Universität Paderborn

Fakultät für
Elektrotechnik, Mathematik und Informatik

Life ist unentscheidbar

Perlen der theoretischen Informatik WS2004/2005
Jens Golombek Matrikelnummer: 3690500
betreut durch Martin Ziegler
vorgelegt bei
Prof. Dr. Friedrich Meyer auf der Heide

Einleitung

Life ist nicht entscheidbar, zumindest das von dem Mathematiker John Horton Conway entworfene *Game of Life*. Beim *Game of Life* handelt es sich um einen zweidimensionalen zellulären Automaten, der nach sehr einfachen Regeln arbeitet. Interessanterweise ist es möglich, anhand der Regeln des *Game of Life* komplexe Strukturen zu erzeugen, mit denen sogar mathematische Probleme gelöst werden können. Noch viel verwunderlicher ist es, dass mit Hilfe des `Game of Life` ein Computer simuliert werden kann, der die Eigenschaften einer Turingmaschine besitzt.
Die vorliegende Seminararbeit, die im Rahmen des Seminars „Perlen der theoretischen Informatik“ der Arbeitsgruppe Algorithmen und Komplexität der Universität Paderborn entstanden ist, stellt diese Phänomene und ihre Ursachen dar.
Der Text besteht aus drei Kapiteln. Im ersten Kapitel werden die Regeln und einige Spielsituationen dargestellt. Das zweite Kapitel beschäftigt sich mit der Berechenbarkeit des *Game of Life*, und im dritten Kapitel wird die Erstellung eines *Life Computers*, und damit die Unentscheidbarkeit von Life gezeigt.

1 Das Spiel

Um Life nachvollziehen zu können müssen erst die Regeln bekannt sein. In diesem Kapitel werden die Regeln und einige Figuren, die bei Life entstehen können, dargestellt.

1.1 Die Spielregeln

Dass es sich bei dem *Game of Life* nicht um ein gewöhnliches Spiel handelt, zeigt die Tatsache, dass es keine Spieler gibt. Die einzige mögliche Interaktion ist die Bestimmung der Startkonfiguration. Gespielt wird Life auf einer unendlich großen quadratischen Fläche, die aus sogenannten Zellen besteht. Diese Zellen können zu jedem Zeitpunkt tot oder lebendig

sein. Das Spiel findet in Zeitintervallen t, beginnend mit $t = 0$, statt. Die Startkonfiguration ist die Bestimmung der zum Zeitpunkt $t = 0$ lebendigen Zellen. Jeder Zustand im Zeitpunkt $t + 1$ wird unerbittlich aus dem Zustand t unter der Anwendung folgender Regeln erzeugt.

Geburt: eine tote Zelle wird zum Zeitpunkt $t + 1$ lebendig wenn zum Zeitpunkt t genau drei der acht Nachbarn leben.

Tod durch Überbevölkerung: wenn eine Zelle zum Zeitpunkt t mehr als drei Nachbarn besitzt ist sie zum Zeitpunkt $t + 1$ tot.

Tod durch Unterkühlung: Eine lebendige Zelle, die einen Nachbarn zum Zeitpunkt t hat, ist zum Zeitpunkt $t + 1$ tot.

Überleben: Eine lebendige Zelle überlebt, wenn sie zum Zeitpunkt t zwei oder drei Nachbarn hat.

Diese Regeln lassen sich folgendermaßen zusammenfassen, wenn eine tote Zelle in ihrer nächsten Nachbarschaft drei Nachbarn besitzt, lebt sie. Ein lebende Zelle überlebt, wenn sie zwei oder drei Nachbarn besitzt.

Die Abbildung 1 zeigt als Startkonfiguration eine Linie bestehend aus fünf Zellen. Diese Linie wird unter der Beachtung der obigen Regeln zu einer Figur, die unter dem Namen *Traffic Lights* bekannt ist. Diese Figur verwandelt sich kontinuierlich im Intervall von 1 aus einem Kreis zu einem Kreuz und wieder zu einem Kreis.

Zum Zeitpunkt 0 haben die mittleren drei Zellen über der Linie drei Nachbarn und die drei Zellen unter der Linie ebenfalls drei Nachbarn, daher werden sie zum Zeitpunkt 1 lebendig. Die Zelle, die das rechte Ende der Linie sowie die Zelle, die das linke Ende der Linie darstellt, sterben zum Zeitpunkt 1, da sie nur einen Nachbarn besitzen. Daher wird aus der Linie zum Zeitpunkt 1 ein $3x3$ Zellen umfassendes Quadrat. Die inneren Elemente dieses Quadrates sterben aufgrund der Überbevölkerung, und nur die Ecken können überleben. Über der Mitte jeder Seite werden insgesamt vier neue Elemente geboren. Daraus wird dann zum Zeitpunkt 2 ein Kreis. Die Zellen diese Kreises überleben und in den Innenflächen entstehen vier neue Zellen. Daraus wird zum Zeitpunkt 3 ein jeweils um eine Zelle dickeres Kreuz. Dieses Kreuz verwandelt sich zum Zeitpunkt 4 zu einem Kreis, da zum Zeitpunkt 3 alle inneren Elemente unter Überbevölkerung leiden. Es bleiben nur die vier Äußeren lebendig, und an deren

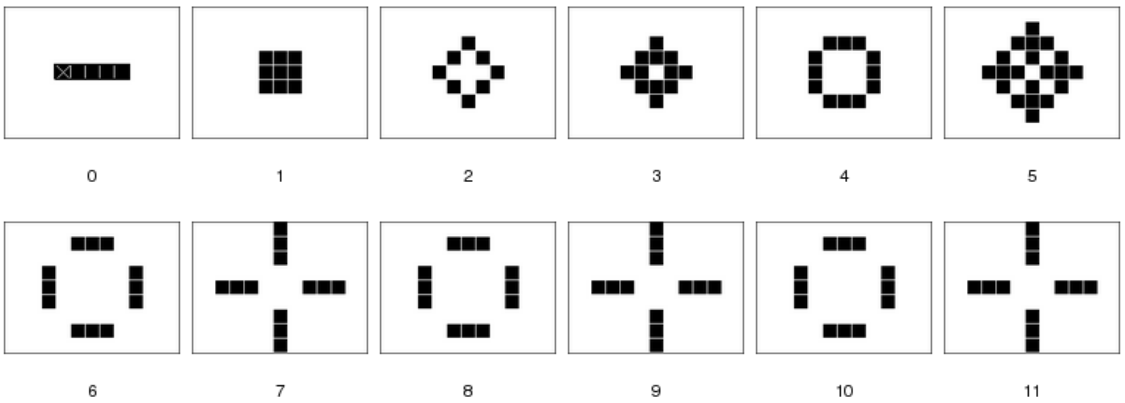

Abbildung 1: Eine Zeile mit fünf Zellen wird zum *Traffic Light*

Rändern entstehen jeweils zwei neue Zellen,so dass zum Zeitpunkt 4 ein kleiner Ring aus vier Linien mit drei Zellen entsteht. Um die Mitte dieser Linien bilden sich jeweils zwei neue Zellen, so dass die Figur zum Zeitpunkt 5 einem Kreis ähnelt. Daraus wird zum Zeitpunkt 6 ein großer Ring, der aus vier Linien mit drei Zellen besteht. Ein Linie, die aus drei Zellen besteht, ändert in jedem Zeitschritt ihre Ausrichtung von vertikal zu horizontal und umgekehrt. Die Figur zum Zeitpunkt 5 besteht aus vier *Blinker* und wird *Traffic Light* genannt.

Wenn, wie im obigen Beispiel, drei Zellen überleben, kann sich unter anderem ein Blinker bilden. Neben dem Blinker werden in der Abbildung 2 zwei weitere Figuren, die aus drei Zellen entstehen, dargestellt.

1.2 Einfache Konfigurationen

In der Abbildung 2 sind drei mögliche Schicksale von drei überlebenden Zellen dargestellt. Dabei sind die Zellen verschwunden, oszillieren oder wurden zu stabilen Konfigurationen. Im folgenden werden die stabilen und die oszillierenden Konfigurationen sowie noch eine viertes neues Schicksal dargestellt.

1.2.1 Stabile Konfigurationen

Neben der, aus der Abbildung 2 bekannten stabilen Konfiguration *Block*, die aus drei Zellen entsteht, gibt es gibt noch weitaus mehr stabile Konfigurationen. Meist ähneln diese einem Kreis. In der Abbildung 3 werden einige stabile Konfigurationen mit ihrem Namen dargestellt.

1.2.2 Oszillierende Life Konfigurationen

Die in 1.1 dargestellte Figur des *Blinkers* ist das einfachste Beispiel einer Struktur, die immer zwischen Figuren wechselt, also oszilliert. Neben dem *Blinker* gibt es noch unzählige Strukturen mit diesen Eigenschaften, einige davon werden in der Abbildung 4 dargestellt.

1.2.3 Glider und andere Raumschiffe

Neben den stabilen und oszillierenden Konfigurationen gibt es auch Konfigurationen, die über das Spielfeld wandern. Die einfachste bekannte Konfiguration ist der sogenannte *Glider*, der in Abbildung 5 dargestellt wird. Dieser *Glider* besitzt die Eigenschaft, dass die 4. Generation der 0. Generation, mit der Ausnahme, dass sich der *Glider* um eine Zelle diagonal nach rechts unten bewegt hat, gleicht. Die Ordnung der Zellen in der $2., 6., 12., \cdots$ Generation ist symmetrisch zu der Ordnung in der $0., 4., 8., \cdots$ Generation. Diese Symmetrie wird *glide reflexion* genannt, weswegen diese Figur *Glider* genannt wird. Es gibt noch weitere Figuren, die sich über das Spielfeld bewegen können, wie beispielsweise Raumschiffe. Raumschiffe bewegen sich horizontal oder vertikal über das Spielfeld. Es gibt *lightwight*, *middlewight* und *heavywight* Raumschiffe, die sich in der Anzahl der Zellen unterscheiden. Die Abbildung 6 zeigt ein *lightwight* Raumschiff, das sich horizontal von links nach rechts über das Spielfeld bewegt.

In diesem Kapitel sind die Regeln und einige einfache Figuren von Life dargestellt worden. Dabei wurden einige Eigenschaften des Game of Life vorgestellt. Im folgenden Kapitel wird der Frage, der Berechenbarkeit der Zukunft von Life Figuren nachgegangen.

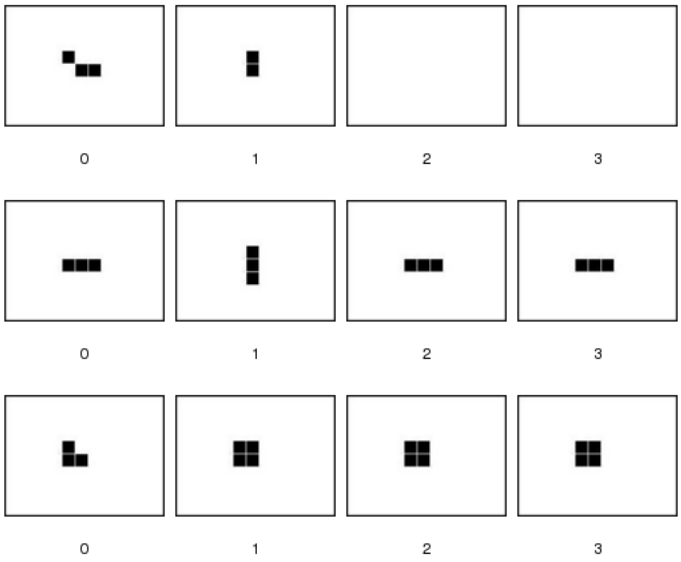

Abbildung 2: Drei überlebende Zellen erzeugen entweder einen Blanker, Blinker oder einen Block

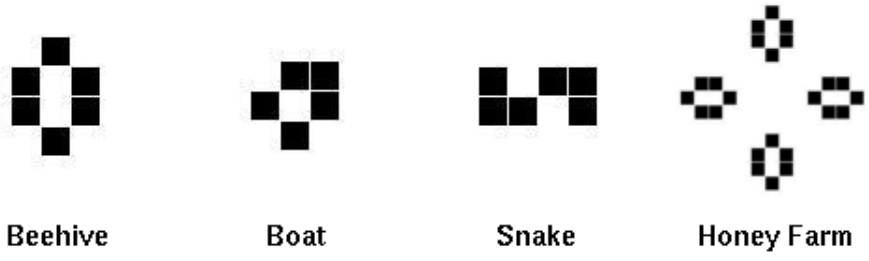

Abbildung 3: Einige stabile Konfigurationen

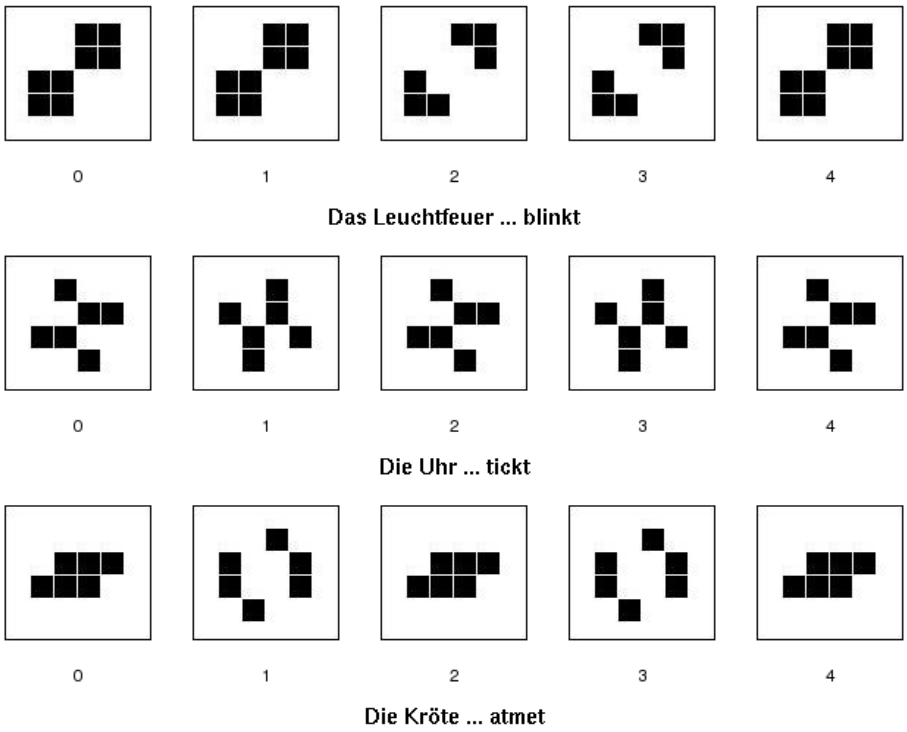

Abbildung 4: Drei oszillierende Lifezyklen mit einer Periode von zwei

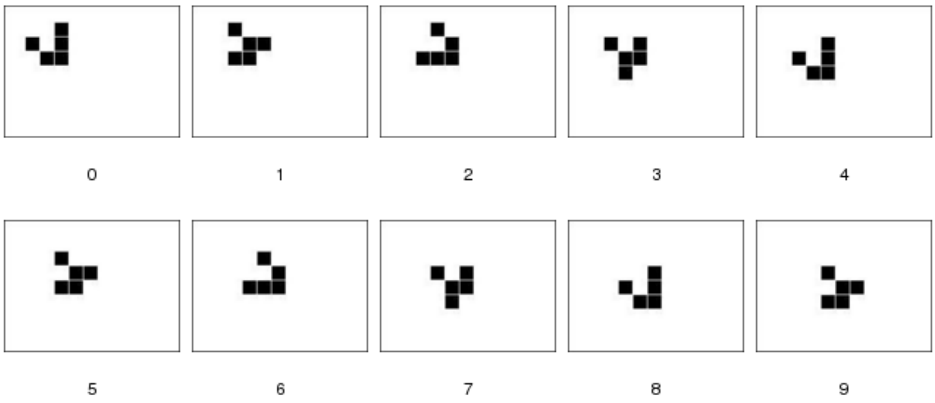

Abbildung 5: Ein Glider, der sich in vier Generationen um eine Zelle diagonal vorwärtsbewegt.

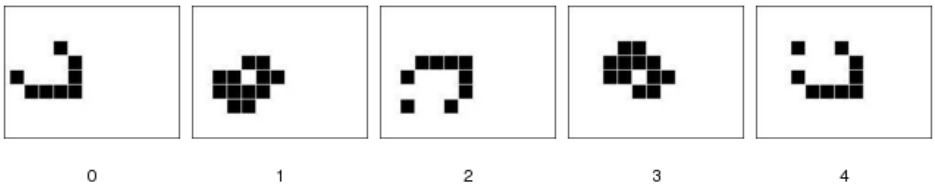

Abbildung 6: Ein lightwight Raumschiff, dass sich von links nach rechts bewegt.

2 Berechenbarkeit von Life

Die Zukunft einer Life Konfiguration kann, wie im vorigen Kapitel gesehen, wenn sie nicht vollkommen verschwindet, entweder eine statische Einheit, eine oszillierende Figur oder eine sich ständig fortbewegende Einheit sein. Aus dieser Tatsache ergibt sich die Frage, ob es möglich ist, die Zukunft einer Startkonfiguration vorherzusagen. Um diese Frage beantworten zu können, werden in den folgenden Abschnitten einige Konfigurationen und deren Lebensläufe betrachtet, danach wird der Frage nachgegangen, ob es eine Konfiguration gibt, die niemals aus einer anderen Konfiguration entstanden sein kann, also ein *Garten Eden*. Abgeschlossen wird das Kapitel mit der Antwort auf die Frage, ob eine Life Konfiguration ohne Grenzen wachsen kann.

2.1 Vorhersagbarkeit von Life Mustern

Ein möglicher Ansatz eine Berechnungsvorschrift für Life Muster zu finden, ist die Beobachtung des Schicksals einfacher Figuren. Eine einfache Figur kann beispielsweise eine gerade Linie aus n-Zellen sein. Bei $n = 1$ oder $n = 2$ verschwindet die Linie. Eine Linie mit 3 Elementen wird zu einem ,aus dem Kapitel 1.1 bekannten, *Blinker* und eine Linie mit $n = 4$ wird zum Zeitpunkt 2 zu einem *Beehive*. Aus der Länge $n = 5$ wird, wie in Abbildung 1 zu sehen, ein *Traffic Light*. Die Linie mit $n = 6$ verschwindet zum Zeitpunkt $t = 12$. Bei $n = 8$ ergeben sich vier Blöcke und vier *Beehives*. Dies kann beliebig so weitergeführt werden, ohne dass eine generelle Aussage getroffen werden kann.
Auch bei der Untersuchung anderer einfacher Muster, wie *Golombs Pentamonioes*, die aus 12 miteinander verbundenen Regionen mit jeweils 5 Zellen bestehen, ist es schwierig oder gar unmöglich, eine generelle Aussage zu treffen.

2.2 Garten Eden

Aus dem vorigen Abschnitt geht hervor, dass es schwer oder unmöglich ist das Schicksal von Life Konfigurationen vorherzusagen. Nun stellt sich die Frage, ob es eine Konfiguration gibt, die nur als Anfangszustand existieren kann, da sie aus keinem Vorgänger erzeugt werden kann. Der folgende Beweis zeigt, dass es diese Konfiguration gibt: Sei n genügend groß und es gibt eine Konfiguration in einem $5*n-2$ mal $5*n-2$ großen Quadrat, die keinen Vorfahren besitzt. Nun kann zu jedem der $5x5$ großen Quadrate, in denen sich keine Zelle befindet, eine Zelle in der Mitte eingefügt werden, ohne die nachfolgenden Generationen zu beeinflussen. Daher müssen nur $(2^25-1)^{n^2} = 2^{(24.9999999...*n^2)}$ der 2^25n2 Konfigurationen betrachtet werden.
In einem $5^{(5n-2)^2}$ zu $5^{(5n-2)^2}$ Quadrat gibt es $5^{(5n-2)^2} = 2^{(25n^2-20n+4)}$ Möglichkeiten Zellen zu platzieren. Nun muss nur noch ein n gefunden werden für das gilt, dass $24.99999... * n^2 < 25n^2 - 20n + 4$. Dies ist bei $n = 465163200$. Dies beweist, dass es Konfigurationen gibt, die aus keiner Vorgängerkonfiguration entstanden sein können. Die Abbildung 7 zeigt noch eine weitere Konfiguration, die nicht aus einer Vorgängerkonfiguration gewonnen werden kann.

2.3 Gospers Glider Gun

Bislang sind nur Life Konfigurationen betrachtet worden, die endlich wachsen, doch gibt es auch Life Konfigurationen, die unendlich groß werden können. Dies zeigt die Glider Gun von Gosper siehe Abbildung 8. Diese erzeugt alle 30 Generationen einen neuen Glider.
Damit wurde gezeigt, dass Life nicht berechenbar ist, da nicht für jede Eingabe eine Ausgabe

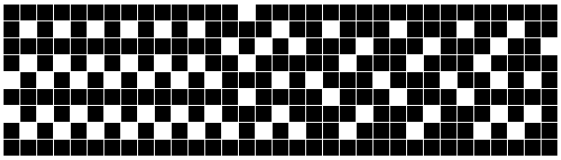

Abbildung 7: Garten Eden von Roger Banks, Mike Beeler, Rich Schroppel, Steve Ward,et al.

Abbildung 8: Gospers Glider Gun

berechnet werden kann. Die Frage nach dem ultimativen Ziel von Life hört sich nicht gerade mathematisch an, doch im folgenden Kapitel wird gezeigt, dass sich mit Hilfe von Life jedes gut definierte mathematische Problem berechnen lässt. Daher können die einfachen Probleme über Life beliebig kompliziert werden.

3 Ein Life Computer

In den letzten Jahrzehnten wurde das *Game of Life* auf den verschiedenen Computer-Plattformen implementiert. Doch mit Hilfe von Life lässt sich selber ein Computer erzeugen. In diesem Kapitel werden die Bauteile, die für die Erstellung eines Life Computers benötigt werden, dargestellt. Um die Bauteile zu erzeugen, wird erst einmal der Frage nachgegangen, wie mit Hilfe von Life mathematische Probleme gelöst werden können. Zusätzlich wird die Frage beantwortet, ob es einen Algorithmus gibt der entscheiden kann, ob eine Life Konfiguration verschwinden kann.

3.1 Ein endliches Zustandsmodell

Wie im vorigen Kapitel erwähnt, lassen sich mit Life wohl definierte mathematische Probleme lösen. Um ein mathematisches Problem zu lösen, wird ein Muster erstellt, das verschwindet, wenn das Problem gelöst ist oder es bleibt, wenn es sich nicht lösen lässt. Dieses Muster kann beliebig kompliziert werden und aus simulierten Computerbauteilen bestehen.
Ein Computer arbeitet aufgrund von elektronischen Signalen, die entweder 0 oder 1 sein können. Diese Signale können anhand von Gliedern, wie die Abbildung 9 zeigt, simuliert werden. Bei der Simulation einer Sequenz wird zu jedem Takt für jedes High-Signal ein Glider

erzeugt. Durch dieses Verfahren wird eine Folge von Bits dargestellt. Anhand der Glider entstehen auch die Kabelverbindungen, da sich Glider diagonal über das Spielfeld bewegen. Eine weitere Eigenschaft eines Computers ist die Uhr, die bestimmt, wann ein Signal gesendet oder empfangen wird. Dieser Takt kann anhand einer Glider Gun bestimmt werden, die alle 30 Generationen einen neuen Glider erzeugt. Die Tabelle 1 zeigt alle bislang besprochenen Komponenten eines Life Computers.

3.2 Komponenten eines Life Computers

Da anhand von Glidern die Signale eines Computers simuliert werden können, ist es möglich, bei der Konstruktion von Komponenten eines Life Computers ihre Eigenschaften auszunutzen. Eine der wichtigsten Eigenschaften ist das Verhalten von kollidierenden Glidern. Da die Anzahl von Anordnungen und Zeitpunkten des Zusammentreffens von Glidern enorm groß ist, ergibt sich eine große Anzahl an möglichen Geschehnissen. Glider können beispielsweise Blinker, Blöcke oder Pounds erzeugen. Sie können aber auch ganz verschwinden oder sich zu einem Eater, der beliebig viele Glider, aber auch Blinker oder Spaceships verschlingen kann ohne sich zu verändern, formieren.
Eine weiteres interessantes mögliches Resultat einer Kollision von Glidern ist die so genannte *Kickback Reaction*, dies bedeutet, dass ein Glider beim Auftreffen auf einem anderen Glider zurückgeworfen wird und der andere Glider verschwindet.
Durch eine bestimmte Kombination von Glidern kann auch wieder eine neue Glider-Gun erzeugt werden.
Zusammenfassend werden alle dynamischen Teile, wie die Eingaben des Life Computers, durch Gliderströme dargestellt. Diese Gliderströme werden anhand der *Kickback Reaction* oder der Zerstörungsreaktion verändert. Dagegen werden statische Teile anhand von Glider-Guns und Eatern gebildet.

3.2.1 Negation, Konjunktion und Disjunktion von Sequenzen

Die Zerstörung beider Glider aufgrund einer Kollision kann ausgenutzt werden, um die Negation einer Glidersequenz zu erzeugen. Die Eingabesequenz ist ein Strom von Glidern, die sich im gleichen Takt, wie ein von einer Glider-Gun erzeugter Gliderstrom, fortbewegt. Eine 1 wird durch einen Glider und eine 0 durch Fehlen eines Glider's ausgedrückt. Jeder Freiraum der Eingabesequenz lässt einen Glider durch, wogegen ein Glider aus der Eingabesequenz einen Glider der Glider-Gun sowie sich selbst zerstört. Dadurch wird die Eingabesequenz invertiert

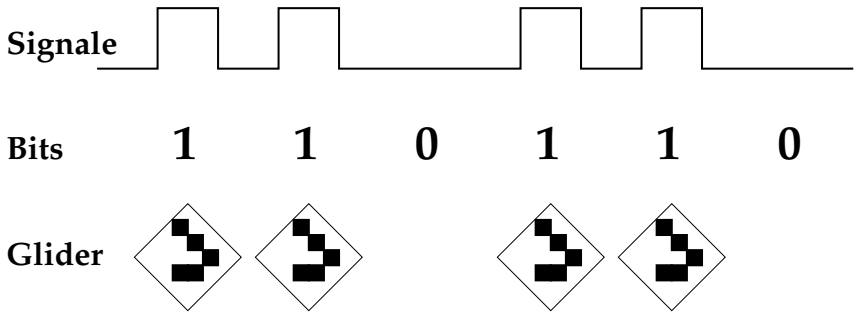

Abbildung 9: Darstellung von Signalen anhand von Glidern.

Komponente	Life Muster
Kabel	Diagonaler Weg der Glider
Signale	Glider
Uhr	Glider Gun
Logische Gatter	Menge von Glider Gun

Tabelle 1: Komponenten eines Life Computers

und gleichzeitig noch umgeleitet. In Abbildung 10 wird eine Eingabesequenz, bestehend aus 111011010101 zu einer Ausgabesequenz 000100101010. Da die Sequenz noch nicht ganz durchgelaufen ist, ist der Anfang mit 101010 zu erkennen. Nicht nur die Negation einer Sequenz kann mit Hilfe der Zestörungsreaktion von Glidern erzeugt werden, sondern auch eine Konjunktion sowie eine Disjunktion. In der Abbildung 11 wird ein UND- und ein ODER-Gatter dargestellt. Beide Gatter haben A und B als Eingabestrom. Ein UND-Gatter besteht aus einer Glider-Gun G und einem Eater E. Dagegen besteht ein Odergatter aus zwei Guns G. Die Darstellung wurde um 45° gedreht, um sie übersichtlicher zu gestalten.

3.2.2 Ausdünnen eines Gliderstroms

Die von den Glider-Guns erzeugten Gliderströme sind so dicht, dass sie nicht von einem Glider durchdrungen werden können. Um dies zu umgehen, muss ein Gliderstrom ausgedünnt werden. Um einen Gliderstrom Gs auszudünnen, werden zwei weitere Glider-Guns $G1$ und $G2$ sowie ein Eater E und ein neuer Glider g verwendet. Die Glider-Guns $G1$ und $G2$ werden parallel in entgegengesetzte Richtungen positioniert. Zwischen den beiden Gliderströmen befindet sich ein Glider, der sich aufgrund der *Kickback Reaction* zwischen dem Gliderstrom A, der von $G1$ erzeugt wird, und dem Gliderstrom B, der von $G2$ erzeugt wird, bewegt. Der Takt, in dem der Glider von A und B zurückgeworfen wird, wird so gewählt, das in jeder Runde ein Glider eines Gliderstroms entfernt wird. Aufgrund der Entfernung beider Ströme kann die Frequenz so bestimmt werden, dass jedes Nte Glied aus den Strömen A und B entfernt wird. Da der Gliderstrom A nicht mehr benötigt wird, wird er von einem Eater verschluckt. Der Gliderstrom B trifft senkrecht auf den Gliderstrom von Gs und erzeugt dort eine Zerstörungsreaktion. Nun kann jeder Nte Glider von Gs den Strom B passieren. Um die Phasen korrekt zu erzeugen, muß N durch vier teilbar sein. Nun kann ein zweiter ausgedünnter Strom den Strom von Gs passieren, ohne dass sich beide Ströme beeinflussen. Dieses Verfahren wird *Thining* gennant und wird in der Abbildung 12 dargestellt.

3.2.3 Duplizieren von Gliderströmen

Aus den Abbildungen 11 und 10 ist erkennbar, dass ein Gliderstrom nicht einfach negiert in UND/ODER-Gatter hineinfließen kann. Der Strom muß dafür umgeleitet werden. Dieses Problem kann mit dem Kopieren eines Stromes gelöst werden. Um einen Strom zu kopieren, werden die Eigenschaften eines, in seinen, von einer Glider-Gun erzeugten, Strom, zurückgeworfenen Gliders, ausgenutzt. Dafür wird ein ausgedünnter Gliderstrom, der alle 120 Generationen einen Glider besitzt, verwendet. Beim Auftreffen des ersten Glider's auf einen anderen Glider wird dieser wieder zurückgeworfen und formt mit seinem Nachfolger einen Block. Der dritte Glider vernichtet diesen Block und alle nachfolgenden können sich weiterbewegen.

Abbildung 10: Ein Not-Gatter. Die Eingabe 111011010101 auf der linken Seite wird negiert zu 000100101010 auf der rechten Seite

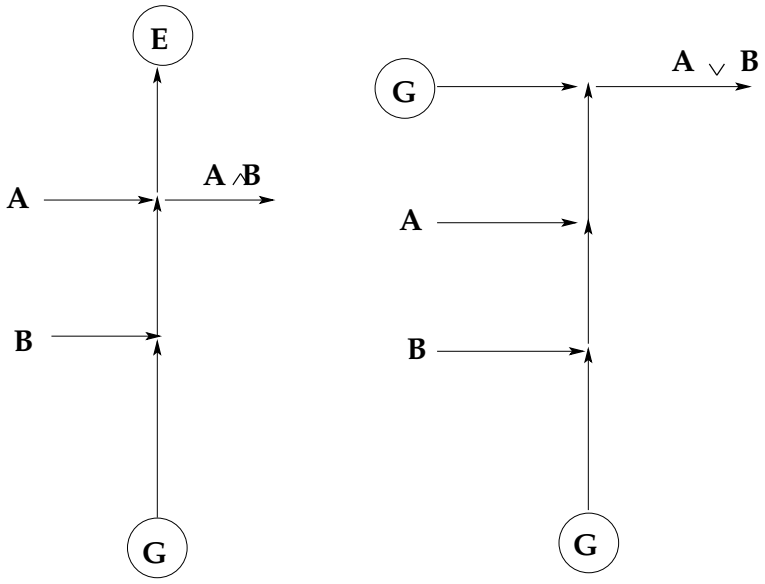

Abbildung 11: Ein UND und ein ODER-Gatter

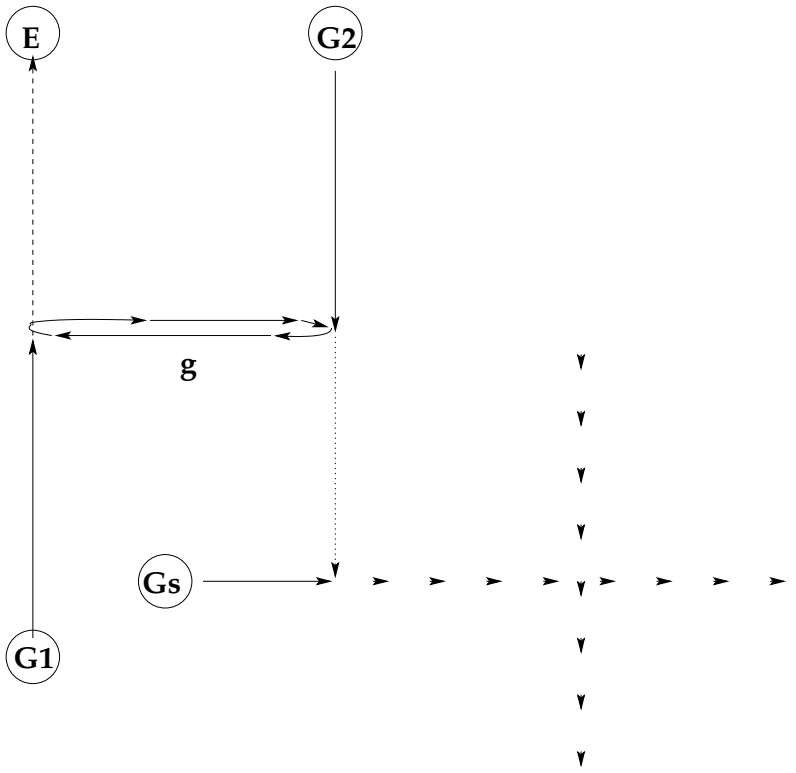

Abbildung 12: Ausdünnen eines Gliderstroms

Diese Eigenschaft kann folgendermaßen verwendet werden. Sei S ein informationstragender Strom, der an jeder zehnten Stelle eine Information beinhaltet und bei dem die anderen 9 Stellen leer sind. Dieser Strom wird mit einem Strom, der bis auf die 9. Stelle leer ist, verknüpft. Als Resultat dieser Verknüpfung befindet sich hinter jedem informationstragenden Block ein Glider. Auf Strom S trifft senkrecht ein Strom N, mit von eins bis zehn durchnummerierten Gliedern, so dass die Glider von N wieder zurückgeworfen werden. Wenn nun im ersten Zehntel von S eine Information A vorhanden ist, werden die ersten drei Glider von N und der informationstragende Glider zerstört. Ansonsten kann der erste Glider von N passieren und die nächsten drei Glider von N werden zerstört. Da sich hinter dem informationstragenden Glider noch ein Glider befindet wird die Information an der zweiten Stelle kopiert. Der Strom N trifft nun auf einen weiteren Strom bei dem sich an jeder zehnten Stelle ein Glider befindet. Dieser wird zerstört falls A nicht vorhanden ist. Da die vierte Stelle von N nur existiert, wenn A vorhanden ist, kann mit Hilfe eines Stromes, der einen Glider an der vierten Stelle besitzt, die Information invertiert werden. Die restlichen Glider von N können nicht mehr verwendet werden und werden von einem Eater verschluckt. Wie in der Abbildung 13 zu sehen ist wurde der Strom S zweimal kopiert und einmal invertiert.

3.2.4 Hilfsspeicher

Um Zwischenergebnisse ablegen zu können benötigt der Life-Computer noch einen Hilfsspeicher, in dem beliebig große Zahlen aufbewahrt werden können. Der Speicher besteht aus Registern, die jeweils ein statischen Block beinhalten. Eine Zahl wird anhand der Entfernung eines Blocks vom Computer dargestellt. Um den Speicher hochzuzählen wird eine geeigne-

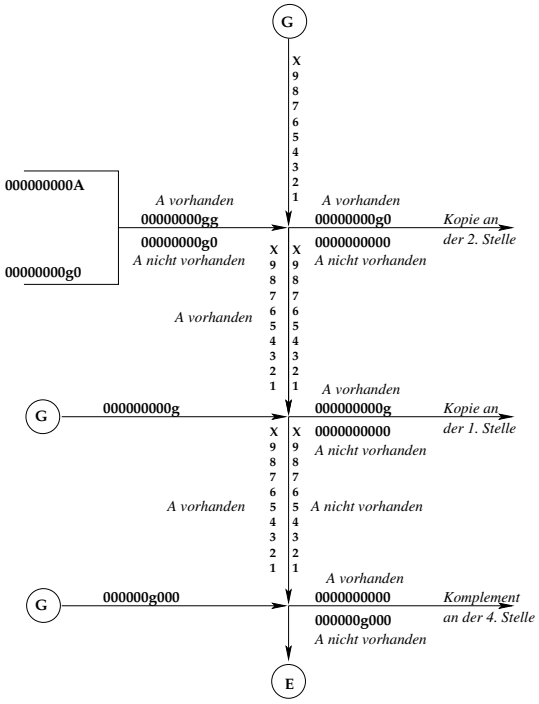

Abbildung 13: Kopieren eines Gliderstroms

te Flotte, bestehend aus Glidern, erzeugt. Diese besteht aus zwei Wellen. Die erste Welle zerstört den Block und erzeugt eine Honey-Farm, die zweite Welle zerstört die Honey-Farm und erzeugt genau ein Feld vom ursprünglichen Block entfernt einen neuen Block.

3.2.5 Side Tracking

Aufgrund der vielen Bauteile, die ein Computer besitzt, ist es manchmal nötig, dass Glider parallel laufen müssen. Da der Abstand aber durch die Größe einer Glider-Gun beschränkt ist, wird das *Side Tracking* verwendet. Beim *Side-Tracking* werden drei Glider-Guns verwendet. Diese Glider-Guns werden vom Computer kontrolliert und sind so programmiert, dass sie Glider nur dann durchlassen, wenn das Programm diese benötigt. Die beiden computergesteuerten Ströme der Glider-Guns $G2$ und $G3$ laufen parallel, senkrecht dazu befindet sich eine weitere Glider-Gun $G1$. Der Ablauf des *Side-Tracking* besteht aus drei Schritten. Im ersten Schritt erzeugt $G1$ den Glider s, der durch $G3$ geht. Dieser Glider s wird nun von $G2$ bis zum richtigen Zeitpunkt wiederzurückgeschickt ansonsten wird er durchgelassen. Im dritten Schritt schickt $G3$ s wieder zu $G2$.

3.2.6 Lösung gefunden

Das *Side Tracking* kann nun auch ausgenutzt werden um den Computer zu zerstören, wenn ein Ergebnis gefunden wurde. Dafür kann ein Gliderstrom sich innerhalb der zwei Ströme aufhalten und wenn ein Ergebnis gefunden wird kann er freigelassen werden und den Computer zerstören.

Hiermit sind alle nötigen Bauteile vorhanden um einen Life Computer zu bauen. Dieser

würde sich zerstören, wenn er eine Lösung findet ansonsten immer weiterarbeiten. Mit Hilfe dieser Bauteile lassen sich nun Computer bauen, die beliebige wohl definierte mathematische Probleme lösen können, daraus folgt, dass Life universal ist. Damit ist auch die Frage der Entscheidbarkeit beantwortet, da die Entscheidung, ob eine Figur verschwindet, sich auf das Halteproblem reduzieren lässt.

4 Zusammenfassung

Das Game of Life ist kein gewöhnliches Spiel, es gibt keine Spieler und nur drei einfache Regeln. Trotzdem können in Life komplexe Strukturen und Mechanismen entstehen. Strukturen in Life können entweder verschwinden, oszillieren, sich fortbewegen oder als statische Struktur vorhanden bleiben. Es ist sogar ein unbegrenztes Wachstum möglich. Doch gibt es keinen Algorithmus, der als Eingabe zwei beliebige Life Konfigurationen erhält und entscheiden kann, ob die eine aus der anderen entstehen kann sowie keinen Algorithmus der entscheiden kann, ob eine Life Konfiguration statisch wird, ewig wächst oder verschwindet, daher ist Life unentscheidbar. Es ist aber möglich logische Strukturen zu erstellen, die beispielsweise UND bzw. ODER Funktionen nachbilden können. Mit Hilfe dieser Strukturen können Life Computer erstellt werden, die beliebige berechenbare Probleme bearbeiten können. Dieser Life Computer kann sich sogar duplizieren. Damit läßt sich das Entscheidungsproblem des *Game of Life* auf das Halteproblem reduzieren.

Literatur

[1] Berlekam Erwin R., Conway John H., Guy Richard K: Winning ways for your mathematical plays(Volume 4), A.K Peters Ltd, Wellesly, 2004