

**Thema:** Mathematische Darstellung der Strategien zum  
Verlassen eines Irrgartens

**Verfasser:** Alexandra Thelen

**Kurs:** GK Mathematik

**Schule:** Gustav-Heinemann-Gesamtschule Alsdorf

**Abgabe:** 28 .02. 2007

## **0. Inhaltsverzeichnis**

### **1. Einleitung**

### **2. Strategien zum Verlassen eines Irrgartens**

#### 2.1 Zum Irrgarten

##### 2.2.1 Zufallsmethode

##### 2.2.2 Anwendung der Zufallsmethode

##### 2.3.1 Methode von Trémaux

##### 2.3.2 Anwendung der Tiefensuche

##### 2.4.1 Linke/Rechte Hand Regel

##### 2.5.1 Pledge-Algorithmus

### **3.1 Zusammenfassung der Ergebnisse und persönliche Stellungnahme**

## 1. Einleitung

Diese Facharbeit befasst sich mit der mathematischen Darstellung der Strategien zum Verlassen eines Irrgartens. Dabei lege ich mein Hauptaugenmerk auf eine praktikable und verständliche Umsetzung mithilfe qualitativer Erläuterungen.

Hier soll zuerst die Herkunft des Begriffs Irrgarten erläutert werden. Dabei versteht man unter einem gemeinen Irrgarten eine Gartenanlage aus einem Wegesystem, das mit Hecken, Mauern, Bretterwänden oder ähnlichem abgegrenzt wird. Diese Gartenanlagen haben ihren historischen Weg von der späten Renaissance über den Barock hin zur Moderne gefunden. Fußend auf der Faszination dieser Anlagen ist es Gegenstand dieser Arbeit, Irrgärten im verallgemeinerten Sinne zu betrachten. Dabei ist ein Irrgarten ein mathematisches Gebilde, welches hauptsächlich in graphentheoretischen Diskussionen zu finden ist. Hier sei zu erwähnen, dass ein Labyrinth ein Spezialfall eines Irrgartens ist, in dem nur ein Weg ohne Verzweigungen vom Eingang bis zum Ziel führt. Wie aber oben schon erwähnt, betrachte ich in dieser Arbeit die Probleme nicht aus streng mathematischer Sicht. Dabei werde ich in dieser Arbeit die Lösungen zum Entkommen aus einem Irrgarten diskutieren.

### 1. Strategien zum Verlassen eines Irrgartens

#### 1.1 Zum Irrgarten

Die hier vorgestellten Methoden bzw. Systematiken zum Entkommen aus einem Irrgarten fundieren auf der Annahme, dass der vorliegende Irrgarten zeitunabhängig ist. Was heißt, dass sich zu keinem Zeitpunkt das „Aussehen“ des Irrgartens ändert. Aus mathematisch präziser Sicht ist ein Irrgarten eine Datenstruktur vom Typ Graph. *Ein Graph  $G$  ist ein Paar von endlichen Mengen  $E$  und  $K$ .  $E$  ist nicht die leere Menge. Jedem Element von  $K$  ist eine zweielementige oder einelementige Teilmenge von  $E$  zugeordnet. Die Elemente von  $E$  heißen Ecken, die Elemente von  $K$  heißen Kanten.*<sup>1</sup>

Bei den Ecken wird eine Unterscheidung in Form eines Grads so vorgenommen, als dass dieser die zur Ecke hinführende Anzahl der Kanten bezeichnet.

### 2.2.1 Zufallsmethode

Als Erstes sei die Strategie des Zufalls zu erwähnen. Dabei wird beim Vorgang, das Ziel zu finden keine besondere Systematik verwendet, sondern die Tatsache, dass das Ziel erreicht wird, ist rein statistisch begründet und könnte gemein auch Glück genannt werden. Dabei geht man so vor, dass man beim Erreichen einer Kreuzung oder Weggabelung zufällig eine Richtung wählt. Eine bestimmte Systematik wird dadurch erreicht, indem man beim Abgehen der Wege nicht mehr solche Wege geht, die man zuvor schon gegangen ist. Dabei ist die Erfolgswahrscheinlichkeit davon abhängig, wie viele Möglichkeiten es gibt, das Ziel zu erreichen. Diese Methode ist wegen der hohen Anzahl an Möglichkeiten, die schon bei Irrgärten mit wenigen Abzweigungen entstehen nicht sehr effizient, führt aber dennoch in endlicher Zeit zum Ziel.

### 2.2.2 Anwendung der Zufallsmethode

Die Zufallsmethode wird bei Laborversuchen mit unintelligenten Robotern oder auch Mäusen angewendet, um einfache Verhaltensweisen zu untersuchen.

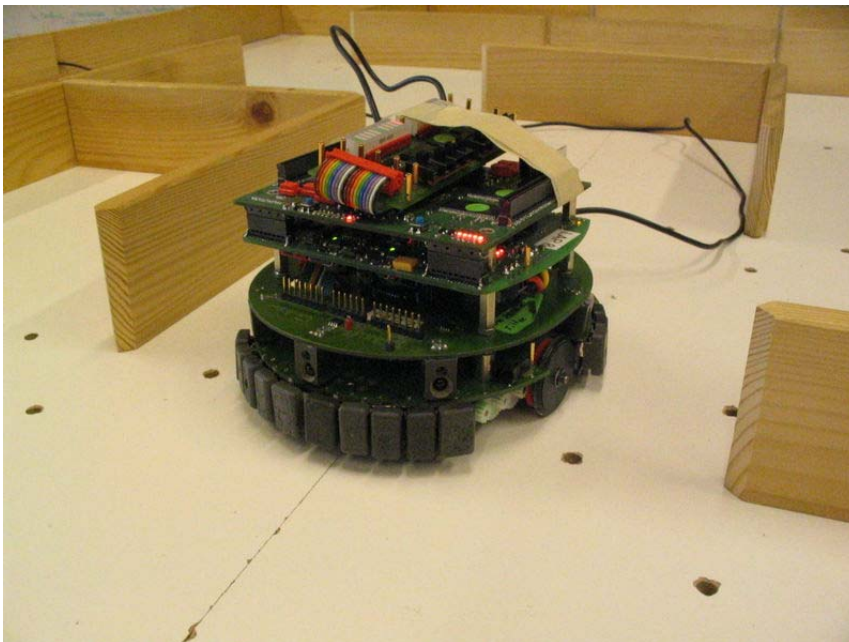


Abb. 1: Roboter in einem Irrgarten bei der Anwendung des Zufallsprinzips (Quelle: [http://en.wikipedia.org/wiki/Image:Cyclope\\_robot.jpg](http://en.wikipedia.org/wiki/Image:Cyclope_robot.jpg))

### 2.3.1 Methode von Trémaux

Die Methode von Trémaux auch bekannt als Tiefensuche gehört zu der Klasse der Suchalgorithmen. Ihr Name rührt daher, dass man sich eine Startecke sucht, von dem aus man in die Tiefe sucht. Bei der praktischen Anwendung in

einem Irrgarten gilt es einen Satz von Regeln zu befolgen, um den Algorithmus der Suche des Ziels anzuwenden:

- 1.) Beim Ankommen an einer „Sackgasse“ bzw. einer **Ecke vom Grad 1** kehrt man um und geht zur letzten Ecke zurück.
- 2.) Beim Ankommen an einer „Kreuzung“ bzw. **Ecke vom Grad  $\geq 3$**  kennzeichnet man die aktuelle Kante beispielsweise mit einem Haken. Zur Richtungsentscheidung ergeben sich folgende zu betrachtende Fakten:
  - Man läuft im Kreis: Die aktuelle Kante wurde zum ersten Mal markiert und die übrigen an der Ecke liegenden Kanten sind ebenfalls markiert. In dem Fall markiert man die aktuelle Kante ein zweites Mal und geht zur letzten Ecke zurück.
  - Es sind noch Kanten an der aktuellen Ecke, die nicht markiert sind: Man wählt von den unmarkierten Kanten die Erste von links und markiert diese.
  - Sonst: Man wählt die Kante an der aktuellen Ecke, die bisher nur einmal markiert wurde und markiert diese ebenfalls ein zweites Mal.

Zur Veranschaulichung ein Beispiel.

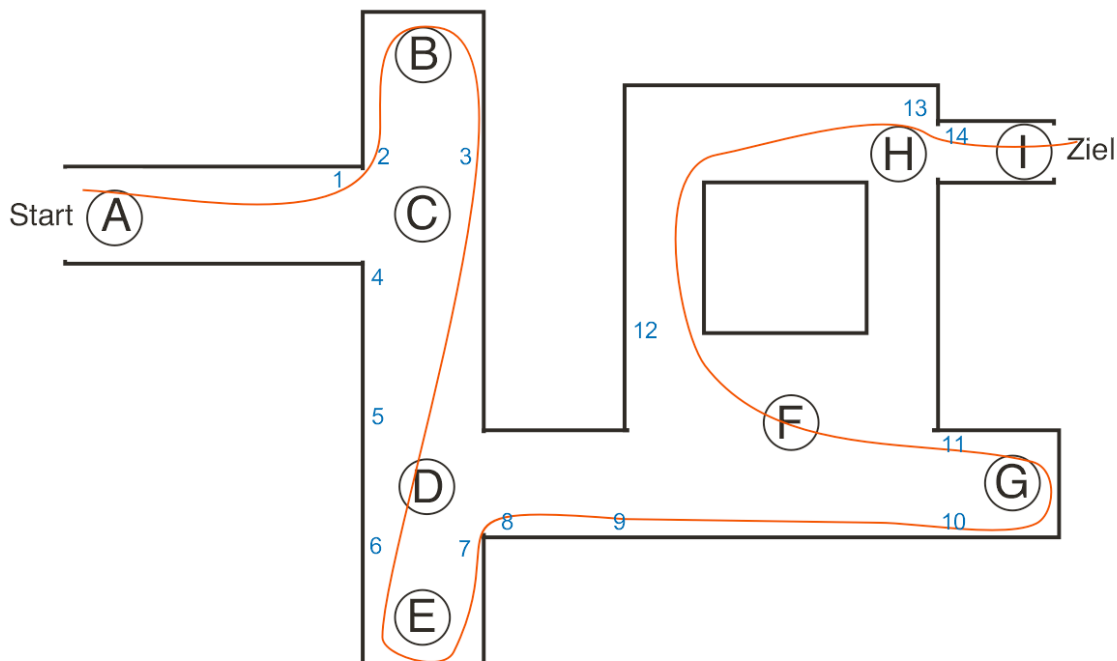


Abb. 2: Beispiel zur Anwendung des Tiefen-Algorithmus in einem Irrgarten

Wir starten bei Ecke A und gehen Richtung Osten. Wir sind an der Ecke C angekommen und markieren die Ecke mit einer 1. Wir bewegen uns in Richtung Norden zur Ecke B, wir kennzeichnen diese Kante mit einer 2. Beim Ecke B angekommen, erkennen wir eine Sackgasse und kehren um. Beim Verlassen dieser Kante markieren wir ein zweites Mal mit einer 3. An Ecke C bewegen wir uns Richtung Süden, nachdem wir die Kante mit einer 4 gesehen haben. Bevor wir an Ecke D gelangen markieren wir die verlassene Kante mit 5, bewegen uns weiter Richtung Süden und nachdem wir die Kante mit einer 6 gesehen haben, erkennen wir schließlich wieder, dass Ecke E ebenfalls eine Sackgasse ist und markieren diese beim Verlassen mit 7. Wieder an Ecke D angekommen, bewegen wir uns Richtung Osten, nachdem wir die Kante mit einer 8 gesehen haben. Beim Verlassen der Kante kennzeichnen wir diese mit einer 9. Wir erreichen Ecke F und bewegen uns weiter Richtung Osten und markieren die weitere Kante mit 10. Beim Erreichen der Ecke G erkennen wir ebenfalls, dass diese eine Sackgasse ist und markieren die Kante wieder beim Verlassen mit 11. An Ecke F angekommen, bewegen wir uns Richtung Norden und versehen die Kante mit einer 12. Wir gehen diese Kante entlang bis wir Ecke H erreichen und markieren die verlassene Kante mit 13. Wir bewegen uns weiter Richtung Osten und durchqueren nach der Markierung mit 14 eine weitere Kante bis zum Erreichen von Ecke I, dem Ziel.

### **2.3.2 Anwendung der Tiefensuche**

Der Algorithmus der Tiefensuche findet seine Anwendung in der Informatik zum Durchsuchen von Netzwerken. Er wird dahingehend genutzt, als dass er nach endlicher Zeit ans Ziel führt und konstruktiv ist, d. h. , dass es nicht nötig ist, mehrere Male denselben Weg zu gehen, wobei Hin- und Rückweg unterschiedliche Wege hinsichtlich dieses Kontextes darstellen. Netzwerke können in dem Zusammenhang vor allem das Internet mit verlinkten Seiten, aber auch einfachere Konstrukte, wie das Telefonnetz darstellen.

### **2.4.1 Linke/Rechte Hand Regel**

Die Idee dieses Verfahrens liegt darin, dass man entweder die linke oder die rechte Hand benutzt und sich so orientiert an den Wänden des Irrgartens

entlang bewegt. Dabei tritt das Problem auf, dass dieses Verfahren nicht immer zum Ziel führen muss und deswegen auch nicht immer endlich ist.

Dazu betrachte man folgendes Beispiel:

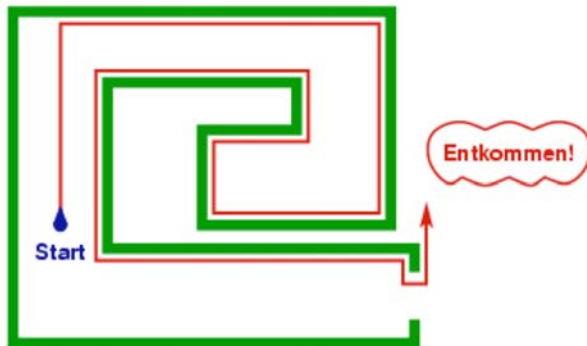


Abb. 3: Beispiel zur Anwendung der Linken/Rechten Hand Regel in einem Irrgarten (Quelle: [www-i1.informatik.rwth-aachen.de/~algorithmus/Algorithmen/algo6/algo6.pdf](http://www-i1.informatik.rwth-aachen.de/~algorithmus/Algorithmen/algo6/algo6.pdf))

Wie man hier sieht, funktioniert es wunderbar, wenn man sich vom Startpunkt weg immer mit der linken Hand entlang der Wand bewegt. Man gelangt zum Ziel, auch wenn es im Allgemeinen kürzer gewesen wäre, wenn man sich umgedreht hätte und nachdem man einmal links abgebogen wäre strikt auf das Ziel zugegangen wäre.

In einem weiteren Beispiel sieht man, dass dieser Algorithmus auch zu Problemen führen kann, aus denen man nicht mehr heraus kommt, wenn man sein Verfahren, um aus dem Irrgarten zu entflüchten nicht ändert.

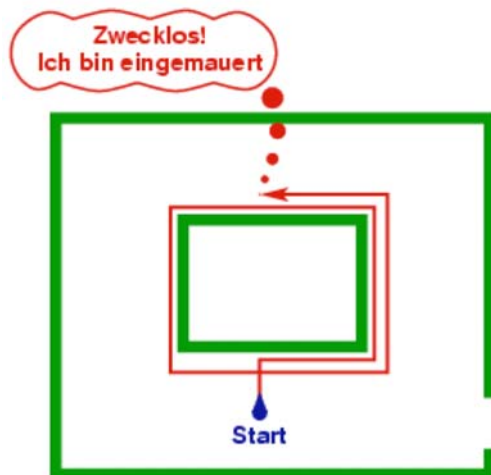


Abb. 4: Beispiel zur Anwendung der Linken/Rechten Hand Regel in einem Irrgarten (Quelle: [www-i1.informatik.rwth-aachen.de/~algorithmus/Algorithmen/algo6/algo6.pdf](http://www-i1.informatik.rwth-aachen.de/~algorithmus/Algorithmen/algo6/algo6.pdf))

Wie man hier sieht, folgert man falsch und erhält eine fehlerhafte Vorstellung des Aufbaus des Irrgartens. Das heißt, es hängt vom Zufall ab, ob man mit diesem Verfahren ans Ziel kommt oder nicht. Anders gesprochen kann man mit dem Verfahren schneller oder langsamer (gar nicht ans Ziel kommen) an das Ziel kommen, als mit der Zufallsmethode. Um diese Problematik zu vermeiden

gilt es diese Regel zu erweitern, sodass man auf den Pledge-Algorithmus kommt.

### 2.5.1 Pledge-Algorithmus

Die Idee, die erst mal hinter diesem Algorithmus steckt, ist die Idee, dass man zwar einen Ansatz macht, der auf 2.4.1 zurückzuführen ist, aber eine zusätzliche Idee hinzufügt. Und zwar ist es als sinnvoll zu erachten, dass man eine Wand, an der man sich entlang bewegt dann wieder verlässt, wenn man sich wieder in die Richtung bewegt, in die man gegangen ist, bevor man sich an der Wand orientiert hat. An dem obigen Beispiel erkennt man, dass dies zum Erfolg führt:

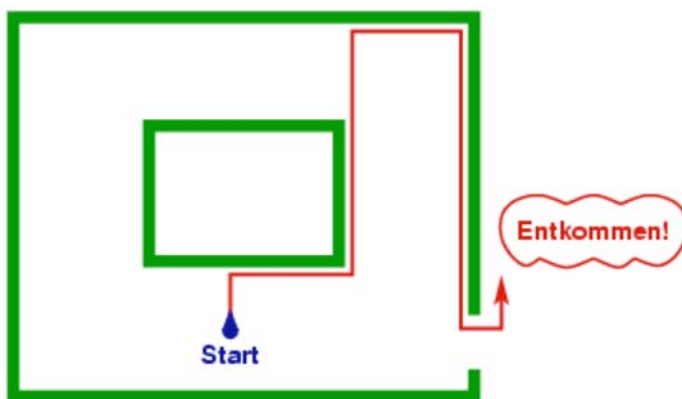


Abb. 5: Beispiel zur Anwendung der Linken/Rechten Hand Regel in einem Irrgarten mit Verlassen der Mauer (Quelle: [www-il.informatik.rwth-aachen.de/~algorithmus/Algorithmen/algo6/algo6.pdf](http://www-il.informatik.rwth-aachen.de/~algorithmus/Algorithmen/algo6/algo6.pdf))

Im Gegensatz dazu ist das andere Beispiel von oben aufzuführen, bei dem das Anwenden dieser vermeintlich verbesserten Lösung eben nicht mehr zum Erfolg führt:

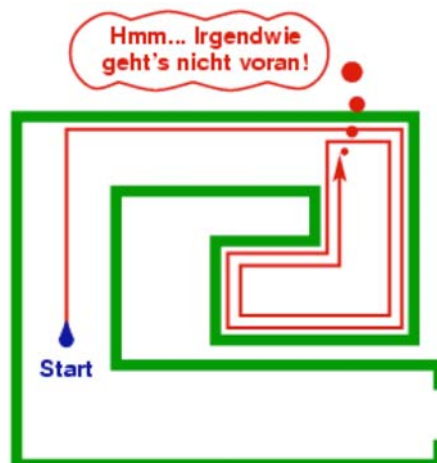


Abb. 6: Beispiel zur Anwendung der Linken/Rechten Hand Regel in einem Irrgarten mit Verlassen der Mauer (Quelle: [www-il.informatik.rwth-aachen.de/~algorithmus/Algorithmen/algo6/algo6.pdf](http://www-il.informatik.rwth-aachen.de/~algorithmus/Algorithmen/algo6/algo6.pdf))

Hier bedarf es einer Verbesserung der Idee, die Wand, an der man sich orientiert wieder los zu lassen. Dies ist dann eben der nach einem angeblich 12-jährigen Jungen benannten Pledge-Algorithmus. Er sieht so aus, dass man





Uhrzeigersinn, also rechts herum durchlaufen wird. Bei diesem Fall nimmt dann der Zähler gemäß dem Bild oben bei jedem gesamten Durchlauf um  $4 \cdot 90^\circ$  Winkel ab, sodass die Winkelsumme irgendwann nur noch negativ ist. Nach den Anweisungen des Algorithmus würde man dann aber ständig mit der linken Hand an der Wand entlang laufen ohne sie zu verlassen. Das hätte aber auch zur Folge, dass es gar keinen Weg nach draußen gibt. Das kann man sich an unten stehendem Beispiel einfach verdeutlichen:

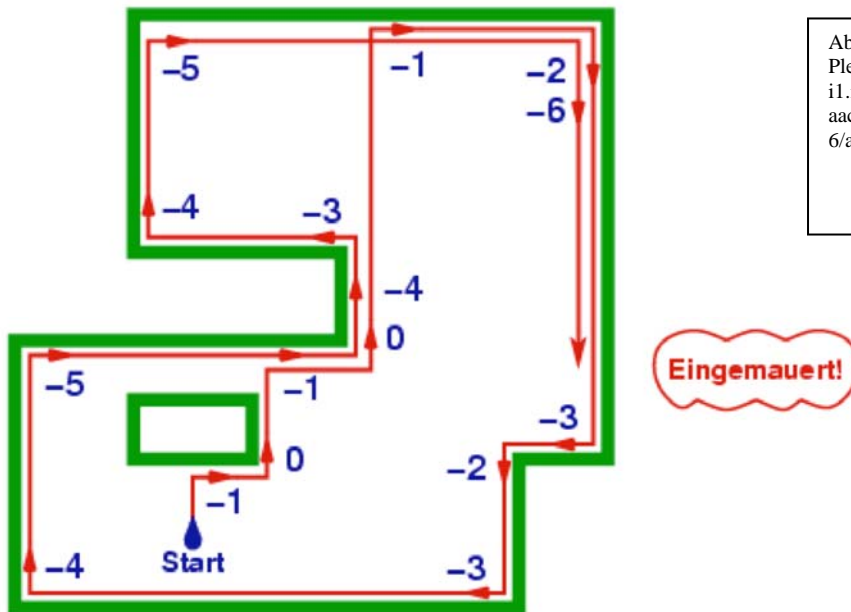


Abb. 8: Motivation zum Beweis des Pledge Algorithmus (Quelle: [www-il.informatik.rwth-aachen.de/~algorithmus/Algorithmen/algorithm6/algo6.pdf](http://www-il.informatik.rwth-aachen.de/~algorithmus/Algorithmen/algorithm6/algo6.pdf))

### 3.1 Zusammenfassung der Ergebnisse und persönliche Stellungnahme

Zusammenfassend ist zu nennen, dass allerhöchstens zwei der fünf genannten Algorithmen als effektiv zu bezeichnen sind. Der erste ist der Algorithmus zur Tiefensuche, der immer zum Ziel führt und mit etwas Glück auch sehr schnell ist. Etwas hinderlich ist vielleicht das Markieren der Kanten. Dahingegen kommt der Pledge-Algorithmus ohne Markierungen aus. Bei ihm muss im Gegenzug die Summe der Winkel betrachtet werden, was aber bei Winkeln, die  $90^\circ$  betragen nicht weiter schwierig ist. Bei Winkeln, die von  $90^\circ$  abweichen, gestaltet sich die Nutzung des Algorithmus in der Realität schon etwas schwieriger. Der Pledge-Algorithmus dauert vom Verfahren im Vergleich zum Algorithmus der Tiefensuche wahrscheinlich etwas länger, da bei ihm mehr Strecke zurückgelegt wird. Somit würde ich bei Nutzung eines Algorithmus für mich persönlich den Algorithmus der Tiefensuche prädestinieren.