Rheinische Friedrich-Wilhelms-Universität Bonn
Institut für Informatik I

Florian Kunze

Lemmings ist NP-vollständig

19. November 2007

Seminararbeit im WS 2007/2008

Zusammenfassung

Diese Ausarbeitung beschäftigt sich mit dem bekannten Computerspiel Lemmings, welches erstmals 1990 von Psygnosis veröffentlicht wurde. Das Spiel wird zunächst formalisiert, um es daraufhin auf dessen Komplexität zu untersuchen. Dabei hat sich herausgestellt, dass das Lösen der Lemmings-Level NP-vollständig ist, selbst dann, wenn es nur einen einzelnen Lemming zu retten gilt. Des Weiteren wird jedoch auch gezeigt, dass es unter gewissen Restriktionen an das Spiel möglich ist, jedes Level in polynomieller Zeit zu lösen.

Inhaltsverzeichnis

1 Einführung

Lemmings hat sich seit seiner Veröffentlichung zu einem der populärsten Computerspiele entwickelt. Es wurde oft kopiert und es existieren inzwischen viele verschiedene Versionen und Klone des Spiels. Der Kern des Spiels ist jedoch immer der selbe: Ziel ist es in jedem Level eine gewisse Anzahl von Lemmingen zu einem Ausgang zu führen. Die Lemminge werden durch eine oder mehreren Falltüren an bestimmten Stellen ins Level gelassen. Ihnen kann der Spieler verschiedene Fähigkeiten geben und ihnen so helfen, den Ausgang zu finden, um dadurch ein Level zu bestehen. In dieser Ausarbeitung wird gezeigt, dass das Lösen der Level NP-Vollständig ist. Des Weiteren wird gezeigt, dass alle Level, die gewissen Einschränkungen bezüglich der Fähigkeiten der Lemminge unterliegen, in polynomieller Zeit lösbar sind.

1.1 Das Spielprinzip

Zu Beginn bedarf es erst einmal einer genaueren Erläuterung des Spiels und dessen Ablaufs. Lemminge sind kleine, in ihrer Auffassungsgabe etwas eingeschränkte Kreaturen, die durch den Spieler heil durch die verschiedenen Level geführt werden müssen. Ihre Welt ähnelt der des Menschen. So herrschen dort auch Anziehungskraft und es gibt verschiedene Elemente wie Stein und Wasser. Jedoch ist ihre Welt nur zwei-dimensional. Lemminge beginnen ihr Leben am Beginn jeden Levels mit dem Fall aus einer Falltür. Sobald Sie auf dem Boden gelandet sind, fangen sie an in eine Richtung los zu laufen. Bei Lemmingen handelt es sich um relativ einfache Kreaturen, ihre Umwelt ist ihnen gänzlich egal. Laufen sie gegen eine Mauer oder eine zu steile Anhöhe, drehen sie sich einfach um und laufen in die entgegen gesetzte Richtung weiter. Liegt ein Graben oder ein Loch im Weg, so fallen sie hinein. Ist der Fall noch niedrig genug, laufen sie dort unten einfach weiter, ab einer gewissen Höhe jedoch kommt der Lemming ums Leben. In den verschiedenen Leveln gibt es eine Reihe von Möglichkeiten des Ablebens der Lemminge, darunter ertrinken, in Lava verglühen oder von einer Falle zerdrückt werden. Für die spätere Komplexitätsanalyse wird sich jedoch auf den Tod durch Herunterfallen beschränkt. Wie bereits erwähnt gibt es in ihrer Welt auch verschiedenste Materialien. Diese lassen sich in zwei Klassen unterteilen, solides und durchdringliches Material. Bei durchdringlichen Materialien kann sich ein Lemmming durch Buddeln einen Weg hindurch verschaffen, bei soliden ist dies nicht möglich. Die Zeit in der Lemming-Welt ist in Einheiten diskretisiert. Jedes Ereignis braucht eine Zeiteinheit. Der Raum der Level ist ebenso in feste Abstände diskretisiert. Um nun als Spieler einen positiven Einfluss auf das Leben der Lemminge und somit auf den Verlauf des Spieles zu haben, kann man den Lemmingen verschiedene Fähigkeiten geben. Diese werden im folgenden Kapitel genauer erläutert. Jedes Level besteht aus einer Zusammenstellung von Elementen, die zusam-

men eine zwei-dimensionale Welt bilden, welche durch ein endliches Rechteck begrenzt ist. Ein Level hat mindestens eine Falltür als Eingang, und mindestens einen Ausgang zur Rettung der Lemminge. Des Weiteren besteht jede Zelle des Levels aus solidem sowie durchdringlichen Material, Luft, Wasser oder Lava. Schafft es ein Lemming zum Ausgang, so gilt er als gerettet und wird aus dem Spiel entfernt. Jedes Level hat zusätzlich folgende Parameter:

- Die Anzahl der Lemminge, die am Beginn des Levels aus den Falltüren in das Level gelassen werden, sowie der Zeitabstand zwischen den Lemmingen
- Die Anzahl der Lemminge, die mindestens zum Ausgang geführt werden müssen um das Level erfolgreich zu beenden.
- Ein Zeitlimit, in der die benötigte Anzahl Lemminge befreit werden muss.
- Für jede Fähigkeit die jeweilige Anzahl, die dem Spieler zur Verfügung steht

1.2 Die Fähigkeiten der Lemminge

Die Fähigkeiten, die man den Lemmingen geben kann, lassen sich in drei Klassen aufteilen: permanente, semi-permanente und temporäre Fähigkeiten. Permanente Fähigkeiten legt ein Lemming, der diese bekommen hat, über den gesamten Zeitraum des Levels nicht wieder ab:

Der Kletter-Lemming kann an allen vertikalen Objekten hochklettern.

Der Fallschirmspringer zieht im Falle eines zu tiefen Absturzes die Reißleine seines Fallschirms und kommt so aus jeder Höhe heil herunter.

An semi-permanenten Fähigkeiten gibt es folgende:

Der Bomber-Lemming explodiert nach einem kurzen Countdown und fügt durchdringlichem Material dabei Schaden zu. Der Lemming ist danach Tod und wird aus dem Spiel entfernt

Der Blocker-Lemming bleibt stehen und fungiert so als Hindernis für andere Lemminge, die umkehren sobald sie gegen ihn laufen. Diese Fähigkeit kann er nur wieder verlieren, in dem ein anderer Lemming unter ihm her gräbt. Der Lemming fällt so ein kleines Stück herunter. Die Blocker-Fähigkeit verfällt, und der Lemming läuft normal weiter.

Temporäre Fähigkeiten hat ein Lemming nur über einen begrenzten Zeitraum. Permanente Fähigkeiten bleiben dem Lemming auch erhalten nachdem eine temporäre verfällt. An temporären Fähigkeiten gibt es folgende:

Der Konstrukteur fängt sofort nach dem Erhalt dieser Fähigkeit an eine Brücke zu bauen. Diese besteht aus zwölf Stufen und geht immer im selben Winkel nach oben. Sie wird in die Richtung gebaut in die der Lemming gerade läuft, und besteht aus durchdringlichem Material.

Der Abbruch-Lemming versucht sich waagerecht durch Hindernisse zu graben. Ist das Hindernis durchdringlich, so gräbt der Lemming so lange weiter, bis er entweder auf Luft oder auf solides Material stößt. Ist das Hindernis solide oder es befindet sich gar kein Hindernis vor ihm, so verfällt die Fähigkeit sofort wieder.

Der Buddel-Lemming buddelt vertikal nach unten. Er hört auf sobald er auf Luft oder auf solides Material stößt.

Der Minen-Lemming arbeitet sich auch durch durchdringliches Material, er allerdings diagonal nach unten. Auch er hört auf sobald er auf Luft oder auf solides Material stößt.

2 Formalisierung

Für eine genauere Analyse des Spiels werden wir es zunächst formalisieren. Dafür beschreiben wir ein Level als folgendes Tupel:

$$L = (limit, save, lems, start, width, height, grid, exit, skills)$$

- $limit$, die Zeitbeschränkung in diskreten Einheiten. Wir verlangen hier, dass diese Zeitbeschränkung polynomiell abhängig von der Größe des Levels ist. Diese Bedingung muss aus beweistechnischen Gründen vorhanden sein, auch wenn sie wahrscheinlich nicht nötig ist. Denn es ist anzunehmen, dass ein Level, welches in polynomiell abhängiger Zeit zu seiner Größe nicht lösbar ist, überhaupt nicht lösbar ist.
- $save$, die Anzahl der Lemminge, die gerettet werden müssen.
- $lems$, die Anzahl der Lemminge, die in das Level gelassen werden.
- $start$, ein Array der Länge $lems$. Jedes Element ist ein Tupel $start[i] = (pos, t)$, welches definiert, dass der $i$te Lemming zur Zeit t an der Position pos das Level betreten wird.
- $width$ und $height$ beschreiben die Breite und Höhe Levels.
- $grid$ ist ein zwei-dimensionales Array mit den Dimensionen $width$ und $height$. In $grid$ ist die Startkonfiguration gespeichert. Für jede Position ist gespeichert ob sich dort Luft, durchdringliches oder solides Material befindet.

- *exit*, die Position des Ausgangs.
- *skills* ist ein Array der Länge 8, der für jede der in Abschnitt 1.2 beschriebenen Fähigkeiten die verfügbare Anzahl speichert.

Ein Spieler kann nun den Ablauf des Levels beeinflussen, indem er den Lemmingen zu frei wählbaren Zeiten Fähigkeiten gibt. Dies werden wir als Spielzug m (vom englischen „move“) beschreiben.

$$m = (t, x, y, l, s)$$

- t ist der Zeitpunkt an dem der Spielzug gemacht wurde.
- x, y geben die Position des Lemmings an, dem die Fähigkeit gegeben werden soll.
- l gibt den Bezeichner des Lemmings an, dem die Fähigkeiten gegeben werden soll ($1 \leq l \leq lems$).
- s gibt den Bezeichner der Fähigkeit (1-8) an.

Wir definieren als *Strategie* S für ein *Level* L eine zeitlich geordnete Folge von *Spielzügen* m. Jedem Lemming kann pro Zeiteinheit nur eine Fähigkeit gegeben werden. Ein *gültiger Spielzug* ist ein Spielzug, in dem der Lemming, der die Fähigkeit erhalten soll, zu dem entsprechenden Zeitpunkt sich an der angegebenen Position befindet, die Fähigkeit noch verfügbar ist, und der Lemming diese Fähigkeit auch erhalten kann. Eine *gültige Strategie* ist eine Strategie, die nur aus gültigen Spielzügen besteht. Eine *siegreiche Strategie* ist eine gültige Strategie, bei dessen Anwendung auf das Level die vorgeschriebene Menge an geretteten Lemmings mindestens erreicht wird. Zu einem gegebenen Level L haben wir nun das Entscheidungsproblem, ob eine siegreiche Strategie S für dieses Level existiert. Wir definieren als LEMMINGS die Menge aller Level L, zu der die Antwort auf dieses Entscheidungsproblem „ja“ist. Nun definieren wir noch eine Untermenge 1-LEMMINGS, in die wir nur Level aufnehmen, die einen einzigen Start-Lemming ($lems = 1$) haben. Diese Menge werden wir in Abschnitt 4 gesondert untersuchen.

3 Lemmings ist NP-Vollständig

Wir wollen nun zeigen, dass LEMMINGS NP-Vollständig ist. Dafür zeigen wir zunächst, dass LEMMINGS in NP liegt und daraufhin, dass LEMMINGS NP-hart ist.

3.1 Lemmings $\in NP$

Hier reicht es zu zeigen, dass wir in polynomieller Zeit überprüfen können, ob eine Strategie S für ein Level L eine siegreiche Strategie ist.

Lemma 1 LEMMINGS $\in NP$

Beweis. Um heraus zu finden, ob es sich bei einer Strategie S für ein Level L um eine siegreiche Strategie handelt, wenden wir S schrittweise an. Wir überprüfen pro Zeiteinheit ob es sich um gültige Spielzüge handelt, und wenden diese Spielzüge auf L an. Nachdem die Zeitbeschränkung (*limit* von L) abgelaufen ist, vergleichen wir die Anzahl der geretteten Lemminge mit der Vorgabe (*save* von L). Dass dieses schrittweise Überprüfen und Anwenden in polynomieller Zeit ausgeführt werden kann, wird aus folgenden Erläuterungen ersichtlich.

- Alle Änderungen pro Zeiteinheit können in polynomieller Zeit auf L angewandt werden. Die Startzustände der Lemminge sind fest angegeben, und die Berechnung der neuen Positionen in Abhängigkeit zur alten Position, der Richtung und der Fähigkeiten ist in polynomieller Zeit berechenbar.

- Die Überprüfung der Spielzüge auf ihre Gültigkeit ist in polynomieller Zeit möglich, da uns durch die schrittweise Anwendung immer der aktuelle Levelzustand zur Verfügung steht.

- Die Anzahl der Zeiteinheiten, die simuliert werden müssen, sind polynomiell in ihrer Eingabegröße. Diese Anzahl ist genau unsere Zeitbeschränkung *limit* des Levels, und diese Zeitbeschränkung haben wir bei unser Formalisierung so definiert, dass sie polynomiell abhängig von der Levelgröße ist.

- Die Anzahl der Spielzüge ist polynomiell in ihrer Eingabegröße. Denn damit es sich um eine siegreiche Strategie handelt, muss jeder Spielzug gültig sein, das heißt jedem Lemming darf pro Zeiteinheit nur eine Fähigkeit zugeschrieben werden. Da die Anzahl der Lemminge, die in das Spiel gelassen werden, begrenzt ist (durch *lems* in L), und, wie oben beschrieben, die Anzahl der Zeiteinheiten polynomiell in ihrer Eingabegröße ist, ist auch die Anzahl der gültigen Spielzüge, und somit alle Spielzüge einer siegreichen Strategie, polynomiell in ihrer Eingabegröße.

□

3.2 Lemmings ist NP-hart

Um zu zeigen, dass LEMMINGS NP-hart ist, werden wir dieses Problem auf das bekannte 3SAT Problem reduzieren. Für jede Formel wird ein korrespondierendes Lemmings-Level generiert, welches genau dann lösbar ist, wenn auch die Formel erfüllbar ist.

3.2.1 Level-Konstruktion

Für die Konstruktion des Levels verwenden wir Level-„Bausteine“, die wir entsprechend der 3SAT-Formel deterministisch zusammenfügen werden.

Gegeben sei eine 3SAT-Formel mit n Variablen $v_1, v_2, ..., v_n$ sowie m Klauseln $c_1, c_2, ..., c_m$. Das entsprechende Level hat pro Variable und pro Klausel jeweils einen Start-Lemming, insgesamt also $(m + n)$ Lemminge, wobei zum erfolgreichen Beenden des Levels alle Lemminge den Ausgang erreichen müssen. Die Zeitbeschränkung für unser Level wählen wir groß genug, sodass genug Zeit für jeden Lemming bleibt, zum Ausgang laufen zu können. Die Größe des Levels ist polynomiell abhängig von der 3SAT-Formel. Für das Level legen wir folgende Fähigkeiten fest: n Abbruch-Lemminge, n Konstrukteure und m Buddel-Lemminge.

Abbildung 1: Elemente die wir für unser Level verwenden. (a) Ein Lemming, (b) der Eingang, (c) solides Material, (d) durchdringliches Material, (e) der Ausgang.

Folgend werden alle für das Level verwendeten Bausteine genauer erläutert. Im Anschluss wird dann erklärt wie diese Bausteine zusammenspielen werden.

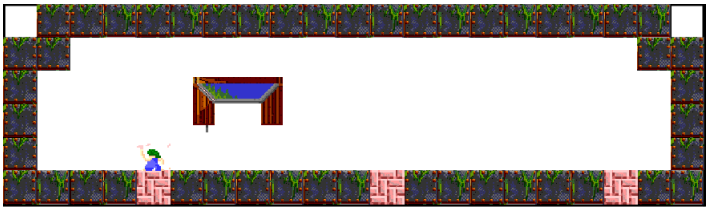
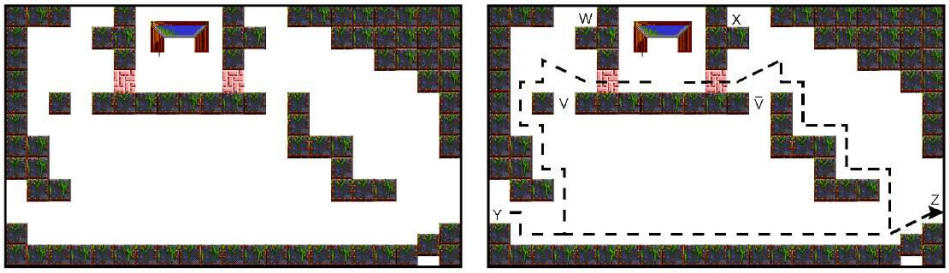


Abbildung 2: Klausel-Baustein.

Pro Klausel in der 3SAT-Formel wird ein *Klausel-Baustein* erzeugt. In

diesem Baustein startet ein Lemming, der ohne Anwendung eines Spielzugs dort gefangen ist. Die einzige Möglichkeit diesen Lemming zu befreien ist, ihn zu einem Buddel-Lemming zu verwandeln und ihn durch einen der drei durchdringlichen Blöcke durch buddeln zu lassen. Da nur m Buddel-Fähigkeiten zur Verfügung stehen und alle Lemminge gerettet werden sollen, müssen alle Buddel-Fähigkeiten in diesen Bausteinen aufgebraucht werden. Die Semantik dahinter ist die, dass durch das Aussuchen eines der drei möglichen Ausgänge ein Literal in der Klausel gewählt wird, welches die Klausel erfüllen soll.



(a) (b)

Abbildung 3: (a) Variabel-Baustein, (b) Mögliche Wege aus dem Baustein.

Pro Variable in der 3SAT-Formel wird ein *Variable-Baustein* erzeugt. Auch in diesem Baustein startet jeweils ein Lemming, der erst einmal gefangen ist. Zur Befreiung braucht dieser Lemming zunächst die Fähigkeit des Abbruch-Lemmings, um sich entweder nach links oder nach rechts durch zu graben. Da der Fall durch das Loch (links und auch rechts) zu tief ist um Überleben zu können, muss der Lemming hier mit der Konstrukteur-Fähigkeit eine Brücke über dieses Loch bauen. Egal für welche Seite man sich entschieden hat, der Lemming wird den Baustein nach rechts (und in die sichere Freiheit) verlassen. Da insgesamt nur n Abbruch-Lemminge und n Konstrukteure gegeben sind, können diese Fähigkeiten pro Baustein, also auch pro Lemming, nur jeweils einmal angewandt werden. Man muss sich also für eine Seite entscheiden. Durch die Brücke, die man entweder links oder rechts gebaut hat, hat man sich für eine Belegung entschieden. Links wird die Variabel auf „wahr“, rechts auf „falsch“gesetzt.

Nun wollen wir die Klausel-Bausteine mit den Variabel-Bausteinen derart verknüpfen, dass unser Klausel-Lemming, der sich zur Befreiung für eine Variable entschieden hat, bei genau dem dafür entsprechenden Variabel-Baustein herauskommt. Gegeben sei zum Beispiel eine 3SAT-Klausel der Form $(v_1 \vee \neg v_2 \vee v_3)$. Entscheidet sich der Klausel-Lemming für den mittleren Ausgang, so landet er in dem Variablen-Baustein von v_2, und zwar durch den

Eingang ‚x', wie in Abbildung 3 zu sehen. Nur wenn der Variabel-Lemming sich für dir Belegung $\neg v$, also „falsch", entschieden hat und dementsprechend nach rechts geflohen ist, landet der Klausel-Lemming sicher auf der Brücke und überlebt. Stimmt die Belegung nicht, ist die Klausel nicht erfüllt und der Klausel-Lemming fällt in den Tod. Würde sich der Klausel-Lemming für den rechten Ausgang entscheiden, würde er im Variablen-Baustein v_3 bei dem Eingang ‚w'herauskommen und nur überleben, falls dieser Variabel-Lemming sich für die Belegung „wahr"entschieden hätte.

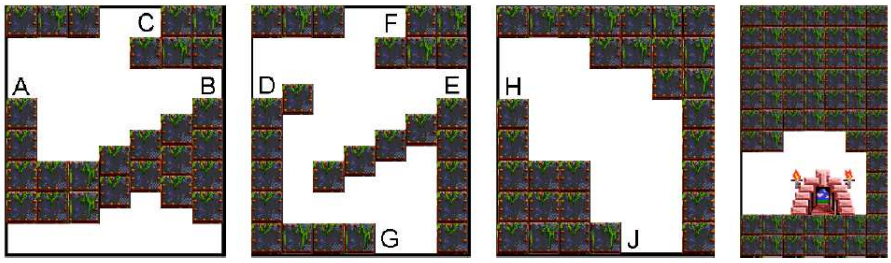


Abbildung 4: (a) Lenk-Baustein, (b) Verbindungs-Baustein, (c) Eck-Baustein, (d) Ausgangs-Baustein.

Um die Bausteine nun wie gerade beschrieben zu verbinden, bedarf es verschiedener Verknüpfungs-Bausteine (siehe Abbildung 4). Der dabei am meisten verwendete ist der *Verbindungs-Baustein*. Jeder Lemming der ihn von oben (F) betritt, verlässt ihn immer unten (G). Jeder Lemming der ihn von links (D) betritt, wird ihn hingegen rechts verlassen (E). Dieser Baustein bildet also eine Art Kreuzung. Würde ein Lemming den Baustein von rechts (E) betreten, würde er ihn unten verlassen. Dieser Fall ist allerdings nicht vorgesehen und wird im Normalfall auch nicht eintreten. Der *Lenk-Baustein* ist ähnlich konzipiert, jedoch wird hier der von oben kommende Lemming auch nach rechts geleitet. Das heißt also jeder Lemming, der von links (A) oder oben (C) kommt, wird den Baustein rechts (B) verlassen. Auch hier ist das Betreten von rechts (B) nicht vorgesehen. Der *Eck-Baustein* lenkt die Lemminge von der linken Seite (H) nach unten (J). Jeder Lemming der den *Ausgangs-Baustein* erreicht, wird erfolgreich das Level verlassen und die Freiheit erlangen.

Im folgenden verwenden wir als Einheiten die Breite (respektive Höhe) der Verbindungs-Bausteine (Abbildung 4). Der Aufbau des Levels erfolgt nun folgendermaßen. Alle Klausel-Bausteine werden nebeneinander platziert. Unter jedem möglichen Ausgang platzieren wir $2n$ Verbindungs-Bausteine. Die dadurch entstandenen $2n$ Reihen stellen dabei die Variablen dar, wobei pro Variable je eine Reihe für v und eine für $\neg v$ steht. Diese Reihen werden wir nun mit den Eck-Bausteinen in Spalten lenken, die wir dann mit unse-

ren Variabel-Bausteinen verbinden können. Um dies zu erreichen platzieren wir hinter der ersten Reihe einen Eck-Baustein, der damit die erste Spalte bildet. Hinter der zweiten Reihe platzieren wir erst einen Verbindungs- und dann den Eck-Baustein. Nun haben wir die Spalten für unseren ersten Variabel-Baustein geschaffen. Da die Variabel-Bausteine eine Länge von $3n$ haben, müssen wir für unsere nächste Reihe erst drei Verbindungs-Bausteine setzen, bevor wir wieder mit der Spaltenbildung beginnen können. Auf diese Weise arbeiten wir alle Reihen ab. Unter die auf der rechten Seite neu entstandenen Spalten platzieren wir nun nebeneinander unsere entsprechenden Variabel-Bausteine, und zum Schluss kommt rechts daneben der Ausgangs-Baustein.

Dies bildet nun das Gerüst für unsere 3SAT-Formel. Da wir allerdings bisher nur mit Verbindungs-Bausteinen gearbeitet haben, würde jeder Klausel-Lemming, egal für welchen Ausgang er sich entscheidet, ganz nach unten durchlaufen und dort nicht mehr weiter kommen. Um jeden Klausel-Lemming zum richtigen Eingang des jeweiligen Variable-Bausteins zu befördern, ersetzen wir an bestimmten Stellen die Verbindungs-Bausteine durch Lenk-Bausteine. Jeder möglich Ausgang für den Klausel-Lemming steht für ein Literal in der Klausel. In der jeweiligen Spalte darunter steht jede Höhe für eine Belegung $v_1, \neg v_1$ bis $v_n, \neg v_n$. Wir ersetzen nun auf der Höhe den Verbindungs-Baustein durch einen Lenk-Baustein, die für die Belegung steht, die in der 3SAT-Formel vorgegeben ist. In Abbildung 5 ist zur Veranschaulichung ein komplettes Beispiel dargestellt. Formal gesprochen muss ein Level wie folgt aufgebaut werden.

Definition 2 *Für jede Klausel der* 3SAT*-Formel wird nebeneinander ein Klausel-Baustein erzeugt. Unter diesen Bausteinen kommt ein* $[3m+3n, 2n]$ *großer Block. Dieser Block wird aufgefüllt mit Verbindungs-Bausteinen, außer einer der folgende Fälle tritt ein (*c_{ij} *bezeichnet das* j*-te Literal in der* i*-ten Klausel):*
Im Fall $c_{ij} = v_k$ *kommt an die Stelle* $[3i + j, 2k - 1]$ *ein Lenk-Baustein.*
Im Fall $c_{ij} = \neg v_k$ *kommt an die Stelle* $[3i + j, 2k]$ *ein Lenk-Baustein.*
Für $k = 1$ *bis* n*, kommt an Stelle* $[3m + 3k - 2, 2k - 1]$ *ein Eck-Baustein.*
Für $k = 1$ *bis* n*, kommt an Stelle* $[3m + 3k - 1, 2k]$ *ein Eck-Baustein.*
An Stelle $[3m + 1, 2n + 1]$ *kommen nebeneinander die Variabel-Bausteine* v_1 *bis* v_n*. An Stelle* $[3m + 3n + 1, 2n + 1]$ *kommt ein Ausgangs-Baustein.*

3.2.2 Beweis

Um nun zu zeigen, dass das Finden einer Lösungsstrategie NP-hart ist, müssen wir die Reduktion in beide Richtungen beweisen:

Lemma 3 *Für jede gültige Belegung der* 3SAT*-Formel gibt es eine siegreiche Strategie in* LEMMINGS*.*

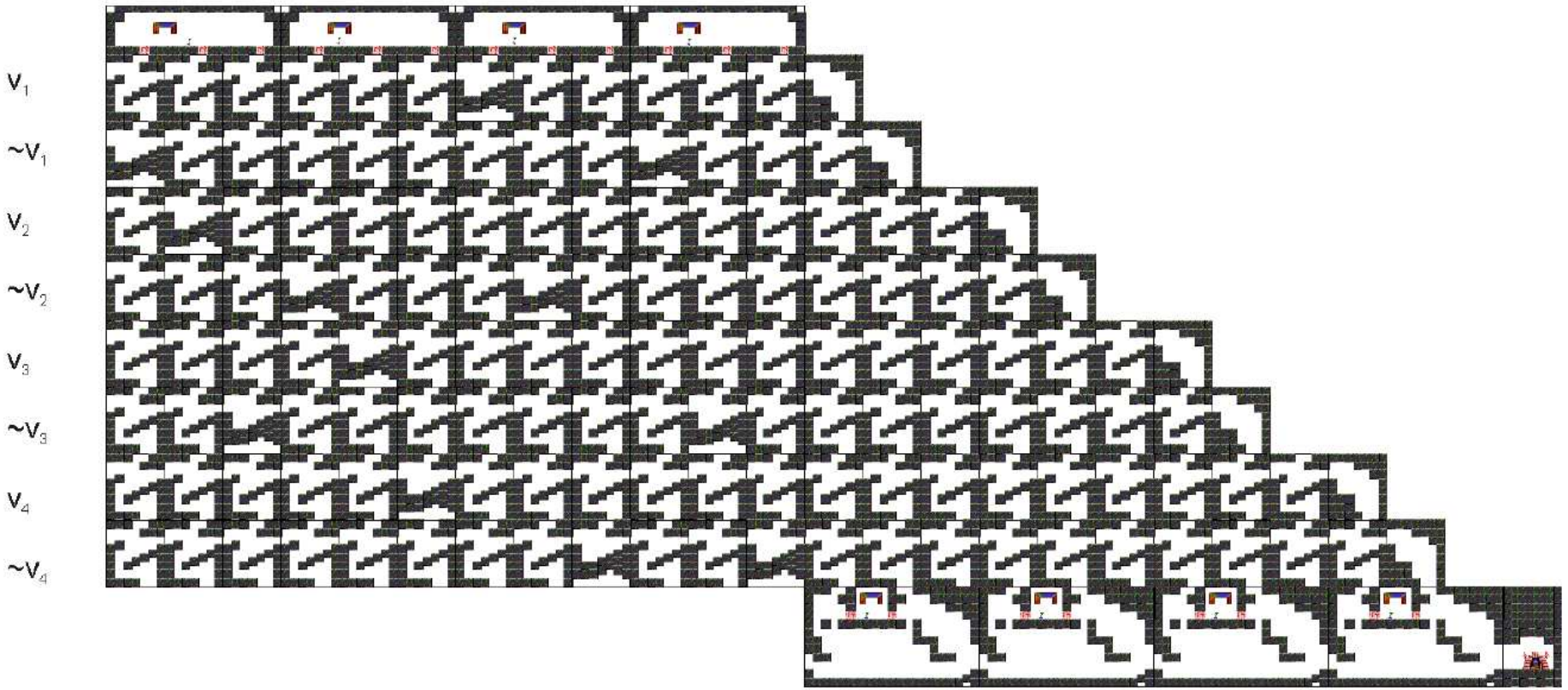


Abbildung 5: Ein LEMMINGS-Level generiert mit folgender Formel: $(\neg v_1 \vee v_2 \vee \neg v_3) \wedge (\neg v_2 \vee v_3 \vee v_4) \wedge (v_1 \vee \neg v_2 \vee \neg v_4) \wedge (\neg v_1 \vee \neg v_3 \vee \neg v_4)$

Beweis. Gegeben sei eine 3SAT-Formel mit einer gültigen Belegung und das korrespondierende LEMMINGS-Level. Jeder Variabel-Lemming bricht aus der Seite aus, die der gültigen Belegung dieser Variabel entspricht. Ist also in der gültigen 3SAT-Belegung v_1 wahr, bricht der entsprechende Variabel-Lemming auf der linken Seite mit Hilfe der Eigenschaft des Abbruch-Lemmings aus und baut mit der Konstrukteur-Fähigkeit eine Brücke über das linke Loch. Würde v_1 bei der Belegung auf falsch stehen, würde der Lemming links ausbrechen. Auf diese Weise sind alle Variabel-Lemmings befreit und auch gerettet. Da eine gültige Belegung für die 3SAT-Formel existiert, muss pro Klausel mindestens ein Literal diese Klausel erfüllen. Der Klausel-Lemming wählt entsprechend diesem Literals seinen Ausgang. Ist c_{i1} das erfüllende Literal, so buddelt der Klausel-Lemming sich durch den Ausgang auf der linken Seite. Ist es das Literal c_{i2}, wählt der Lemming den mittleren, und ist es das Literal c_{i3}, wählt er den rechten Ausgang um sich zu befreien. Die Klausel-Lemminge wandern nun durch das Level und kommen an ihren entsprechenden Variabel-Bausteinen heraus. Da sie sich jeweils für ein erfüllendes Literal entscheiden haben, werden sie an der Seite des Variabel-Bausteins herauskommen, an der der Variabel-Lemming eine Brücke über die Schlucht gebaut hat. Die Klausel-Lemminge werden also sicher auf der Brücke landen und das Level über den Ausgang verlassen. Alle Lemminge sind nun befreit. Dabei wurden alle Fähigkeiten aufgebraucht. □

Lemma 4 *Jede siegreiche Strategie in* LEMMINGS *entspricht einer gültigen Belegung in der* 3SAT*-Formel.*

Beweis. Gegeben sei ein LEMMINGS-Level mit einer siegreichen Strategie. Für ein erfolgreiches Beenden dieses Levels müssen alle Lemminge befreit werden. Da alle Lemminge zuerst in ihrem Baustein gefangen sind, müssen zur Befreiung aller Lemminge alle Fähigkeiten, die zur Verfügung stehen, in diesen Bausteinen aufgebraucht werden. Das heißt für jeden Variabel-Lemming, dass er nur auf einer Seite ausbrechen und eine Brücke bauen kann. Für die Klausel-Lemminge bedeutet dies, dass, sobald sie sich für einen Ausgang entschieden haben, sie in den Verbindungs-Bausteinen keine Fähigkeiten mehr zur Verfügung haben, um ihren Weg zu beeinflussen. Sie werden also zwangsweise bei der entsprechenden Seite des Variabel-Bausteins herauskommen. Jeder Variabel-Lemming hat sich vorher für eine Belegung seiner Variabel entscheiden müssen. Ein Klausel-Lemming wird also nur überleben, wenn er sich für ein erfüllendes Literal entschieden hat, und somit auf der Brücke des entsprechenden Variabel-Lemmings landet. Nur wenn alle Klausel-Lemminge überleben, kann das LEMMINGS-Level erfolgreich beendet werden. Nur dann wurde jede Klausel erfüllt, und somit eine gültige Belegung der 3SAT-Formel gefunden. □

4 1-Lemmings ist NP-vollständig

Der Beweis, dass LEMMINGS NP-vollständig ist, mag zwar allgemein gelten, jedoch kann man argumentieren, dass die gewählte Level-Konstruktion doch sehr untypisch für dieses Spiel ist. Dies liegt zum Beispiel daran, dass es sehr viele Falltüren gibt aus denen je nur ein Lemming startet, und somit viele Lemminge an verschiedenen Teilen des Levels starten. Deswegen betrachten wir nun unsere Untermenge 1-LEMMINGS, und zeigen (wieder durch eine Reduktion auf 3SAT), dass auch diese NP-vollständig ist. Die Besonderheit an 1-LEMMINGS-Leveln ist, dass nur ein einziger Lemming existiert, der gerettet werden muss. Es gilt also $lems = save = 1$. Das 1-LEMMINGS in NP liegt brauchen wir nicht mehr zu beweisen, da der Beweis für LEMMINGS auch hier anwendbar ist.

4.1 Level-Konstruktion

Der Aufbau des Levels ähnelt dem in Abschnitt 3.2.1 sehr. Die Zeitbeschränkung werden wir wieder so wählen, dass der Lemming genug Zeit hat, das Level zu passieren. Für dieses Level wird allerdings nur die Fähigkeit des Buddel-Lemmings zur Verfügung gestellt. Auch hier arbeiten wir mit Bausteinen, die wir für die Konstruktion einfach zusammen setzen. Zur späteren Veranschaulichung an einem Beispiel werden den Bausteinen Symbole zugeordnet, wobei ein Baustein mehrere Symbole haben kann. Dadurch kann man in dem Beispiel auf Abbildung 8 einen besseren Überblick behalten.

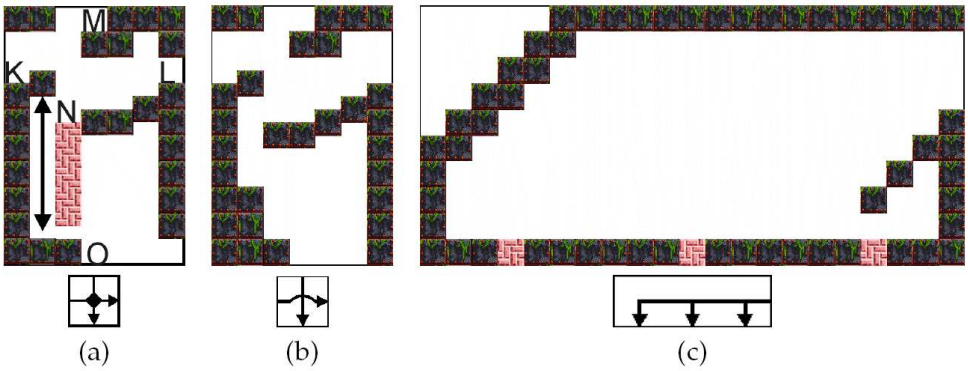


Abbildung 6: (a) Fallen-Baustein, (b) Verbindungs-Baustein, (c) 3-Split-Baustein.

Der wichtigste neue Baustein ist wohl der *Fallen-Baustein*. Betritt ein Lemming ihn von oben (M) oder von rechts (L), so ist seine einzige Chance zu überleben die, dass er sich durch das durchdringliche Material buddelt und ihn unten wieder verlässt (O). Ist dies vorher schon einmal geschehen, so ist der Lemming, sobald er diesen Baustein erneut betritt, dem Tode geweiht,

da ein Sturz dieser Höhe (dargestellt durch den Pfeil) für einen Lemming nicht mehr verkraftbar ist. Besteht das durchdringliche Material allerdings noch, so kann ein Lemming, der den Baustein von links (K) betritt, ihn unversehrt auf der rechten Seite (L) wieder verlassen.
Der *Verbindungs-Baustein* erfüllt den selben Zweck wie bei der vorherigen Konstruktion, hier wurde nur die Höhe angepasst. Ein Lemming, der diesen Baustein von oben betritt, kann ihn nur nach unten wieder verlassen. Betritt er ihn von links, verlässt er ihn wieder auf der rechten Seite. Theoretisch kann ein Lemming diesen Baustein auch von rechts betreten und dann nach unten hin verlassen, dieses Betreten wird aber durch den Level-Aufbau nicht möglich sein und bringt ihn auch nicht zum Ziel. Der *3-Split-Baustein* ähnelt sehr dem Verbindungs-Baustein aus Abschnitt 3.2.1. Auch hier besteht der Zweck darin, dass sich der Lemming für einen der drei Ausgänge entscheiden muss.

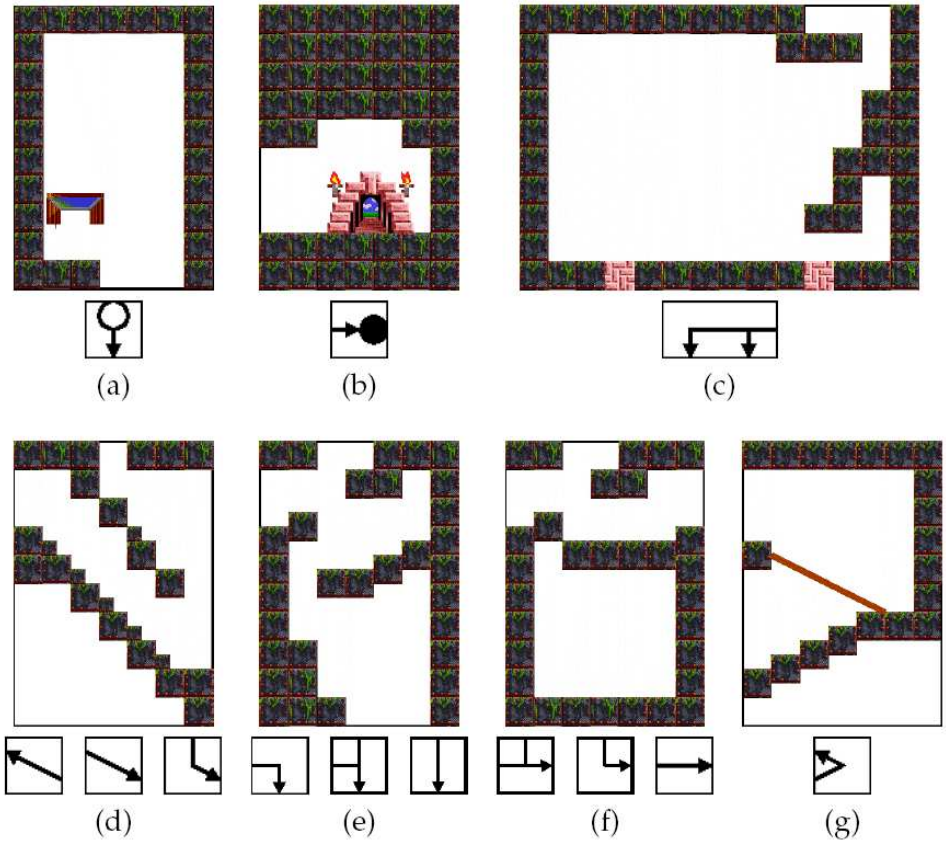


Abbildung 7: (a) Eingang, (b) Ausgang, (c) 2-Split-Baustein, (d) Treppen-Baustein, (e) Nach-Unten-Baustein, (f) Nach-Rechts-Baustein, (g) Eck-Treppen-Baustein.

Für *Eingang* und *Ausgang* bedarf es wohl keiner weiteren Erläuterung. Der *2-Split-Baustein* hat den Zweck, dass der Lemming sich zwischen zwei Ausgängen entscheiden kann, der Eingang befindet sich hier, anders als beim 3-Split-Baustein, rechts oben. Mit dem *Treppen-Baustein* landet der Lemming wenn er von oben oder auch von links kommt, unten rechts. Betritt er ihn von unten rechts, verlässt er ihn auf der linken oberen Seite. Der *Nach-Unten-Baustein* veranlasst den Lemming, egal ob er den Baustein von oben oder von links betritt, ihn nach unten wieder zu verlassen. Ähnliche Aufgabe hat der *Nach-Rechts-Baustein*, der dem Lemming, ob von oben oder links eingetreten, nach rechts den Ausgang weist. Eine besondere Stellung hat der *Eck-Treppen-Baustein*. Hier ist eine Brücke, wie sie eigentlich von der Eigenschaft eines Konstrukteurs gebaut würde, bereits vorhanden. Dies hat den Grund, das Brücken eine besondere Eigenschaft haben: Ein Lemming, der diesen Baustein von unten links betritt, würde erst vorbei an der Brücke gegen die rechte Wand laufen, auf dem Rückweg die Brücke dann hoch steigen, und so den Baustein oben links verlassen. Man könnte dem Lemming auch gerade genug Konstrukteur-Fähigkeiten geben, dass er sich diese Brücke selber bauen kann, dies würde aber dazu führen, dass wir in unserem späteren Beweis zu viele Fälle, die entstehen können, berücksichtigen müssten. Deswegen wird hier davon ausgegangen, dass diese Brücke bereits vorhanden ist.
Alle vorgestellten Bausteine können auch in spiegelverkehrter Weise vorkommen, um den Lemming in die entsprechend andere Richtung lotsen zu können.

Gegeben sei nun wieder eine 3SAT-Formel mit n Variablen und m Klauseln. Für die n Variablen werden $2n$ Spalten nebeneinander aus Verbindungs-Bausteinen erzeugt, als Einheit nehmen wir also wieder Breite und Höhe der Verbindungs-Bausteine. Die $(2i-1)$-ten Spalten stehen hier für die v_i's, die $(2i)$-ten Spalten jeweils für $\neg v_i$. Diese Spalten haben die Höhe $4m$. Über diesen Spalten, also in der ersten Reihe, platzieren wir n 2-Split-Bausteine. Den Eingang setzen wir über den ersten 2-Split-Baustein von links. Kommt der Lemming nun in das Level, muss er sich zunächst für v_1 oder $\neg v_1$ entscheiden, diese Spalte wandert er bis nach unten durch. Unten werden pro Variable beide Spalten wieder zusammengeführt. Über ein Treppensystem, wie man es in dem Beispiel in Abbildung 8 erkennen kann, wird der Lemming nun zu der nächsten Variable geführt, wo er sich wieder für v_2 oder $\neg v_2$ entscheiden muss. So führen wir den Lemming durch alle Variable-Spalten durch. Nach der letzten Variablen führen wir ihn noch einmal nach oben, rechts neben die zweite Reihe. Diese wandert er einmal quer durch. Links neben der zweiten Reihe platzieren wir einen 3-Split-Baustein, indem der Lemming nun landet. Jeder der drei Ausgänge entspricht einer Reihe der Klausel, sprich einem Literal. Hat sich der Lemming für ein Literal entschie-

den, wandert er die entsprechende Reihe nach rechts durch. Von dort aus lotsen wir ihn nun wieder durch die erste Spalte der nächsten Klausel nach links, wo wieder ein 3-Split-Baustein auf seine Entscheidung wartet. So geht es durch alle m Klauseln, bis rechts hinter der letzten Klausel der Ausgang wartet.
Nach dieser Konstruktion würde der Lemming, egal für welche Variablen-Belegung und Literale er sich auch entscheidet, den Ausgang erreichen und das Level erfolgreich beenden. Um nun unsere 3SAT-Formel in dem Level abzubilden, ersetzen wir bestimmte Verbindungs-Bausteine durch Fallen-Bausteine. Sei unsere i-te 3SAT-Klausel $(\neg v_1 \vee v_2 \vee \neg v_4)$, dann setzen wir in der $3i+1$-ten Reihe, die unserem ersten Literal entspricht an der ersten Spalte, die der Variable v_1 entspricht, einen Fallen-Baustein. Dementsprechend setzen wir für das zweite Literal in Reihe $3i+2$, Spalte 4 (für $\neg v_2$) und für das dritte Literal in Reihe $3i+3$, Spalte 7 (für v_4) einen Fallen-Baustein. Allgemein setzen wir einen Fallen-Baustein an Stelle $(2i-1, 3i+j)$, falls $c_{ij} = \neg v_k$. Ist $c_{ij} = v_k$, setzen wir den Fallen-Baustein an Stelle $(2i, 3i+j)$. Sind wir so die gesamte 3SAT-Formel durchgegangen und haben pro Literal ein Fallen-Baustein gesetzt, kann der Lemming nur noch lebend das Ziel erreichen, wenn er sich für die richtige Belegung der Variablen und das jeweils richtige Literal pro Klausel entschieden hat.
Die Fallen-Bausteine, die dem Lemming auf dem Weg durch die Spalten entgegenkommen, zwingen ihn sich durch das durchdringliche Material zu buddeln, jetzt ist dieser Baustein nicht mehr passierbar, weder vertikal noch horizontal. Betrachten wir das Beispiel in Abbildung 8. Entscheidet sich der Lemming nach dem Start bei der Variable v_1 für die Belegung „wahr“, nimmt er den linken Ausgang. Dort begegnet er an der zweiten Stelle einem Fallen-Baustein, da dieses Literal in der 3SAT-Formel auf $\neg v_1$ steht. Entscheidet er sich später, wenn er bei dem ersten 3-Split-Baustein für die erste Klausel angekommen ist für v_1, kommt für diese Klausel ein „falsch“heraus. Er muss den Fallen-Baustein horizontal passieren, und da er dort schon gebuddelt hat fällt er in den Tod.

4.2 Beweis-Skizze

Theorem 5 *Das zu einer* 3SAT*-Formel erzeugte* 1-LEMMINGS*-Level kann nur erfolgreich beendet werden, wenn auch die* 3SAT*-Formel durch eine Belegung erfüllbar ist.*

Beweis-Skizze. Wir können annehmen, dass alle Klauseln, in dem die selbe Variabel einmal als v und einmal als $\neg v$ vorkommt, vor der Level-Erzeugung entfernt wurden, da diese trivialerweise immer wahr sind. Um nun das Theorem zu beweisen, gehen wir genau so vor wie in Abschnitt 3.2.2. Wir zeigen, dass es für eine gültige Belegung der 3SAT-Formel eine siegreiche Strategie in unserem Level existiert, und dass eine siegreiche Strategie einer gültigen

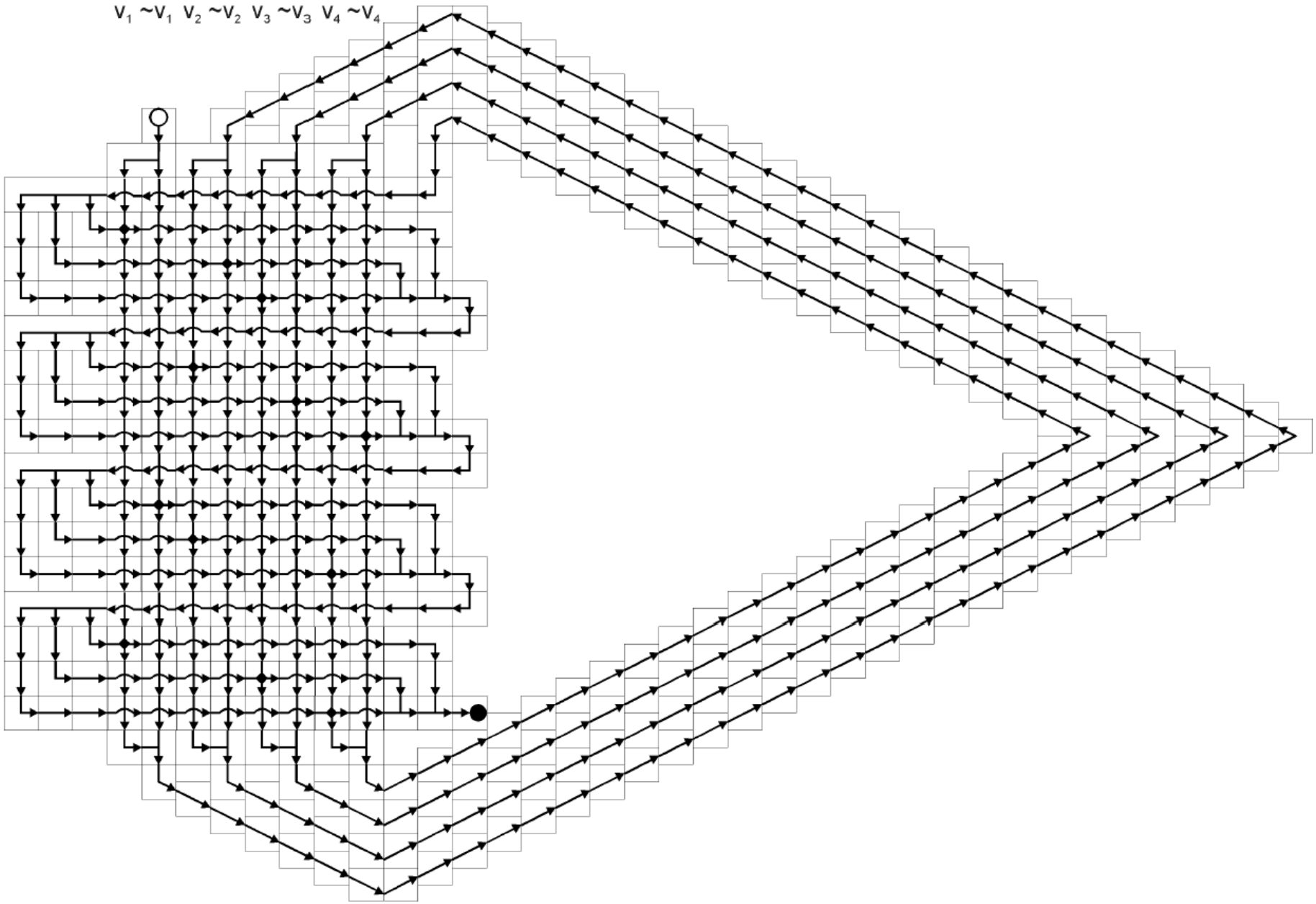


Abbildung 8: Ein 1-Lemmings-Level generiert mit folgender Formel: $(\neg v_1 \vee v_2 \vee \neg v_3) \wedge (\neg v_2 \vee v_3 \vee v_4) \wedge (v_1 \vee \neg v_2 \vee \neg v_4) \wedge (\neg v_1 \vee \neg v_3 \vee \neg v_4)$

Belegung in unserer 3SAT-Formel entspricht. Dies wird erreicht durch die Fallen-Bausteine. Entscheidet sich der Lemming für eine Belegung v_i, so sind in allen Klauseln jeweils die Reihen, in denen $\neg v_i$ vorkommt, nicht mehr passierbar. An diesen Stellen hatte der Lemming vorher die Fallen-Bausteine durch gebuddelt. Nachdem sich der Lemming für eine Belegung entschieden hat, sind die einzigen Reihen die noch passierbar sind Reihen die Literalen entsprechen, die jeweils ihre Klausel erfüllen. Diese Reihen sind der einzig mögliche Weg für den Lemming zu dem Ausgang zu gelangen. Da dem Lemming keine anderen Fähigkeiten außer dem Buddel-Lemming zur Verfügung stehen, hat dieser keine Möglichkeit einen anderen Weg durch dieses Level zu finden. Dies bedeutet, dass das zu einer 3SAT-Formel erzeugte 1-LEMMINGS-Level genau dann erfolgreich beendet werden kann, wenn auch die 3SAT-Formel durch eine Belegung erfüllbar ist. □

5 Eine Lemmings-Variante, die in P liegt

Da wir nun wissen, dass Lemmings allgemein NP-vollständig ist, stellt sich die Frage, ob es durch bestimmte Einschränkungen möglich ist, Lemmings praktisch lösbar zu machen. Bezogen auf die Fähigkeiten ist dies tatsächlich möglich.

Theorem 6 LEMMINGS *in dem nur permanente Fähigkeiten zur Verfügung stehen liegt in P.*

Beweis. Dadurch, dass nur noch permanente Fähigkeiten zur Verfügung stehen, können die Lemminge nicht mehr das Level, und somit sich auch nicht mehr gegenseitig beeinflussen. Dadurch können wir das Level mit einem gerichteten Graphen modellieren. Jeder Knoten ist ein Tupel der Form $(loc, dir, climb, float)$. loc bezeichnet die aktuelle Position des Lemmings, dir gibt dir Richtung an, in die der Lemming aktuell läuft (links oder rechts), $climb$ und $float$ sind jeweils boolesche Werte, die angeben, ob der Lemming zur Zeit die Fähigkeit des Kletter-Lemmings bzw des Fallschirmspringers besitzt. Die Kanten des Graphen entsprechen dem nächsten Schritt, den ein Lemming machen wird. Ein Lemming, der gerade ohne Hindernis vor sich nach rechts läuft, entspräche dem Knoten $(loc, rechts, climb, float)$ mit einer ausgehenden Kante zu dem nächsten Knoten $(loc', rechts, climb, float)$, wobei loc' eine Einheit weiter rechts wäre. Würde der Lemming direkt vor einem Hindernis stehen (und keine Fähigkeiten haben), wäre der nächste Knoten $(loc, links, climb, float)$, da er sich nun umdreht um zurück zu laufen. Hätte er allerdings die Fähigkeit des Kletter-Lemmings (also $climb = wahr$), wäre der nächste Knoten $(loc', rechts, wahr, float)$, wobei hier loc' nun die nächst-höhere Position wäre. Zusätzlich kommen ausgehende Kanten hinzu, in denen dem Lemming eine Fähigkeit hinzugefügt wurde, die er nicht vorher besessen hat. Ein Knoten hat also maximal drei ausgehende Kanten, eine

für den einfachen Positionswechsel, eine für Positionswechsel mit Fähigkeit Kletter-Lemming hinzugefügt, und eine Positionswechsel mit Fähigkeit Fallschirmspringer hinzugefügt. Für den Fall, dass ein Lemming ums Leben kommt, führen wir einen weiteren Knoten ein, der einen toten Lemming repräsentiert und keine ausgehenden Kanten mehr hat. Ein Lemming, der eine Fähigkeit erhalten hat, kann diese nicht mehr verlieren. Dementsprechend unterteilt sich der Graph in vier Teile. Diese Teile sind alle Wahr/Falsch-Kombinationen des $(climb, float)$ Paares. Zischen den Teilen gibt es Knoten zwischen $(falsch, falsch)$ und $(falsch, wahr)$, zwischen $(falsch, falsch)$ und $(wahr, falsch)$, und schließlich zwischen $(falsch, wahr)$ und $(wahr, wahr)$ bzw $(wahr, falsch)$ und $(wahr, wahr)$. Innerhalb der vier Teile gehen alle Lemminge immer den selben Weg, pro Position können sie entweder nach links oder nach rechts gehen. Das heißt pro Position im Level kann der Graph maximal acht Knoten haben, damit ist er linear zu der Größe des Levels.
Um nun herauszufinden, ob ein Level lösbar ist oder nicht, betrachten wir die möglichen Wege der Lemminge durch das Level. Dabei können wir alle Lemminge mit selber Startposition zusammenfassen, da alle durch die selbe Strategie gerettet werden können. Zunächst betrachten wir den Teil des Graphen, in dem *climb* und $float$ auf „falsch“gesetzt sind und schauen, wieviele Lemmminge es bis zum Ausgang schaffen. Dies geht sehr schnell, da es immer nur einen Ausgangsknoten zu betrachten gilt. Können wir ohne weitere Fähigkeiten schon so viele Lemminge retten, wie in *save* gefordert, sind wir fertig. Ansonsten betrachten wir nun die beiden Teilgraphen in denen einmal *climb*, und einmal $float$ auf „wahr“ist, in denen also ein Lemming genau eine der beiden Fähigkeiten besitzt. Hier müssen wir beachten, dass es einen Unterschied macht, zu welcher Zeit einem Lemming die Fähigkeiten geben wurden. Die Berücksichtigung aller Wege im Graphen ist dennoch polynomiell zur Größe des Graphen, da ein Lemming wie oben erwähnt, eine einmal erhaltene Fähigkeit nicht wieder verlieren kann. Mit den übrig gebliebenen Lemmingen berechnen wir nun von der Startposition aus, welche Lemminge zum Ausgang finden. Wir bezeichnen als d die Anzahl Lemminge die in beiden Teilgraphen gerettet werden, die also nur durch Geben der Fähigkeit des Fallschirmspringers oder nur durch Geben der Fähigkeit des Kletter-Lemmings den Ausgang gefunden haben. Alle Lemminge, die nur mit der Fähigkeit des Fallschirmspringers gerettet werden konnten, bezeichnen wir mit f, und alle, die als Kletter-Lemming befreit wurden mit c. Des Weiteren bezeichnen wir mit F die Anzahl an Fallschirmspringer-Fähigkeiten, und mit C die Anzahl der Fähigkeit als Kletter-Lemming. Aus der Menge der immer noch nicht geretteten Lemminge berechnen wir nun die, die unter Zuschreibung beider Fähigkeiten den Ausgang finden können, und bezeichnen diese mit e. Zum Schluss definieren wir noch folgende Hilfsvariablen.

Variable	Wert	Bedeutung
C'	$C - \min(c, C)$	Übrig gebliebene Kletter-Lemming Fähigkeiten nachdem c Kletter-Lemminge gerettet wurden
F'	$F - \min(f, F)$	Übrig gebliebene Fallschirmspringer Fähigkeiten nachdem f Fallschirmspringer gerettet wurden
d'	$\max(d - \lvert C' - F' \rvert, 0)$	Anzahl übrig gebliebener Kletter-Lemming und Fallschirmspringer Fähigkeiten nach Ausbalancierung
D'	$\min(C', F') - \lceil \frac{d'}{2} \rceil$	Anzahl übrig gebliebener Kletter-Lemming und Fallschirmspringer Fähigkeiten nachdem d Lemminge mit einer der beiden Fähigkeiten gerettet wurden.

Wir berechnen D' um zu erfahren, wie vielen Lemmingen zum Schluss noch beide Fähigkeiten zu Verfügung stehen, nach dem wir alle von unseren d Lemmingen gerettet haben. Zunächst brauchen wir die Fähigkeiten auf, von der wir im Vergleich zu der anderen einen Überschuss haben (dies entspricht der Berechnung von d'). Um nun möglichst viele Lemminge aus e retten zu können, teilen wir für die restlichen d Lemminge die Fähigkeiten gleich auf. Nun bleibt die Anzahl an Fähigkeiten über, mit der wir die restlichen Lemminge retten können. Nach diesen Berechnungen ist ein Level nun lösbar falls folgende Ungleichung erfüllt ist:

$$x + \min(c, C) + \min(f, F) + \min(d, C' + F') + \min(e, D') \geq save$$

All diese Berechnungen können in einer Zeit polynomiell zur Eingabegröße berechnet werden. Damit liegt diese eingeschränkte LEMMINGS-Version in P. □

6 Quellenangabe

Alle hier vorgestellten Erkenntnisse sowie alle Abbildungen stammen aus folgendem Paper:

G. Cormode. The hardness of the lemmings game, or Oh no, more NP-completeness proofs. In Proceedings of Third International Conference on Fun with Algorithms, pages 65-76, 2004.