

# Suchbaum-Modellierung

Gerhard Röhner

Studienseminar für Gymnasien Darmstadt  
Paul-Wagner-Str. 44A  
64285 Darmstadt  
GRoehner@t-online.de

**Abstract:** Die Modellierung von Problemen mit Suchbäumen wird vorgestellt. Die Suchbäume können mit unterschiedlichen Methoden nach Lösungen abgesucht werden. Für das Backtracking-Verfahren wird gezeigt, wie sich geometrische Strukturen eines Suchbaums in algorithmische Strukturen transformieren lassen. Der explizite Bezug zwischen der ikonischen Modellierungsebene und der symbolischen Implementierungsebene lässt im Gegensatz zu einer unreflektierten Übernahme fachwissenschaftlicher Darstellungen in den Informatikunterricht eine fruchtbare unterrichtliche Auseinandersetzung mit NP-Problemen zu.

## 1 Suchbaum-Modellierung

Die fachdidaktische Diskussion ist in den letzten Jahre in großem Maße durch die Auseinandersetzung mit dem Thema Modellierung geprägt. Entscheidenden Anteil daran hat die Entwicklung in der Fachwissenschaft, welche mit einem vereinheitlichten Modellierungsansatz repräsentiert durch die Unified Modelling Language (UML) eine standardisierte objektorientierte Modellierung ermöglicht. Fachdidaktisch wird Modellierung durch den informationszentrierten Ansatz von Hubwieser (vgl. [Hu00]) in das Zentrum des Interesses gestellt. Dabei wird von Modellierung als inhaltlichem Kern der Schulinformatik ausgegangen. Das Spektrum der Modellierungstechniken umfasst neben der objektorientierten Modellierung auch die Datenmodellierung, die zustandsorientierte und die funktionale Modellierung.

Der wesentliche fachdidaktische Mehrwert der Modellierung besteht darin, dass beim Problemlösen mit dem Computer eine planerische, konzeptionelle Modellierungsphase explizit durchlaufen wird, die zwischen dem zu lösenden Ausgangsproblem und der algorithmischen Lösung liegt. Unter lernpsychologischen Gesichtspunkten ist bedeutsam, dass die Modellierung zum großen Teil auf der ikonischen Ebene und die Implementierung auf symbolischen Ebene stattfindet. Durch die Nutzung zweier Repräsentationsebenen wird der Lernprozess erheblich unterstützt. Transformationen zwischen ikonischer und symbolischer Ebene leisten einen bedeutsamen Beitrag zur Steigerung der Lesekompetenz und Problemlösefähigkeit.

Modellierungen können auch sehr hilfreich bei der Vermittlung von Inhalten sein. So lässt sich zum Beispiel die Client-Server-Kommunikation hervorragend am einem Sequenzdiagramm darstellen [Rö98a]. Eine rein textliche Beschreibung kann die komplexen Zusammenhänge nur bedingt vermitteln. Beim Unterrichtsthema Datenbanken ist die Nutzung von Entity-Relationship-Diagrammen eigentlich selbstverständlich. Aber auch Zustandsdiagramme lassen sich über den bekannten Rahmen der Automaten hinaus immer wieder sinnvoll einsetzen, z.B. beim Verbindungsaufbau nach dem POP3-Protokoll [Rö98b].

Modellierungstechniken sind dann wertvoll, wenn sie ein breites Anwendungsspektrum aufweisen, also zur Lösung einer großen Klasse von Problemen eingesetzt werden können. Die künstliche Intelligenz hat sich mit ganz allgemeinen Problemlösungsansätzen auseinandergesetzt, wobei Suchbäume und Suchverfahren eine ganz bedeutsame Rolle spielten. In der Lehre und im Informatikunterricht werden dabei meist didaktisch reduzierte, realitätsferne Beispiele herangezogen. Damit stehen diese Beispiele potenziell unter dem Verdacht, dass durch die Reduktion ein Großteil der Modellierung praktisch schon vorweg genommen ist. Wie wir sehen werden, ist aber auch bei solchen Beispielen noch genügend Modellierungsspielraum drin. Ihr Vorzug besteht in der didaktischen Reduktion auf den Kern, der für die Auseinandersetzung mit der Suchbaum-Modellierung zentral ist. Im Folgenden wird ein neuer Zugang zum klassischen Themenbereich Suchverfahren über die Suchbaum-Modellierung dargestellt.

## 2 Der Suchbaum

An einem kleinen Labyrinth-Problem – die Maus möchte zum Käse – lässt sich ein erster Suchbaum anschaulich entwickeln.

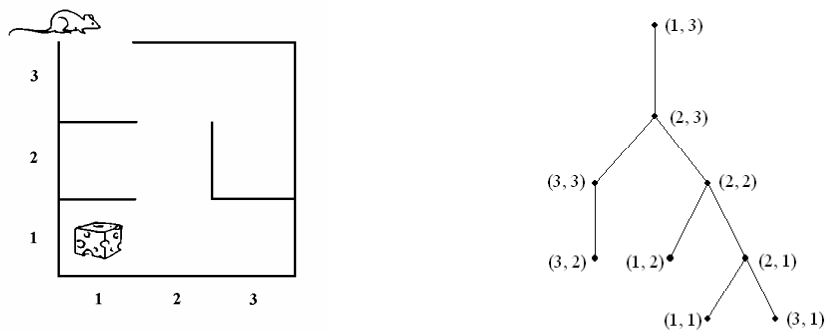


Abbildung 1: Labyrinth-Problem als Suchbaum modelliert

Die Knoten des Suchbaums stellen die Position der Maus im Labyrinth dar. Der Wechsel der Maus von einem auf ein benachbartes Feld im Labyrinth wird durch eine Kante zwischen dem bisherigen und dem neuen Knoten repräsentiert. Der maximale Verzweigungsgrad beträgt bei einem Labyrinth drei, weil man natürlich nicht zu dem

Feld zurückgeht, von dem man gerade gekommen ist. Lediglich die Wurzel eines Labyrinth-Suchbaums kann sich in alle vier Richtungen verzweigen.

In diesem überschaubaren Beispiel lässt sich der Suchbaum komplett hinzeichnen. In der Regel ist dies nicht möglich, wie wir bei den nächsten Beispielen sehen werden. Daher ist es um so wichtiger, dass die Suchbaum-Modellierung soweit zeichnerisch ausgeführt wird, dass die charakteristische Struktur des Suchbaums deutlich wird. Wir betrachten nun zwei weitere Probleme, die sich mit Suchbäumen modellieren lassen und werden erst im Anschluss an diese Beispiele Verfahren betrachten mit denen Suchbäume nach Lösungen abgesucht werden können.

### 3 8-Damen-Problem

Beim 8-Damen-Problem geht es darum, acht Damen so auf ein Schachbrett aufzustellen, dass sich keine zwei gegenseitig schlagen können.

							D
	D						
			D				
D							
						D	
				D			
		D					
					D		

Abbildung 2: Eine Lösung des 8-Damen-Problems

Da nicht erkennbar ist wie eine oder alle Lösungen berechnet werden können kommen Schülerinnen und Schüler bald darauf, dass man Lösungen suchen muss und zwar systematisch. Die erforderliche Systematik bekommt man am besten durch eine unterstützende Zeichnung in den Griff, woraus sich im Unterricht schnell ein erster Suchbaum ergibt.

Im schlimmsten Fall weist der Suchbaum auf der ersten Stufe 64 Verzweigungen, auf der zweiten Stufe 63, auf der dritten Stufe 62 usw. auf. Er modelliert also das Setzen der ersten Dame auf eines der 64 verfügbaren Felder, der zweiten Dame auf eines der 63 restlichen Felder und das der dritten auf eines der übrigen 62 Felder. Diese Modellierung führt zu einem Suchbaum mit  $64 \cdot 63 \cdot 62 \cdot \dots \cdot 57 = 178.462.987.637.760$  zu untersuchenden Stellungen. Eine Kante im Suchbaum repräsentiert bei der Suchbaum-Modellierung das Setzen einer Dame. Der Knoten am unteren Ende einer Kante kann dementsprechend mit der Bezeichnung des Schachfeldes versehen werden, auf das die Dame gesetzt wird. Längs eines Pfades von der Wurzel zu einem Blatt stehen dann in den Knoten alle Felder die zu einer zu untersuchenden Stellung gehören.

Eine bessere Modellierung ergibt sich, wenn man beachtet, dass in jeder Spalte des Schachbretts nur eine Dame stehen kann. Für jede Dame bleiben dann nur noch 8 mögliche Positionen übrig, der Suchbaum reduziert sich auf  $8^8 = 16.777.216$  zu untersuchende Stellungen.

Natürlich darf auch in jeder Reihe nur eine Dame stehen. Wenn also die erste Dame in der ersten Spalte gesetzt ist, so bleiben für die nächste Dame in der zweiten Spalte nur noch sieben Möglichkeiten übrig. Bei jeder weiteren Dame reduziert sich die Anzahl übrigbleibenden Setzmöglichkeiten um eins. Insgesamt weist der Suchbaum noch  $8! = 40.320$  zu untersuchende Stellungen aus.

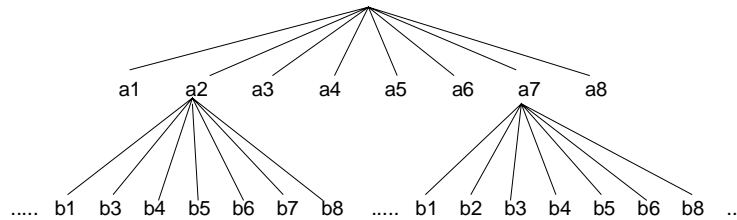


Abbildung 3: Suchbaum für das 8-Damen-Problem

Im Bild sind die Knoten mit den Bezeichnungen der Schachfelder versehen auf die die jeweilige Dame gesetzt wird. Wenn auf a2 eine Dame steht kann die Dame in Spalte b nicht auf b2 gestellt werden. Die Güte der Modellierung ist in diesem Beispiel durch die Anzahl der Pfade von der Wurzel zu einem Blatt bzw. durch die Anzahl der Blätter im Suchbaum bestimmt, was der Anzahl der zu untersuchenden Stellungen entspricht.

## 4 Erbteilungsproblem

Eine Einkleidung des Partitionsproblems ist das Erbteilungsproblem aus dem 7. Bundeswettbewerb Informatik von 1988/89:

Die Baronin von Birlinghoven hat ihren beiden Töchtern eine Truhe voller Goldmünzen hinterlassen. Ihr Testament bestimmt, dass das Gold einem benachbarten Kloster zukommt, falls es den Töchtern nicht gelingt, den Inhalt der Truhe wertmäßig genau in zwei Hälften untereinander aufzuteilen. Die Goldmünzen haben nur ganzzahlige Werte.

Beispiel: Eine Truhe Goldmünzen mit den Werten 1, 9, 5, 3, 8 Taler könnten die Töchter in die Hälften 1, 9, 3 Taler und 5, 8 Taler teilen.

Aufgabe: Schreibe ein Programm, das bei Eingabe einer Folge ganzer Zahlen für die in der Truhe vorkommenden Werte die beiden Erbteile aufzählt, anderenfalls das Erbe dem Kloster zuspricht, wenn eine Aufteilung nicht möglich ist. Schicke uns mindestens 5 Beispiele mit verschiedenen Truheninhalten.

Der Inhalt einer Truhe sei: 15, 27, 39, 7, 23, 56, 13, 39, 22, 5, 42, 34

Wir betrachten mehrere Modellierungsansätze, denen alle die Idee des sukzessiven Hinzunehmens einer Münze zum Erreichen der Erbteihälfte zu Grunde liegt. Im Beispiel stehen fünf Möglichkeiten für die erste Münze zur Verfügung. Für die zweite Münze stehen jeweils noch vier, dann drei, dann zwei und zuletzt eine. Daraus ergibt sich folgender Suchbaum:

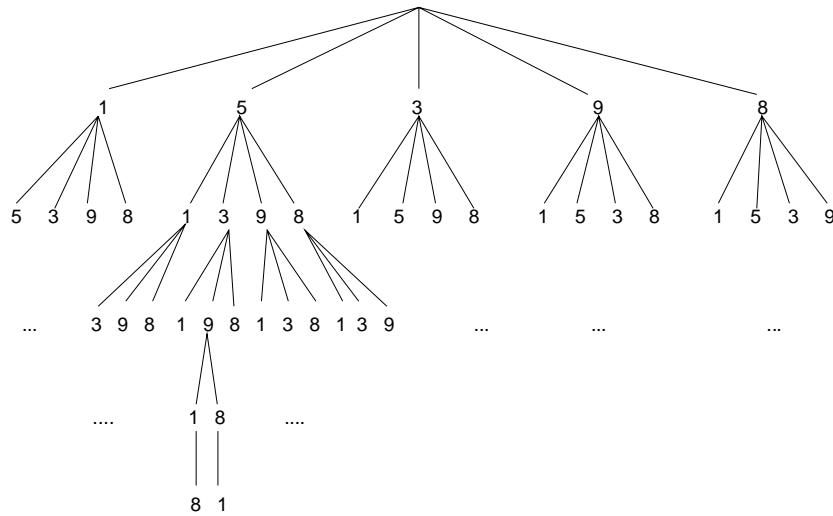


Abbildung 4: Suchbaum für das Erbteilungsproblem mit Wiederholungen

Eine Lösung des Erbteilungsproblems entspricht nicht wie beim Layrinth- oder Damenproblem einem Pfad von der Wurzel zu einem Blatt, sondern zu einem inneren Knoten, wobei die Summe der Knoten längs des Lösungspfades gleich der Erbteihälfte ist. Ein solcher Suchbaum muss daher nicht komplett abgesucht werden.

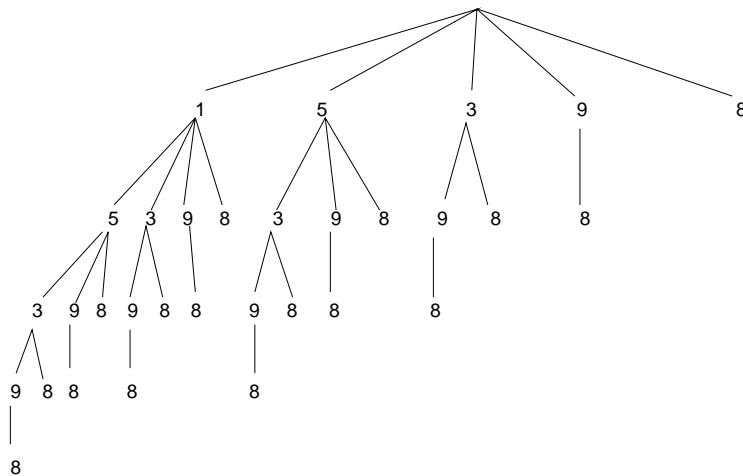


Abbildung 5: Suchbaum für das Erbteilungsproblem ohne Wiederholungen

Die Diskussion dieser Modellierung zeigt, dass unnötig viele Kombinationen untersucht werden, denn schließlich kommt es auf die Reihenfolge der Münzen nicht an. Eine Modellierung, die diesen Umstand berücksichtigt, führt auf den Suchbaum von Abbildung 5.

Er ist asymmetrisch, die Zahl der Verzweigungen nimmt nach rechts hin jeweils um eins ab. Die Söhne eines Knotens entsprechen den rechten Brüdern des Knotens, während bei der vorherigen Modellierung auch die linken Brüder als Sohnknoten auftreten. Betrachten wir den an der Wurzel hängenden Knoten 5. Dadurch dass sein linker Bruder 1 nicht als Sohnknoten benutzt wird, werden Wiederholungen ausgeschlossen, bei denen erst die Münze 5 und später die Münze 1 ausgewählt wird.

In der Literatur findet man auch gänzlich andere Suchbaum-Modellierungen. In [He90] wird beispielsweise ein binärer Suchbaum verwendet, dort allerdings am Beispiel des Rucksackproblems. Übertragen auf das Erbteilungsproblem würde sich der Suchbaum aus Abbildung 6 ergeben.

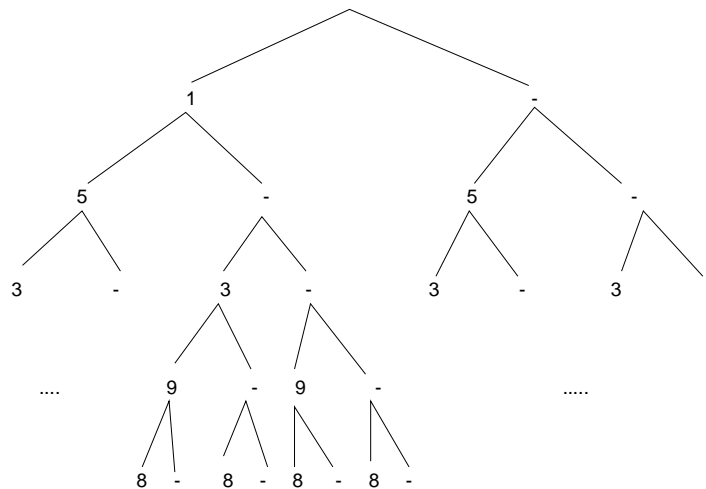


Abbildung 6: Binärer Suchbaum für das Erbteilungsproblem

Ausgehend von der Wurzel wird bei jedem Knoten entschieden, ob die nächste Münze genommen wird (linke Kante) oder nicht (rechte Kante). Bei dieser Suchbaum-Modellierung sind Lösungen Wurzel-Blatt-Pfade. Da es sich um binäre Suchbäume handelt, allerdings nicht zu verwechseln mit der Datenstruktur binärer Suchbaum, könnte man meinen, dies sei eine einfachere und bessere Modellierung. Doch dies trifft nicht zu, denn die erste Modellierung führt bei  $n$  Münzen auf  $n!$  Suchpfade, die zweite hat wegen der Elimination von Wiederholungen deutlich weniger, während die Modellierung als binärer Suchbaum auf  $2^n$  Pfade kommt. Binäre Suchbäume lassen sich durch Elimination der jeweils rechten Knoten in einen allgemeinen Suchbaum nach der zweiten Modellierung zurückführen. Dazu müssen die linken Söhne der eliminierten Knoten mit den übergeordneten Knoten verbunden werden, wodurch diese sich zu mehrfach verzweigten Knoten wandeln.

## 5 Backtracking

Anhand dreier Beispiele haben wir gesehen, wie Probleme durch Suchbäume modelliert werden können. Am 8-Damen- und Erbteilungsproblem wurde zusätzlich klar, dass ein Problem auf verschiedene Arten als Suchbaum modelliert werden kann und dass es sich lohnt, problemspezifisches Wissen für die Modellierung zu benutzen, um möglichst kleine Suchbäume zu erhalten. Ähnlich wie bei objektorientierter Modellierung, ER- oder Zustands-Diagrammen kann, wie wir anschließend sehen werden, die Umsetzung eines gefundenen Modells in ein Programm nach festen Regeln erfolgen. Ein Suchbaum kann mit verschiedenen Verfahren, z. B. Tiefensuche, Breitensuche oder heuristische Suche, nach Lösungen abgesucht werden. Wir betrachten das auf der Tiefensuche basierende Backtracking-Verfahren.

Backtracking ist ein Verfahren zum Absuchen von Suchbäumen, das nach dem Prinzip der Tiefensuche einen Suchbaum von links nach rechts hin absucht. Führt ein Suchpfad in eine Sackgasse, dann wird der jeweils letzte Schritt rückgängig gemacht und mit der nächsten möglichen Alternative die Suche fortgesetzt. Das Zurücknehmen von Schritten und erneute Vorangehen wird solange wiederholt, bis eine Lösung des vorliegenden Problems gefunden ist oder man erkennt, dass das Problem keine Lösung besitzt.

Im Maus-Käse-Labyrinth lässt sich das Backtracking-Verfahren anschaulich demonstrieren. Ein Beispiel aus der Erfahrungswelt der Schülerinnen und Schüler ist die Arbeit mit einem Browser. Mit dem „Zurück“-Button wird ein Suchschritt zurückgenommen.

Ist man nur an einer Lösung interessiert, kann die Suche mit der ersten Lösung abgebrochen werden. Werden alle Lösungen gesucht oder soll eine optimale Lösung gefunden werden, so muss der Suchbaum abgesucht werden.

Im Unterschied zu den mir bekannten Darstellungen in der Fach- oder fachdidaktischen Literatur wird im Folgenden der enge Zusammenhang zwischen der ikonischen Repräsentation des Suchbaums und dem Backtracking-Algorithmus hergestellt. Die konsequente Umsetzung der geometrischen Zusammenhänge in eine äquivalente algorithmische Ablaufstruktur schafft eine enge Verzahnung zwischen ikonischer und symbolischer Ebene, die für ein echtes Verständnis des Backtracking-Algorithmus unverzichtbar ist.

Während fachwissenschaftliche Darstellungen meist ganz auf die ikonische Darstellung verzichten, findet man in fachdidaktischer Literatur hin und wieder problematische Darstellungen. Beim schon erwähnten Rucksackproblem aus [He90] ist der binäre Suchbaum eine ungünstige Modellierung. Beim CD-Problem von [Ta95] wird sogar der Suchbaum teilweise als Datenstruktur aufgebaut und in den Knoten die komplette Pfadinformation abgelegt, wodurch die Problemstellung unnötig verkompliziert wird. Ein Bezug zwischen Suchbaum und Backtracking-Algorithmus wird nicht hergestellt, stattdessen wird einfach auf das Backtracking-Schema von [Wi83] verwiesen.

## 6 Absuchen des Suchbaums

Der Zusammenhang zwischen Suchbaum und Backtracking-Algorithmus wird anhand von Abbildung 7 entwickelt, die einen Ausschnitt aus dem Suchbaum des Erteilungsproblems darstellt. Jeder Teilbaum besteht aus einem Vaterknoten, im Beispiel ist es der Knoten 1 und den Verzweigungskanten, an denen im Beispiel vier Sohnknoten dranhängen.

Damit die Suche systematisch und von links nach rechts den Suchbaum absucht, müssen wir dafür sorgen, dass der Reihe nach alle Kanten benutzt werden. Da wir aus der Modellierung heraus wissen, wie viele Verzweigungen jeweils vorhanden sind, können wir mittels einer for-Schleife alle Verzweigungen nacheinander untersuchen. Wird nur eine Lösung gebraucht, so nimmt man eine while-Schleife, die schon dann terminiert, wenn eine erste Lösung gefunden wurde.

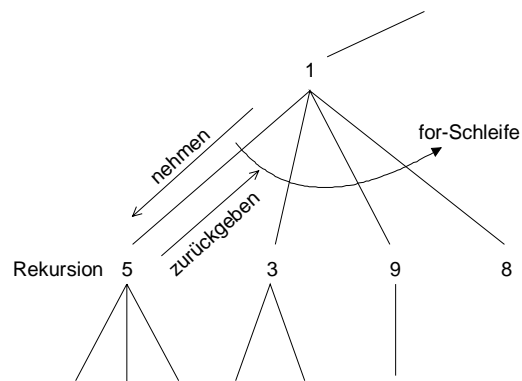


Abbildung 7: Beziehung zwischen Suchbaum und Backtracking-Algorithmus

Mit der for-Schleife kommen der Reihe nach alle Kanten dran. Im Bild ist die Situation für die erste Kante dargestellt. Um den Suchbaum längs dieser Kante abzusuchen müssen wir die Goldmünze 5 in unsere aktuelle Auswahl aufnehmen. Der bisherige Pfad von der Wurzel zum aktuellen Vaterknoten wird um eine Kante verlängert, der Sohnknoten wird zum aktuellen Knoten.

Im Allgemeinen wird man nur dann längs einer Kante zum Sohnknoten gehen, wenn die Hinzunahme des Sohnknotens immer noch eine Lösung erlaubt. Würde im Beispiel durch die Hinzunahme einer Münze die bis dahin akkumulierte Münzsumme die Gesamt Münzhälfte überschreiten, so muss die zugehörige Kante nicht besritten werden. Eine Kante wird daher nur dann besritten, wenn der Sohnknoten eine Lösung zulässt. Wir sagen dazu in Wirth-Manier, dass der Sohnknoten *annehmbar* sein muss.

Wie wir bei der Modellierung gesehen haben, kann eine Lösung aus einem Pfad von der Wurzel zu einem inneren Knoten bestehen. Daher kann der bisherige Pfad bis zum Sohnknoten eine Lösung sein. Dies muss geprüft werden. Liegt eine Lösung vor, so wird diese ausgegeben. Ansonsten wird die Suche vom Sohnknoten aus nach dem gleichen Verfahren fortgesetzt. Es liegt somit eine verschachtelte Struktur von for-Schleifen vor.



Da Wiederholung durch Schachtelung nichts anderes als Rekursion ist – die im konkreten Unterrichtsgeschehen vorab behandelt worden sein muss – wird der Backtracking-Algorithmus rekursiv. Ist der komplette Teilbaum mit Wurzel 5 abgesucht geht es zum Vaterknoten 1 zurück. Dabei muss die Goldmünze wieder zurück gegeben werden. Die for-Schleife sorgt dann dafür, dass es mit der nächsten Kante – die dann zum Knoten 3 führt – weitergeht.

Die Übersetzung der Verzweigungen an einem Knoten im Suchbaum in einen systematischen Suchalgorithmus führt also auf diesen Algorithmus:

#### Algorithmus SuchbaumAbsuchen(Vaterknoten)

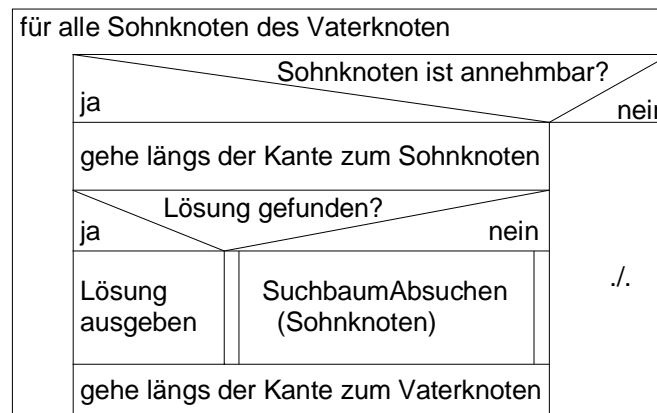


Abbildung 8: Backtracking-Algorithmus zum Absuchen von Suchbäumen

Bei der Programmierung dieses Verfahrens auf ein spezielles Suchproblem sollte darauf geachtet werden, dass das Gehen längs einer Kante zum Sohn- bzw. Vaterknoten explizit gemacht wird und dabei jeweils konkrete Anweisungen ausgeführt werden.

Hat man wie in Abbildung 6 einen binären Suchbaum modelliert, so entfällt die for-Schleife. Stattdessen verwendet man eine Sequenz für das Absuchen der beiden Teilbäume. Für die linke Kante muss eine Annehmbarkeitsprüfung stattfinden, für die rechte nicht. Zweimal müssen die Kanten beschritten werden und ein rekursiver Aufruf erfolgen. Da beim binären Suchbaum Lösungen Pfade von der Wurzel zu einem Blatt sind, wird nur beim Erreichen eines Blatts geprüft werden, ob eine Lösung vorliegt.

## 7 Lösung

Hat man mittels Backtracking eine Lösung gefunden, so soll diese ausgegeben werden. Die Lösung wird geometrisch repräsentiert durch einen Pfad im Suchbaum. Das Absteigen längs einer Kante bedeutet die Hinzunahme eines weiteren Knotens, das Zurückgehen längs einer Kante das Wegnehmen des letzten Knotens. Demnach passt am besten ein Stack mit seinen Push- und Pop-Operationen zur Verwaltung des Pfades.

Allerdings verfügt der abstrakte Datentyp Stack über keine Standardoperation zum Ausgeben der Elemente. Lediglich bei der Suche nach einer einzigen Lösung könnte die Lösung beim Zurückgehen in umgekehrter Form ausgegeben werden. Ist die Stack-Klasse von einer Listen-Klasse abgeleitet, kann man über die geerbten Zugriffsmethoden die Ausgabe realisieren. Alternativ speichert man den Pfad in einem Feld, das wie ein Stack benutzt wird. Die Eingangsdaten des Problems müssen in einer eigenen Datenstruktur verwaltet werden.

Wenn wie im Beispiel des Erbteilungsproblems alle Knoten des Lösungspfades einen additiven Beitrag zur Lösung liefern, so ist es sinnvoll, die bisher erreichte Teilsumme bei jedem rekursiven Aufruf als weiteren Parameter anzugeben. Dies vereinfacht auch die Prüfung, ob ein Sohnknoten annehmbar ist.

Beim asymmetrischen Suchbaum aus Abbildung 5 muss man sich Gedanken machen, wie man die unterschiedliche Anzahl von Verzweigungen in den Griff bekommt. Sind die verfügbaren Münzen in einem Feld gespeichert so, geht das einfach über den Feldindex des Sohnknotens.

		↓			
Münzindex	1	<b>2</b>	3	4	5
Münzwert	1	<b>5</b>	3	9	8

Wenn der Münzindex des aktuellen Knotens 2 ist, so sind lediglich die rechts davon stehenden Knoten als Sohnknoten zu betrachten, also alle ab dem um eins größeren Münzindex 3.

## 8 Fazit

Die Suchbaum-Modellierung stellt ein allgemeines Modellierungsverfahren dar. Mit einem Backtracking-Algorithmus kann ein Suchbaum abgesucht werden. Andere Verfahren wie zum Beispiel Breitensuche oder heuristische Suche sind auch möglich. Viele schwierige Probleme, insbesondere NP-vollständige Probleme, lassen sich mit einem Suchbaum und dem Backtracking-Algorithmus prinzipiell lösen. Die exponentielle Zeitkomplexität setzt der praktischen Einsetzbarkeit allerdings enge Grenzen.

Im Informatikunterricht können Suchbäume und Backtracking als Zugang zu Fragen der Berechenbarkeit genutzt werden. Rekursion muss vorausgesetzt werden und erfährt hier eine Vertiefung. Der Zusammenhang zwischen Rekursion und Datenstruktur Stack kann am Beispiel der Verwaltung des Lösungspfades erläutert und somit ein zentrales Wirkprinzip verdeutlicht werden. Kenntnisse von Suchbäumen und deren algorithmischer Abarbeitung lassen sich auf die Arbeit mit Bäumen als Datenstrukturen transferieren.

Schülerinnen und Schüler sollten sich mit mindestens drei einfachen Beispielen auseinander setzen, wenn sie diese Methodik in ihr verfügbares Problemlöserpertoire aufnehmen sollen. Dabei sollten sie unterschiedliche Fälle kennen lernen: die Suche nach einer, allen und der besten Lösung.

Das aus der Mathematikdidaktik von Bruner bekannte Prinzip der Nutzung mehrerer Repräsentationsebenen enaktiv, ikonisch und symbolisch ist auch im Informatikunterricht ertragreich. Es kommt allerdings entscheidend darauf an die Beziehungen zwischen diesen Repräsentationsebenen explizit zu machen. In der Hochschulformatik wird zu oft nur auf der symbolischen Repräsentationsebene gearbeitet.

Die klare Trennung zwischen der Modellierung von Suchbäumen und den Verfahren zum Absuchen von Suchbäumen akzentuiert den Modellierungsaspekt in besonderer Weise. Auf diese Weise wird ein bekanntes Gebiet der Schulformatik in zeitgemäßer Form erschlossen.

## **Literaturverzeichnis**

- [He90] Hermes, A., Stobbe, D.: Informatik zwei. Klett, 1990.
- [Hu00] Hubwieser, P.: Didaktik der Informatik. Springer, 2000.
- [Rö98a] <http://www.bildung.hessen.de/abereich/inform/skii/material/delphi/interne3.htm>
- [Rö98b] <http://www.bildung.hessen.de/abereich/inform/skii/material/delphi/interne5.htm>
- [Ta95] Taake, G.: Modulzerlegung von Backtracking-Algorithmen. In: LOG IN 5/6 1995.
- [Wi83] Wirth, N.: Algorithmen und Datenstrukturen. B. G. Teubner, 1983.