

Studiengang: Informatik
Prüfer: Prof. Dr. Ulrich Hertrampf
Betreuer: Prof. Dr. Ulrich Hertrampf

begonnen am: 29. April 2003
beendet am: 30. Juli 2003

CR-Klassifikation: F.2.2, I.2.1

Studienarbeit Nr. 1887

Die Menge der Virtuellen Verbindungen im Spiel Hex ist PSPACE-vollständig

Stefan Kiefer

Fakultät für Informatik, Elektrotechnik und Informationstechnik
Universität Stuttgart
Institut für Formale Methoden der Informatik
Abteilung Theoretische Informatik
Universitätsstraße 38
D-70569 Stuttgart

Inhaltsverzeichnis

1	Einleitung	3
1.1	Das Spiel Hex	3
1.2	Das Programm Hexy und Virtuelle Verbindungen	4
1.3	Die Motivation dieser Studienarbeit	6
2	Ein formaler Rahmen für Hex	7
2.1	Hex-Graphen	7
2.2	Spielverläufe ausgehend von Hex-Graphen	8
3	Definition Virtueller Verbindungen	9
4	Wichtige Eigenschaften Virtueller Verbindungen	10
5	Eine Einschränkung der Operatoren	15
5.1	Die eingeschränkte Rolle der AV-Regel	15
5.2	Informelle Überlegungen zur AND-Regel	17
6	Strategien	19
6.1	Nicht-lokale Strategien	19
6.2	Lokale Strategien	21
6.2.1	Lokalitäten in Spielen	21
6.2.2	Lokale Strategien formal	22
6.3	Direkt-verbindende LS und Virtuelle Verbindungen	24
7	Die PSPACE-Härte von HEXY	30
7.1	Vorwärts-verbindende LS	30
7.2	Die Reduktion von QBF auf HEX	32
7.3	Überblick über die Konstruktion	32
7.4	Überblick über die Gewinnstrategie	33
7.5	Die erste Phase	35
7.5.1	Der Weiße Auswahlgraph	35
7.5.2	Der Schwarze Auswahlgraph	36
7.5.3	Der Treffpunktgraph	38

<i>INHALTSVERZEICHNIS</i>	2
7.5.4 Der Entscheidungsgraph	39
7.5.5 Der Kreuzungsgraph	41
7.5.6 Fazit von Phase 1	42
7.6 Die zweite Phase	42
7.7 Weitere komplexitätstheoretische Überlegungen	46
7.7.1 Sprachen zwischen <i>HEXY</i> und <i>HEX</i>	46
7.7.2 Die Sprache <i>HEX</i> \ <i>HEXY</i>	47
8 Zusammenfassung	48

1 Einleitung

1.1 Das Spiel Hex

Das Brettspiel Hex wurde 1942 vom dänischen Mathematiker P. Hein erfunden. M. Gardner stellte es 1959 in *Scientific American* einer breiteren Öffentlichkeit vor. Seine Regeln sind sehr einfach und können einem Anfänger in einer Minute erklärt werden. Wird Hex aber auf hohem Niveau gespielt, stellt es eine Herausforderung dar sowohl in taktischer als auch in strategischer Hinsicht. Erst kürzlich erschien ein Artikel in *Spektrum der Wissenschaft* [Ste02] über dieses interessante Brettspiel.

Hex wird auf einem rautenförmigen Spielbrett, das aus Sechsecken besteht, gespielt (siehe Abbildung 1).

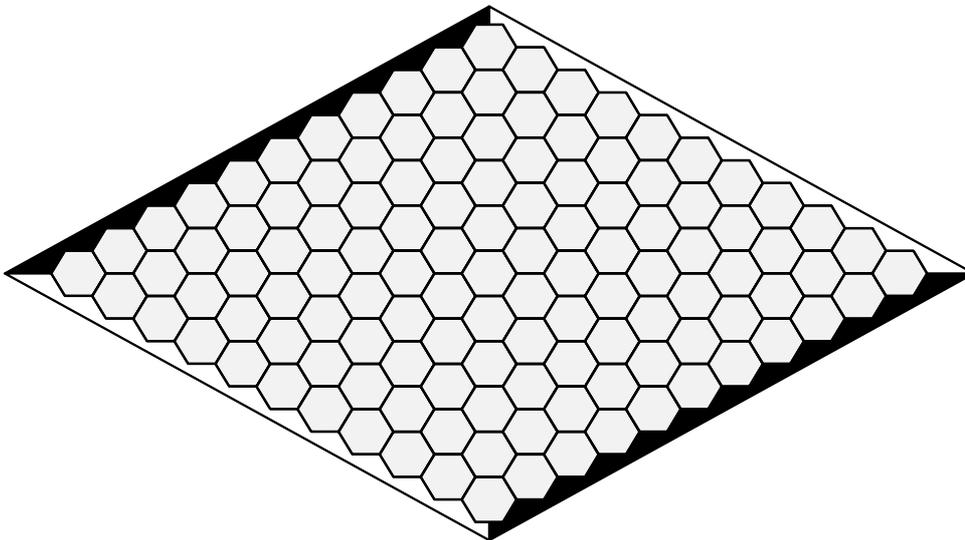


Abbildung 1: Ein Hex-Spielbrett der Größe 11×11

Es gelten folgende Regeln: Zwei Spieler, hier „Weiß“ und „Schwarz“ genannt, belegen abwechselnd eines der sechseckigen Spielfelder mit Steinen ihrer Farbe. Von den vier Seiten des Spielbrettes werden jedem Spieler zwei gegenüberliegende Seiten zugeordnet, wie es in Abbildung 1 durch die Färbung der Ränder angedeutet ist. Ziel jedes Spielers ist es, seine beiden Seiten durch eine Kette von Steinen seiner Farbe zu verbinden (siehe Abbildung 2). Schon liegende Steine dürfen weder fortgenommen noch bewegt werden, auch darf kein Feld zweimal belegt werden. Die Größe des Brettes ($n \times n$) kann variieren; die Größe 11×11 (vgl. Abbildung 1) ist ein gewisser Standard.

Aufgrund dieser einfachen Regeln lassen sich die beiden folgenden grundlegenden Sätze relativ leicht beweisen.

Satz 1.1 *Hex endet nie unentschieden.*

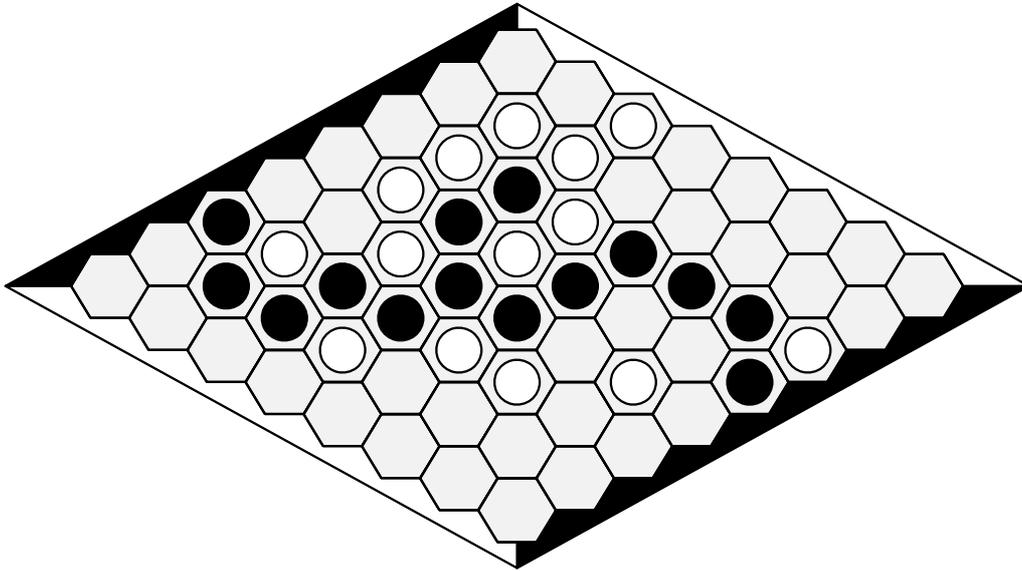


Abbildung 2: Schwarz hat gewonnen.

Satz 1.2 *Auf einem leeren Spielbrett der Größe $n \times n$ hat der beginnende Spieler eine Gewinnstrategie.*

Beweise zu diesen Sätzen finden sich z. B. unter [jav03].

Wir interessieren uns im Folgenden für Hex-Spielsituationen, bei denen das Spiel bereits begonnen hat. Im Unterschied zur Situation bei Satz 1.2 ist es hier i. A. schwierig festzustellen, welcher der beiden Spieler bei optimalem Spiel gewinnen kann.

In [Rei81] wurde 1981 bewiesen, dass die Entscheidung, welcher Spieler bei gegebener Spielsituation eine Gewinnstrategie hat, **PSPACE**-vollständig ist. Damit gilt für Hex ähnliches wie für viele andere bekannte Spiele für zwei Personen (z. B. Go, Dame und Reversi).¹

Hex-Spielsituationen sind ein Spezialfall des sogenannten *Shannon Switching Game*. Die Entscheidung, welcher Spieler bei gegebenem Graphen das Shannon Switching Game gewinnt, ist ebenfalls **PSPACE**-vollständig. Dies wurde bereits einige Jahre vor [Rei81] bewiesen [ET75].

1.2 Das Programm Hexy und Virtuelle Verbindungen

Seine **PSPACE**-Härte macht das Spiel Hex nicht-trivial und interessant für die Künstliche Intelligenz. Das momentan beste Hex-spielende Computerprogramm

¹Man muss natürlich variable Spielbrett-Größen zugrundelegen. Sonst wird das Problem endlich.

heißt *Hexy*. Es ist in einer Windows-Version zum Download erhältlich. Erfreulicherweise hat der Autor (V.V. Anshelevich) einige Prinzipien der Funktionsweise von Hexy in der Zeitschrift *Artificial Intelligence* veröffentlicht [Ans02].

Er betont dabei ein Konzept, das er *Virtuelle Verbindungen* (*virtual connections*) nennt. Die Bedeutung Virtueller Verbindungen ist folgende: Wenn ein Spieler zwei Felder x und y durch eine Virtuelle Verbindung miteinander verbunden hat, besitzt dieser Spieler eine Strategie, diese beiden Felder im weiteren Spielverlauf tatsächlich miteinander zu verbinden, auch wenn der Gegner am Zug ist.

Wenn man auch die beiden Ränder, die dem Spieler zugeordnet sind, als (belegte) Spielfelder auffasst, so bedeutet eine Virtuelle Verbindung der Ränder, dass der Spieler mit der Virtuellen Verbindung eine Gewinnstrategie besitzt. Abbildung 3 (aus [Ans02] entnommen) zeigt beispielhaft eine solche Situation. Die Buchstaben markieren Felder, die Schwarz zum Sieg evtl. benötigt.

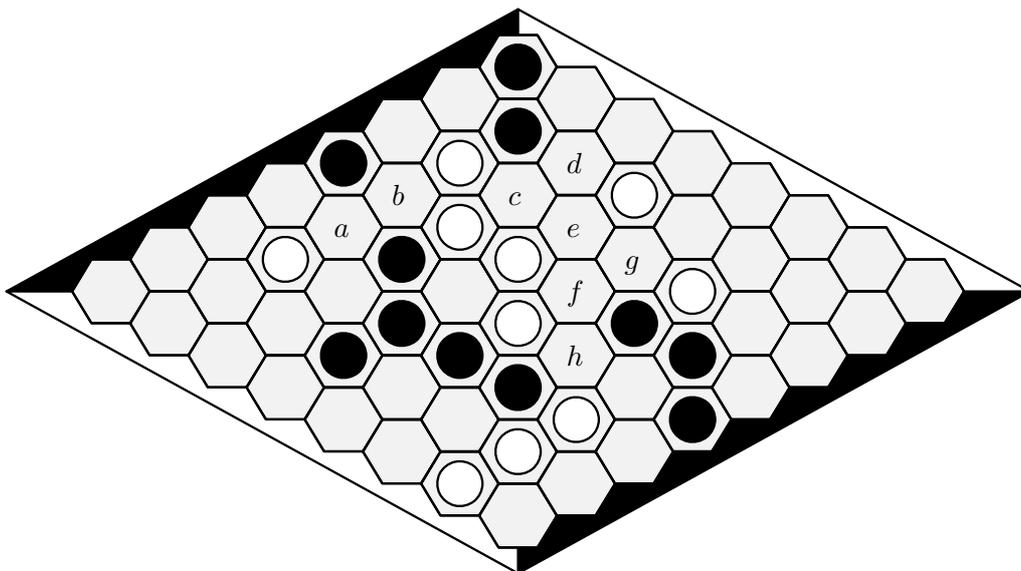


Abbildung 3: Schwarz hat seine Ränder virtuell verbunden.

Das Programm Hexy ist in der Lage, Virtuelle Verbindungen zu berechnen. Entdeckt es eine Virtuelle Verbindung seiner beiden Ränder, so verfolgt es eine Gewinnstrategie. Aber auch ohne eine solche *winning virtual connection* kann das Auffinden Virtueller Verbindungen hilfreich für die Bewertung von Spielsituationen und damit für Standard-Techniken wie α - β -Suche sein (siehe [Ans02]).

Zur Berechnung Virtueller Verbindungen setzt das Programm Hexy aus „kleineren“ Virtueller Verbindungen „größere“ mittels Ableitungsregeln (*AND-/OR-deduction rules*) zusammen. Die „kleinsten“ Virtueller Verbindungen bestehen zwischen direkt benachbarten Spielfeldern. Wir werden in dieser Studienarbeit

Virtuelle Verbindungen und ihre „Zusammenbau-Regeln“ genau beschreiben und definieren.

Es ist notwendig, an dieser Stelle eine Begriffsbestimmung vorzunehmen. In der Sprechweise von Anshelevich hat ein Spieler eine Virtuelle Verbindung zwischen zwei Feldern *genau dann*, wenn er im weiteren Spielverlauf diese beiden Felder verbinden kann. In unserer Sprechweise aber ist eine Virtuelle Verbindung nur das, was mittels der Ableitungsregeln aus den kleinsten Virtuellen Verbindungen (den direkten Nachbarschaften) abgeleitet werden kann. Wir betrachten also nur diejenigen Virtuellen Verbindungen als solche, die das Programm Hexy berechnet.² Es stellt sich heraus, dass diese Definitionen nicht äquivalent sind.

1.3 Die Motivation dieser Studienarbeit

Ein Algorithmus, der Virtuelle Verbindungen berechnet, wird in [Ans02] *H-search* (*H-Suche*) genannt. Es wird auch ein Algorithmus angegeben, der eine erschöpfende H-Suche durchführt, der also alle Virtuellen Verbindungen in unserem (engeren) Sinn in einer Spielsituation berechnet.

Wären Virtuelle Verbindungen im engeren Sinn und Virtuelle Verbindungen im weiteren Sinn identisch, könnte eine solche erschöpfende H-Suche nach allgemeiner Ansicht nicht in polynomieller Zeit durchgeführt werden. Denn mit der Erkennung von allgemeinen Virtuellen Verbindungen ist insbesondere auch die Feststellung des Siegers bei optimalem Spiel verbunden. Dies ist jedoch – wie schon erwähnt – **PSPACE**-hart.

Es ist allerdings bekannt, dass Virtuelle Verbindungen im weiteren und im engeren Sinn verschieden sind. Daher wäre es möglich, dass erschöpfende H-Suche „leichter“ ist als die Feststellung des Siegers.

In [Ans02] wird die Laufzeit des Algorithmus zur H-Suche nicht allgemein untersucht. Es wird aber für eine bestimmte Menge von Hex-Graphen (die eine bestimmte Menge von Spielsituationen repräsentiert) gezeigt, dass H-Suche auf diesen Graphen in polynomieller Zeit möglich ist. Andererseits sind nach [Ans02] beim Programm Hexy aus Zeitgründen Heuristiken erforderlich, die die H-Suche hinsichtlich bestimmter Kriterien einschränken.

In dieser Studienarbeit soll die Komplexität der Erkennung Virtueller Verbindungen geklärt und dadurch eine gewisse Lücke in [Ans02] geschlossen werden. Dies ist insbesondere auch deshalb von Interesse, weil die praktische Bedeutung Virtueller Verbindungen beim Computer-Hex durch die Überlegenheit des Programms Hexy erwiesen ist.

²Dies ist auch die Sprechweise in anderer Sekundärliteratur. Das Problem rührt daher, dass Anshelevich gleichzeitig Virtuelle Verbindungen und die Ableitungsregeln erfunden hat.

2 Ein formaler Rahmen für Hex

2.1 Hex-Graphen

Bevor wir erklären, was wir unter *Virtuellen Verbindungen* verstehen, definieren wir mithilfe von *Hex-Graphen* eine geeignete Repräsentation von Hex-Spielsituationen. Dabei stellt ein Hex-Graph eine Spielsituation aus der Sicht *eines* der beiden Spieler (im Folgenden *eigener Spieler* genannt) dar.

Definition 2.1 (Hex-Graph) *Ein Hex-Graph G ist ein Tripel (V, E, β) , wobei gilt:*

- V ist eine endliche Menge von Spielfeldern,
- $E \subseteq \{\{x, y\} \mid x, y \in V, x \neq y\}$ ist die Nachbarschaftsmenge,
- $\beta : V \rightarrow \{f, b\}$ ist die Besetzungsfunktion mit

$$\{x, y\} \in E \implies \beta(x) = f \vee \beta(y) = f.$$

Die Nachbarschaftsmenge E gibt an, welche Paare von Spielfeldern benachbart sind. Die Besetzungsfunktion β gibt an, ob ein Spielfeld x frei ist ($\beta(x) = f$) oder mit einem Stein der *eigenen* Farbe (d. h. der Farbe des eigenen Spielers) besetzt ist ($\beta(x) = b$). Dabei verlangen wir, dass zwei benachbarte Spielfelder nicht beide besetzt sind. Dies lässt sich erreichen, indem zwei oder mehr benachbarte besetzte Spielfelder zu *einem* besetzten Spielfeld verschmolzen werden. Dabei übernimmt dieses Spielfeld alle äußeren Nachbarschaften (d. h. Nachbarschaften zu freien Feldern), die vorher mit einem der zusammengefassten Spielfelder bestanden.³

Man beachte, dass Felder, die vom anderen Spieler (im Folgenden *Gegner* genannt) besetzt sind, nicht dargestellt werden. Dies lässt sich erreichen, indem solche Felder mitsamt ihren Nachbarschaften entfernt werden.

Es kann also jede Hex-Spielsituation auf diese Weise repräsentiert werden. Insbesondere lassen sich die beiden Ränder der eigenen Farbe als besetzte Felder betrachten. Zu Anfang des Spiels sind diese Randfelder zu den an die Ränder grenzenden Feldern benachbart. Als Beispiel zeigen Abbildung 4 (Seite 13) und Abbildung 5 eine solche Repräsentation einer Spielsituation durch einen Hex-Graphen aus der Sicht von Weiß.

Wir verlangen aber nicht, dass ein Hex-Graph eine Hex-Spielsituation repräsentieren *muss*.

³Wir halten eine solche Darstellung im Hinblick auf eine Repräsentation in Computern für zweckmäßig bzw. effizient. Dies wird später noch erläutert.

2.2 Spielverläufe ausgehend von Hex-Graphen

Wir möchten nun mögliche Spielverläufe betrachten, ausgehend von einer Spielsituation, die als Hex-Graph (siehe Abschnitt 2.1) modelliert ist. Für die Beschreibung der Folge-Spielsituationen weichen wir aber von Hex-Graphen wie folgt ab:

- Die Zielmenge von β wird erweitert auf $\{f, b, \bar{b}\}$.
- Vom eigenen Spieler besetzte Felder ($\beta(x) = b$), die benachbart zu vom eigenen Spieler besetzten Feldern sind, werden nicht mehr verschmolzen.
- Vom Gegner besetzte Felder werden nicht mehr aus dem Graphen entfernt, sondern mit $\beta(x) = \bar{b}$ markiert.

Die Hex-Regeln können wir dann wie folgt beschreiben. Die Spieler ziehen abwechselnd. Ein Zug des Gegners besteht darin, ein freies Feld zu besetzen. Das heißt, er wählt ein $z \in V$ mit $\beta(z) = f$ und setzt $\beta(z)$ auf \bar{b} . Ein Zug des eigenen Spielers besteht analog darin, dass er für ein $z \in V$ den Wert von $\beta(z)$ von f auf b setzt.

Das heißt, V und E bleiben während des Spiels unverändert.

Kann der eigene Spieler die Felder x, y verbinden und repräsentieren x, y die Ränder des eigenen Spielers, so hat der eigene Spieler gewonnen. Die Felder x, y zu *verbinden* bedeutet, dass es im Verlauf des Spiels ein k und Spielfelder z_1, \dots, z_k gibt mit

$$\beta(z_1) = \dots = \beta(z_k) = b \text{ und} \\ \{\{x, z_1\}, \{z_1, z_2\}, \dots, \{z_{k-1}, z_k\}, \{z_k, y\}\} \subseteq E.$$

Dies ermöglicht uns folgende Definition:

Definition 2.2 (σ_{HEX})

Das Prädikat σ_{HEX} trifft genau dann auf das Quintupel (V, E, β, x, y) zu, wenn es alle folgenden Eigenschaften besitzt:

1. (V, E, β) ist ein Hex-Graph, der eine Hex-Spielsituation repräsentiert.
2. Die Felder $x, y \in V$ repräsentieren die beiden Randfelder dieser Hex-Spielsituation.
3. In dieser Hex-Spielsituation ist der Gegner am Zug.
4. Der eigene Spieler kann x und y verbinden und somit gewinnen.

Weil wir später auch über komplexitätstheoretische Sachverhalte reden möchten, setzen wir für das Quintupel aus Definition 2.2 und ähnliche Fälle eine geeignete Codierung über ein endliches Alphabet (z. B. $\{0, 1\}$) voraus. Das Prädikat σ_{HEX} kann so auch auf Wörter aus $\{0, 1\}^*$ zutreffen. In diesem Sinn ist auch die folgende Definition zu verstehen.

Definition 2.3 (Sprache HEX)

Die Sprache HEX ist diejenige Sprache, für die σ_{HEX} die charakteristische Funktion ist.

Bemerkung 2.4 Informell ist HEX die Sprache der Gewinnstellungen.

Bemerkung 2.5 Die erste Eigenschaft aus Definition 2.2 könnte eine echte Einschränkung sein, da nicht jeder Hex-Graph in ein Hex-Spielbrett einbettbar sein muss.

Satz 2.6 HEX ist **PSPACE**-vollständig.

Beweis: siehe [Rei81]. □

Zwischen zwei Spielfeldern x, y (die nicht notwendigerweise Randfelder in einer Hex-Spielsituation repräsentieren müssen) eines Hex-Graphen (V, E, β) kann, wie wir später sehen werden, eine *Virtuelle Verbindung* bestehen. Eine solche Virtuelle Verbindung bedeutet u. a., dass der eigene Spieler die Möglichkeit hat, x und y zu verbinden, auch wenn der Gegner am Zug ist. Repräsentieren x und y dabei Randfelder einer Hex-Spielsituation, ist $(V, E, \beta, x, y) \in HEX$.

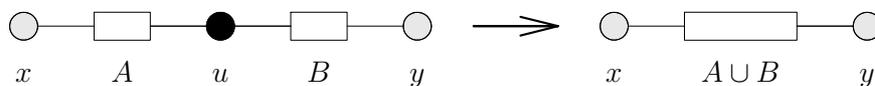
3 Definition Virtueller Verbindungen

Für die folgende Definition sei ein Hex-Graph $G = (V, E, \beta)$ gegeben.

Definition 3.1 (Virtuelle Verbindung)

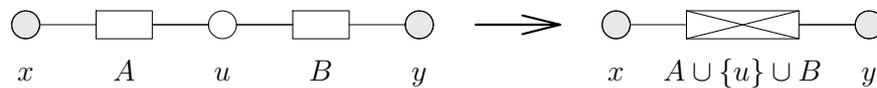
1. Ein Tripel $\mathcal{VC} = (x, C, y) \in V \times 2^V \times V$ bildet genau dann eine Virtuelle Verbindung, wenn es eine der drei folgenden Bedingungen erfüllt:
 - (a) $\{x, y\} \in E$ und $C = \emptyset$.
 - (b) \mathcal{VC} ist mittels der AV-Regel als Virtuelle Verbindung ableitbar.
 - (c) \mathcal{VC} ist mittels der OR-Regel als Virtuelle Verbindung ableitbar.
2. Ein Tripel $\mathcal{SC} = (x, C, y) \in V \times 2^V \times V$ bildet genau dann eine Virtuelle Semi-Verbindung, wenn es mittels der AS-Regel als Virtuelle Semi-Verbindung ableitbar ist.
3. Seien $(x, A, u), (u, B, y)$ zwei Virtuelle Verbindungen mit $x \neq y$, $x \notin B$, $y \notin A$ und $A \cap B = \emptyset$.

- (a) Wenn $\beta(u) = b$ gilt, dann ist $(x, A \cup B, y)$ eine Virtuelle Verbindung.



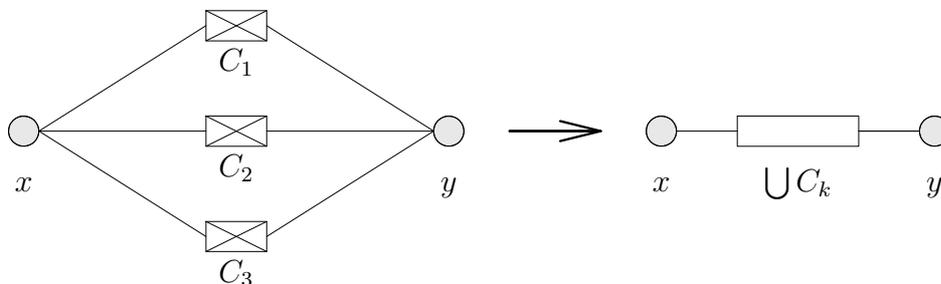
Wir nennen diese Regel AND-Regel für Virtuelle Verbindungen oder AV-Regel.

(b) Wenn $\beta(u) = f$ gilt, dann ist $(x, A \cup \{u\} \cup B, y)$ eine Virtuelle Semi-Verbindung.



Wir nennen diese Regel AND-Regel für Virtuelle Semi-Verbindungen oder AS-Regel.

4. Seien $n > 1$ und $(x, C_1, y), \dots, (x, C_n, y)$ Virtuelle Semi-Verbindungen mit $\bigcap_{k=1}^n C_k = \emptyset$. Dann ist $(x, \bigcup_{k=1}^n C_k, y)$ eine Virtuelle Verbindung.



Wir nennen diese Regel OR-Regel.

Bemerkungen:

- Die auf der linken Seite der Zeichnungen in Definition 3.1 verwendeten Darstellungen zeigen eine genauere Struktur der abgeleiteten Virtuellen Verbindungen bzw. Semi-Verbindungen. Auf diese Weise werden wir teilweise auch im Folgenden Virtuelle Verbindungen bzw. Semi-Verbindungen darstellen.
- Sei (x, C, y) eine Virtuelle Verbindung. Dann sagen wir auch, x und y seien *virtuell verbunden*.
- Sei (x, C, y) eine Virtuelle Verbindung oder Semi-Verbindung. Dann nennen wir C *Träger*. Träger sind immer Mengen unbesetzter Felder. Außerdem gilt $\{x, y\} \cap C = \emptyset$.

4 Wichtige Eigenschaften Virtueller Verbindungen

Die Sätze in diesem Abschnitt sind alle in [Ans02] bewiesen. Wir wiederholen sie aus Gründen der Vollständigkeit und um einen Beweis etwas genauer zu führen als in [Ans02].

Satz 4.1

1. Sei $\mathcal{VC} = (x, C, y)$ eine Virtuelle Verbindung. Dann kann der eigene Spieler die Felder x und y im weiteren Spielverlauf verbinden, auch wenn der Gegner am Zug ist.
2. Sei $\mathcal{SC} = (x, C, y)$ eine Virtuelle Semi-Verbindung. Dann kann der eigene Spieler die Felder x und y im weiteren Spielverlauf verbinden, wenn er selbst am Zug ist.

In beiden Fällen muss der eigene Spieler nur auf Felder in C spielen. Im Fall der Virtuellen Verbindung ist der eigene Spieler nur zur Reaktion gezwungen, wenn der Gegner auf ein Feld in C spielt. Im Fall der Virtuellen Semi-Verbindung ist nach seinem ersten Zug der eigene Spieler nur zur Reaktion gezwungen, falls der Gegner auf ein Feld in C spielt.

Beweis: Wir führen den Beweis durch Induktion über die Ableitung von Virtuellen Verbindungen.

1. Sei \mathcal{VC} mittels der AV-Regel ableitbar, d. h., seien (x, A, u) und (u, B, y) Virtuelle Verbindungen mit $A \cap B = \emptyset$, $C = A \cup B$ und $\beta(u) = b$. Nach Induktionsvoraussetzung kann der eigene Spieler die Felder x und u sowie die Felder u und y verbinden. Wegen $A \cap B = \emptyset$ kann der eigene Spieler nämlich seine Strategie, die Felder x und u zu verbinden, und seine Strategie, die Felder u und y zu verbinden, unabhängig voneinander verfolgen. Verbindungen zwischen den Feldern x , u und y stellen auch eine Verbindung zwischen x und y dar.

Sei \mathcal{VC} mittels der OR-Regel ableitbar, d. h., seien $n > 1$ und $(x, C_1, y), \dots, (x, C_n, y)$ Virtuelle Semi-Verbindungen mit $\bigcap_{k=1}^n C_k = \emptyset$. Falls der Gegner im nächsten Zug ein Feld aus C bespielt, gibt es danach wegen $\bigcap_{k=1}^n C_k = \emptyset$ immer noch eine intakte Virtuelle Semi-Verbindung (x, C_i, y) . Nach Induktionsvoraussetzung kann der eigene Spieler die Felder x und y verbinden, denn er ist dann am Zug.

2. Sei \mathcal{SC} mittels der AS-Regel ableitbar, d. h., seien (x, A, u) und (u, B, y) Virtuelle Verbindungen mit $A \cap B = \emptyset$, $C = A \cup B$ und $\beta(u) = f$. Spielt nun der eigene Spieler auf u , tritt dieselbe Situation auf, die am Anfang des ersten Teils des Beweises beschrieben wurde. D. h. im weiteren Verlauf kann der eigene Spieler x und y verbinden.

□

Definition 4.2 (σ_{HEXY})

Das Prädikat σ_{HEXY} trifft genau dann auf das Quintupel (V, E, β, x, y) zu, wenn es alle folgenden Eigenschaften besitzt:

1. (V, E, β) ist ein Hex-Graph, der eine Hex-Spielsituation repräsentiert.

2. Die Felder $x, y \in V$ repräsentieren die beiden Randfelder dieser Hex-Spielsituation.
3. In dieser Hex-Spielsituation ist der Gegner am Zug.
4. Die Felder x und y sind virtuell verbunden.

Definition 4.3 (Sprache $HEXY$)

Die Sprache $HEXY$ ist diejenige Sprache, für die σ_{HEXY} die charakteristische Funktion ist.

Bemerkung 4.4 Informell ist $HEXY$ die Sprache der Gewinnstellungen, für die es mittels Virtueller Verbindungen einen Beweis gibt.

Mit diesen Definitionen können wir aus Satz 4.1 das folgende Korollar folgern:

Korollar 4.5 $HEXY \subseteq HEX$

Es stellt sich die Frage, ob $HEXY = HEX$ gilt. Wir können das mit folgendem Satz verneinen.

Satz 4.6 $HEX \setminus HEXY \neq \emptyset$.

Beweis: Zu zeigen ist:

Es gibt einen Hex-Graphen $G = (V, E, \beta)$ und Felder $x, y \in V$, sodass gilt:

1. Der Hex-Graph G repräsentiert eine Hex-Spielsituation.
2. Die Felder $x, y \in V$ mit $\beta(x) = \beta(y) = b$ repräsentieren die beiden Randfelder dieser Hex-Spielsituation.
3. In dieser Hex-Spielsituation ist der Gegner am Zug.
4. Der eigene Spieler kann die Felder x und y verbinden.
5. Die Felder x und y sind nicht virtuell verbunden.

Der Beweis erfolgt durch Angabe eines geeigneten Hex-Graphen G (siehe Abbildung 4) und einer dazugehörigen Hex-Spielsituation (siehe Abbildung 5).⁴ Die weißen Ränder stellen die Felder x und y dar. Der Gegner (Schwarz) ist am Zug.

Wir zeigen zunächst, dass der eigene Spieler (Weiß) die Ränder x und y verbinden kann. Schwarz besetze o.B.d.A. ein Feld aus der unteren Hälfte von G .

⁴In [Ans02] wird auf die Angabe einer Spielsituation verzichtet.

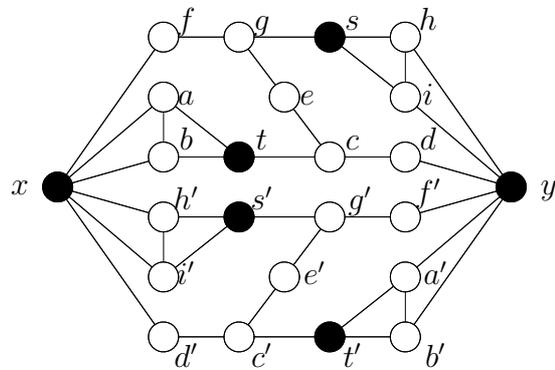


Abbildung 4: Der Hex-Graph G

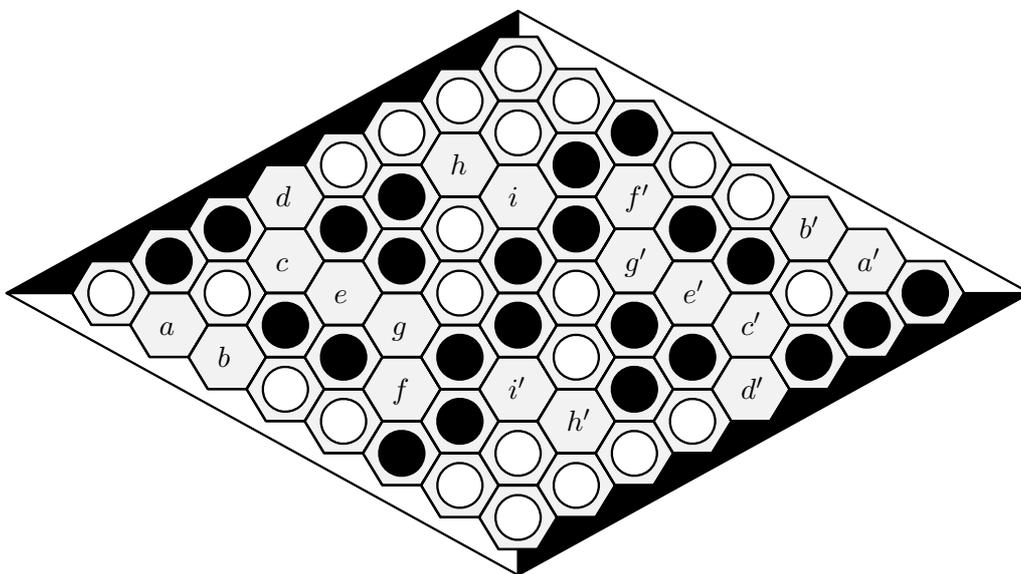


Abbildung 5: Eine zu G gehörige Spielsituation

Dann kann Weiß g belegen und zwingt damit Schwarz, im nächsten Zug auf f zu spielen.⁵

Symmetrisch dazu zwingt anschließend Weiß mit einem Zug auf c Schwarz dazu, auf d zu spielen. Wenn Weiß jetzt e belegt, entsteht ein weißer Pfad von x über t, c, e, g und s nach y .

Es ist nur noch zu zeigen, dass x und y nicht virtuell verbunden sind. In [Ans02] wird anstelle eines Beweises auf die Benutzung eines Computerprogramms verwiesen, welches alle Virtuellen Verbindungen in G berechnet. Wir führen hier einen Beweis ohne Computerhilfe.

Dazu zeigen wir zunächst, dass eine Virtuelle Verbindung von x und y nicht mit der AV-Regel (über s, s', t oder t') abgeleitet werden kann.

Angenommen, es gäbe Virtuelle Verbindungen (x, A, t) und (t, B, y) mit $A \cap B = \emptyset$ (wobei aus $\{s, s', t, t'\}$ das Feld t o.B.d.A. gewählt ist). Einer der Träger A und B enthält dann nur höchstens ein Feld aus $\{a, b, c\}$. Wenn Schwarz nun auf dieses Feld spielt, wird die Virtuelle Verbindung zu x oder zu y abgeschnitten, d. h. es handelt sich überhaupt nicht um eine Virtuelle Verbindung. Widerspruch. Dieselben Überlegungen treffen auch für d' anstelle von x zu; das wird später benötigt.

Eine Virtuelle Verbindung von x und y kann also nur mit der OR-Regel abgeleitet werden. Sei dementsprechend (x, C, y) eine Virtuelle Semi-Verbindung mit $g' \notin C$. Wir können uns vorstellen, Schwarz spiele auf g' . Dann gibt es Virtuelle Verbindungen (x, C_1, z) und (z, C_2, y) mit $z \neq g'$, $\{z, g'\} \cap (C_1 \cup C_2) = \emptyset$, $C_1 \cap C_2 = \emptyset$ und $\beta(z) = f$. Wir können uns vorstellen, dass Weiß als Reaktion auf die Belegung von g' durch Schwarz nun auf z spielt. Wegen Satz 4.1 hat Weiß so eine Gewinnstrategie (x mit z über C_1 verbinden, z mit y über C_2 verbinden).

Es muss $z \neq e'$ gelten, denn sonst kann Schwarz $z (= e')$ durch Spiel auf c' isolieren. Wäre $z \in \{e, f, a, b, h, i, f', a', b', h', i'\}$, könnte Schwarz g belegen und im Folgenden x und y durch komplementäres Spiel auf die Felderpaare (c, d) und (c', d') trennen. Wäre $z = d$, könnte Schwarz c belegen und im Folgenden x und y durch komplementäres Spiel auf die Felderpaare (f, g) und (c', d') trennen.

Wäre $z = c'$, enthielte (wegen $C_1 \cap C_2 = \emptyset$) C_1 oder C_2 höchstens ein Feld aus $\{d', a', b'\}$. Dies könnte Schwarz im nächsten Zug besetzen und so c' von x oder von y abschneiden. (Schwarz hat g' schon belegt.)

Wäre $z = g$, spielt Schwarz auf f und trennt im Folgenden g von x durch komplementäres Spiel auf die Felderpaare (e, c) und (d', c') . Ähnliches gilt für den Fall $z = c$. (Man beachte hier den Unterschied zu einer allgemeinen Strategie.)

Es bleibt also nur der Fall $z = d'$ übrig. D. h. es bestehen Virtuelle Verbindungen (x, C_1, d') und (d', C_2, y) . Man kann $C_1 = \emptyset$ wählen. Wie kann aber die Virtuelle Verbindung (d', C_2, y) abgeleitet werden?

⁵Schwarz könnte auch zunächst auf h bzw. i spielen, was Weiß mit i bzw. h beantworten würde. Ähnliches gilt während des ganzen Spiels für h' und i' , a und b sowie a' und b' . Wir nennen das *komplementäres Spiel* auf die Felderpaare $(h, i), (h', i'), (a, b)$ und (a', b') .

Wir wissen bereits, dass eine Virtuelle Verbindung von d' und y nicht mit der AV-Regel (über s, s', t oder t') abgeleitet werden kann. Also muss (d', C_2, y) über die OR-Regel abgeleitet werden. Wir stellen uns vor, Schwarz spiele auf c' . Jetzt muss es Virtuelle Verbindungen (d', C_3, z') und (z', C_4, y) mit $z' \notin \{c', g', d'\}$, $\{z', c', g', d'\} \cap (C_3 \cup C_4) = \emptyset$, $C_3 \cap C_4 = \emptyset$ und $\beta(z') = f$ geben.

Durch Anwendungen der AV-Regel können Virtuelle Verbindungen (d', \emptyset, i') , (d', \emptyset, h') , (d', \emptyset, b) , (d', \emptyset, a) und (d', \emptyset, f) abgeleitet werden. Da c' von Schwarz bereits besetzt ist, entspricht dies gerade den Nachbarschaften von x . Daher kann die Aufgabe, Virtuelle Verbindungen (d', C_3, z') und (z', C_4, y) abzuleiten (mit o.g. Eigenschaften für z', C_3, C_4), höchstens analog zur Aufgabe, Virtuelle Verbindungen (x, C_1, z) und (z, C_2, y) abzuleiten, gelöst werden.⁶ Für Letzteres war aber die Wahl $z = d'$ zwingend. Da $z' = d'$ nicht möglich ist, sind beide Aufgaben unlösbar. D. h. es gibt keine Virtuelle Verbindung (x, C, y) . \square

Korollar 4.7 $HEXY \subsetneq HEX$.

5 Eine Einschränkung der Operatoren

Das Ziel dieses Abschnitts ist es, für die weiteren Betrachtungen die Anwendungsfälle der Operatoren, die in Abschnitt 3 definiert wurden, einzuschränken. Genauer gesagt, möchten wir teilweise auf die Möglichkeit verzichten, mit der AV-Regel Virtuelle Verbindungen abzuleiten. Wir werden zeigen, dass mit dieser Einschränkung nicht weniger Virtuelle Verbindungen erkannt werden.

5.1 Die eingeschränkte Rolle der AV-Regel

Wir werden den Beweis des zentralen Satzes dieses Abschnitts per Induktion führen. Dazu benötigen wir folgende partielle Ordnung auf der Menge der Virtuellen Verbindungen.

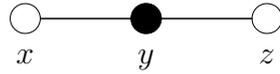
Definition 5.1 Sei $G = (V, E, \beta)$ ein Hex-Graph. Seien darin $\mathcal{VC}_1 = (x_1, C_1, y_1)$ und $\mathcal{VC}_2 = (x_2, C_2, y_2)$ Virtuelle Verbindungen. Dann gilt $\mathcal{VC}_1 \leq \mathcal{VC}_2$ per Definition genau dann, wenn gilt:

- $\mathcal{VC}_1 = \mathcal{VC}_2$ oder
- $|C_1| \leq |C_2|$ oder
- $|C_1| = |C_2|$ und $|\{z \mid z \in \{x_1, y_1\}, \beta(z) = f\}| \leq |\{z \mid z \in \{x_2, y_2\}, \beta(z) = f\}|$

⁶Tatsächlich ist die Situation jetzt schwieriger, da neben g' auch c' von Schwarz besetzt ist.

Definition 5.2 Wird die AV-Regel auf zwei Virtuelle Verbindungen mit leeren Trägern angewandt, sprechen wir auch von einer AV_0 -Regel. Bei sonstigen Anwendungen der AV-Regel sprechen wir auch von einer AV_+ -Regel.

Bemerkung 5.3 Die AV_0 -Regel kann nur in Fällen folgender Art angewandt werden:



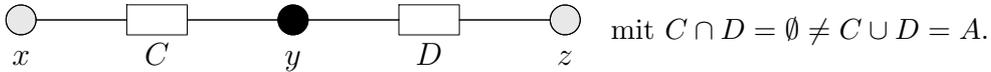
D. h., x und z sind frei. Denn Nachbarn von besetzten Feldern sind nach Definition 2.1 (3. Eigenschaft) frei.

Satz 5.4 Sei $\mathcal{VC} = (x, A, z)$ eine Virtuelle Verbindung in einem Hex-Graphen G . Dann ist zu deren Ableitung die AV_+ -Regel nicht erforderlich.

Beweis: Wir zeigen durch Induktion über die Ordnung aus Definition 5.1: Wenn \mathcal{VC} über die AV_+ -Regel entstanden ist, ist \mathcal{VC} auch ohne Verwendung dieser Regel ableitbar.

Induktionsanfang: $A = \emptyset$. Dann kann \mathcal{VC} nicht über die AV_+ -Regel entstanden sein. Die Behauptung ist dann also leer.

Induktionsschritt: $A \neq \emptyset$. Sei im Folgenden \mathcal{VC} über die AV_+ -Regel entstanden. (Andernfalls ist nichts zu zeigen.) Dann hat \mathcal{VC} die Form



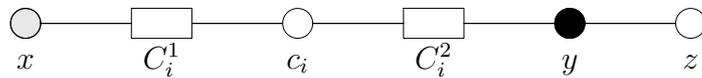
Die Felder x, z könnten besetzt oder frei sein (dargestellt durch graue Kreise).

Fall 1: $C \neq \emptyset = D$ oder $D \neq \emptyset = C$.

Wir nehmen o.B.d.A. ersteres an. Weil G ein Hex-Graph ist, gilt dann $\beta(z) = f$.

Weil $\beta(y) = b$ gilt, ist die Virtuelle Verbindung nach Induktion ohne AV_+ -Regel ableitbar, denn sie ist kleiner als \mathcal{VC} .

Das bedeutet, es gibt ein $k > 1$ und C_i für $i \in \{1, \dots, k\}$ mit $C = \bigcup_i C_i$ und $\bigcap_i C_i = \emptyset$, wobei sich die C_i jeweils disjunkt in $C_i^1 \cup \{c_i\} \cup C_i^2$ partitionieren lassen:

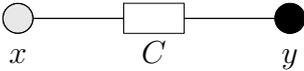
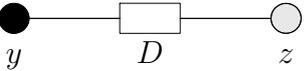


Die Virtuellen Verbindungen und sind ohne AV_+ -Regel ableitbar,

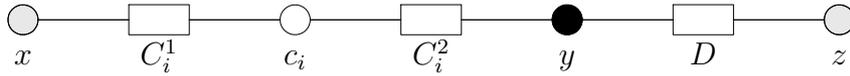
weil kleiner als \mathcal{VC} , also auch die Virtuellen Semi-Verbindungen . Wegen $\bigcap_i C_i = \emptyset$ ist die OR-Regel anwendbar und $\mathcal{VC} =$

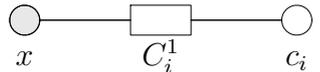
ist ohne AV_+ -Regel ableitbar.

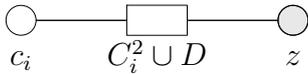
Fall 2: $C \neq \emptyset$ und $D \neq \emptyset$ mit $C \cap D = \emptyset$.

Dann sind  und  ohne

AV₊-Regel ableitbar, denn sie sind kleiner als \mathcal{VC} . Das bedeutet, es gibt wieder ein $k > 1$ und C_i für $i \in \{1, \dots, k\}$ mit $C = \bigcup_i C_i$ und $\bigcap_i C_i = \emptyset$, wobei sich die C_i jeweils disjunkt in $C_i^1 \dot{\cup} \{c_i\} \dot{\cup} C_i^2$ partitionieren lassen:



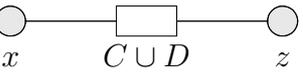
Die Virtuellen Verbindungen  und



sind ohne AV₊-Regel ableitbar, weil kleiner als \mathcal{VC} ,

also auch die Virtuellen Semi-Verbindungen .

Symmetrisch gilt dies auch für Virtuelle Semi-Verbindungen . Wegen $\bigcap_i (C_i \cup D) \cap \bigcap_j (C \cup D_j) = \emptyset$ ist die

OR-Regel anwendbar und $\mathcal{VC} =$  ist ohne AV₊-

Regel ableitbar. □

5.2 Informelle Überlegungen zur AND-Regel

Satz 5.4 wirft zwei Fragen auf:

1. Inwieweit ist es überhaupt sinnvoll, Hex-Graphen mit besetzten Feldern zu betrachten?
2. Welche Bedeutung hat die AV₊-Regel möglicherweise im Hinblick auf die praktische Berechnung Virtueller Verbindungen mit Computern?

Wir wollen zuerst die erste Frage erörtern. Zur Erläuterung sei angemerkt, dass man folgendes Verfahren durchführen könnte, um besetzte Felder aus einem Hex-Graphen zu entfernen:

- Wähle ein besetztes Feld z . Seien x_1, \dots, x_n dessen Nachbarn, die nach Definition von Hex-Graphen frei sind.
- Entferne z und seine Nachbarschaften. Füge die Nachbarschaften $\{\{x_i, x_j\} \mid 1 \leq i, j \leq n, i \neq j\}$ hinzu.
- Wiederhole, bis alle besetzten Felder entfernt sind.

Das Verfahren fügt also Cliques hinzu. Die Verbindung zur AV-Regel ergibt sich daraus, dass die Durchführung des Verfahrens genau der Anwendung der

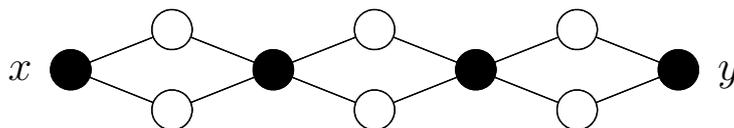
AV₀-Regel entspricht. Man könnte jedoch die Anwendung der AV-Regel ganz verbieten, wenn man vorher das obige Verfahren durchführt.

Ein offensichtlicher Nachteil dieser Vorgehensweise ist aber, dass sich die Ränder nicht mehr als besetzte „Randfelder“ repräsentieren lassen, da besetzte Felder ja entfernt werden. Um das zu vermeiden, könnte man die Randfelder als frei betrachten. Allerdings müsste man trotzdem konsequenterweise zwei Felder x_i, x_j , die an den Rand angrenzen als benachbart (bzw. virtuell verbunden mit leerem Träger) betrachten. Diese Vorgehensweise erscheint insgesamt etwas unnatürlich.

Ein weiterer Nachteil obiger Vorgehensweise ergibt sich daraus, dass wir auch die Einbettung von Hex-Graphen in konkrete Hex-Bretter im Auge haben. In [Rei81] wird beschrieben, wie planare Hex-Graphen in Hex-Bretter eingebettet werden können, falls freie Felder höchstens drei Nachbarn haben. Durch die Einführung weiterer Nachbarschaften durch obiges Verfahren geriete diese Eigenschaft in vielen Fällen in Gefahr.

Praktische Erwägungen bei der konkreten Berechnung Virtueller Verbindungen mit Computern führen uns zur zweiten Frage und zu einem weiteren Nachteil des völligen Verzichts auf die AV-Regel. Es stellt sich nämlich heraus, dass die Berechnung Virtueller Verbindungen rechenintensiv ist.⁷ Trotzdem widmet beispielsweise das Programm *Hexy* viel seiner Bedenkzeit der Berechnung Virtueller Verbindungen. Jedoch können nicht alle Virtuellen Verbindungen berechnet werden. Es sind daher (heuristische) Regeln erforderlich, unter welchen Bedingungen welche Regeln angewandt werden sollen. Naheliegender ist, u. U. die Anwendung der OR-Regel auf zu viele (beispielsweise mehr als fünf) Virtuelle Semi-Verbindungen zu verbieten.

Andererseits sollen aber (im Sinne der „Künstlichen Intelligenz“) zumindest diejenigen Virtuellen Verbindungen erkannt werden, die ein menschlicher Spieler schnell erkennt. Beispielsweise erkennen menschliche Spieler leicht „Ketten“ aus „Zweier-Brücken“:



Es ist leicht nachzuprüfen, dass viele Regelanwendungen erforderlich sind, um die Virtuelle Verbindung zwischen x und y nachzuweisen, wenn man auf die AV₊-Regel verzichtet. Wenn man jedoch die AV₊-Regel hinzuzieht, kann man die Anwendung der OR-Regel auf zwei Virtuelle Semi-Verbindungen gleichzeitig beschränken.

Diese Gründe motivieren nachträglich unsere Definition von Hex-Graphen mit besetzten Feldern.

Trotzdem werden wir der Einfachheit halber bei den folgenden theoretischen Überlegungen auf die Anwendung der AV-Regel verzichten. Das heißt, wir set-

⁷Das soll ja auch theoretisch in dieser Arbeit gezeigt werden.

zen die Anwendung des obigen Verfahrens voraus, ohne allerdings die Randfelder zu entfernen. Ob diese Randfelder als besetzt oder frei betrachtet werden, spielt keine Rolle. Mit Satz 5.4 schränken wir die Mächtigkeit Virtueller Verbindungen durch den Verzicht auf die AV-Regel nicht ein.

6 Strategien

Für die Überlegungen in diesem Abschnitt setzen wir einen festen Hex-Graphen $G = (V, E, \beta)$, wie er in Abschnitt 2.1 definiert wurde, voraus. Mit den Überlegungen aus Abschnitt 5.2 können wir die Spielfelder als frei annehmen, auch wenn wir uns für Virtuelle Verbindungen interessieren.⁸

Wir interessieren uns in diesem Abschnitt für *Strategien*. Strategien sind Pläne eines Spielers (hier: des eigenen Spielers), die beschreiben, wie er sich im weiteren Spielverlauf verhalten wird. Wir definieren Strategien formal. Dazu bedienen wir uns einer informatik-typischen Beschreibung mit hierarchisch-baumartigem Charakter.

Ziel dieses Abschnittes ist, die *Strategie*, die zur Realisierung von Virtuellen Verbindungen nötig ist, zu charakterisieren. So können wir das Vorhandensein einer Virtuellen Verbindung der Spielfelder x und y beweisen bzw. widerlegen, indem wir das Vorhandensein eines bestimmten Typs von Verbindungs*strategie* beweisen bzw. widerlegen.

6.1 Nicht-lokale Strategien

Definition 6.1 (Nicht-lokale Strategie (NLS))

Sei $C \subseteq V$ eine Menge freier Spielfelder. Dann ist eine nicht-lokale Strategie (NLS) mit Träger C ein Baum mit folgenden Eigenschaften:

1. Die Wurzel ist ein Knoten w , der mit $\text{Carr}(w) = C$ markiert ist.
2. Die Wurzel w hat $n \geq 0$ Söhne s_i , welche ihrerseits NLS mit Träger $\text{Carr}(s_i) = C_i$ sind.
3. Die Kanten d_i zu den Söhnen s_i sind mit $\text{zug}(d_i) = c_i \in V$ markiert.
4. $\bigcup_{i=1}^n (\{c_i\} \cup C_i) \subseteq C$.
5. $\forall i \in \{1, \dots, n\} : c_i \notin C_i$.
6. $n > 0 \Rightarrow \bigcap_{i=1}^n (\{c_i\} \cup C_i) = \emptyset$.

Um einen Zusammenhang mit der umgangssprachlichen Bedeutung des Worts *Strategie* herzustellen, beachte man folgendes:

⁸Zur Kennzeichnung von Randfeldern zeichnen wir in manchen Hex-Graphen Randfelder als belegte Felder (schwarz).

- Der Träger gibt an, auf welche Felder der eigene Spieler im weiteren Spielverlauf möglicherweise spielen wird.
- Die Söhne geben Strategien für das weitere Spiel an, nachdem Gegner und eigener Spieler je einmal gezogen haben.
- Im Fall $n = 0$ „endet“ die Strategie in dem Sinn, dass es für den weiteren Spielverlauf keine Strategie mehr gibt.
- Die Kantenmarkierungen geben den Zug des eigenen Spielers an.
- Die vierte Eigenschaft von Definition 6.1 garantiert, dass der Spieler auch weiterhin nur auf Felder des Trägers spielen wird.
- Die fünfte Eigenschaft garantiert, dass die Strategie nicht verlangt, dass der Spieler mehr als einmal auf ein Feld spielt.
- Die sechste Eigenschaft garantiert, dass der eigene Spieler auf jeden möglichen Zug des Gegners eine Antwort (d.h. einen eigenen Zug und eine Fortsetzungsstrategie mit einem Träger noch freier Felder) hat.

Es ist möglich, dass eine NLS zwei verschiedene, aber identisch markierte Unterbäume enthält. Man kann dann solche Teilstrategien miteinander identifizieren, wodurch Speicherbedarf und graphische Darstellungen der Strategien kleiner werden. Auf diese Weise entsteht ein gerichteter azyklischer Graph (DAG). Wir betrachten so entstandene DAGs auch als Strategien.

Der „Erfolg“ einer Strategie misst sich bei Hex an den vom eigenen Spieler belegten Feldern. Dabei kommt es typischerweise nicht auf die Reihenfolge der Belegung an. Nur aufgrund der bequemeren Notation benutzt die folgende Definition das Konzept der *Wörter*, die eine *Sprache* bilden. Wir können in diesem Zusammenhang aber von Wörtern ausgehen, die keine Reihenfolge der Buchstaben induzieren.

Definition 6.2 (Sprache einer NLS)

Sei w die Wurzel einer NLS mit den Bezeichnungen aus Definition 6.1. Dann ist die Sprache dieser NLS definiert als

$$L(w) := \bigcup_{i=1}^n (\{\text{zug}(d_i)\} \circ L(s_i)) \quad (\subseteq C^*).$$

Im Fall $n = 0$ ist $L(w) = \emptyset$.

Die Sprache gibt die möglichen Belegungen des eigenen Spielers an, wenn er der Strategie folgt. Dabei entscheidet i. A. der Gegner, welche Belegung der eigene Spieler tatsächlich erzielt, d. h. der Gegner wählt das Wort aus der Sprache.

Definition 6.3

Eine NLS mit Wurzel w heißt (x, y) -verbindend genau dann, wenn

$$\forall u \in L(w) \exists \text{ Buchstaben } u_1, \dots, u_k \text{ aus } u : \\ \{\{x, u_1\}, \{u_1, u_2\}, \dots, \{u_{k-1}, u_k\}, \{u_k, y\}\} \subseteq E.$$

In den Fällen $L(w) = \emptyset$ und $k = 0$ muss $\{x, y\} \in E$ gelten.

Diese Definition entspricht der Intuition. Als Beispiel betrachte man die Abbildungen 6 und 7. Die Sprache dieser NLS ist $\{bae, baf, bce, bcf, bea, bec, bfa, bfc, dg, dh\}$. Die NLS ist somit (x, y) -verbindend.

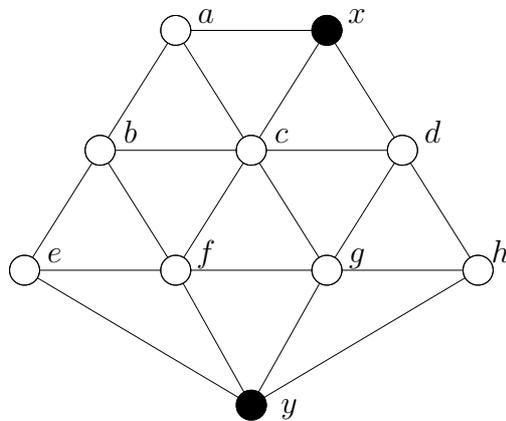


Abbildung 6: Ein Hex-Graph

6.2 Lokale Strategien

6.2.1 Lokalitäten in Spielen

Ein moderner Zweig der KI, der sich mit Spielen befasst, ist die *Combinatorial Game Theory*. Diese befasst sich damit, inwieweit Spielsituationen (z. B. im japanischen Brettspiel *Go*) sich in sog. *local subgames* zerlegen lassen. Das betrifft insbesondere die Endphase (*endgames*) [Mue99].

Dieser Ansatz entspricht dem menschlichen Vorgehen bei der Beurteilung von Spielsituationen. Auch Hex-Spielsituationen lassen sich oftmals vorteilhaft in sog. *local patterns* zerlegen. Z. B. besteht die Darstellung der Gewinnstrategie für ein 7×7 -Hex-Spiel in [YLP01] im Wesentlichen in der Beschreibung von 41 *local patterns*.

Dies motiviert uns, auch in unserem Formalismus lokale Strategien explizit beschreiben zu können.

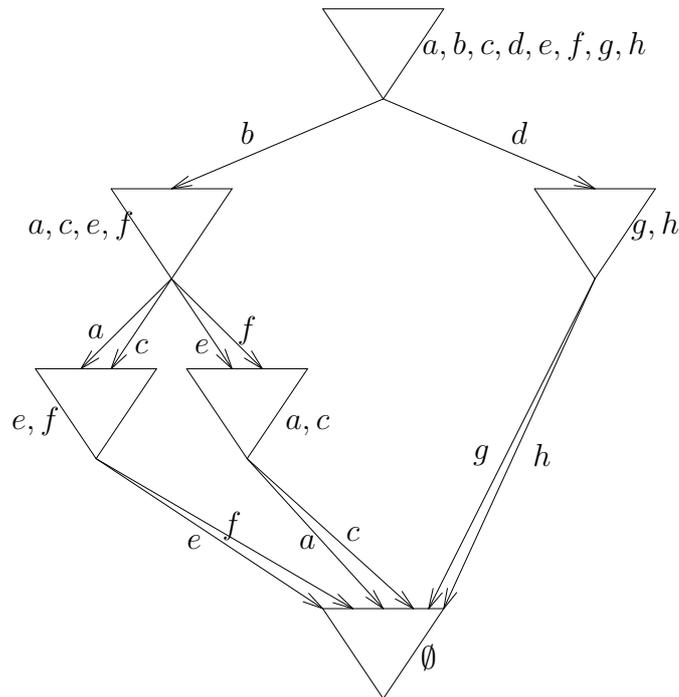


Abbildung 7: Eine NLS zum Hex-Graphen aus Abbildung 6

6.2.2 Lokale Strategien formal

Die Idee zur Einbindung lokaler Strategien besteht darin, dass wir erlauben, dass sich die NLS aus Abschnitt 6.1 an den Knoten in lokale Teilstrategien aufspalten. Dabei spaltet sich entsprechend der Träger disjunkt in Träger für die lokalen Teilstrategien auf. Dazu führen wir *Spaltungsknoten* ein: Knoten einer lokalen Strategie können Spaltungsknoten als Söhne haben, deren Söhne wiederum lokale Strategien sind.

Die unter einem Spaltungsknoten liegenden Strategien können parallel verfolgt werden, da ihre Träger disjunkt sind. Der Gegner kann immer nur den Träger einer darunterliegenden Strategie angreifen. Der eigene Spieler reagiert auf solche Angriffe entsprechend derjenigen Strategie, dessen Träger angegriffen worden ist.

Definition 6.4 (Lokale Strategie (LS))

Sei $C \subseteq V$ eine Menge freier Spielfelder. Dann ist eine lokale Strategie (LS) mit Träger C ein Baum mit folgenden Eigenschaften:

1. Die Wurzel ist ein Knoten w , der mit $\text{Carr}(w) = C$ markiert ist.
2. Die Wurzel w hat $n \geq 0$ Söhne s_i , welche ihrerseits LS oder Spaltungsknoten mit Träger $\text{Carr}(s_i) = C_i$ sind.

3. Die Kanten d_i zu den Söhnen s_i sind mit $\text{zug}(d_i) = c_i \in V$ markiert.
4. $\bigcup_{i=1}^n (\{c_i\} \cup C_i) \subseteq C$.
5. $\forall i \in \{1, \dots, n\} : c_i \notin C_i$.
6. $n > 0 \Rightarrow \bigcap_{i=1}^n (\{c_i\} \cup C_i) = \emptyset$.

Definition 6.5 (Spaltungsknoten)

Ein Knoten w ist ein Spaltungsknoten mit Träger $C \subseteq V$, wenn er folgende Eigenschaften besitzt:

1. Er ist mit $\text{Carr}(w) = C$ markiert.
2. Er hat $n > 0$ Söhne s_i , die LS mit Träger $\text{Carr}(s_i) = C_i$ sind.
3. Die C_i sind paarweise disjunkt, und es gilt $\bigcup_{i=1}^n C_i = C$.
4. Die Kanten zu den Söhnen sind unmarkiert.

Als Beispiel betrachte man wieder Abbildung 6. Die NLS aus Abbildung 7 können wir jetzt wie in Abbildung 8 gezeigt lokal formulieren. Dabei symbolisiert der trapezförmige Knoten einen Spaltungsknoten.

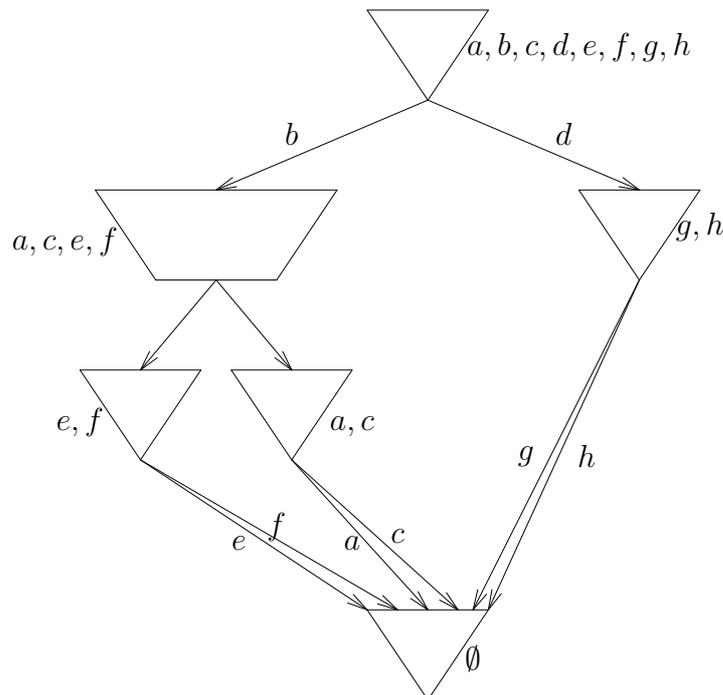


Abbildung 8: Eine LS zum Hex-Graphen aus Abbildung 6

Es stellt sich analog zu NLS die Frage, welche Belegungen der eigene Spieler erreichen kann. Dazu definieren wir die Sprache einer LS gleich wie die Sprache einer NLS:

Definition 6.6 (Sprache einer LS)

Sei w die Wurzel einer LS mit den Bezeichnungen aus Definition 6.4. Dann ist die Sprache dieser LS definiert als

$$L(w) := \bigcup_{i=1}^n (\{\text{zug}(d_i)\} \circ L(s_i)) \quad (\subseteq C^*).$$

Im Fall $n = 0$ ist $L(w) = \emptyset$.

Zur Vollständigkeit dieser Definition benötigen wir auch die Sprache eines Spaltungsknotens. Die folgende Definition berücksichtigt die oben beschriebene Funktion von Spaltungsknoten.

Definition 6.7 (Sprache eines Spaltungsknotens)

Sei w ein Spaltungsknoten mit den Bezeichnungen aus Definition 6.5. Dann ist die Sprache dieses Knotens definiert als

$$L(w) := L(s_1) \circ \dots \circ L(s_n).$$

Das bedeutet, der eigene Spieler erzielt Belegungen entsprechend *aller* darunterliegenden lokalen Strategien s_i . Wie zuvor bei NLS wählt jedoch i. A. der Gegner die Wörter aus den $L(s_i)$ und somit das Wort aus $L(w)$.

Wie NLS, können LS (x, y) -verbindend sein. Diese Eigenschaft sei genau analog zur Definition 6.3 definiert.

Als Beispiel mag die LS aus Abbildung 8 dienen: Die Sprache des Spaltungsknotens ist $\{e, f\} \circ \{a, c\} = \{ea, ec, fa, fc\}$. Die Sprache der LS ist dann $\{bea, bec, bfa, bfc, dg, dh\}$, was im Wesentlichen (modulo Reihenfolge) der Sprache der NLS aus Abbildung 7 entspricht. Die LS ist ebenfalls (x, y) -verbindend.

6.3 Direkt-verbindende LS und Virtuelle Verbindungen**Definition 6.8 (direkt-verbindende LS, DVLS)**

Eine LS mit Wurzel w heißt (x, y) -direkt-verbindend genau dann, wenn eine der beiden folgenden Bedingungen erfüllt ist:

1. Die Wurzel w hat keine Söhne und es gilt $\{x, y\} \in E$.
2. Die Wurzel w hat $n > 0$ Spaltungsknoten s_i als Söhne, sodass
 - (a) die Kanten zu den Söhnen mit c_i markiert sind und
 - (b) die Spaltungsknoten s_i jeweils zwei LS mit Wurzeln v_i^1 bzw. v_i^2 als Söhne haben und
 - (c) die v_i^1 Wurzeln von (x, c_i) -direkt-verbindenden LS sind und
 - (d) die v_i^2 Wurzeln von (c_i, y) -direkt-verbindenden LS sind.

Man kann sich (x, y) -direkt-verbindende Strategien so vorstellen, dass der Spieler x und y verbindet, indem er im Spielverlauf einen einzigen Pfad von x nach y aufbaut. Alle Felder, die er im Spielverlauf besetzt, sind Bestandteil dieses Pfades. Spielt er etwa in seinem ersten Zug auf ein Feld z , so könnte er seinem Gegner schon garantieren, dass er x und y verbindet, indem er x und z sowie z und y verbindet. Außerdem könnte er dem Gegner die beiden disjunkten Mengen freier Felder nennen, die er für die genannten Verbindungen eventuell benutzen wird. Im nächsten Zug besetzt er dann ein Feld z' , welches er entweder mit x und mit z verbinden wird oder mit z und mit y . So „konkretisiert“ er mit jedem Zug den Pfad von x nach y , bis dieser vollständig ist. Abbildung 9 versucht eine Illustration dieses Sachverhalts.

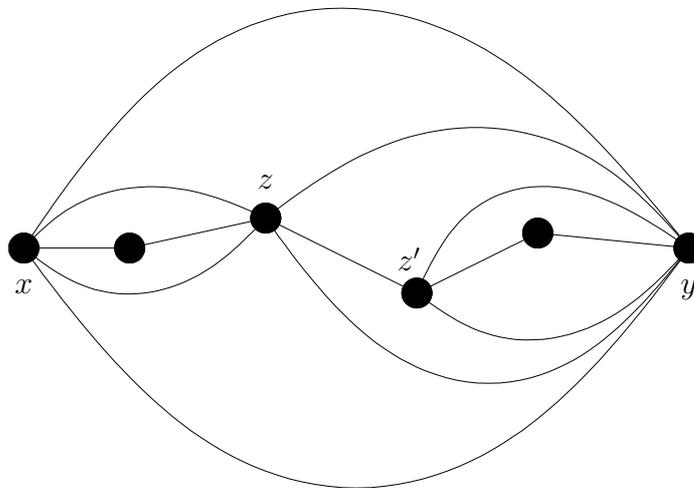


Abbildung 9: Eine Illustration einer (x, y) -DVLS

Es ist klar (Induktion über die Definition von DVLS), dass jede (x, y) -direkt-verbindende Strategie (x, y) -verbindend ist.

Als Beispiel betrachte man wieder den Graph aus Abbildung 6. Die (x, y) -verbindende LS aus Abbildung 8 lässt sich, wie Abbildung 10 zeigt, auch als DVLS formulieren.

Mit DVLS lassen sich Strategien, die zu Virtuellen Verbindungen „gehören“, charakterisieren. Dies besagen die folgenden beiden Sätze.

Satz 6.9 *Wenn die Knoten x und y vermöge der Virtuellen Verbindung (x, C, y) virtuell verbunden sind, existiert eine (x, y) -DVLS mit Wurzel w und $\text{Carr}(w) = C$.*

Beweis: Der Beweis erfolgt induktiv über den Aufbau Virtueller Verbindungen ohne AV-Regel. Im Induktionsanfang gilt $C = \emptyset$, und x, y sind benachbart. Also bildet ein sohnloser Knoten w mit $\text{Carr}(w) = \emptyset$ eine (x, y) -DVLS.

Ist $C \neq \emptyset$, liegt folgende Situation vor:

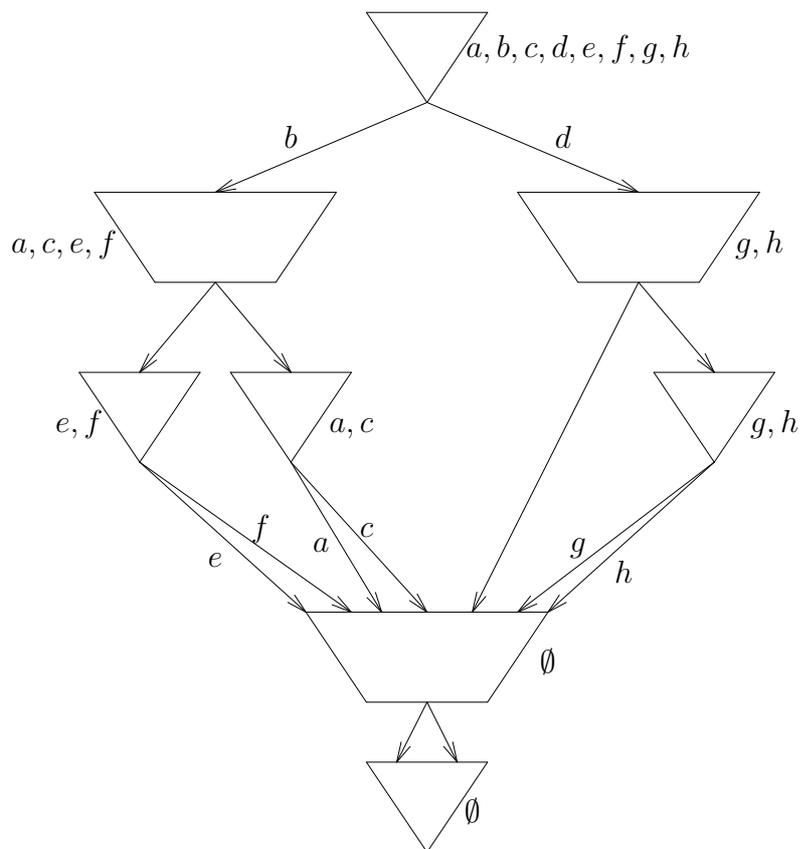
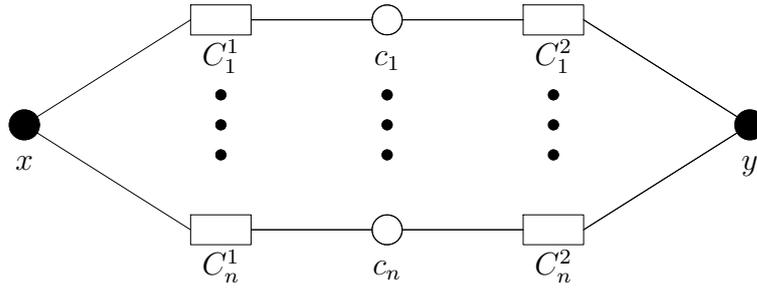


Abbildung 10: Eine DVLS zum Hex-Graphen aus Abbildung 6

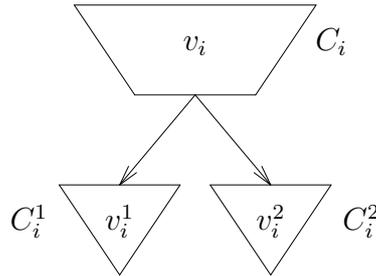


Dabei sind (siehe Abschnitt 3) $C_i^1, \{c_i\}$ und C_i^2 für alle i paarweise disjunkt. Außerdem gilt:

$$\bigcap_{i=1}^n C_i^1 \dot{\cup} \{c_i\} \dot{\cup} C_i^2 = \emptyset \quad \text{und}$$

$$\bigcup_{i=1}^n C_i^1 \dot{\cup} \{c_i\} \dot{\cup} C_i^2 = C.$$

Induktiv existieren (x, c_i) -DVLS mit Wurzeln v_i^1 und $\text{Carr}(v_i^1) = C_i^1$, sowie (c_i, y) -DVLS mit Wurzeln v_i^2 und $\text{Carr}(v_i^2) = C_i^2$. Bilde Spaltungsknoten v_i mit $\text{Carr}(v_i) = C_i := C_i^1 \dot{\cup} C_i^2$:

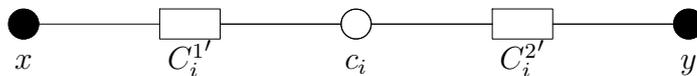


Bilde daraus die in Abbildung 11 gezeigte (x, y) -DVLS mit Träger $C = \bigcup_{i=1}^n C_i \dot{\cup} \{c_i\}$. \square

Satz 6.10 Wenn eine (x, y) -DVLS mit Wurzel w und $\text{Carr}(w) = C$ existiert, dann existiert auch eine Virtuelle Verbindung (x, C', y) mit $C' \subseteq C$.

Beweis: Der Beweis erfolgt induktiv über den Aufbau von DVLS. Im Induktionsanfang hat w keine Söhne. Da eine (x, y) -DVLS vorliegt, müssen x und y benachbart sein. In diesem Fall ist (x, \emptyset, y) nach Definition eine Virtuelle Verbindung. Es gilt $\emptyset \subseteq C$.

Hat w Söhne ($n > 0$ an der Zahl), so habe die DVLS die in Abbildung 11 gezeigte Struktur. Induktiv gibt es dann für alle $i \in \{1, \dots, n\}$ Virtuelle Verbindungen $(x, C_i^{1'}, c_i)$ und $(c_i, C_i^{2'}, y)$ mit $C_i^{j'} \subseteq C_i^j$ ($j \in \{1, 2\}$). Insbesondere gilt daher auch $C_i^{1'} \cap C_i^{2'} = \emptyset$:



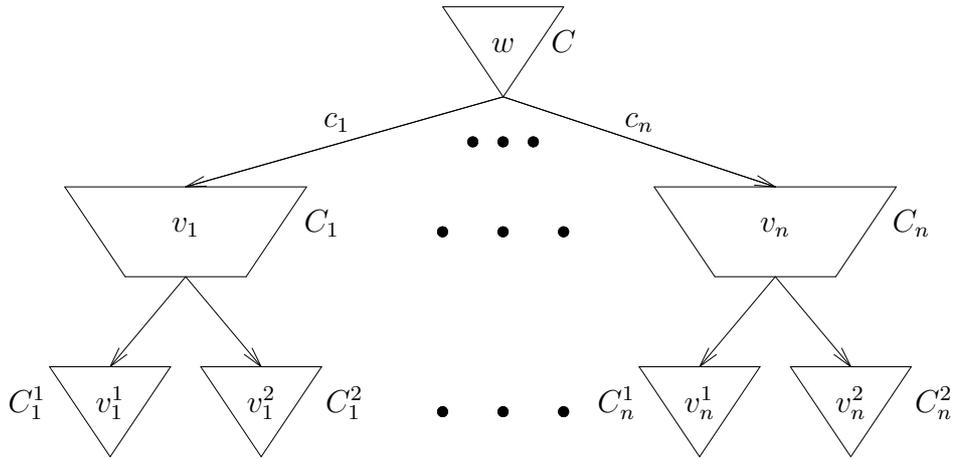
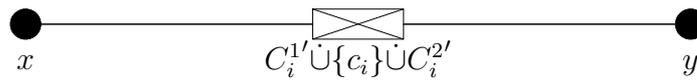


Abbildung 11: Eine DVLS

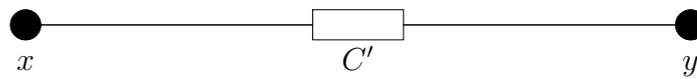
Mittels der AS-Regel lassen sich dann für alle i Virtuelle Semi-Verbindungen von folgender Form ableiten:



Es gilt

$$\begin{aligned} \bigcap_{i=1}^n C_i^{1'} \cup \{c_i\} \cup C_i^{2'} &\subseteq \bigcap_{i=1}^n C_i^1 \cup \{c_i\} \cup C_i^2 \\ &= \bigcap_{i=1}^n C_i \cup \{c_i\} \\ &= \emptyset. \end{aligned}$$

Folglich lässt sich mittels der OR-Regel die Virtuelle Verbindung (x, C', y) ableiten:



Dabei gilt wie gefordert

$$\begin{aligned} C' &= \bigcup_{i=1}^n C_i^{1'} \cup \{c_i\} \cup C_i^{2'} \\ &\subseteq \bigcup_{i=1}^n C_i^1 \cup \{c_i\} \cup C_i^2 \\ &\subseteq C. \end{aligned}$$

□

Korollar 6.11 Zwei Felder x, y sind genau dann virtuell verbunden, wenn es eine (x, y) -DVLS gibt.

Satz 6.12 $HEXY \in \text{PSPACE}$.

Beweis: Sei (V, E, β, x, y) das Quintupel, dessen Mitgliedschaft in $HEXY$ getestet werden soll.

Die Eigenschaften 1. – 3. aus Definition 4.2 überprüft man durch Raten einer entsprechenden Hex-Spielsituation.

Zur Überprüfung, ob x und y virtuell verbunden sind, entfernt man zunächst aus dem Hex-Graphen (V, E, β) die besetzten Felder außer den beiden Randfeldern nach dem Verfahren aus Abschnitt 5.2. Es entsteht ein neuer Hex-Graph (V', E', β') . Mit Korollar 6.11 genügt es jetzt zu überprüfen, ob eine (x, y) -DVLS mit Träger $V' \setminus \{x, y\}$ existiert. Dazu verwenden wir den nicht-deterministischen Algorithmus 6.13.

Algorithmus 6.13 *check-DVLS*

function check-DVLS(V, E, β, C, x, y) **return** boolean

(* Das Tripel (V, E, β) bildet einen Hex-Graphen mit $\beta(V \setminus \{x, y\}) = \{f\}$. *)

(* Es wird zurückgegeben, ob es eine (x, y) -DVLS mit Träger C gibt. *)

var

n, i : $\{1, \dots, |V|\}$

c : array of V

C^1, C^2 : array of 2^V

begin

if $\{x, y\} \cap C \neq \emptyset$ **then return** false

elsif $\{x, y\} \in E$ **then return** true

endif

guess $n, c_1, \dots, c_n, C_1^1, \dots, C_n^1, C_1^2, \dots, C_n^2$ **such that**

$\bigcup_{i=1}^n (\{c_i\} \cup C_i^1 \cup C_i^2) \subseteq C$

$\bigcap_{i=1}^n (\{c_i\} \cup C_i^1 \cup C_i^2) = \emptyset$

$\forall i \in \{1, \dots, n\} : c_i \notin C_i^1 \cup C_i^2$

$\forall i \in \{1, \dots, n\} : C_i^1 \cap C_i^2 = \emptyset$

else return false

endguess

for $i := 1$ **to** n **do**

if not check-DVLS($V, E, \beta, C_i^1, x, c_i$) **or not** check-DVLS($V, E, \beta, C_i^2, c_i, y$)

then return false

endif

endfor

return true

end

Die Korrektheit von Algorithmus 6.13 ergibt sich aus den Überlegungen dieses Abschnitts. Sein Platzbedarf ist polynomiell beschränkt:

Sei $n := |V|$. Die C_i^j benötigen jeweils $O(n)$ Platz. Von ihnen gibt es $O(n)$ viele. Daher benötigt eine Rekursionsstufe $O(n^2)$ Platz. Die Rekursion bricht nach $O(n)$ rekursiven Aufrufen ab, weil zumindest ein Feld (nämlich c_i) in jeder Stufe entfernt wird. Insgesamt liegt der Platzbedarf also in $O(n^3)$. \square

Bemerkung 6.14 Aus Algorithmus 6.13 kann man (z.B. mit dem Satz von Savitch) einen deterministischen polynomiell platzbeschränkten Algorithmus entwickeln.

7 Die PSPACE-Härte von HEXY

Wir wollen in diesem Abschnitt zeigen, dass *HEXY* PSPACE-hart ist. Mit Satz 6.12 folgt damit, dass *HEXY* PSPACE-vollständig ist.

7.1 Vorwärts-verbindende LS

Als Vorbereitung für die PSPACE-Härte von *HEXY* definieren wir *vorwärts-verbindende lokale Strategien (VVLS)* als Spezialfall von direkt-verbindenden lokalen Strategien (DVLS, siehe Definition 6.8).

Definition 7.1 (vorwärts-verbindende lokale Strategie, VVLS)

Eine LS mit Wurzel w heißt (x, y) -vorwärts-verbindend genau dann, wenn eine der beiden folgenden Bedingungen erfüllt ist:

1. Die Wurzel w hat keine Söhne und es gilt $\{x, y\} \in E$.
2. Die Wurzel w hat $n > 0$ Spaltungsknoten s_i als Söhne, sodass
 - (a) die Kanten zu den Söhnen mit c_i markiert sind und
 - (b) die Spaltungsknoten s_i jeweils zwei LS mit Wurzeln v_i^1 bzw. v_i^2 als Söhne haben und
 - (c) die v_i^1 keine Söhne haben und $\{x, c_i\} \in E$ gilt und
 - (d) die v_i^2 Wurzeln von (c_i, y) -vorwärts-verbindenden LS sind.

Es ist klar, dass VVLS auch DVLS sind, da Knoten v_i^1 , die Eigenschaft 2.(c) erfüllen, auch Wurzeln von (trivialen) (x, c_i) -direkt-verbindenden LS sind.

Die Abbildungen 12 und 13 stellen beispielhaft einen Hex-Graphen und eine (x, y) -vorwärts-verbindende lokale Strategie dar.

Man kann sich (x, y) -vorwärts-verbindende LS so vorstellen, dass sich der Spieler von x ab einen einzigen Pfad aufbaut, den er im Spielverlauf in Richtung y verlängert, bis er schließlich y erreicht. Erreicht der Pfad während des Spiels ein Feld z , d. h. existiert schon ein Pfad von x nach z , so besetzt der Spieler im

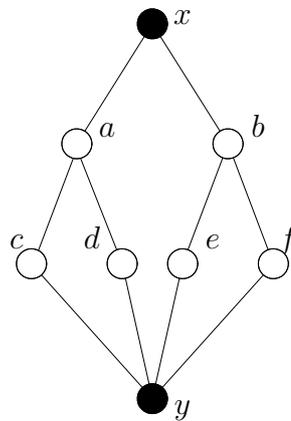


Abbildung 12: Ein Binärbaum als Hex-Graph

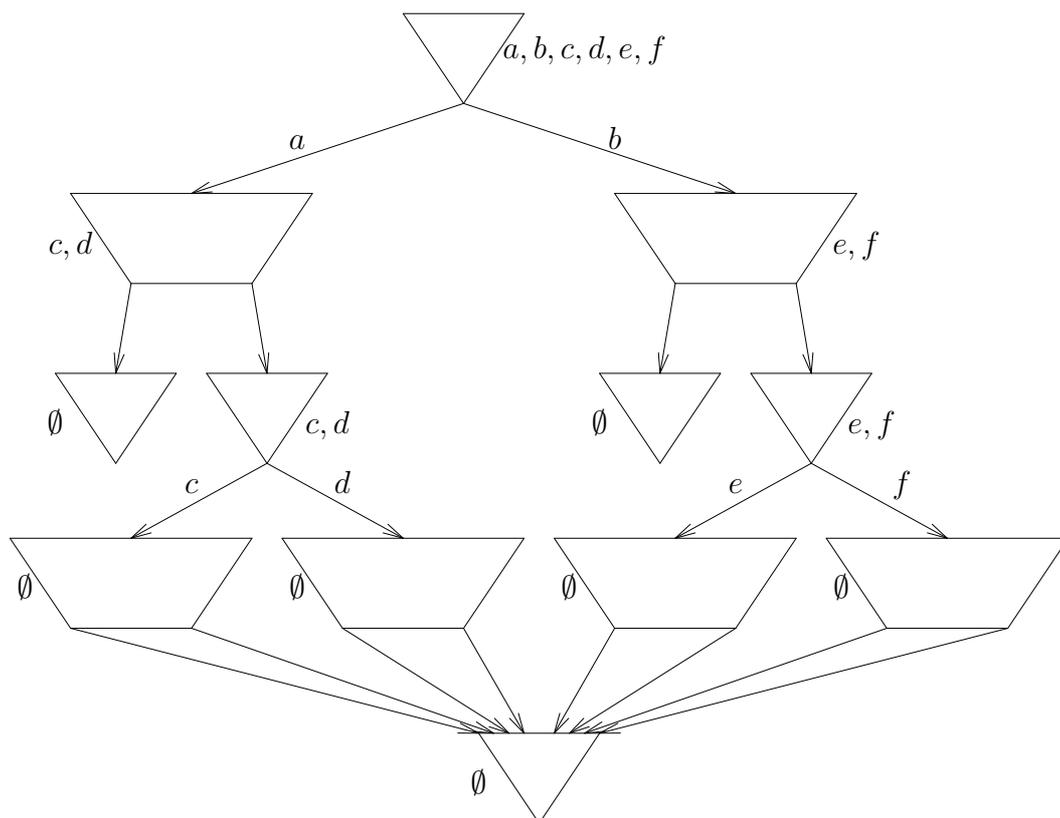


Abbildung 13: Eine vorwärts-verbindende lokale Strategie zu Abbildung 12

nächsten Zug ein Feld z' , das benachbart zu z ist. So verlängert er den Pfad, der dann von x bis z' reicht.

Wenn man in einer (x, y) -VVLS einen Teilbaum, der selbst eine (c_i, y) -VVLS ist, ersetzt durch eine andere (c_i, y) -direkt-verbindende LS, erhält man zwar keine (x, y) -VVLS, aber immer noch eine (x, y) -DVLS. Man kann solche Ersetzungen beliebig häufig durchführen, ohne dass die (x, y) -DVLS-Eigenschaft verloren ginge.

Wir werden in den folgenden Teilabschnitten solche DVLS angeben: Zunächst verfolgt der Spieler eine „ (x, y) -VVLS“ in Anführungszeichen, denn er verlängert nur bis zu einem gewissen Zeitpunkt in jedem Zug einen bereits existierenden Pfad von x nach z um ein zu z benachbartes Feld z' . Wir nennen das ab jetzt *Pfadverlängerungsstrategie*. So erreicht er (bis zu dem gewissen Zeitpunkt) einen Pfad von x bis zu einem z_k . Danach verfolgt er keine (z_k, y) -VVLS, wie es bei einer reinen VVLS der Fall wäre, sondern eine (z_k, y) -DVLS. Insgesamt verfolgt er so eine (x, y) -DVLS.

7.2 Die Reduktion von QBF auf HEX

In [Rei81] wird bewiesen, dass *HEX PSPACE*-vollständig ist. Für die *PSPACE*-Härte wird *QBF*, das bekannteste *PSPACE*-vollständige Problem, auf *HEX* reduziert: Wahre Formeln werden auf Gewinnstellungen des eigenen Spielers abgebildet, falsche auf Verlierstellungen.

Um zu beweisen, dass auch *HEXY PSPACE*-hart ist, übernehmen wir fast genau die Konstruktion aus [Rei81]. Wir zeigen also, dass wahre Formeln nicht nur auf Gewinnstellungen abgebildet werden, sondern auf Stellungen (Hex-Graphen), für die es eine (s, t) -DVLS gibt, wobei s und t die Ränder repräsentieren.⁹ Für die falschen Formeln müssen wir uns nicht interessieren, da sie (nach [Rei81]) auf Verlierstellungen abgebildet werden, die ja auf keinen Fall in *HEXY* liegen. Im Folgenden gehen wir immer vom Fall einer wahren Formel aus, die auf eine Gewinnstellung abgebildet wird.

Die Konstruktion in [Rei81] ist zu umfangreich, als dass sie hier vollständig wiedergegeben werden könnte. Wir bemühen uns aber, die Eigenschaften der Konstruktion, die für die *PSPACE*-Härte von *HEXY* von Bedeutung sind, explizit zu nennen.

7.3 Überblick über die Konstruktion

Die Konstruktion in [Rei81] verläuft zweistufig. Es werden in der ersten Stufe Hex-Graphen¹⁰ als Bauteile („Elementargraphen“) konstruiert, die dann in der zweiten Stufe zusammengesetzt werden. Auf beiden Stufen werden planare

⁹Wir übernehmen aus [Rei81] die Bezeichnungen s bzw. t für die Ränder.

¹⁰In [Rei81] werden im Gegensatz zu unserer Definition von Hex-Graphen benachbarte belegte Felder nicht verschmolzen. Wir übernehmen in diesem Abschnitt diese Darstellung.

Graphen erzeugt, sodass auch insgesamt ein planarer Graph entsteht. Weiterhin wird gezeigt, dass der entstehende Graph in ein Hex-Brett einbettbar ist. Bei unserem Beweis, dass es zu Gewinnstellungen DVLS gibt, werden wir uns an die Zweistufigkeit anlehnen.

In den Elementargraphen gibt es Eingänge und Ausgänge. Die Eingänge sind entweder mit Ausgängen anderer Elementargraphen oder mit s verbunden; die Ausgänge sind entweder mit Eingängen anderer Elementargraphen oder mit t verbunden.

In der Darstellung aus [Rei81] enthalten die Elementargraphen auch vom eigenen Spieler besetzte Felder. Wir übernehmen diese Darstellung der Übersicht halber. Wir können uns aber die besetzten Felder (nach dem Verfahren aus Abschnitt 5.2) wegdenken, damit wir auch von *benachbart* sprechen können, wenn zwei freie Felder über ein besetztes Feld miteinander verbunden sind. Wegen Satz 5.4 und weil wir nicht am Layout der Konstruktion interessiert sind, sind für uns beide Darstellungen gleichbedeutend. Besetzte Felder bringen uns also nur eine bessere Optik.

In [Rei81] ist die Sprache *HEX* etwas anders definiert als bei uns: Der eigene Spieler ist – anders als bei uns – am Zug. Dazu ändern wir die Konstruktion minimal – wie in Abbildung 14 gezeigt – ab. Die Einbettbarkeit in ein Hex-Brett bleibt davon unberührt.

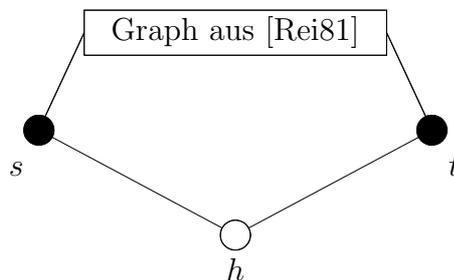


Abbildung 14: Änderung der Konstruktion aus [Rei81]

Auf diese Weise ist der Gegner gezwungen, in seinem ersten Zug auf das Feld h zu setzen, will er eine sofortige Niederlage vermeiden.

7.4 Überblick über die Gewinnstrategie

Die Gewinnstrategie ist zweiphasig.

Der eigene Spieler verfolgt in Phase 1 eine Pfadverlängerungsstrategie, d. h. er belegt in jedem Zug ein Feld, das benachbart zu dem Feld ist, das er im letzten Zug belegt hat.¹¹ In dieser ersten Phase verläuft das Spiel so, dass nacheinander (benachbarte) Elementargraphen bespielt werden. Die Spieler spielen dabei für

¹¹Am Anfang belegt er ein zu s benachbartes Feld.

einige Züge beide auf denselben Elementargraphen, bis sie dann beide das Spiel auf einem benachbarten Elementargraphen fortsetzen.

Genauer gesagt leitet der eigene Spieler die Wechsel der Elementargraphen ein. Dabei behält er seine Pfadverlängerungsstrategie bei. D. h. er verlängert seinen Pfad bis zum Ausgang des alten Elementargraphen, wartet den Zug des Gegners auf den alten Elementargraphen ab und macht dann den ersten Zug auf ein Eingangsfeld des neuen Elementargraphen, welches benachbart ist zu dem von ihm im letzten Zug besetzten Feld im alten Elementargraphen.

Die in dieser ersten Phase bespielten Elementargraphen werden im weiteren Verlauf des gesamten Spiels nicht wieder bespielt, nachdem das Spiel einmal auf einen benachbarten Elementargraphen übergegangen ist. Daher können wir bei einigen Überlegungen die jeweiligen Elementargraphen als bislang unbespielt annehmen, wenn das Spiel in den betreffenden Elementargraphen eintritt. Wir werden die Spielverläufe in Elementargraphen noch detaillierter untersuchen.

In jedem Elementargraph hat der Gegner die Möglichkeit, Phase 1 zu beenden. Tut er dies nie, kann der eigene Spieler schließlich Phase 1 beenden.

Wenn Phase 1 endet, hat der eigene Spieler seinen von s beginnenden Pfad bis zu einem s' geführt, welches über einen sogenannten „t-Pfad“ (Bezeichnung aus [Rei81]) (und evtl. über eine „Zweierbrücke“ mit t verbunden ist (siehe Abbildung 15).¹² In Phase 2 verfolgt der eigene Spieler dann eine (s', t) -DVLS. Wir werden diese Strategie explizit angeben.

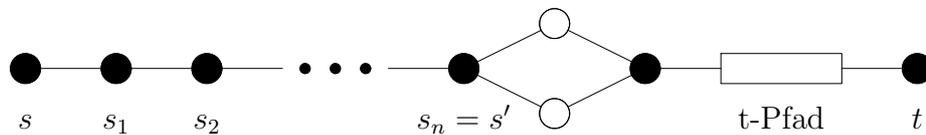


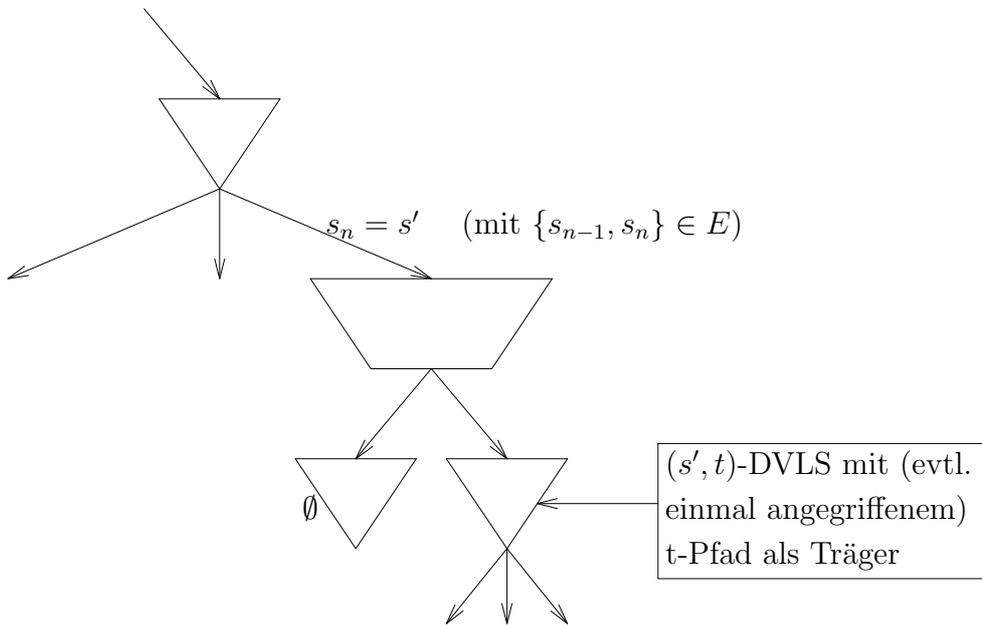
Abbildung 15: Die Situation beim Wechseln der Phasen

Wenn der Gegner in der ersten Phase von der in den letzten Abschnitten beschriebenen Spielweise abweicht und auf diese Weise Phase 1 beendet, muss man damit rechnen, dass er einen t-Pfad angreift. Dann kann der eigene Spieler seinen von s beginnenden Pfad in einem einzigen Zug zu einem s' verlängern, welches über einen (erweiterten) t-Pfad mit t verbunden ist. Wir werden das für jeden Elementargraphen nachweisen.

Im Allgemeinen müssen wir also damit rechnen, dass der t-Pfad, mit dem der eigene Spieler s' und t verbinden möchte, vom Gegner bereits angegriffen worden ist, dass der Gegner also bereits ein Feld des t-Pfads besetzt hat. Nachdem der eigene Spieler auf s' gesetzt hat, ist der Gegner ein weiteres Mal am Zug.

Abbildung 16 zeigt einen Ausschnitt der (s, t) -DVLS beim Übergang zwischen den beiden Phasen.

¹²Ist s' über eine Zweierbrücke mit dem t-Pfad verbunden, sagen wir auch, s' ist mit einem erweiterten t-Pfad verbunden.

Abbildung 16: Die (s, t) -DVLS beim Phasenwechsel

7.5 Die erste Phase

Wir untersuchen jetzt das Spiel in den einzelnen Elementargraphen. Dazu geben wir die Elementargraphen an und beschreiben jeweils, welche Situation vor Eintritt des Spiels in den betreffenden Elementargraphen vorliegt, welche Situationen beim Verlassen des Elementargraphen möglich sind und wer über den Fortgang entscheidet. Um zu zeigen, dass der eigene Spieler seine Pfadverlängerungsstrategie weiterführen kann, gehen wir jeweils auf mögliche Spielverläufe im Einzelnen ein.

Die Bezeichnungen der Elementargraphen stammen aus [Rei81].

7.5.1 Der Weiße Auswahlgraph

Abbildung 17 zeigt den Weißen Auswahlgraphen, einen Elementargraphen.

Situation vor Eintritt:

Das Spiel befindet sich in Phase 1. Der eigene Spieler hat von s bis zum Eingang E einen Pfad gelegt und ist jetzt wieder am Zug. Am Ausgang A_t befindet sich ein t-Pfad, der noch nicht bespielt wurde. An den Ausgängen A_1 und A_2 befinden sich Eingänge weiterer unbespielter Elementargraphen.

Mögliche Situationen beim Verlassen:

Der Gegner wählt, ob Phase 1 dadurch beendet wird, dass der eigene Spieler seinen Pfad bis zum Feld 2 bzw. 2a bzw. 7 bzw. 7a weiterführt, die (evtl. über eine „Zweierbrücke“) mit Ausgang A_t verbunden sind. Entscheidet sich der

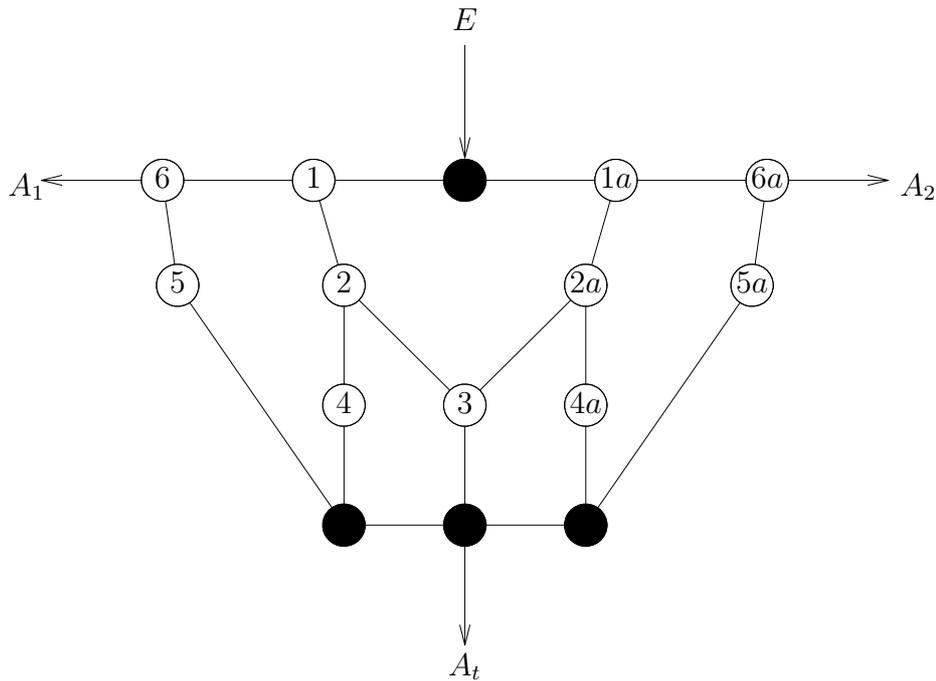


Abbildung 17: Der Weiße Auswahlgraph

Gegner für diese Option, kann er den t-Pfad einmal angreifen, noch bevor der eigene Spieler seinen Pfad zu Feld 2, 2a, 7 oder 7a verlängert hat. Entscheidet sich der Gegner gegen diese Option, wählt der eigene Spieler, ob er seinen Pfad zum Ausgang A_1 oder zum Ausgang A_2 weiterführt.

Spielverläufe im Elementargraphen:

Der eigene Spieler wählt, welchen Ausgang er ansteuert, falls der Gegner Phase 1 noch nicht beendet. O.B.d.A. wähle er Ausgang A_1 . Dazu spielt er auf Feld 1.

Falls der Gegner auf ein Feld außerhalb von $\{2, 3, 4\}$ spielt, antwortet der eigene Spieler mit Feld 2 und beginnt dann Phase 2.

Falls der Gegner aber auf ein Feld aus $\{2, 3, 4\}$ spielt, beantwortet dies der eigene Spieler mit der Verlängerung seines Pfades zu A_1 über Feld 6. Falls jetzt der Gegner nicht auf Feld 5 spielt, verlängert der eigene Spieler seinen Pfad über Feld 5 zum Ausgang A_t und beginnt dann Phase 2. Spielt der Gegner jedoch auf Feld 5, kann der eigene Spieler Phase 1 mit dem an A_1 angeschlossenen Elementargraphen fortsetzen.

7.5.2 Der Schwarze Auswahlgraph

Abbildung 18 zeigt den Schwarzen Auswahlgraphen, einen Elementargraphen.

Situation vor Eintritt:

Die Situation vor Eintritt ist identisch zur Situation vor Eintritt beim Weißen

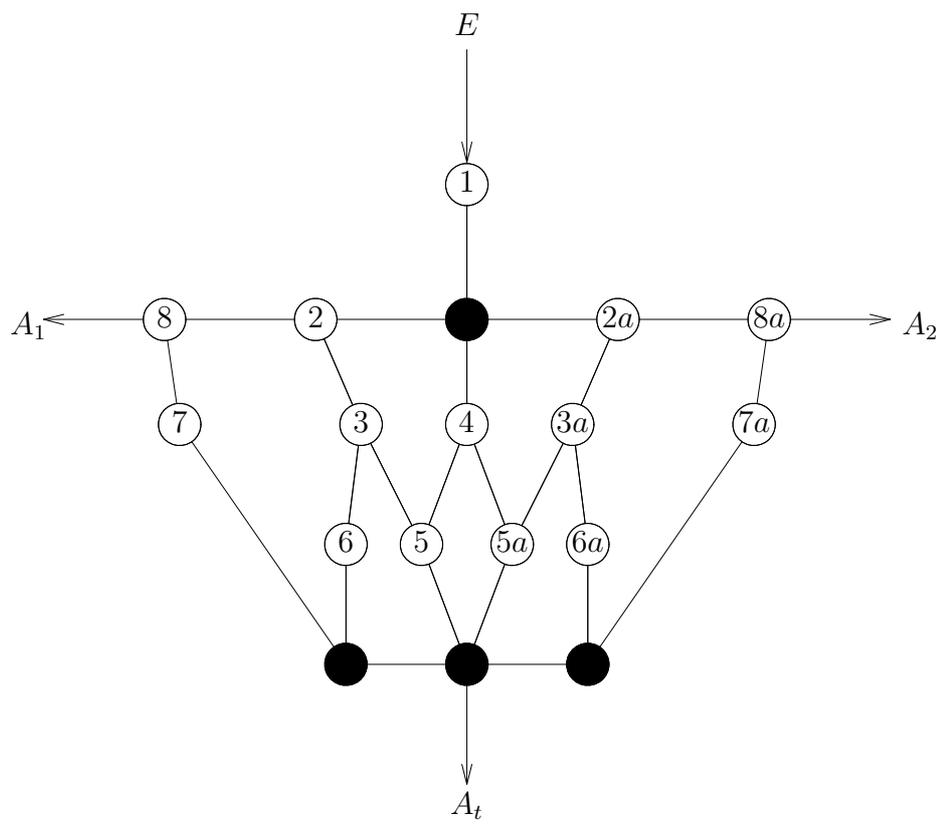


Abbildung 18: Der Schwarze Auswahlgraph

Elementargraphen.

Mögliche Situationen beim Verlassen:

Der Gegner wählt, ob Phase 1 dadurch beendet wird, dass der eigene Spieler seinen Pfad bis zum Feld 7 bzw. 7a bzw. 3 bzw. 3a bzw. 4 bzw. 4a weiterführt, das (evtl. über eine „Zweierbrücke“) mit Ausgang A_t verbunden ist. Entscheidet sich der Gegner für diese Option, kann er den t-Pfad einmal angreifen, noch bevor der eigene Spieler seinen Pfad zu Feld 7, 7a, 3, 3a, 4 oder 4a verlängert hat. Entscheidet sich der Gegner gegen diese Option, kann der Gegner wählen, ob der eigene Spieler seinen Pfad zum Ausgang A_1 oder zum Ausgang A_2 weiterführt.

Spielverläufe im Elementargraphen:

Zunächst spielt der eigene Spieler auf Feld 1. Danach sind drei Fälle zu unterscheiden:

- Falls der Gegner nun nicht auf Feld 4, 5 oder 5a spielt, besetzt der eigene Spieler Feld 4 und beginnt dann Phase 2.
- Falls der Gegner auf Feld 4 spielt, kann der eigene Spieler wählen, ob er A_1 oder A_2 ansteuert. O.B.d.A. wähle er Ausgang A_1 . Dazu spielt er auf Feld 2. Falls der Gegner nun auf ein Feld außerhalb von $\{3, 6, 5\}$ spielt, beantwortet dies der eigene Spieler mit Feld 3, um Phase 2 einzuleiten. Wenn der Gegner aber auf Feld 3, 6 oder 5 spielt, verlängert der eigene Spieler seinen Pfad um Feld 8. Falls der Gegner nun Feld 7 besetzt, kann der eigene Spieler Phase 1 mit dem an A_1 angeschlossenen Elementargraphen fortsetzen. Ansonsten besetzt der eigene Spieler das Feld 7, um danach mit Phase 2 zu beginnen.
- Ansonsten kann der Gegner wählen, ob der eigene Spieler seinen Pfad zum Ausgang A_1 oder zum Ausgang A_2 weiterführt. Entscheidet er sich (o.B.d.A.) für ersteres, besetzt er Feld 5a. Danach spielt der eigene Spieler auf Feld 2 und setzt fort wie im zweiten Fall beschrieben.

7.5.3 Der Treffpunktgraph

Abbildung 19 zeigt den Treffpunktgraphen, einen Elementargraphen.

Situation vor Eintritt:

Das Spiel befindet sich in Phase 1. Der eigene Spieler hat von s bis zum Eingang E_1 oder von s bis zum Eingang E_2 einen Pfad gelegt (o.B.d.A. ersteres) und ist jetzt wieder am Zug. Am Ausgang A_t befindet sich ein t-Pfad, der noch nicht bespielt wurde. Am Ausgang A_1 befindet sich der Eingang eines weiteren unbespielten Elementargraphen.

Mögliche Situationen beim Verlassen:

Der Gegner wählt, ob Phase 1 dadurch beendet wird, dass der eigene Spieler seinen Pfad bis zum Ausgang A_t weiterführt. Entscheidet sich der Gegner für diese Option, kann er den t-Pfad einmal angreifen, noch bevor der eigene Spieler

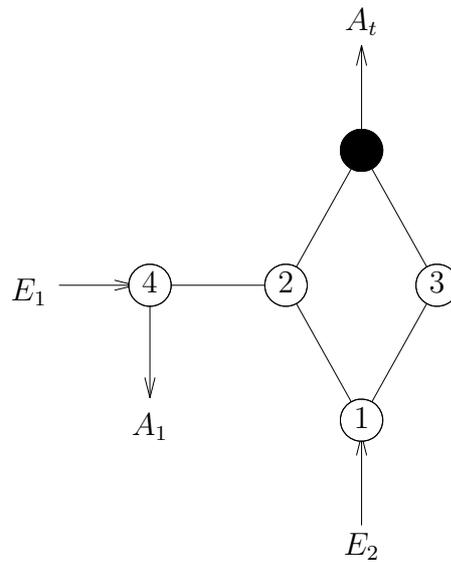


Abbildung 20: Der Entscheidungsgraph

Der Gegner wählt, ob Phase 1 dadurch beendet wird, dass der eigene Spieler seinen Pfad bis zum Ausgang A_t weiterführt. Entscheidet sich der Gegner für diese Option, kann er den t-Pfad einmal angreifen, noch bevor der eigene Spieler seinen Pfad zum Ausgang A_t verlängert hat. Entscheidet sich der Gegner gegen diese Option, führt der eigene Spieler seinen Pfad zum Ausgang A_1 weiter.

Spielverläufe im Elementargraphen nach Situation 1:

Der eigene Spieler spielt auf Feld 4.

Falls der Gegner nun nicht auf Feld 2 spielt, verlängert der eigene Spieler seinen Pfad über Feld 2 zum Ausgang A_t und beginnt danach Phase 2.

Falls der Gegner aber auf Feld 2 spielt, kann der eigene Spieler Phase 1 mit dem an A_1 angeschlossenen Elementargraphen fortsetzen.

Situation 2 vor Eintritt:

Das Spiel befindet sich in Phase 1. Der eigene Spieler hat von s bis zum Eingang E_2 einen Pfad gelegt und ist jetzt wieder am Zug. Am Ausgang A_t befindet sich ein t-Pfad, der noch nicht bespielt wurde.

Mögliche Situationen beim Verlassen, wenn zuvor Situation 2 eingetreten ist:

Phase 1 wird beendet. Der eigene Spieler ist über eine Zweierbrücke an den t-Pfad hinter Ausgang A_t angeschlossen.

Spielverläufe im Elementargraphen nach Situation 2:

Der eigene Spieler spielt auf Feld 1 und hat schon eine Zweierbrücke (über die Felder 2 und 3) zum Ausgang A_t . Danach beginnt Phase 2.

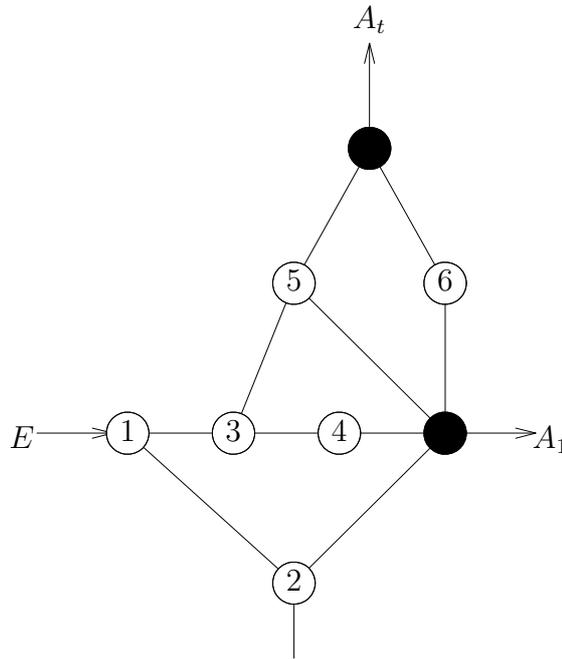


Abbildung 21: Der Kreuzungsgraph

7.5.5 Der Kreuzungsgraph

Abbildung 21 zeigt den Kreuzungsgraphen, einen Elementargraphen.

Situation vor Eintritt:

Das Spiel befindet sich in Phase 1. Der eigene Spieler hat von s bis zum Eingang E einen Pfad gelegt und ist jetzt wieder am Zug. Am Ausgang A_t befindet sich ein t-Pfad, der noch nicht bespielt wurde.

Mögliche Situationen beim Verlassen:

Der Gegner wählt, ob Phase 1 dadurch beendet wird, dass der eigene Spieler seinen Pfad bis zum Feld 2 bzw. 5 bzw. 6 weiterführt, das (evtl. über eine „Zweierbrücke“) mit Ausgang A_t verbunden ist. Entscheidet sich der Gegner für diese Option, kann er den t-Pfad einmal angreifen, noch bevor der eigene Spieler seinen Pfad zu Feld 2, 5 oder 6 verlängert hat. Entscheidet sich der Gegner gegen diese Option, führt der eigene Spieler seinen Pfad zum Ausgang A_1 weiter.

Spielverläufe im Elementargraphen:

Zunächst spielt der eigene Spieler auf Feld 1. Danach sind 3 Fälle zu unterscheiden:

- Falls der Gegner nun nicht auf Feld 2, 5 oder 6 spielt, besetzt der eigene Spieler Feld 2 und beginnt danach Phase 2, weil er über die Zweierbrücke der Felder 5 und 6 mit Ausgang A_t verbunden ist.

- Falls der Gegner auf Feld 5 oder 6 (o.B.d.A. Feld 5) spielt, belegt der eigene Spieler das Feld 2. Falls nun der Gegner noch Feld 6 besetzt, kann der eigene Spieler Phase 1 mit dem an A_1 angeschlossenen Elementargraphen fortsetzen. Falls der Gegner das Feld 6 nicht besetzt, tut dies der eigene Spieler, der damit seinen Pfad zum Ausgang A_t verlängert hat und danach mit Phase 2 beginnt.
- Falls der Gegner auf Feld 2 spielt, belegt der eigene Spieler Feld 3.
 - Falls nun der Gegner auf Feld 5 spielt, belegt der eigene Spieler Feld 4. Falls nun der Gegner auf Feld 6 spielt, kann der eigene Spieler Phase 1 mit dem an A_1 angeschlossenen Elementargraphen fortsetzen. Falls der Gegner nicht auf Feld 6 spielt, tut dies der eigene Spieler und beginnt dann mit Phase 2.
 - Falls der Gegner nicht auf Feld 5 spielt, tut dies der eigene Spieler und beginnt dann mit Phase 2.

7.5.6 Fazit von Phase 1

Phase 1 muss schließlich beendet werden, da es nur endlich viele unbespielte Elementargraphen gibt. Der eigene Spieler erreicht mittels einer Pfadverlängerungsstrategie den Anschluss an einen (erweiterten) t-Pfad, der vom Gegner höchstens mit der Belegung *eines* Feldes im t-Pfad angegriffen worden sein kann.¹³

7.6 Die zweite Phase

In Phase 2 werden s' und t mit einer DVLS über einen (erweiterten) t-Pfad verbunden. Abbildung 22 zeigt allgemein einen solchen erweiterten t-Pfad.

Man beachte, dass jede „Ebene“ des t-Pfades aus drei Kreuzungsgraphen besteht. Der gesamte t-Pfad kann einmal angegriffen worden sein, also sind auf jeder Ebene noch mindestens zwei Kreuzungsgraphen unbespielt. Weiterhin ist der Träger unserer (s', t) -DVLS nur eine Teilmenge der noch freien Felder. Insgesamt können wir uns auf den Graphen in Abbildung 23 beschränken.

Zu diesem Graphen geben wir jetzt explizit eine (s', t) -DVLS an. Dazu definieren wir für $1 \leq i \leq n$ (siehe Abbildung 23):

$$T_i := \{t_{i11}, t_{i12}, t_{i13}, t_{i21}, t_{i22}, t_{i23}\}$$

Die Abbildungen 24 bzw. 25 zeigen eine (r_i, r_j) -DVLS ($0 \leq i < j \leq n$) bzw. eine (t_{ie1}, r_j) -DVLS ($0 \leq i \leq j \leq n, e \in \{1, 2\}$).

Dabei bezeichnet $[xy]$ die Wurzel einer (x, y) -DVLS. Die eingerahmten Wertebereiche für die Parameter sind so zu verstehen, dass *für alle* Parameterwerte

¹³Im Falle eines erweiterten t-Pfades kann die Zweierbrücke nicht angegriffen worden sein.

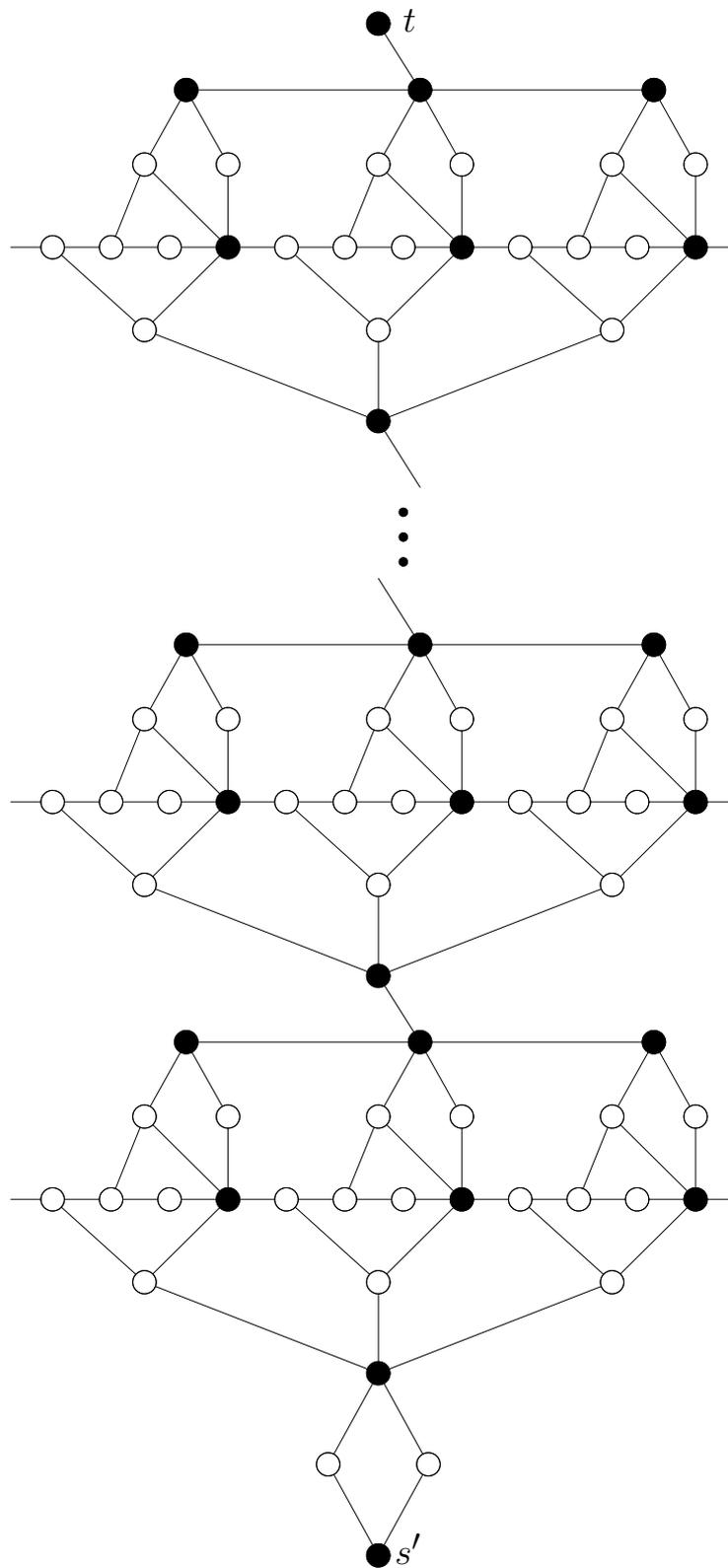


Abbildung 22: Erweiterter t -Pfad

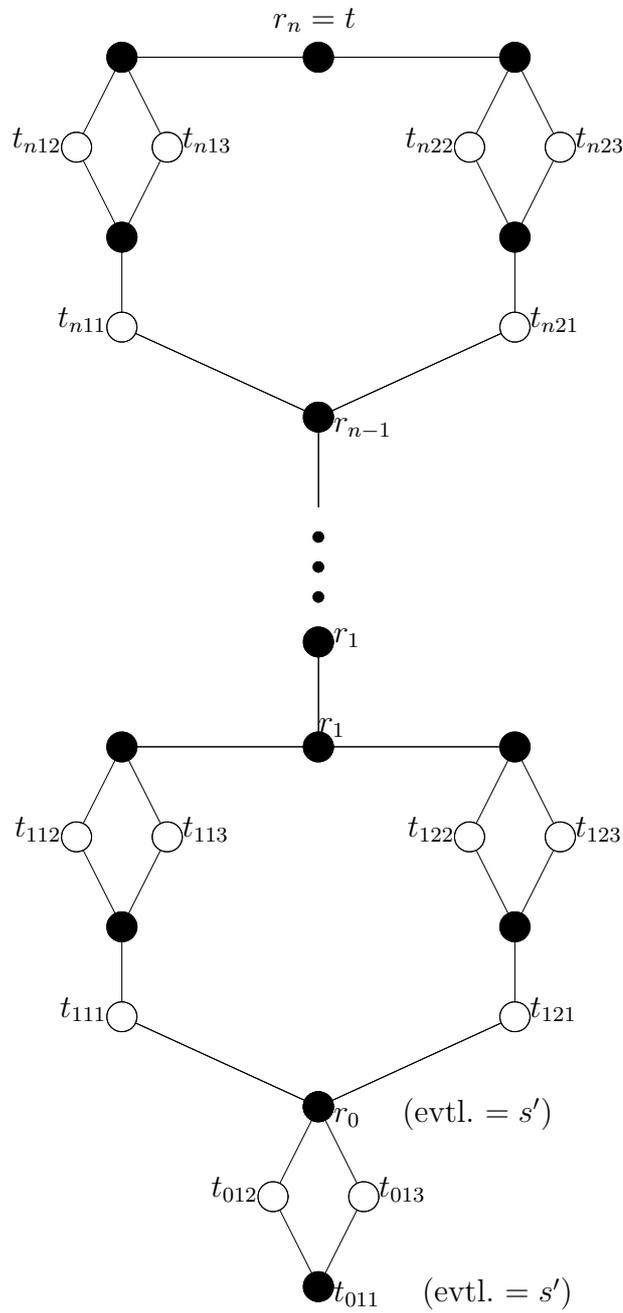


Abbildung 23: Wesentlicher Teil des t-Pfads

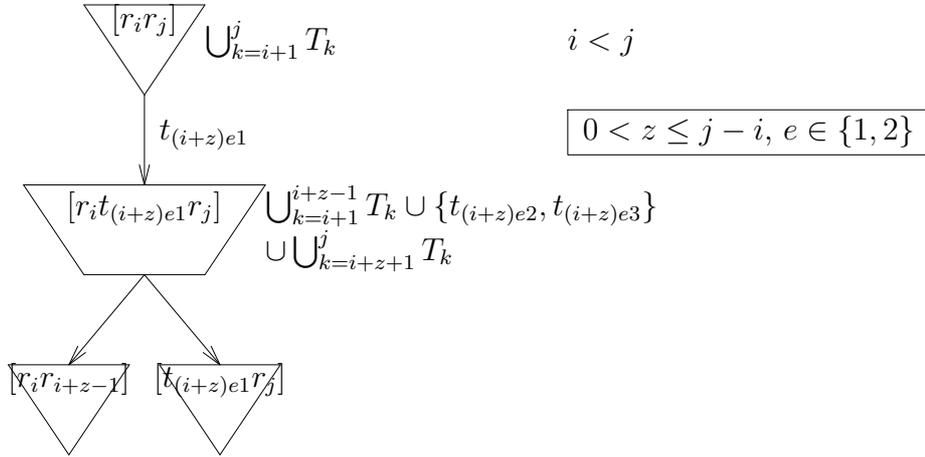


Abbildung 24: Die (r_i, r_j) -DVLS

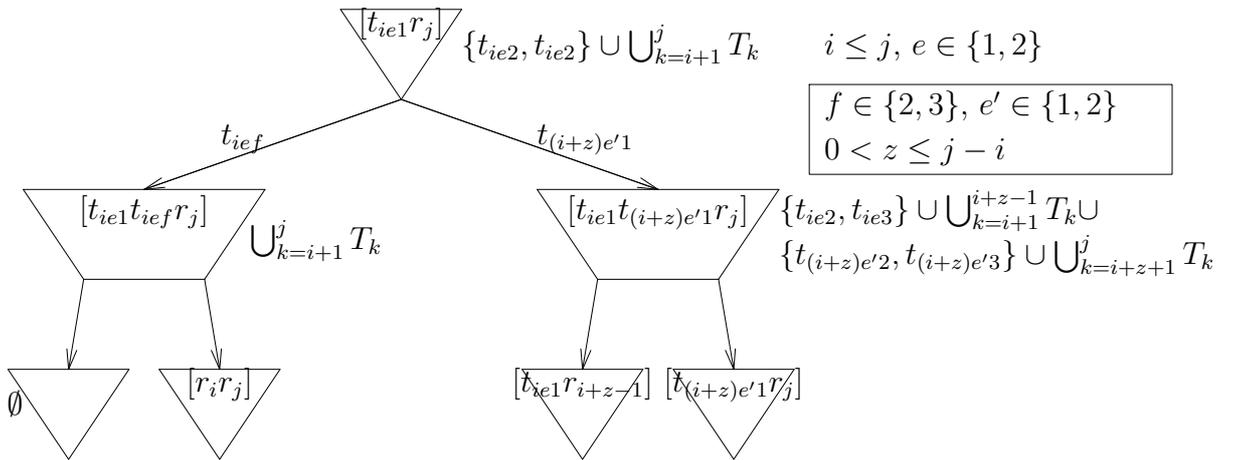


Abbildung 25: Die (t_{ie1}, r_j) -DVLS

entsprechend markierte Kanten zu entsprechend markierten Spaltungsknoten führen.

Für die Enkel der Wurzeln in den beiden Abbildungen sind entsprechende Strategien rekursiv einzusetzen. Im Rekursionsanfang sind die $[r_i r_i]$ Strategien mit leerem Träger und ohne Söhne.

Mit $s' = t_{011}$ oder $s' = r_0$ und $t = r_n$ ergibt sich die gesuchte (s', t) -DVLS.

Damit ist folgender Satz bewiesen:

Satz 7.2 *HEXY ist PSPACE-hart.*

Mit Satz 6.12 folgt:

Korollar 7.3 *HEXY ist PSPACE-vollständig.*

7.7 Weitere komplexitätstheoretische Überlegungen

Nach Korollar 4.7 verhalten sich die Mengen HEX und $HEXY$ zueinander wie in Abbildung 26 skizziert.

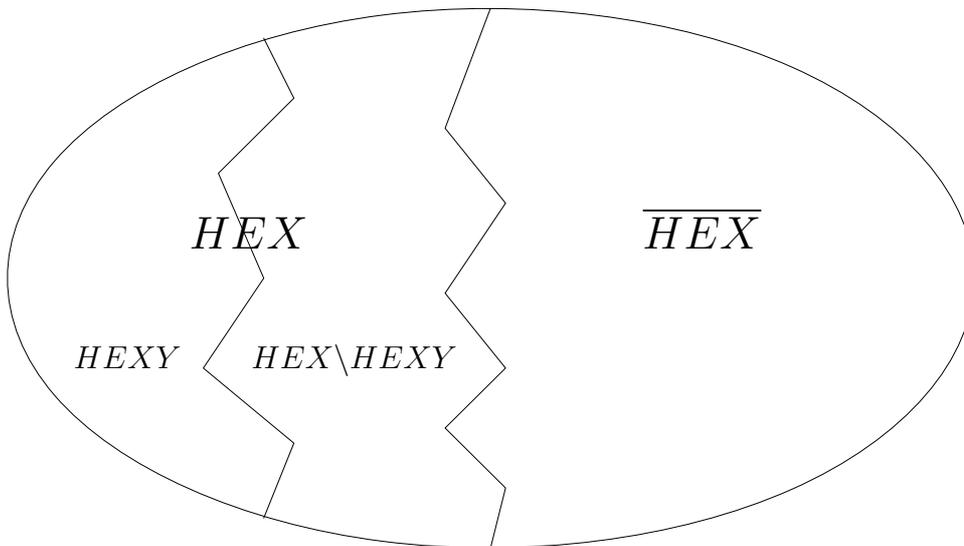


Abbildung 26: Die Sprachen HEX und $HEXY$

7.7.1 Sprachen zwischen $HEXY$ und HEX

In den letzten Abschnitten wurde gezeigt, dass $HEXY$ **PSPACE**-hart ist. Dazu wurde QBF derart auf $HEXY$ reduziert, dass wahre Formeln in $HEXY$ abgebildet werden und falsche in \overline{HEX} . Damit können wir auch folgendes Korollar folgern:

Korollar 7.4 *Alle Sprachen L mit $HEXY \subseteq L \subseteq HEX$ sind PSPACE-hart.*

Zur Illustration stelle man sich eine Erweiterung des Konzeptes Virtueller Verbindungen vor mit folgenden Eigenschaften:

- Jede Virtuelle Verbindung ist auch eine Erweiterte Virtuelle Verbindung.
- Sind zwei Felder x und y „erweitert virtuell verbunden“, gibt es eine (x, y) -verbindende NLS.

Die Menge, die dieses Konzept induziert (analog zu Virtuellen Verbindungen und $HEXY$), ist nach Korollar 7.4 ebenfalls PSPACE-hart.

Man kann Korollar 7.4 so verstehen, dass das Spiel Hex durch eine Konzentration auf Virtuelle Verbindungen aus komplexitätstheoretischer Sicht nicht einfacher wird.

7.7.2 Die Sprache $HEX \setminus HEXY$

Abbildung 26 wirft evtl. die Frage auf, ob die Menge $HEX \setminus HEXY$ ebenfalls PSPACE-hart ist. Wir können die Frage bejahen.

Satz 7.5 *Die Sprache $HEX \setminus HEXY$ ist PSPACE-hart.*

Für einen Beweis dieses Satzes muss man die Konstruktion für die PSPACE-Härte von $HEXY$ nur minimal abändern. Man betrachte dazu noch einmal Abbildung 14 auf Seite 33. Diese Konstruktion ändern wir wie in Abbildung 27 gezeigt.

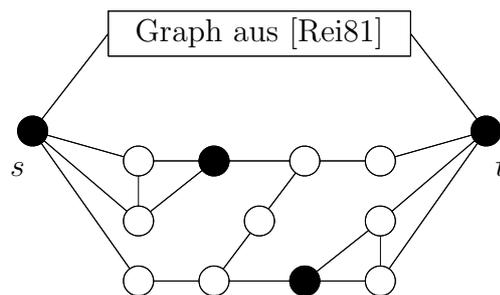


Abbildung 27: Änderung der Konstruktion

Man beachte, dass der untere Teil dieser Abbildung aus Abbildung 4 auf Seite 13 entnommen wurde.

Mit dieser Änderung werden weiterhin falsche Formeln in \overline{HEX} abgebildet, denn die Situation kann sich für den eigenen Spieler durch die Ersetzung des unteren Teilgraphen nur verschlechtern haben.

Wahre Formeln werden weiterhin in *HEX* abgebildet, denn wenn der Gegner in seinem ersten Zug auf den Graphen aus [Rei81] setzt, kann der eigene Spieler, wie im Beweis zu Satz 4.6 beschrieben, über den unteren Teilgraphen eine Verbindung herstellen. Tut der Gegner dies nicht, hat der eigene Spieler den ersten Zug im Graphen aus [Rei81] und gewinnt also.

Der Beweis, dass s und t nicht virtuell verbunden sind, verläuft analog zum Beweis zu Satz 4.6. Man beachte, dass der Gegner in seinem ersten Zug bereits alle Möglichkeiten, s und t über den Graphen aus [Rei81] zu verbinden, kappen kann, wenn er selbst beginnt.

8 Zusammenfassung

Wir haben einige neue Eigenschaften von Virtuellen Verbindungen gefunden. U. a. konnten wir zeigen, dass die AND-Regel in vielen Fällen aus theoretischer Sicht entbehrlich ist, wenn sie auch praktisch sein mag bei der tatsächlichen Berechnung Virtueller Verbindungen.

Mithilfe dieses Ergebnisses konnten wir die Strategie, mit der eine Virtuelle Verbindung realisiert wird, genau (und relativ einfach) charakterisieren (als direkt-verbindende lokale Strategie). Das lieferte uns zunächst die Zugehörigkeit von *HEXY* zu **PSPACE**.

Das zentrale Ergebnis ist die **PSPACE**-Vollständigkeit von *HEXY*. Dafür konnten wir wieder die Charakterisierung Virtueller Verbindungen durch Strategien verwenden und uns ansonsten stark an die Konstruktion aus [Rei81] anlehnen.

Damit ist – beziehentlich auf die Einleitung – die Komplexität einer H-Suche geklärt: Leider ist eine erschöpfende H-Suche im Allgemeinen nicht effizient möglich.

Dies tut jedoch der Schönheit und der praktischen Relevanz des Konzeptes Virtueller Verbindungen keinen Abbruch.

Literatur

- [ET75] Even, S.; Tarjan, R. E.: A combinatorial problem which is complete in polynomial space. 7th annual ACM Symposium on Theory of Computing, 66-71 (1975)
- [Rei81] Reisch, Stefan: Hex ist PSPACE-vollständig. Acta Informatica 15, 167-191 (1981)
- [Mue99] Müller, Martin: Decomposition search: A combinatorial games approach to game tree search, with applications to solving Go endgames. IJCAI-99, 578-583 (1999)
<http://www.cs.ualberta.ca/~mmueller/ps/ijcai1999.ps.gz>

- [YLP01] Yang, Jing; Liao, Simon; Pawlak, Mirek: On a decomposition method for finding winning strategy in Hex game.
<http://www.ee.umanitoba.ca/~jingyang/hexsol.pdf>
- [Ans02] Anshelevich, V.V.: A hierarchical approach to computer Hex. Artificial Intelligence 134, 101-120 (2002), ISSN 0004-3702
<http://home.earthlink.net/~vanshel/VAnshelevich-ARTINT.pdf>
- [Ste02] Stewart, Ian: Der Kampf um die Sechsecke. Spektrum der Wissenschaft September 2002, 118-119 (2002)
<http://www.wissenschaft-online.de/abo/spektrum/archiv/6266>
- [jav03] <http://www.cs.ualberta.ca/~javhar/hex/>

Erklärung

Hiermit versichere ich, diese Arbeit
selbständig verfasst und nur die
angegebenen Quellen benutzt zu haben.

(Stefan Kiefer)