Bedienungsanleitung zum σ-Spiel- und Analyseprogramm

Irene Thesing

19. Dezember 2012

In dieser Bedienungsanleitung gehen wir im ersten Abschnitt zunächst auf die notwendigen Definitionen von σ- und Trisentis-Spiel ein, um zu Erklären, wozu das Programm nützlich ist. Im zweiten Abschnitt wird dann die konkrete Bedienung des Programms beschrieben. Das Programm selber gehört aber zu einer Diplomarbeit und mit dem Lesen dieser Arbeit wird sicher auch bei vielen Programmfunktionen noch klarer, weshalb sie hilfreich für die Analyse des σ-Spiels sind. Diese Bedienungsanleitung besteht selbst zu großen Teilen aus Abschnitten der Arbeit.

1 Das σ- und Trisentis-Spiel

Wir führen zunächst in die Grundbegriffe und Basiskonzepte für das σ-Spiel ein und beschreiben das Spiel und interessante Aspekte für die Analyse.

1.1 Graphen

Zunächst wiederholen wir die graphentheoretischen Grundbegriffe.

Definition 1.1. Ein *gerichteter Graph* ist ein Paar $G = (V, E)$ bestehend aus einer Knotenmenge V und einer Kantenmenge $E \subseteq V \times V$.

Definition 1.2. Ein (ungerichteter) *Graph* ist ein gerichteter Graph $G = (V, E)$, für den für alle $u, v \in V$ gilt, dass $(u, v) \in E$ genau dann, wenn $(v, u) \in E$. Außerdem schließen wir Kanten der Form (v, v) aus. In ungerichteten Graphen werden Kanten als zwei-elementige Mengen der Form $\{v, u\}$ geschrieben.

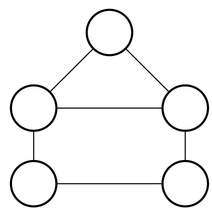


Abbildung 1: Beispielgraph

Definition 1.3. In einem Graphen (V, E) sind die Nachbarn eines Knotens v alle Knoten $u \in V$, sodass $\{v, u\} \in E$. Dann heißen u und v *benachbart*.

1.2 Das σ-Spiel

Wir beginnen mit einer Beschreibung des σ-Spiels auf einem Graphen G. Die Idee ist, dass jeder Knoten zwei Zustände („an“oder „aus“) annehmen kann, dass ein Startspielfeld gegeben ist (zum Beispiel „alle Knoten aus“) und ein Ziel erreicht werden soll (zum Beispiel alle Knoten an). Ein Spielzug auf einen Knoten färbt dabei alle Nachbarn des Knotens um, dass heißt der Zustand der Nachbarknoten ändert sich.

Definition 1.4. Eine *Konfiguration* X auf einem Graphen $G = (V, E)$ ist eine Abbildung $X : V \to \{0, 1\}$. Damit wird jedem Knoten in V eine Farbe 0 (*ungefärbt, ausgeschaltet*) oder 1 (*gefärbt, markiert, eingeschaltet*) zugewiesen. Die Menge aller Konfigurationen bezeichnen wir mit C_G.

Mit 0- (1-) Konfiguration bezeichnen wir die Konfiguration, in der alle Knoten den Wert 0 (1) annehmen.

Definition 1.5. Das *σ-Spiel* auf G ist ein Ein-Personen-Spiel nach folgenden Regeln: Gegeben ist eine Startkonfiguration $X_s \in C_G$ und eine Zielkonfiguration $X_t \in C_G$. Ziel ist es von X_s aus X_t zu erreichen. Ein Zug besteht aus der Auswahl eines Knotens $v \in V$. Er wechselt die Zustände aller benachbarten Knoten aus $\Gamma(v)$ in den jeweils anderen Zustand.

In der Literatur findet sich auch die Bezeichnung *lights-out-game* oder σ^--Spiel für das σ-Spiel.

In Abbildung 2 ist die Konfiguration auf dem Graphen aus dem obigen Beispiel dargestellt, die nach einem Spielzug auf den linken unteren Knoten entsteht, ausgehend von der 0-Konfiguration.

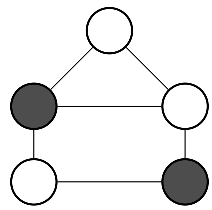


Abbildung 2: Konfiguration nach einem Spielzug auf den linken unteren Knoten

1.3 Lösungen

Definition 1.6. Ein Paar von Konfigurationen (X_s, X_t) auf einem Graphen G heißt *Gewinnpaar (für G)*, falls aus der Startkonfiguration X_S die Zielkonfiguration X_t mittels Spielzügen auf G erreicht werden kann.

Es wird stets von der Startkonfiguration 0 ausgegangen, sodass alle Knoten im Graphen unmarkiert sind. Die gesuchte Zielkonfiguration ist meistens 1.

Definition 1.7. Eine *Lösung* für das σ- oder σ^+-Spiel auf dem Graphen G ist eine Knotenmenge (oder Konfiguration), die bestätigt, dass $(0, 1)$ ein Gewinnpaar für das entsprechende Spiel auf G ist, dass heißt eine Knotenmenge $M \subseteq V$ sodass durch Ziehen (Anwahl) aller Knoten in M der Graph komplett umgefärbt wird.

Eine Lösung des σ-Spiels auf dem Graphen aus Beispiel dem obigen wird durch folgende Zug-Konfiguration in Abbildung 3 gegeben. Durch Anwahl der

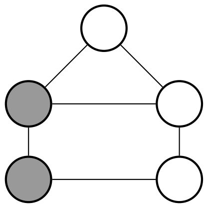


Abbildung 3: Konfiguration, die das σ-Spiel auf dem Graphen löst

beiden markierten Knoten werden in einem komplett unmarkierten Graphen alle Knoten eingefärbt.

1.4 Zusammenhang mit linearen Gleichungssystemen

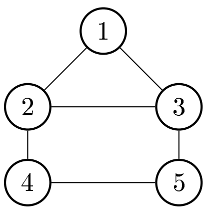
Das σ-Spiel lässt sich durch ein lineares Gleichungssystem über dem Körper mit genau zwei Elementen repräsentieren. Dazu berechnen wir die Adjazenzmatrix eines Graphen.

Definition 1.8. Sei $G = (V, E)$ ein endlicher Graph mit nummerierter Knotenmenge $V = \{1, \ldots, n\}$. Die *Adjazenzmatrix* ist definiert durch $Adj(G) = (a_{i,j})_{(i,j)} \in M_{n\times n}(\mathbb{F}_2)$ wobei gilt

$$a_{i,j} = 1 \quad \Leftrightarrow \quad \{i, j\} \in E.$$

Bei den σ-Spielen nehmen wir stets an, dass kein Knoten mit sich selber verbunden ist, also $a_{i,i} = 0$ für alle i gilt. Da die betrachteten Graphen ungerichtet sind, ist die Adjazenzmatrix stets symmetrisch.

Beispiel 1.9. Zum Verständnis geben wir einen Graphen mit zugehöriger Adjazenzmatrix an.



$$\begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix}$$

Abbildung 4: Graph mit Adjazenzmatrix

Es gilt nun das folgende Lemma zum Zusammenhang zwischen der Lösbarkeit von linearen Gleichungssystemen und der Lösbarkeit der σ-Spiele.

Lemma 1.10. *Ein σ-Spiel auf dem Graphen G ist genau dann lösbar, wenn das lineare Gleichungssystem $Adj(G)X = \mathbb{1}$ eine Lösung besitzt. Dabei bezeichnet $\mathbb{1}$ den Vektor, der nur 1en enthält.*

1.5 Kernelemente

Oft stellt sich auch die Frage: Wie viele Lösungen des σ-Spiels auf einem Graphen gibt es eigentlich? Fragen lässt sich auch nach Auswahlmöglichkeiten, die das Spielfeld nicht verändern. Diese Zugfolgen nennen wir *Kernelemente* des Spiels.

Da das σ-Spiel durch lineare Gleichungssysteme repräsentiert wird und alle Kernelemente den Kern der dadurch repräsentierten Abbildung bilden, gibt die Anzahl der Kernelemente auch an, wie viele Lösungen es gibt. Weiterhin lässt sich eine Basis des Kerns errechnen. Diese kann im Programm repräsentiert werden.

1.6 Das Trisentis-Spiel

Ein spezielles Spiel (für das im Programm eine eigene Oberfläche existiert) ist das σ-Spiel auf Gittern. Dabei gibt es zwei Möglichkeiten der Definition eines Gitters: Einmal über die sogenannte von-Neumann-Nachbarschaft (das heißt ein Knoten hat die vier direkt über die Kanten angrenzenden Nachbarn) und zum anderen die sogenannte Moore-Nachbarschaft, bei der auch die diagonal liegenden Knoten als benachbart gelten, was zu bis zu acht Nachbarn pro Knoten führt.

Mit der Moore-Nachbarschaft ergibt sich das Trisentis-Spiel, auf das wir in einem späteren Kapitel noch näher eingehen werden.

Definition 1.11. Ein $n \times m$-Trisentis-Spiel ist ein Spiel, welches auf einer $n \times m$-Matrix mit zwei Farben gespielt wird. Bei einem Spielzug wird durch Auswahl eines Feldes die Farbe aller benachbarten Felder (auch diagonal) geändert, die des eigenen Feldes nicht. Wir bezeichnen das $n \times m$-Trisentis-Spiel mit $T_{n,m}$ und den entsprechenden Graphen dazu mit $P_n \boxtimes P_m$.

2 Bedienung des Programms

Zum Analysieren des Trisentis-Spiels, sowohl auf rechteckigem Spielfeld als auch auf allgemeinen Graphen, wurde dient das Programm, welches schnell überprüfen kann, ob es eine Lösung gibt und wie groß der Kern ist. Auch das Spielen wird ermöglicht.

Beim Start des Programms wird zunächst gefragt, ob die Oberfläche für Trisentis oder für allgemeine Graphen angezeigt werden soll.

2.1 Oberfläche für allgemeine Graphen

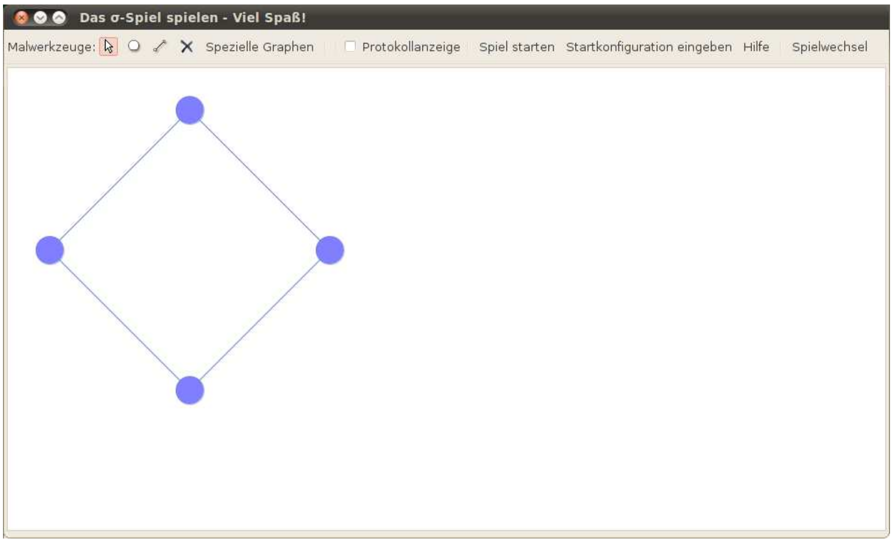


Abbildung 5: Anzeige nach dem Spielstart

Beginnen wir bei den allgemeinen Graphen. Um einen Graphen zu untersuchen, muss dieser zunächst eingezeichnet werden. Dazu befinden sich auf der Werkzeugleiste verschiedene Zeichenwerkzeuge, zum Malen von Knoten und Kanten, Verschieben von Knoten verschieben und Löschen von Objekten. Die jeweiligen Optionen müssen erst angewählt werden und können dann ausgeführt werden, wie bei Zeichenprogrammen üblich. Zudem können einige voreingestellte Typen von Graphen eingefügt werden, die in dieser Arbeit untersucht wurden. Die Anzeige des Programms nach dem Spielstart mit Werkzeugleisten und Startgraphen ist in Abbildung 5 zu sehen.

Ist der Graph fertig eingezeichnet, kann das Spiel mittels „Spielstart“ gestartet werden. Vorher kann ausgewählt werden, ob ein Protokoll mitgeführt werden soll oder nicht. Falls ein Protokoll geführt wird, wird der Bildschirm unterteilt: Auf einer Hälfte können mittels Klicks auf die Knoten Spielzüge ausgeführt werden, auf der anderen werden die gemachten Züge protokolliert. Falls das Spiel mit einer bestimmten Konfiguration gestartet werden soll, ist es möglich mit „Startkonfiguration eingeben“ zunächst eine Startkonfiguration auf dem Graphen zusammen zu klicken.

Nach Start des Spiels erscheint auch eine zweite Leiste, die das Anzeigen von

Lösungen, Kernelementen oder ein paar Fakten über die Dimension des Kerns und die Existenz von Lösungen ermöglicht. Die genaueren Beschreibungen dieser Möglichkeiten werden wir unten anhand der Trisentis-Oberfläche erklären.

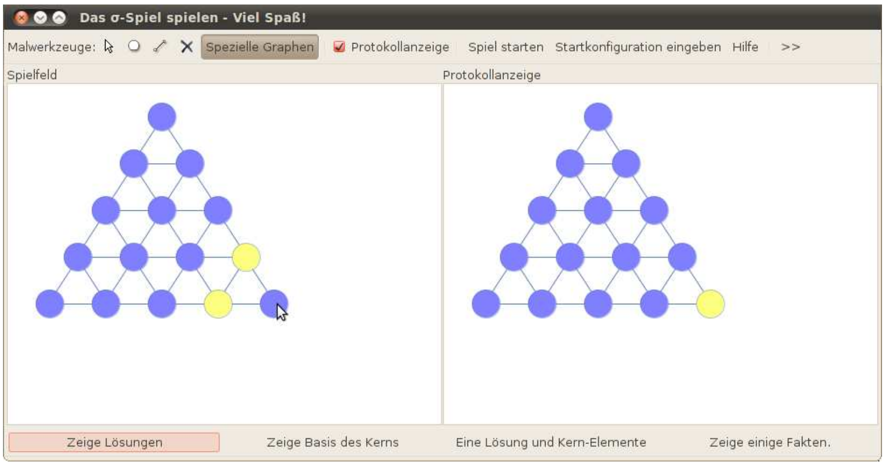


Abbildung 6: Anzeige nach einem Zug

Abbildung 6 zeigt die Programmoberfläche nach Spielstart mit der Anwahl der „Protokollanzeige“-Funktion und einem Spielzug auf den rechten unteren Knoten.

Es gibt außerdem eine Hilfe-Funktion, die das Spiel und die Benutzung des Programms kurz erläutert und es kann auf das Trisentis-Spiel gewechselt werden.

2.2 Oberfläche des Trisentis-Spiels

Kommen wir nun zur Oberfläche des Trisentis-Spiels, die ziemlich ähnlich zur Oberfläche für allgemeine Graphen ist. Dort muss allerdings nicht gezeichnet werden, sondern die Anzahl der gewünschten Zeilen und Spalten eingestellt werden. Es ist auch möglich, die Diagonalnachbarschaften auszustellen, sodass nur Felder als benachbart gelten, die direkt aneinander angrenzen. Danach kann das Spiel analog mit oder ohne Protokollfenster gestartet werden.

2.3 Analysemöglichkeiten

Die Leiste, die nach Spielstart am unteren Spielfeldrand erscheint, dient zur Analyse des Spiels (bei beiden Oberflächen). Es können direkt Lösungen angezeigt werden (falls keine existiert, wird das gemeldet) oder eine Basis der Kernelemente. Alternativ lässt sich auch eine Lösung zusammen mit der Basis des Kerns anzeigen. Daraus können dann Auswahl und Addition alle Lösungen konstruiert werden. Desweiteren ist es möglich, sich einige Eigenschaften des Spiels anzeigen zu lassen. Dabei wird berechnet, wie viele Knoten das Spiel besitzt, wie groß die Dimension des Kerns ist und ob das Spiel eine oder mehrere Lösungen besitzt. Eine Anzeige der Lösungen zum Trisentis-Spiel ist dargestellt in Abbildung 7.

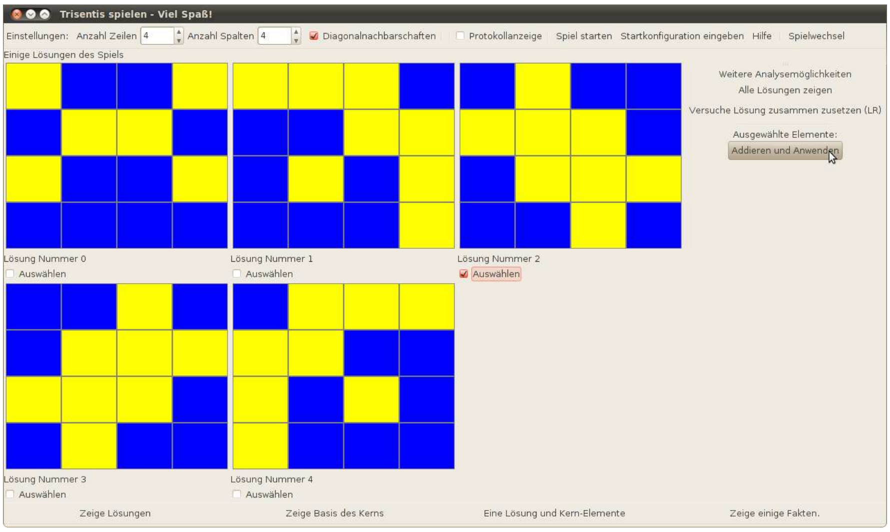


Abbildung 7: Lösungsanzeige für 4x4-Trisentis

Werden Lösungen oder Kernelemente angezeigt, werden gleichzeitig weitere Analysemöglichkeiten bereit gestellt, die von der getroffenen Auswahl abhängen (Kern- oder Lösungselemente, Trisentis-Spiel oder allgemeine Graphen). Beispielsweise ist es möglich, verschiedene Konfigurationen auszuwählen und zu addieren. Das Ergebnis wird in einem neuen Fenster angezeigt. Beim Trisentis-Spiel werden diese Ergebnisse dann zusätzlich auf ein Trisentis-Spiel mit zusätzlichem Rand angewandt. [1] Das soll dabei helfen zu überprüfen, ob zwei Lösungen zusammen gesetzt werden können zu einer Lösung eines größeren Trisentis-Spiels.

Mittels „Versuche Lösungen zusammen zusetzen (LR)“ wird automatisch überprüft, ob für zwei Lösungen des aktuellen $m \times n$-Spiels zu einer Lösung des $m \times (2n+1)$-Spiels zusammen gesetzt werden können, indem sie nebeneinander angewandt werden.

Werden Kernelemente analysiert, können auch verschiedene ausgewählt und auf lineare Unabhängigkeit überprüft werden. Es ist möglich, alle Kernelemente (nicht nur eine Basis und das triviale Kernelement) anzuzeigen. Beim Trisentis-Spiel können auch achsensymmetrische Kernelemente angezeigt werden. Diese spielten bereits in Kapitel 4 eine entscheidende Rolle. Insgesamt gibt es also verschiedenste Anzeigemöglichkeiten für Analyseergebnisse.

[1] Der zusätzliche Rand bedeutet, dass um das Spielfeld herum eine zusätzliche Reihe Felder angefügt wird, so dass das Spielfeld um zwei Spalten und zwei Zeilen wächst. Die Konfiguration wird dabei auf dem mittleren Feld ausgeführt, hat aber auch Auswirkungen auf den Rand. So können eventuell zwei Konfigurationen mit ihren Rändern ineinander geschoben werden und eine neue Lösung entstehen.