. Der Beweis orientiert sich an dem Beweis aus *Tetris is Hard, Made Easy*[1].

Satz 3.1: Das Tetris Problem ist NP-vollständig.

4 Beweis

In diesem Abschnitt wird der Beweis der NP-Vollständigkeit des Tetris Problems behandelt. Vorab wird kurz die Beweisidee vorgestellt, damit die Leserin beziehungsweise der Leser einen Überblick über die einzelnen Beweisschritte erhält. Die Artikel *Tetris is Hard, Even to Approximate*[2], *Tetris is Hard, Made Easy*[1] und *Tetris ist NP-vollständig*[7] sowie das Buch *Computers and intractability: a guide to the theory of NP-completeness*[3] sind die für diesen Abschnitt zu Grunde liegende Literatur.

4.1 Beweisidee

Um die NP-Vollständigkeit des Tetris Problems zu beweisen, sind zwei Punkte zu zeigen. Zum Einen muss bewiesen werden, dass das Problem in der Komplexitätsklasse NP liegt. Zum Anderen müssen alle Probleme, welche in NP liegen, in polynomialer Zeit auf das Tetris Problem reduzierbar sein.

Ersteres ist leicht zu sehen und wird im folgenden Abschnitt 4.2 direkt bewiesen. Für den zweiten Punkt wird ein kleiner Trick verwendet. Anstatt alle Probleme aus der Klasse NP zu betrachten und diese auf Tetris zu reduzieren, wird ein Problem P' , welches bereits NP-vollständig ist, ausgewählt und dieses in polynomialer Zeit auf das Tetris Problem reduziert, das heißt $P' \leq_p \text{Tetris}$. Als Repräsentant wird das 3-Partitionen Problem dienen, welches in Abschnitt 4.3 definiert wird. Für das 3-Partitionen Problem wurde die NP-Vollständigkeit im starken Sinne bereits bewiesen (siehe *Computers and intractability: a guide to the theory of NP-completeness*[3], Abschnitt 4.2.2) und somit gezeigt, dass alle Probleme aus NP auf dieses Problem in polynomialer Zeit reduzierbar sind. Um eine Reduktion vom 3-Partitionen Problem auf das Tetris Problem zu zeigen, wird eine Reduktionsfunktion gesucht, welche in polynomialer Zeit jede lösbare Instanz des 3-Partitionen Problems auf ein gelöschtes Spielfeld des Tetris Problems abbildet, sowie jede unlösbare Instanz auf ein nicht leeres Spielfeld. Hierfür werden mehrere Sätze bewiesen, um sämtliche mögliche Anordnungen der Tetrominos zu betrachten.

4.2 Tetris ist in NP

Satz 4.1: Das Tetris Problem ist in der Komplexitätsklasse NP, $\text{Tetris} \in \text{NP}$.

Beweis: Sei ein beliebiges erlaubtes Tetris-Spiel $S = \langle \mathbb{S}, T_1, T_2, \dots, T_n \rangle$ gegeben. Erlaubt bedeutet, dass die Tetrominos ihren Zustand nur nach den in Abschnitt 2.1 beschriebenen Regeln ändern dürfen. Das Spielfeld \mathbb{S} ist dahingehend begrenzt,

dass es in polynomialer Zeit berechenbar ist. Die Rotation der Spielsteine ist auf die vier Orientierungszustände $o \in \{0^\circ, 90^\circ, 180^\circ, 270^\circ\}$ begrenzt. Dies hat zur Folge, dass ein endloses Rotieren der Tetrominos nicht erlaubt ist, beziehungsweise diese Tetrominos immer wieder den gleichen Zustand erreichen. Somit kann jede Trajektorie T_1, T_2, \dots, T_n maximal $4 \cdot |\mathbb{S}| + 1$ Zustände annehmen. Diese maximale Anzahl sind die unfixierten Zustände in jeder Orientierung, an jedem Punkt des Spielfelds, addiert mit dem fixierten Zustand. Hiermit ergibt sich, dass jedes erlaubte Tetris-Spiel mit einer nichtdeterministischen Turingmaschine in polynomialer Zeit berechenbar ist. \square

4.3 Das 3-Partitionen Problem

Im nachstehenden Abschnitt wird das 3-Partitionen Problem definiert, ein Beispiel dazu ausgeführt und der Beweis der NP-Vollständigkeit dieses Problems behandelt. Hiermit ist dann die notwendige Grundlage geschaffen, um den Trick, wie in Abschnitt 4.1 beschrieben, durchzuführen.

Definition (3-Partitionen Problem *3-Part*): Seien (A, T) gegeben, wobei A eine Folge von a_1, a_2, \dots, a_{3s} , mit $a_i \in \mathbb{N} \setminus \{0\}$ für alle $i = 1, 2, \dots, 3s$, $s \in \mathbb{N}$ und $T \in \mathbb{N} \setminus \{0\}$ ist, so dass

- $\frac{T}{4} < a_i < \frac{T}{2}$ für alle $i = 1, 2, \dots, 3s$ und
- $\sum_{i=1}^{3s} a_i = s \cdot T$ gilt.

Frage: Kann A in s disjunkte Folgen B_1, B_2, \dots, B_s unterteilt werden, so dass $\sum_{a_i \in B_j} a_i = T$ für alle $1 \leq j \leq s$ gilt?

Zu beachten ist, dass hier als disjunkte Folge nicht der Wert der a_i zählt, sondern die a_i selbst. Dies bedeutet, dass es mehrere Folgen B_i geben kann, welche die selben Werte der einzelnen a_i beinhalten können, allerdings darf kein a_i in zwei dieser Folgen gleichzeitig vorkommen. Dies wird anhand des nachfolgenden Beispiels veranschaulicht.

Beispiel: Seien $A = (3, 3, 3, 3, 4, 4)$ und $T = 10$ gegeben.

Der Wert $s = 2$ ist durch Abzählen und $s \cdot T = 20$ durch Multiplikation sofort klar. A muss also in zwei disjunkte Teilfolgen, deren Summe $\sum_{a_i \in B_j} a_i = 10$ ist, unterteilt werden. Eine Möglichkeit hierfür ist $B_1 = (3, 3, 4)$ und $B_2 = (3, 3, 4)$ dementsprechend zu wählen. Es gibt weitere mögliche Lösungen für das Problem, doch um die Fragestellung des 3-Partitionen Problems zu beantworten, ist lediglich eine „Ja“ oder „Nein“ Antwort notwendig. Damit ist dies ein Beispiel für eine lösbare Instanz des 3-Partitionen Problems.

Satz 4.2: Das 3-Partitionen Problem *3-Part* ist NP-vollständig, sogar NP-vollständig im starken Sinne.

NP-vollständig im starken Sinne bedeutet, dass nicht nur das zugrunde liegende Problem NP-vollständig ist, sondern auch eine Version des Problems NP-vollständig ist, welche in der Eingabelänge durch ein Polynom begrenzt ist. Beim 3-Partitionen Problem kann die Eingabelänge, wie in Gleichung 4.1 zu sehen ist, gewählt werden, wobei die Funktion *Länge*³ einen Zahlenwert als Ausgabe besitzt. Dieser Zahlenwert ist abhängig von der gewählten Instanz (A, T) des 3-Partitionen Problems.

$$Länge(A, T) = |A| + \sum_{i=1}^{3s} \log_2(a_i) \quad (4.1)$$

Die Funktion *Maximum*³, welche in Gleichung 4.2 zu sehen ist, ordnet jeder Instanz (A, T) von *3-Part* dem größten Zahlenwert der a_i aus A zu.

$$Maximum(A, T) = \max\{a_i : a_i \in A \text{ für } i = 1, 2, \dots, 3s\} \quad (4.2)$$

Das 3-Partitionen Problem ist NP-vollständig im starken Sinne, da das Problem selbst NP-vollständig ist und außerdem jede Problemstellung, deren Eingabelänge durch ein Polynom beschränkt ist, auch NP-vollständig ist. Anders ausgedrückt muss das 3-Partitionen Problem immer noch NP-vollständig sein, wenn man es auf Instanzen (A, T) beschränkt, welche die Gleichung 4.3³ erfüllen, für ein Polynom p .

$$Maximum(A, T) \leq p(Länge(A, T)) \quad (4.3)$$

In dieser Bachelorarbeit wird die NP-Vollständigkeit im starken Sinne von *3-Part* benötigt, da das Tetris Problem auf ein Spielfeld begrenzt wird, welches in polynomialer Zeit aufgebaut werden kann. Daher werden nur Instanzen (A, T) des 3-Partitionen Problems betrachtet, welche die Gleichung 4.3 erfüllen.

Bemerkung: Da der Beweis, dass das 3-Partitionen Problem NP-vollständig im starken Sinne ist, den Umfang dieser Arbeit sprengen würde, wird hierfür auf die Literatur *Computers and intractability: a guide to the theory of NP-completeness*[3] verwiesen. In Abschnitt 4.2.1 *Some Additional Definitions* werden die Funktionen *Länge* und *Maximum* sowie der Zusammenhang zu Gleichung 4.3, genauer beschrieben. Außerdem kann in diesem Abschnitt des Buches der Begriff *NP-vollständig im starken Sinne* genauer nachgelesen werden. Der Beweis, der NP-Vollständigkeit im starken Sinne des 3-Partitionen Problems, wird in

³Als Grundlage für die Gleichungen 4.1, 4.2 und 4.3 dient das Buch *Computers and intractability: a guide to the theory of NP-completeness*[3] Abschnitt 4.2.1 *Some Additional Definitions*.

Abschnitt 4.2.2 *Proving Strong NP-Completeness Results* durchgeführt. Zuvor wird in diesem Abschnitt das 3-Partitionen Problem definiert.

4.4 Reduktionsfunktion

Wie in der Beweisidee (siehe Abschnitt 4.1) beschrieben wurde, wird nun eine Reduktionsfunktion gesucht, welche das 3-Partitionen Problem 3-Part in polynomialer Zeit auf das Tetris Problem abbildet, also bezeugt, dass $3\text{-Part} \leq_p \text{Tetris}$ gilt. Im Folgenden wird die Funktion, die dies erfüllt, definiert. In Abschnitt 4.5 und Abschnitt 4.6 wird bewiesen, dass diese Funktion jede lösbare Instanz (A, T) von 3-Part auf ein gelöschttes Spielfeld in *Tetris* und jede unlösbare Instanz auf ein nicht leeres Spielfeld abbildet. Um dies zu erfüllen, bildet die Funktion jede Instanz (A, T) auf ein Spielfeld S und eine Folge von Tetrominos ab. Das Spielfeld hat dabei eine spezielle Form, wie in Abbildung 4.1 zu sehen ist.

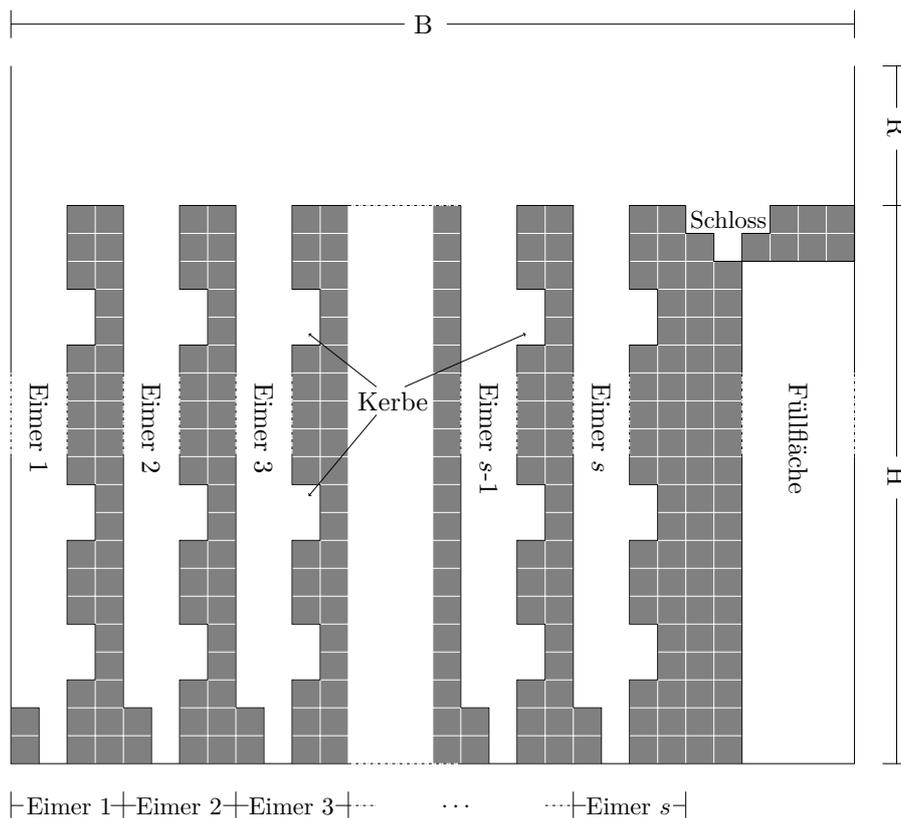


Abbildung 4.1: Spielfeld für die Reduktionsfunktion

Die s Eimer stehen für die s Teilmengen des 3-Partitionen Problems. Die Breite B sowie die Höhe H des Spielfelds variieren mit der Anzahl der Elemente der

Folge A . Da jeder der s Eimer 4 Spalten breit ist und 6 Spalten für die Füllfläche und das Schloss benötigt werden, beträgt die Breite $B = 4 \cdot s + 6$. Die Höhe H ergibt sich aus der Befüllung der Eimer, welche zu einem späteren Zeitpunkt in diesem Abschnitt erklärt wird. Es wird gezeigt werden, dass $H = 5 \cdot T + 18$ ist. Der Platz R dient zur Rotation und Translation der Tetrominos, um diese in die Eimer einzufüllen und kann somit konstant gewählt werden. Das Schloss sowie die Füllfläche dienen dazu, dass erst Zeilen gelöscht werden, wenn das Schloss mit dem richtigen Tetromino , dem Schlüssel, gefüllt wird. Dadurch werden die zwei oberen Zeilen gelöscht und die Füllfläche wird freigegeben, um befüllt zu werden. Die Kerben in den Eimern stellen sicher, dass die Eimer nur auf eine bestimmte Weise gefüllt werden können, ohne dass Freiräume zwischen den Tetrominos entstehen. Wie bei der Höhe H hängt die Anzahl der Kerben von der Anzahl der Elemente der Instanz (A, T) ab. Jeder Eimer hat $T + 3$ Kerben. Zu dieser speziellen Form des Spielfelds S , wird nun die Folge der Tetrominos zu einer beliebigen 3-Partitionen Instanz definiert. Jede Instanz (A, T) wird dabei durch die Reduktionsfunktion folgendermaßen abgebildet:

1. Für jedes $a_i \in A$ wird die nachstehenden Folge („Beginn“, „Mitte“, „Ende“) von Tetrominos erzeugt:

- „Beginn“: 
- a_i -mal „Mitte“: , , 
- „Ende“: , 

2. Anschließend wird s -mal  generiert, um die Eimer zu schließen.

3. Einmal der Schlüssel , um das Schloss zu öffnen.

4. $5 \cdot T + 16$ -mal , zum Auffüllen der Füllfläche und somit Löschen des restlichen Spielfeldes.

Die Reduktionsfunktion wird durch das nachstehende Beispiel veranschaulicht. *Beispiel* (Reduktionsfunktion): Gegeben sei eine lösbare Instanz des 3-Partitionen Problems (A, T) mit $A = (3, 3, 3, 3, 4, 4)$ und $T = 10$. Diese Problemstellung wurde bereits in Abschnitt 4.3 als Beispiel für das 3-Partitionen Problem gelöst. Die gewählte Lösung war $B_1 = (3, 3, 4)$ und $B_2 = (3, 3, 4)$. Die Reduktionsfunktion

bildet diese Instanz auf ein Spielfeld mit Höhe $H = 68$ und Breite $B = 14$ ab. Die bildliche Darstellung der Befüllung des Spielfelds ist in Anhang A, Abbildung A.1 (a) bis (l) zu sehen. Die einzelnen Bilder zeigen Folgendes:

- (a) Das leere Spielfeld, welches für diese Instanz (A, T) erzeugt wird. Da $s = 2$ und $T = 10$ ist, werden 2 Eimer mit jeweils 13 Kerben benötigt, sowie das Schloss und die Füllfläche.
- (b) Jedes $a_i \in A$ erzeugt „Beginn“, a_i -mal „Mitte“ und „Ende“. Hier wird in den ersten Eimer der von $a_1 = 3$ erzeugte „Beginn“ eingefügt.
- (c)-(e) Für $a_1 = 3$ wird der erste Eimer 3-mal mit der „Mitte“ befüllt.
- (f) Es folgt das „Ende“ von a_1 im ersten Eimer.
- (g) $a_2 = 3$ wird in den ersten Eimer eingefüllt.
- (h) $a_3 = 4$ wird in den ersten Eimer eingefüllt.
- (i) Der zweite Eimer wird mit $a_4 = 3$, $a_5 = 3$ und $a_6 = 4$, nach den selben Regeln wie der erste Eimer, befüllt.
- (j) Die zwei Eimer werden jeweils mit einem rechten L  geschlossen.
- (k) Der Schlüssel  wird eingesetzt, um das Schloss zu öffnen und somit die Füllfläche freizugeben.
- (l) Die Füllfläche wird mit 66-mal dem I  aufgefüllt und damit das Spielfeld komplett geleert.

4.5 Lösbare Instanz

Satz 4.3 (Lösbare Instanz): Jede lösbare Instanz des 3-Partitionen Problems 3-Part , wird mit der Reduktionsfunktion auf ein leeres Spielfeld des Tetris Problems $Tetris$ abgebildet.

Beweis: Sei (A, T) eine beliebige lösbare Instanz des 3-Partitionen Problems. Die Reduktionsfunktion erzeugt das zu dieser Instanz gehörige Spielfeld \mathbb{S} , welches in Abbildung 4.1 zu sehen ist. Außerdem wird die Folge der Tetrominos, wie in Abschnitt 4.4 beschrieben, erzeugt. Die Teilfolge „Beginn“, „Mitte“ und „Ende“ der Tetrominos entspricht dabei jeweils den $a_i \in A$. Da (A, T) eine lösbare Instanz des 3-Partitionen Problems ist, lässt sich A in s disjunkte Teilfolgen zerlegen, für die gilt, $\sum_{a_i \in B_j} a_i = T$, für alle $j = 1, 2, \dots, s$. Diese s Teilfolgen entsprechen den s Eimern auf dem Spielfeld \mathbb{S} . Damit dieses Spielfeld lückenlos gefüllt wird, müssen die Tetrominos in die richtigen Eimer gefüllt werden. Befindet sich das Element a_k in der Teilfolge B_j , so müssen die durch die Reduktionsfunktion

für a_k erzeugten Tetrominos in den j -ten Eimer gefüllt werden. Hierzu wird, wie in Abbildung 4.2 (a) bis (c) zu sehen ist, für jedes $a_i \in A$, $i = 1, 2, \dots, 3s$ vorgegangen. Die Abbildung zeigt, wie ein $a_i = 2$ in einen Eimer gefüllt wird. Auf diese Weise können verschiedene Werte der a_i in die Eimer gefüllt werden, da die Form des Eimers, wie in der Abbildung zu sehen ist, nach dem Einfüllen des „Endes“ sowie vor dem Einfüllen des „Beginns“ dieselbe ist.

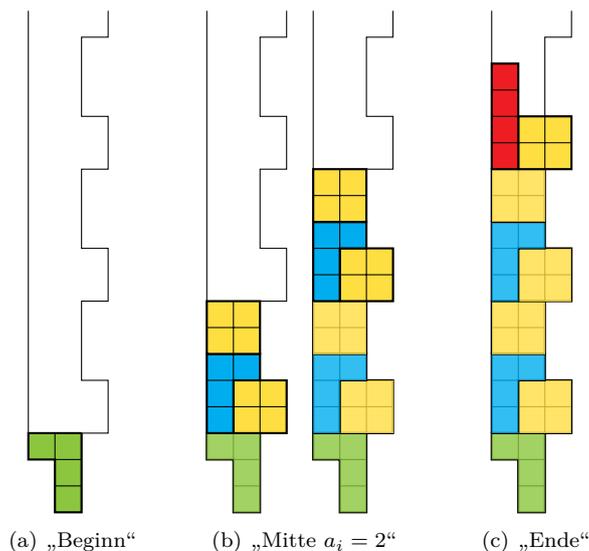


Abbildung 4.2: Füllung eines Eimers

Aus dieser Vorgangsweise ergibt sich folgende Höhe für jeden Wert $a_i \in A$, $i = 1, 2, \dots, 3s$.

$$\begin{aligned} \text{Höhe „Beginn“} + a_i \cdot \text{Höhe „Mitte“} + \text{Höhe „Ende“} &= \\ 3 + a_i \cdot 5 + 4 & \end{aligned} \quad (4.4)$$

Um die Füllhöhe eines beliebigen Eimers zu berechnen, müssen alle Höhen der Tetrominos, die durch die Werte a_i erzeugt werden, welche zu dem entsprechenden Eimer gehören, addiert werden. Hierbei muss auf die diversen Überlappungen der Tetrominos geachtet werden.

$$\begin{aligned} \sum_{a_i \in B_j} (\text{Höhe „Beginn“} + a_i \cdot \text{Höhe „Mitte“} + \text{Höhe „Ende“}) &= \\ = \sum_{a_i \in B_j} (3 + a_i \cdot 5 + 2) & \end{aligned} \quad (4.5)$$

In Gleichung 4.5 wird auf die Überlappung zwischen „Beginn“ und „Ende“ geachtet. Diese wird jeweils zum „Beginn“ gezählt, lediglich das letzte I ■■■■,

welches 2 Gitterhöhen über das „Ende“ herausragt (siehe Abbildung 4.3 (a)), muss noch addiert werden. Hieraus ergibt sich Gleichung 4.6.

$$\sum_{a_i \in B_j} (\text{Höhe „Beginn“} + a_i \cdot \text{Höhe „Mitte“} + \text{Höhe „Ende“}) + \text{Überlappung „Ende“} = \sum_{a_i \in B_j} (5 + a_i \cdot 5) + 2 \quad (4.6)$$

Diese Gleichung kann, da für jede lösbare Instanz des 3-Partitionen Problems $\sum_{a_i \in B_j} a_i = T$ und $|B_j| = 3$ für $1 \leq j \leq s$ gilt, folgendermaßen umgeschrieben werden.

$$\sum_{a_i \in B_j} (5 + a_i \cdot 5) + 2 = 5 \cdot T + 3 \cdot 5 + 2 = 5 \cdot T + 17 \quad (4.7)$$

Die somit errechnete Füllhöhe jedes Eimers ist kleiner als die für das Spielfeld S angegebenen $5 \cdot T + 18$, deswegen ist die Spielfeldhöhe ausreichend.

Nach dem Befüllen jedes Eimers bildet die Reduktionsfunktion noch s -mal ein rechtes L , welches zum Abschließen jedes Eimers dient. Dies ist in Abbildung 4.3 zu sehen, wobei (a) den offenen Eimer und (b) den abgeschlossenen Eimer zeigt.

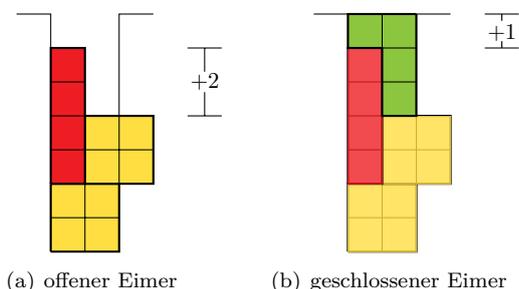


Abbildung 4.3: Abschließen der Eimer

Als Nächstes wird von der Reduktionsfunktion der Schlüssel  erzeugt, um das Schloss zu öffnen. Nachdem dieser korrekt in das Schloss eingesetzt ist, sind die oberen zwei Reihen des Spielfelds gelöscht und die Füllfläche offen. Die $5 \cdot T + 16$ I Tetrominos  werden dazu genutzt, diese zu befüllen. Hierzu werden die Steine horizontal in die Füllfläche gesetzt, um Zeile für Zeile das Spielfeld zu löschen. Nachdem alle Tetrominos, die durch die Reduktionsfunktion erzeugt wurden, eingefüllt sind, ist das Spielfeld leer. \square

4.6 Unlösbare Instanz

Satz 4.4 (Unlösbare Instanz): Jede unlösbare Instanz des 3-Partitionen Problems wird mit der Reduktionsfunktion auf ein nicht leeres Spielfeld des Tetris Problems abgebildet.

Um diesen Satz zu beweisen, werden mehrere Lemmata gezeigt. Die Zusammensetzung all dieser Lemmata beweist schlussendlich den Satz 4.4. Ist dies geschehen, ist der Beweis, dass das Tetris Problem NP-vollständig ist (siehe Satz 3.1), abgeschlossen.

Beweis: Zu Beginn wird gezeigt, dass jedes über der $5 \cdot T + 18$ Linie platzierte Tetromino dazu führt, dass das Spielfeld nicht mehr gelöscht werden kann.

Lemma 4.5: Wann immer ein Tetromino über der $5 \cdot T + 18$ Linie platziert wird, lässt sich das Spielfeld \mathbb{S} nicht leeren.

Beweis: Da Tetrominos all jene Spielsteine sind, welche aus 4 Quadraten bestehen, haben alle denselben Flächeninhalt. Wie bereits in Satz 4.3 bewiesen wurde, entspricht die Fläche aller Tetrominos einer lösbaren Instanz genau der zu füllenden Fläche des Spielfelds, bis zur $5 \cdot T + 18$ Linie. Da die Anzahl der Tetrominos zwischen einer lösbaren und einer unlösbaren Instanz nicht variiert, denn für beide gilt $\sum_{a_i \in A} a_i = s \cdot T$, bleibt auch der Flächeninhalt unverändert. Falls nun ein Spielstein über der $5 \cdot T + 18$ Linie platziert wird, bleibt die entsprechende Fläche unter der Linie frei. Somit ist es nicht möglich, das Spielfeld zu leeren. \square

Bemerkung: Falls ein Tetromino über der $5 \cdot T + 18$ Linie platziert würde, müssten mehr Linien als $5 \cdot T + 18$ gelöscht werden. Hierfür würden zusätzliche Spielsteine benötigt werden, welche nicht zur Verfügung stehen. Außerdem würde ein Loch an jener Stelle, wo das Tetromino fehlt, welches über der $5 \cdot T + 18$ Linie platziert wurde, bleiben. Dies würde zusätzlich das Löschen der Zeilen während des Füllens der Füllfläche beeinträchtigen.

Lemma 4.6: Kein anderes Tetromino, außer der Schlüssel , kann das Schloss öffnen, so dass das Spielfeld noch geleert werden kann.

Beweis: Jedes Tetromino, außer der Schlüssel , ragt über die $5 \cdot T + 18$ Linie heraus, wenn es in das Schloss eingesetzt wird. Dies wird in Abbildung 4.4 (a) bis (d) für alle Tetrominos, die durch die Reduktionsfunktion erzeugt werden, gezeigt. Es handelt sich hierbei um die Tetrominos rechtes L , linkes L , Quadrat  und I . Da all diese Tetrominos über die $5 \cdot T + 18$ Linie heraus ragen, wenn diese als Schlüssel verwendet werden, folgt aus Lemma 4.5 sofort, dass das Spielfeld nicht mehr gelöscht werden kann. \square

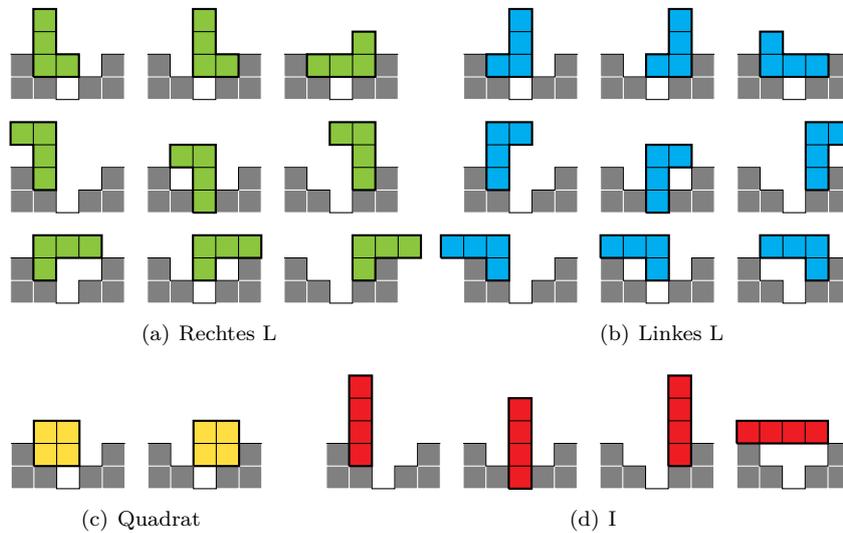


Abbildung 4.4: Schloss mit verschiedenen Tetrominos

Lemma 4.7: Wenn das Einfüllen eines Tetrominos, welches in der Folge der Spielsteine vor dem Schlüssel kommt, einen Freiraum lässt, der nicht durch Rotation oder Translation anderer Tetrominos ausgefüllt werden kann, dann kann das Spielfeld nicht geleert werden.

Beweis: Es können keine Reihen gelöscht werden, bevor der Schlüssel das Schloss öffnet und wegen Lemma 4.6 kann nur das Tetromino  hierfür verwendet werden. Wie schon in Lemma 4.5 erklärt wurde, haben alle Spielsteine zusammen, die in der Folge der Tetrominos vor dem Schlüssel kommen, genau den Flächeninhalt, der benötigt wird, um alle Eimer zu füllen. Entsteht nun ein Freiraum beim Füllen der Eimer, so ragt eine äquivalente Fläche zum Freiraum über die $5 \cdot T + 18$ Linie hinaus. Somit kann, laut Lemma 4.5, das Spielfeld nicht mehr geleert werden. \square

Lemma 4.8: Füllt man ein Tetromino der „Mitte“ oder des „Endes“ eines Wertes a_i in einen anderen Eimer, als in jenen Eimer, in dem der „Beginn“ dieses Wertes gesetzt wurde, so lässt sich das Spielfeld nicht mehr leeren.

Beweis: Der Beweis geht davon aus, dass bis zu dem Einfüllen des rechten L  Tetrominos, alles gemäß der Regeln aus Abschnitt 4.4 eingefüllt wurde. Wann immer das rechte L  eines Wertes a_i als „Beginn“ in einen Eimer eingefüllt wird, sehen sämtliche andere Eimer gleich aus. Abbildung 4.5 zeigt diese Form sowie die neun Möglichkeiten, die Tetrominos der „Mitte“ beziehungsweise des „Endes“ in diese Form einzufüllen. Es handelt sich dabei um das Quadrat , das linke L  und das I . In Abbildung 4.5 steht jedes \times für einen Freiraum, der weder durch Translation noch durch Rotation nachfolgender Tetrominos

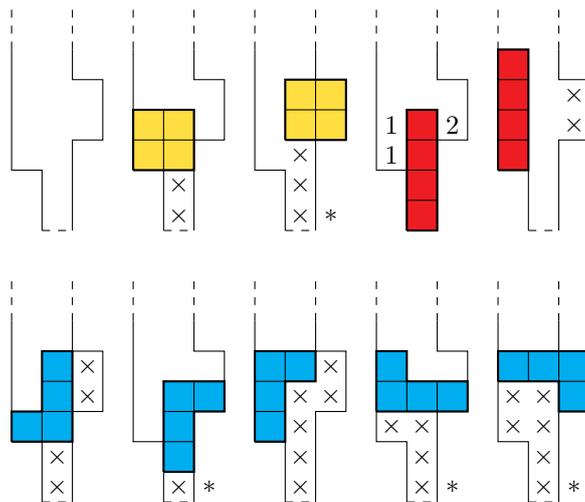


Abbildung 4.5: verschiedene Möglichkeiten der Befüllung

ausgefüllt werden kann. Die Ziffern 1 und 2 stehen für Freiräume, die zwar durch nachfolgende Tetrominos ausgefüllt werden können, wodurch jedoch im Anschluss ein Freiraum entstehen würde. In den mit * gekennzeichneten Bildern der Abbildung 4.5 können die Tetrominos auch in eine beliebige höhere Kerbe gesetzt werden. Dies führt bei Bild 9 und 10 (Reihenfolge gemäß Schreibrichtung) aus Abbildung 4.5 lediglich zu einem noch größeren Freiraum. Bei Bild 3 und 7 würden sich unerreichbare Stellen in den darunterliegenden Kerben ergeben. Aus Lemma 4.7 folgt sofort, dass das Spielfeld nicht mehr geleert werden kann. \square

Zu beachten bei diesem Beweis ist, dass nach jedem Quadrat $\begin{array}{|c|c|} \hline \blacksquare & \blacksquare \\ \hline \blacksquare & \blacksquare \\ \hline \end{array}$ der „Mitte“ ein linkes L $\begin{array}{|c|c|c|} \hline \blacksquare & \blacksquare & \blacksquare \\ \hline \blacksquare & \blacksquare & \blacksquare \\ \hline \end{array}$ und nach jedem linken L $\begin{array}{|c|c|c|} \hline \blacksquare & \blacksquare & \blacksquare \\ \hline \blacksquare & \blacksquare & \blacksquare \\ \hline \end{array}$ wiederum ein Quadrat $\begin{array}{|c|c|} \hline \blacksquare & \blacksquare \\ \hline \blacksquare & \blacksquare \\ \hline \end{array}$ folgt. Außerdem folgt dem Quadrat $\begin{array}{|c|c|} \hline \blacksquare & \blacksquare \\ \hline \blacksquare & \blacksquare \\ \hline \end{array}$ im „Ende“ ein I $\begin{array}{|c|c|c|c|} \hline \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \hline \end{array}$.

Bemerkung: In Abbildung 4.5 wirken die Bilder 9 und 10 unmöglich, doch es kann sein, dass das Spiel Tetris noch eine Rotation zulässt, bevor ein Tetromino fixiert wird. So können diese Möglichkeiten entstehen und somit wurden diese in den Beweis aufgenommen.

Lemma 4.9: Falls die Tetrominos nicht exakt wie in Abschnitt 4.4 beschrieben, in die Eimer gefüllt werden, kann das Spielfeld nicht geleert werden.

Beweis: Bei diesem Beweis werden wiederum alle Möglichkeiten aufgelistet. Angefangen wird mit dem „Beginn“, wie in Abbildung 4.6 zu sehen ist. In Abbildung 4.7 sind die Möglichkeiten für die Tetrominos der „Mitte“ abgebildet, falls der „Beginn“ korrekt eingefüllt wurde. Abbildung 4.8 zeigt sämtliche Möglichkeiten, die Tetrominos I $\begin{array}{|c|c|c|c|} \hline \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \hline \end{array}$ und das Quadrat $\begin{array}{|c|c|} \hline \blacksquare & \blacksquare \\ \hline \blacksquare & \blacksquare \\ \hline \end{array}$, welche zum „Ende“ gehören, einzufüllen.

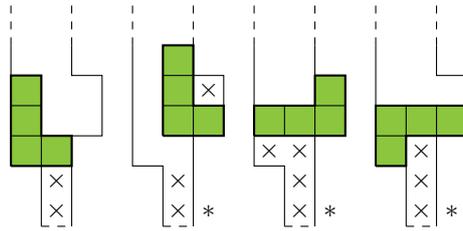


Abbildung 4.6: Möglichkeiten den „Beginn“ einzufüllen

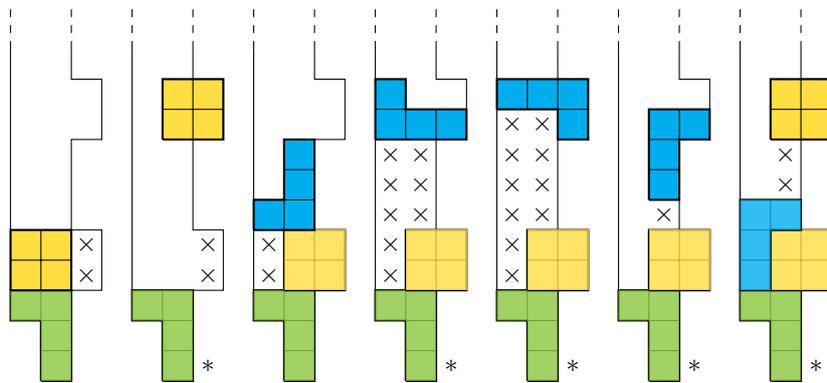


Abbildung 4.7: Möglichkeiten die „Mitte“ einzufüllen

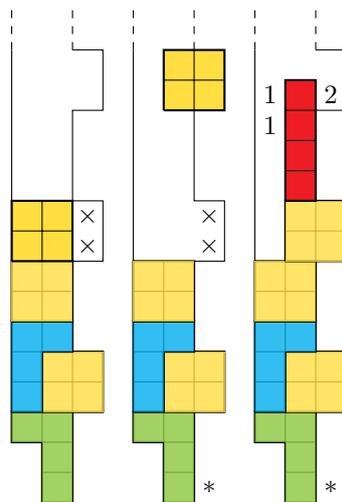


Abbildung 4.8: Möglichkeiten das „Ende“ einzufüllen

In den Abbildungen 4.6 bis 4.8 kennzeichnet ein \times wiederum Freiräume, welche weder durch Translation noch durch Rotation nachfolgender Tetrominos erreicht werden können. In den mit $*$ markierten Bildern können die entsprechenden Spielsteine auch in eine beliebige höhere Kerbe eingesetzt werden, es würden dabei lediglich ähnliche, meist größere Freiräume entstehen. Ziffern weisen auf Freiräume hin, welche zwar durch Rotation beziehungsweise Translation nachfolgender Tetrominos erreicht werden können, allerdings, wenn diese platziert sind, einen Freiraum bilden. Aus Lemma 4.7 und Lemma 4.8 folgt, dass die Tetrominos genau in der Art wie in Abschnitt 4.4 beschrieben wurde, in das Spielfeld eingefüllt werden müssen, sonst kann das Spielfeld nicht geleert werden. \square

Lemma 4.10: Damit das Spielfeld gelöscht werden kann, muss jeder Eimer genau jene Tetrominos aufnehmen, welche durch genau drei Werte $a_i \in A$ gegeben sind.

Beweis: Jeder Eimer hat genau $T + 3$ Kerben. Wie Lemma 4.8 zeigt, müssen die Tetrominos, welche durch die Reduktionsfunktion aus einem Wert a_i erzeugt werden, gemeinsam in einen Eimer gefüllt werden. Aus Lemma 4.9 folgt, dass für das Befüllen nur eine bestimmte Möglichkeit besteht, die in Abschnitt 4.4 beschrieben wurde. Hieraus ergibt sich, dass jeder Wert $a_i \in A$ genau $a_i + 1$ Kerben benötigt, nämlich eine Kerbe für jede „Mitte“ und zusätzlich eine Kerbe für die Kombination aus „Beginn“ und „Ende“. Hieraus folgt, dass die Anzahl der Kerben eines Eimers gleich der Addition der Summe aller Werte des Eimers und der Mächtigkeit von B_j ist, wie Gleichung 4.8 zeigt.

$$\sum_{a_i \in B_j} a_i + |B_j| = T + 3 \quad \text{für } 1 \leq j \leq s \quad (4.8)$$

Um das Spielfeld zu löschen, muss also Gleichung 4.8 erfüllt sein. Angenommen $|B_j| < 3$, so wäre $\sum_{a_i \in B_j} a_i \geq T + 1$ für $1 \leq j \leq s$. Jedoch ist dies nicht möglich, da $a_i < \frac{T}{2}$ für $1 \leq i \leq 3 \cdot s$ gelten muss. Sei andererseits angenommen $|B_j| > 3$, so müsste $\sum_{a_i \in B_j} a_i \leq T - 1$ für $1 \leq j \leq s$ gelten. Dies ist wegen der unteren Schranke $a_i > \frac{T}{4}$ für $1 \leq i \leq 3 \cdot s$ nicht möglich. Es folgt, dass $|B_j| = 3$ für $1 \leq j \leq s$ gelten muss. Mit dieser Erkenntnis kann Gleichung 4.8 zu Gleichung 4.9 umgeschrieben werden, welche der Fragestellung des 3-Partitionen Problems entspricht.

$$\sum_{a_i \in B_j} a_i = T \quad \text{für } 1 \leq j \leq s \quad (4.9)$$

\square

Lemma 4.11: Wird die Reduktionsfunktion auf eine unlösbare Instanz (A, T) des 3-Partitionen Problems *3-Part* angewendet, so kann das Spielfeld \mathbb{S} des Tetris Problems *Tetris* nicht geleert werden.

Beweis: Aus Lemma 4.10 folgt, dass, wenn das Spielfeld geleert wird, jeder Eimer exakt drei Werte $a_i \in A$ aufnehmen muss. Außerdem wurde im Beweis von Lemma 4.10 gezeigt, dass die Summe dieser drei Werte genau T ergibt. Aus der Fragestellung des 3-Partitionen Problems (siehe Abschnitt 4.3) ergibt sich, dass dies nur möglich ist, wenn es sich um eine lösbare Instanz (A, T) handelt. Es folgt, dass das Spielfeld \mathbb{S} bei einer unlösbaren Instanz (A, T) des 3-Partitionen Problems nicht geleert werden kann. □

□

5 Schlussfolgerung

Wie in der Definition der NP-Vollständigkeit aus Abschnitt 2.2 zu lesen ist, muss ein Problem zwei Kriterien erfüllen, um in dieser Klasse zu liegen. Zum einen muss das Problem in NP sein. Zum anderen müssen alle anderen Probleme aus NP auf das neue Problem in polynomialer Zeit reduzierbar sein.

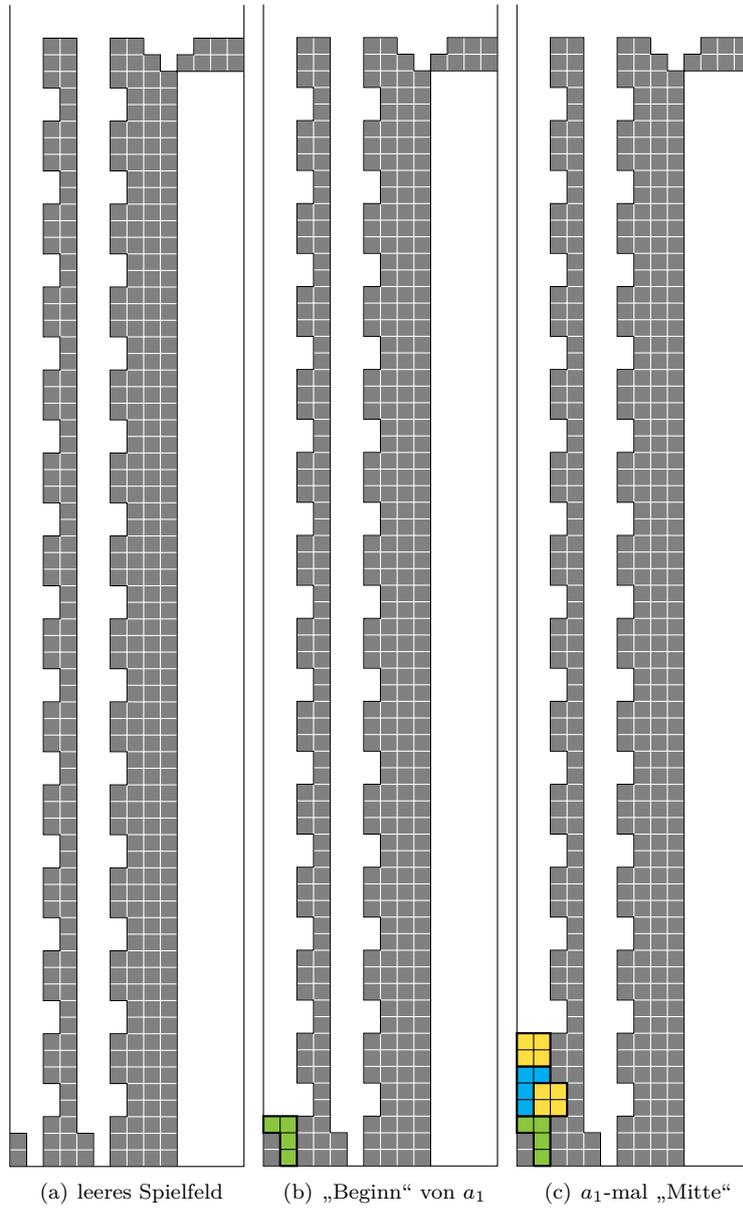
Für das Tetris Problem, welches in Abschnitt 3 beschrieben wurde, treffen beide Bedingungen zu. Dass dieses Problem in NP liegt, zeigt Satz 4.1. Da für das zweite Kriterium ein bereits NP-vollständiges Problem für den Beweis der Reduzierbarkeit verwendet wurde, ist es ausreichend, nur dieses Problem zu betrachten. Hierfür wurde das 3-Partitionen Problem (siehe Abschnitt 4.3) verwendet. Es wurde, mittels Satz 4.3 und Satz 4.4, bewiesen, dass das 3-Partitionen Problem in polynomialer Zeit auf Tetris reduzierbar ist.

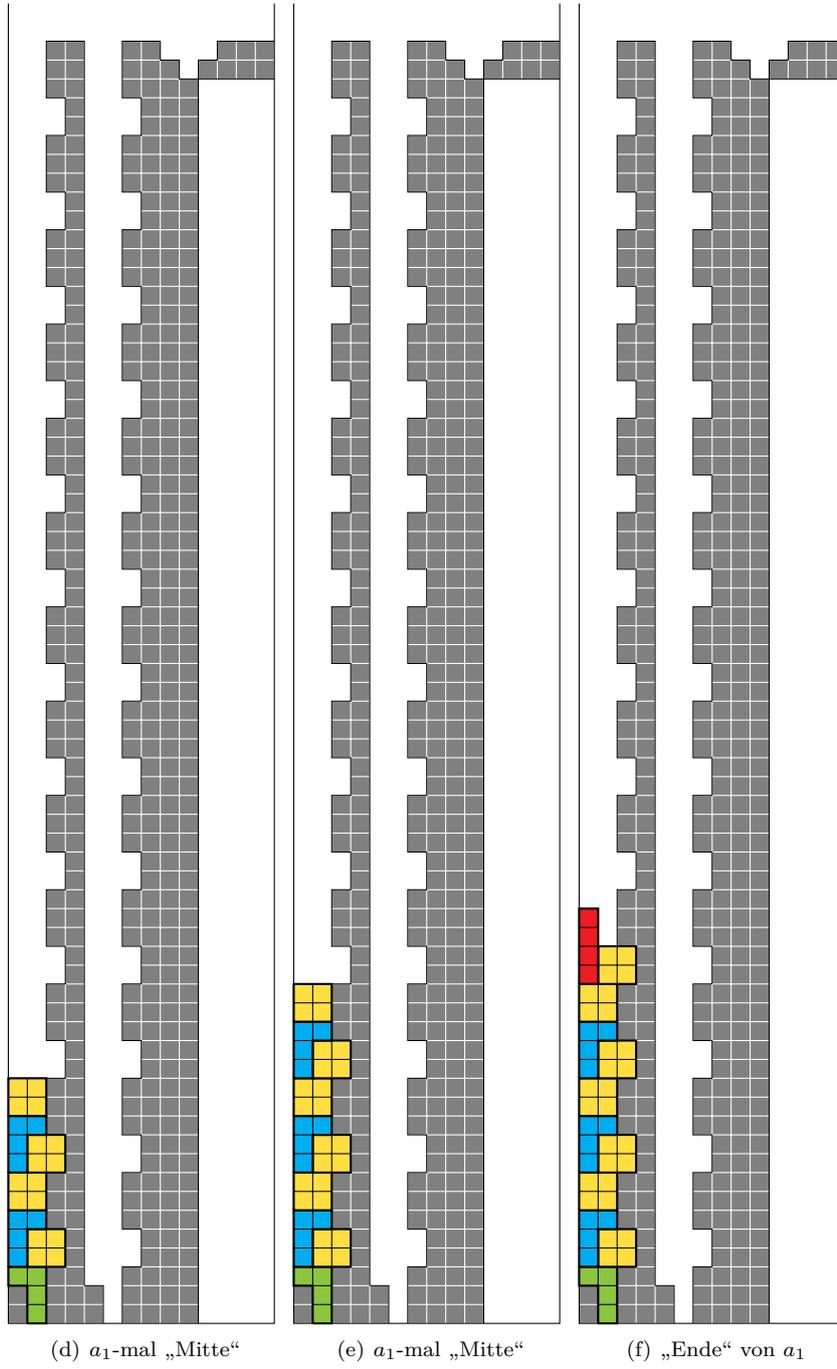
Zusammengefasst konnte die NP-Vollständigkeit von Tetris bewiesen werden.

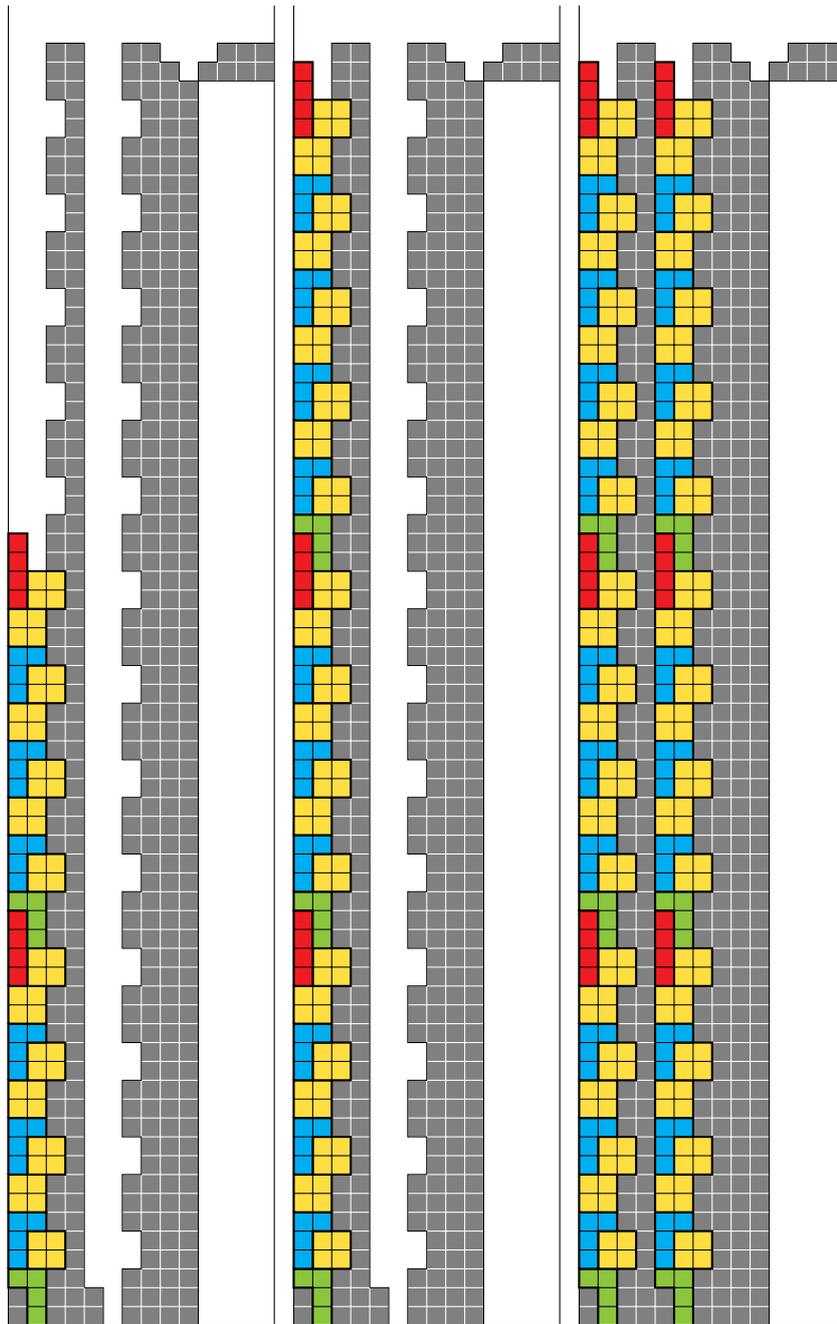
Literatur

- [1] Ron Breukelaar, Hendrik Jan Hoogeboom und Walter A. Kusters. *Tetris is Hard, Made Easy*. In: *Technical report 2003-09*. Leiden Institute of Advanced Computer Science, Universiteit Leiden (2003).
- [2] Erik D. Demaine, Susan Hohenberger und David Liben-Nowell. *Tetris is Hard, Even to Approximate*. In: *Technical Report MIT-LCS-TR-865*. Laboratory of Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139, USA (2002).
- [3] Michael R. Garey und David S. Johnson. *Computers and intractability: a guide to the theory of NP-completeness*. © 1979, 2. [upd. print.] A series of books in the mathematical sciences. San Francisco: Freeman, 1979. ISBN: 071-67-1045-5.
- [4] Prof. Dr. Ulrich Hedtstück. *Einführung in die Theoretische Informatik: Formale Sprachen und Automatentheorie*. Bd. 5., überarbeitete Auflage. Rosenheimer Straße 145, D-81671 München: Oldenbourg Verlag München, 2012. ISBN: 978-3-486-71404-3.
- [5] Clifford A. Pickover. *Das Mathebuch*. Postbus 72, 5330 AB Kerkdriel, Niederlande: Librero IBP, 2014. ISBN: 978-90-8998-280-3.
- [6] David Sheff. *Game Over: Nintendo's Battle to Dominate an Industry*. London: Hodder und Stoughton, 1993. ISBN: 0-340-59-55-74.
- [7] Alexander Wiese. *Tetris ist NP-vollständig*. In: *Universität Stuttgart, Institut für Formale Methoden der Informatik, Seminar im Sommersemester 2006: Spiele in der Informatik* (Mai 2006). URL: <http://www2.informatik.uni-stuttgart.de/fmi/szs/teaching/ss2006/spiele/ausarbeitungen/tetris.pdf> (besucht am 28.11.2017).

A Anhang



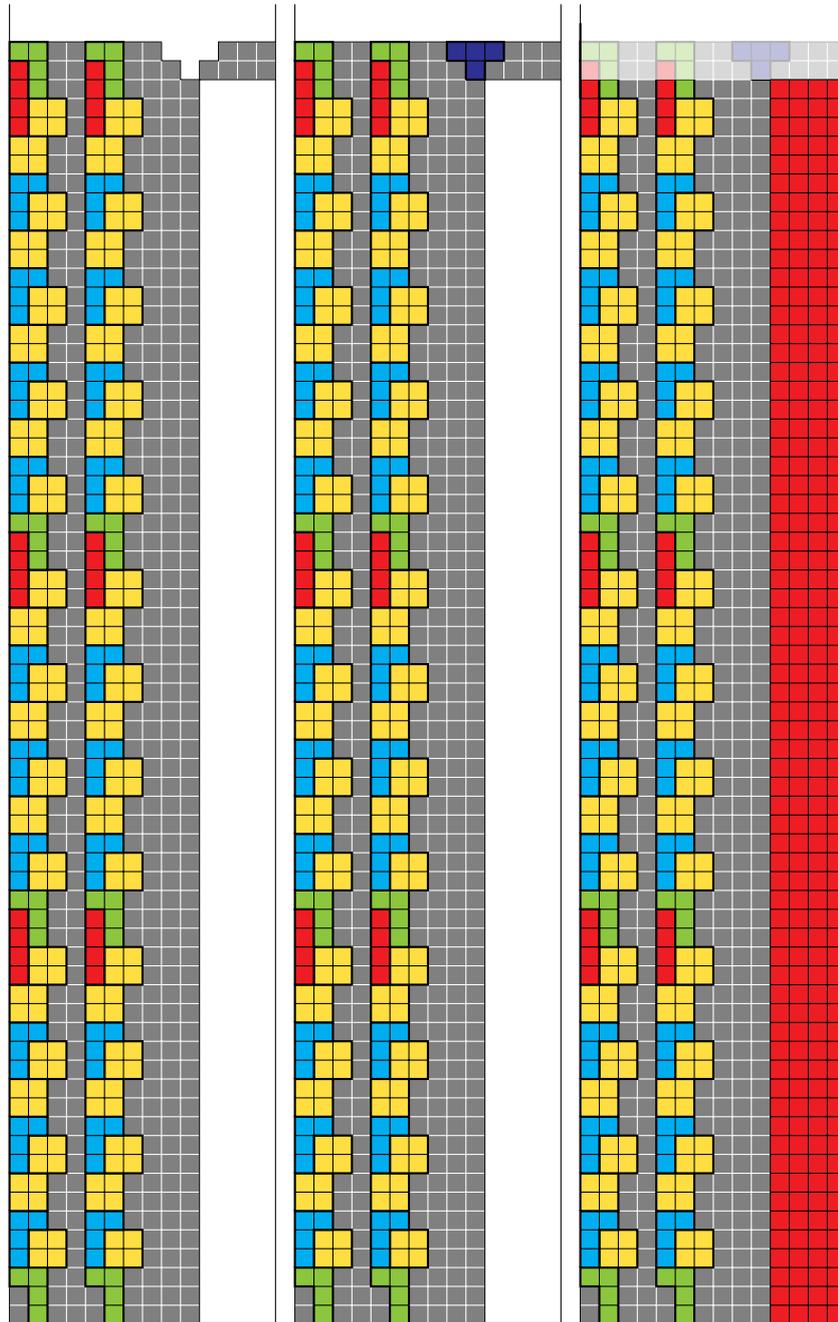




(g) $a_2 = 3$ einfüllen

(h) $a_3 = 4$ einfüllen

(i) Eimer 2 füllen



(j) Eimer schließen

(k) Schloss öffnen

(l) Füllfläche füllen

Abbildung A.1: Füllung des Spielfeldes