

Die Aussagekraft von Beispielen

„... wo es auf die Feststellung einer neuen oder zweifelhaften Meinung ankommt, ein einziges gründlich dargestelltes Ereignis belehrender ist als zehn bloss berührte.“

(Clausewitz, „Vom Kriege“)

Jürg Nievergelt · André Behr

Der Mensch verfügt über die bemerkenswerte Fähigkeit, aus einem einzigen Beispiel universelle Regeln abzuleiten.

Allgemeiner Unterricht sollte dieses Talent ansprechen und Beispiele ins Zentrum rücken. Sind sie gut gewählt, wirken sie

wie Samen, aus denen Pflanzen wachsen, indem an einem konkreten Fall die Vielfalt möglicher Strukturen illustriert wird, die daraus hergeleitet werden können. In der Informatik lassen sich kurze, prägnante und damit lehrreiche Beispiele vor allem in Bereichen finden, die stark algorithmisch und mathematisch geprägt sind. Als Anregung, sich selbst eine Beispielsammlung anzulegen, besprechen wir grundlegende Begriffe der Informatik wie Zustandsraum, Korrektheit und Terminierung von Algorithmen.

Beispiele im Spannungsfeld zwischen Einzelfall und Allgemeinheit

Wissenschaft strebt nach allgemeinen Aussagen. Sie versucht, so viele Einzelfälle wie möglich in umfassende Theorien einzubetten. In unserem Alltag interessieren uns allgemeine Aussagen jedoch nicht in erster Linie. Ein Beispiel ist der Energieerhaltungssatz der Physik. Fährt man mit dem Fahrrad eine Strasse hinunter in ein Tal, muss man in der Regel an einer Kreuzung stoppen. Dadurch geht all die Bewegungsenergie verloren, die einem ohne Zwischenhalt dank des Erhaltungssatzes auf der gegenüberliegenden Talseite Beinarbeit eingespart hätte.

Weil die Relevanz allgemeiner Gesetze für das tägliche Leben oft schwer zu erkennen ist, ergeben sich im Unterricht Probleme. Gesetze in ihrer mathe-

matischen Darstellung gelten als schwer verständlich und ihre Vermittlung erfordert einiges an Aufwand. Am besten helfen da konkrete Beispiele, die für sich selbst sprechen. Sie illustrieren die Problematik und schaffen eine intuitive Grundlage, von der man später auf allgemeine Aussagen schliessen kann.

Unterricht auf Stufe der Allgemeinbildung sollte also, im Unterschied zur Ausbildung von Spezialisten, nicht von Beweisen allgemeiner Gesetze, sondern von Beispielen geprägt sein. Dabei hilft es, wenn man sich eine Sammlung prägnanter Beispiele anlegt. Dann lässt sich auch leichter ein Wochenpensum von vielleicht 20 Stunden bewältigen. Elegante und lehrreiche Beispiele findet man in der Informatik vorwiegend bei mathematisch darstellbaren Themen. In der Folge präsentieren wir einige solcher Beispiele. Sie sind voneinander unabhängig und können jedes in etwa 10 Minuten beschrieben werden. Alle sind einfach und dürften bekannt sein. Wir lenken die Aufmerksamkeit des Lesers auf die allgemein gültigen Ideen, welche das Beispiel illustriert, und nutzen die erstaunliche menschliche Fähigkeit, aus einem einzigen Beispiel auf eine allgemeine Regel zu schliessen.

DOI 10.1007/s00287-006-0062-3
© Springer-Verlag 2006

Prof. em. Jürg Nievergelt
Dept. Informatik,
ETH Zürich,
8092 Zürich,
E-Mail: jn@inf.ethz.ch

André Behr
Schaffhauserstrasse 25,
8006 Zürich,
E-Mail: behr@dia1.eunet.ch

Zustandsräume und die Geschichte der Kannibalen und Missionare

Im Alltag sind wir es uns gewohnt, Ereignisse in ihrem zeitlichen Ablauf wahrzunehmen. Fragt uns ein Fremder, wie er ein bestimmtes Ziel erreichen kann, neigen wir dazu, ihm den Weg als Pfadbeschreibung zu schildern: „Bei der zweiten Verkehrsampel links über die Brücke, dann rechts.“ Darf der Fremde jedoch aus irgendeinem Grunde bei der Ampel nicht links abbiegen, hilft ihm diese Pfadbeschreibung wenig. Natürlich könnte man ihm für solche Fälle Alternativen mitgeben, doch bei weit entfernten Zielen würde diese Art der Wegbeschreibung allzu kompliziert.

Im Unterschied zu Pfadbeschreibungen, die auch von Kindern verstanden werden, liefert eine Landkarte die Beschreibung aller möglichen Wege von einem beliebigen Ausgangspunkt zu einem beliebigen Ziel. Ist einer der Wege gesperrt, ist der Reisende in der Lage, andere selbst zu suchen. Als statische Beschreibung nimmt die Landkarte keinen expliziten Bezug auf zeitliche Abläufe. Dafür verlangt ihre Interpretation etwas Schulung.

In der Technik existiert zum Begriff der Landkarte als statische Beschreibung aller möglichen zeitlichen Abläufe eine Analogie: der Zustandsraum. Die denkbaren Verhaltensweisen technischer Systeme sind so vielfältig, dass eine Aufzählung unmöglich wird. Die Verkehrssteuerung auf einem Bahnknotenpunkt beispielsweise lässt viele zeitliche Abläufe zu wie „Falls Zug Z mit mehr als 3 Minuten Verspätung einfährt, dann sollen Signale A, B und C wie folgt leuchten“. Wie aber überprüft man die Konsistenz solcher Abweichungen vom Fahrplan mit den Anordnungen für alle anderen Züge und Signale? Das erfordert ein statisches Modell aller Zustände des Systems „Bahnknoten“ sowie eine klare Trennung in erlaubte und unerlaubte Zustände.

Obschon der Begriff Zustandsraum natürlich ist und an vielen anschaulichen Beispielen erläutert werden kann, wird er an den Schulen kaum systematisch erklärt und eingesetzt. Das ist schade, denn dieser Begriff erlaubt es, viele Probleme, die mit dem Verhalten unterschiedlichster Systeme zu tun haben, einfacher und genauer zu lösen und zu erklären, als mit der expliziten Aufzählung zeitlicher Abläufe. Wir illustrieren dies am Beispiel der wohl bekannten Denkaufgabe der Kannibalen und Missionare.

Für ihre Reise durch den Dschungel haben drei Missionare drei Kannibalen angeheuert. Sie gelangen an einen Fluss, an dessen Ufer ein Boot liegt, das Platz für höchstens zwei Personen bietet. Ohne weitere Einschränkungen wäre es einfach, alle sechs Reisende über den Fluss zu bringen: Zwei fahren hinüber, einer bringt das Boot zurück, und nach neun Fahrten wären alle sechs auf der anderen Uferseite. Aber die Kannibalen machen Schwierigkeiten. Sie beharren darauf, gegenüber den Missionaren nie in der Minderheit zu sein, denn aus bitterer Erfahrung haben sie gelernt, dass sie in Unterzahl konvertiert werden.

Diese Bedingung erschwert das Problem und verführt viele Schüler nach allerlei vergeblichen Versuchen zu dem Schluss, es gäbe keine Lösung. Eine systematische Analyse mit Hilfe des Begriffs „Zustandsraum“ führt jedoch nach wenigen Minuten zur Klärung (siehe Abb. 1).

Die Matrix zeigt alle möglichen Verteilungen der sechs Protagonisten auf die beiden Ufer, vom Startzustand $(K_L, M_L) = (3, 3)$, also 3 Kannibalen und 3 Missionare am linken Ufer, bis zum Zielzustand $(K_L, M_L) = (0, 0)$, bei dem alle sechs am rechten Ufer angekommen sind. Die 16 Zustände der Personenverteilung definieren den Suchraum für die Lösung. Diejenigen sechs Zustände, in denen sie am einen oder anderen Ufer in der Minderheit wären, haben die Kannibalen als illegal erklärt.

Ein Zustandsraum ist freilich mehr als eine Menge von Zuständen. Entscheidend sind die möglichen Übergänge von einem Zustand in einen nächsten. Um diese möglichen Übergänge zu definieren, müssen wir festhalten, ob sich das Boot am linken oder am rechten Ufer befindet. Dazu führen wir Pfeile ein. Liegt das Boot am linken Ufer, sind die Übergänge, die wir ungerade Züge nennen, durch

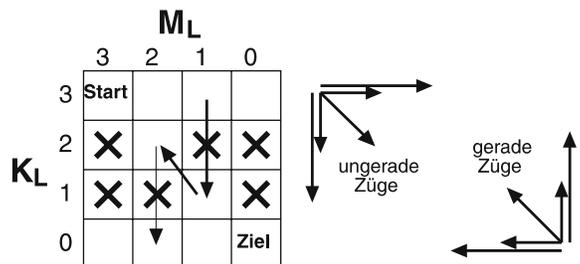


Abb. 1 Der Zustandsraum des Problems der Kannibalen und Missionare, die erlaubten Übergänge und der Kern der Lösung

die 5 Pfeile nach unten rechts dargestellt. Liegt es am rechten Ufer, sind die Übergänge durch die 5 Pfeile nach oben links dargestellt und werden gerade Züge genannt.

Aufgrund dieser Formalisierung stellt sich das Problem als Wegsuche auf einer Landkarte dar. Die Frage lautet: Wie kommen wir vom Startzustand zum Endzustand, indem wir abwechselnd Übergänge der einen und der anderen Art wählen? Dabei erweist sich der Flaschenhals in der Mitte der Matrix als der schwierigste Teil der Flussüberquerung. Er kann, wie abgebildet, nur auf eine einzige Art übersprungen werden. Hat man dies erkannt, wird es leicht, die fehlenden Anfangs- und Endstücke zu finden. Sie entsprechen der simplen Regel „Zwei rudern hinüber, einer bringt das Boot zurück“.

Typisch für ein mathematisches Vorgehen beim Lösen von Problemen ist die Verallgemeinerung. Betrachten wir das Beispiel für k Kannibalen und m Missionare. Während ein direktes Suchen nur auf konkrete Werte von k und m anwendbar ist, hilft uns der Zustandsraum, die allgemeine Frage zu untersuchen. Zeichnet man den Zustandsraum für $k = 4, m = 4$ zeigt sich sofort, dass ein Boot für maximal zwei Passagiere nicht genügt. Mit einem Dreier-Boot hingegen ist die Lösung einfach zu finden. Dabei beachte man, dass ein Dreier-Boot die Anzahl möglicher Übergänge fast verdoppelt. Mit Hilfe des geistigen Werkzeugs „Zustandsraum“ kann der Leser nun sicher die folgende Frage beantworten: Hat das verallgemeinerte Problem „ k Kannibalen und k Missionare mit einem Boot der Kapazität $\lceil (k+1)/2 \rceil$ “ für jeden Wert von k eine Lösung? (Das Symbol $\lceil x \rceil$ bedeutet die kleinste ganze Zahl $\geq x$. Auf Englisch ist es anschaulich als $\text{ceiling}(x)$ bekannt; wir könnten es $\text{Dach}(x)$ nennen.) Und wie viele Flussüberquerungen braucht es?

Der Begriff des Zustandsraums ist so elementar und nützlich, dass er spätestens im Mathematikunterricht des Gymnasiums explizit eingeführt werden sollte. Natürlich tritt dieser fundamentale Begriff in den verschiedensten Varianten auf. Ein Zustandsraum kann zum Beispiel diskret sein, wie in obigem Beispiel, oder kontinuierlich, wie oft in der Physik. Auch Übergänge können verschieden definiert sein, etwa von aussen als System-Eingänge ausgelöst, oder spontan vom System selbst ausgeführt. Sie können deterministisch, nicht-deterministisch oder probabilistisch sein. Auf solche Unterschiede kommt es im Unterricht jedoch nicht an. Wer den

Begriff „Zustandsraum“ verinnerlicht hat, findet sich in allen Varianten zurecht, so wie jemand, der das Kartenlesen gelernt hat, die verschiedensten Landkarten interpretieren kann.

Invarianten:

plus ça change, plus c'est la même chose

Wir haben die enorm grosse Anzahl möglicher zeitlicher Abläufe angesprochen, die selbst bei der algorithmischen Lösung einfacher Probleme auftreten kann. Ansätze, wie man trotz „Komplexitätsexplosionen“ den Überblick behalten kann, bilden einen wichtigen Kern der Informatik. Neben dem Begriff des Zustandsraums gehört der Begriff der Invariante zu den wichtigsten Hilfsmitteln.

Wie das Wort andeutet, ist eine Invariante eine Grösse, die mitten im Getümmel aller Veränderungen konstant bleibt. An ihr kann man sich orientieren, wie an einer Bergspitze, die über treibende Wolken hinausragt. Im Unterschied zu einem Berg, den die Natur hingestellt hat, muss eine Invariante vom Algorithmiker/Programmierer erkannt und gebaut werden. Wir illustrieren den Einsatz von Invarianten, indem wir von wohl bekannten Denkaufgaben ausgehen und zeigen, wie dieser Begriff zu Verallgemeinerungen führt.

Gegeben seien ein Schachbrett, dem zwei diagonal gegenüberliegende Eckfelder weggeschnitten wurden, sowie Dominosteine, von denen jeder so gross ist wie zwei aneinander grenzende Felder des Bretts. Ist es möglich, die übrig gebliebenen Felder des amputierten Schachbretts ohne Überlappung vollständig mit Dominosteinen zu überdecken? Die abwechselnd schwarz und weiss bemalten Felder helfen uns, eine geeignete Invariante zu finden: Jeder Dominostein deckt genau ein schwarzes und ein weisses Feld ab. Somit werden immer gleich viele schwarze wie weisse Felder abgedeckt, wie die Steine auch gelegt werden. Weil das amputierte Schachbrett aber ungleich viele schwarze und weisse Felder hat, kann es nicht vollständig mit Dominosteinen überdeckt werden. QED.

Dieser Gedankengang ist ausbaufähig. Betrachten wir nun ein verallgemeinertes „Schachbrett“, also eine $m \times n$ Matrix von Feldern, wobei von zwei angrenzenden Feldern immer eines schwarz und das andere weiss gefärbt ist. Zusätzlich sei mindestens eine der zwei Dimensionen m, n eine gerade Zahl, damit wir eine gerade Anzahl von Feldern haben.

Aus dieser Matrix schneide man nun ein beliebiges schwarzes und ein beliebiges weisses Feld aus. Die Frage lautet: Kann die amputierte Matrix mit Dominosteinen überdeckt werden?

Abbildung 2 zeigt am Beispiel $m = 8, n = 5$, wie man auf einer beliebigen Matrix (mit m gerade) einen „Kreislauf“ definieren kann. Dazu werden die $k = m \times n$ Felder derart zu einer Folge $\text{Feld}(1), \text{Feld}(2), \dots, \text{Feld}(k)$ sequenziert, dass $\text{Feld}(i)$ und $\text{Feld}(i + 1)$ ebenso aneinander grenzen wie $\text{Feld}(k)$ und $\text{Feld}(1)$. Der Abstand zwischen einem beliebigen schwarzen und einem beliebigen weissen Feld, entlang dieses Kreislaufs gemessen, ist gerade. Das ist unsere Invariante. Daher können die zwei Lücken zwischen den herausgeschnittenen Feldern entlang des Kreislaufs mit Dominosteinen überdeckt werden und somit die gesamte Matrix.

Das folgende Beispiel verknüpft Invarianten mit dem algorithmischen Paradigma „Teile und herrsche“ sowie der Rekursion. Gegeben sei eine quadratische Matrix mit Seitenlänge 2^n , aus der ein beliebiges Feld herausgeschnitten ist. Wir nennen eine derart amputierte Matrix ein „amputiertes Potenz-Quadrat“. Unsere „Dominosteine“ sollen jetzt drei Felder abdecken, die rechtwinklig aneinander grenzen (siehe Abb. 3). Wir nennen sie „Dreierkacheln“. Ist es möglich, jedes amputierte Potenz-Quadrat mit Dreierkacheln zu überdecken?

Beim Überprüfen einer Vermutung ist es oft sinnvoll zu untersuchen, ob zumindest eine schwächere Aussage zutrifft; falls nicht, ist die stärkere Aussage bereits widerlegt. Stimmt es, dass die Anzahl Felder auf einem amputierten Potenz-Quadrat durch 3 teilbar ist? Wir untersuchen diese Frage aber

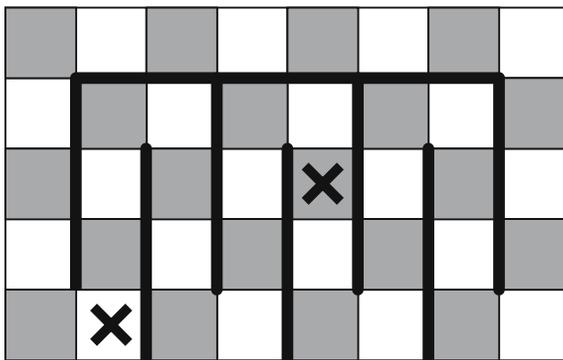


Abb. 2 Die Felder eines verallgemeinerten Schachbretts werden zu einem Kreislauf geordnet

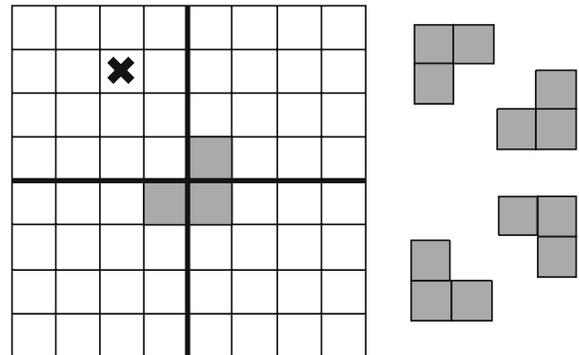


Abb. 3 Ein Potenz-Quadrat, dem das angekreuzte Feld herausgeschnitten ist, kann mit Dreierkacheln überdeckt werden

nicht eigens, denn die Antwort „Ja“ folgt aus dem Induktionsbeweis, dass eine Überdeckung möglich ist.

Als Basis der Induktion verifizieren wir, dass das kleinste 1×1 amputierte Potenz-Quadrat mit Dreierkacheln überdeckt werden kann: Wird das einzige Feld entfernt, bleibt nichts übrig, und dieser nicht existente Rest kann durch null Dreierkacheln abgedeckt werden. Als Induktionshypothese nehmen wir an, dass für einen bestimmten Wert von $n \geq 0$ die Überdeckung möglich sei. Nun betrachten wir ein beliebiges amputiertes Potenz-Quadrat mit Wert $n + 1$, also mit Seitenlänge $2^{(n+1)}$. Abbildung 3 zeigt dieses Quadrat in Quadranten aufgeteilt, wobei die Dreierkachel in der Mitte so gelegt ist, dass jeder Quadrant ein amputiertes Potenz-Quadrat mit Wert n ist. Da gemäss Induktionshypothese jeder Quadrant abgedeckt werden kann, kann auch das ganze Quadrat gekacheln werden.

Solche Überdeckungsprobleme, auf Englisch „tiling problems“ genannt, sind in der Literatur über Denkaufgaben weit verbreitet. Sicher kann der Leser das Beispiel des Potenzquadrats auf höhere Dimensionen verallgemeinern, angefangen mit dem Potenzwürfel: eine 3-dimensionale Matrix, deren Seitenlänge eine Zweierpotenz ist.

Terminierung: die verkappte Invariante

Für den Beweis der Korrektheit eines Algorithmus sind Invarianten unerlässlich. Sie stellen sicher, dass eine wichtige Aussage, die wir beim Start eines Algorithmus verifiziert haben und die uns die Korrektheit garantiert, während des ganzen Ablaufs des Algorithmus gilt. Invarianten implizieren die Aussage: „Falls der Algorithmus terminiert, ist das Ergebnis korrekt“.

Wie in den besprochenen Beispielen ist Terminierung in vielen Fällen so offensichtlich, dass wir gar nicht explizit darauf hingewiesen haben. Terminierung in diesen Beispielen ist deshalb so trivial, weil Grössen auftreten, die monoton abnehmen. Mit jedem hingelegten Dominostein verkleinert sich die Fläche, die noch abgedeckt werden muss. Dieser Prozess läuft nur so lange, bis alles abgedeckt ist. Mit jeder Aufteilung eines Quadrats in vier Quadranten werden die noch zu bearbeitenden Quadrate kleiner, und beim 1×1 Quadrat hört das Teilen auf.

Terminierungsbeweise stützen sich also auf Grössen, die sich monoton verändern, abnehmend oder wachsend, und die nach einer endlichen Anzahl von Schritten an Schranken stossen. Abstrakt ausgedrückt ist eine monoton wachsende oder abnehmende Grösse auch nur eine verkleidete Invariante, denn die Differenz zwischen zwei aufeinander folgenden Werten hat immer dasselbe Vorzeichen, positiv bei einer wachsenden Grösse, negativ bei einer abnehmenden Grösse.

Nicht bei allen Problemen drängt sich allerdings eine monotone Grösse derart offensichtlich auf wie bei den Abdeckungsaufgaben. Der wahre Wert dieses Begriffes zeigt sich erst bei Problemen, wo eine Terminierung nicht auf den ersten Blick erkennbar ist. Wir demonstrieren hier ein solches Beispiel, das auf E.W. Dijkstra zurückgeht, und nennen es das Problem der Punktpaarung in der Ebene.

In der Ebene sind n rote Punkte und n blaue Punkte gegeben (in Abb. 4 als kreisförmig und quadratisch gekennzeichnet), unter denen keine drei Punkte kollinear sind, das heisst, auf derselben Geraden liegen. Bei einer Paarung wird je ein roter und ein blauer Punkt durch eine geradlinige Strecke derart verbunden, dass alle $2n$ Punkte gepaart sind. Eine Paarung heisst kreuzungsfrei, falls es unter den n Verbindungsstrecken keine Schnittpunkte gibt. Die Frage lautet: Existiert immer eine kreuzungsfreie Paarung, unabhängig davon, wie die $2n$ Punkte liegen? Und falls ja, wie findet man solche Paarungen?

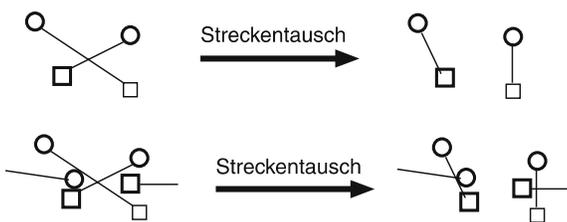


Abb. 4 Eine Kreuzung kann durch Streckentausch immer eliminiert werden, dabei können aber viele neue Kreuzungen auftreten

Es ist keineswegs offensichtlich, dass das Paarungsproblem immer eine kreuzungsfreie Lösung hat. Eine Kreuzung zwischen zwei roten und zwei blauen Punkten kann zwar durch „Streckentausch“ eliminiert werden, wie Abb. 4 zu entnehmen ist, aber die Elimination einer Kreuzung kann viele neue Kreuzungen zur Folge haben, wie dasselbe Bild zeigt.

Dijkstra diskutiert folgenden „gierigen“ Algorithmus: Solange eine Paarung Kreuzungen aufweist, eliminiere man eine beliebige Kreuzung durch einen Streckentausch.

Dieser Algorithmus terminiert nur, wenn eine kreuzungsfreie Paarung entsteht, und falls er terminiert, liefert er ein korrektes Ergebnis. Wird er aber immer terminieren? Da die Anzahl Kreuzungen, wie erwähnt, auch zunehmen kann, taugt sie nicht als monoton veränderliche Grösse. Suchen wir eine andere: Könnte es zutreffen, dass während des Ablaufs dieses Algorithmus eine eliminierte Kreuzung nie wieder auftreten kann? Diese Vermutung ist plausibel, aber leider falsch, wie uns die Sequenz in Abb. 5 lehrt.

Abbildung 5.1 zeigt 4 Strecken, die sich alle schneiden. Eine Kreuzung wird durch das betroffene Streckenpaar identifiziert, es spielt also keine Rolle, dass sich in Abb. 5.1 alle Strecken im selben Schnittpunkt treffen. Mit dicken Strichen wird jeweils das Streckenpaar hervorgehoben, dessen Kreuzung eliminiert wird – dadurch entsteht aus 5.1 die Konfiguration 5.2. Man prüfe, dass die gewählten Streckentausch-Operationen in Abb. 5.6 dieselbe

Abbildung 5.1 zeigt 4 Strecken, die sich alle schneiden. Eine Kreuzung wird durch das betroffene Streckenpaar identifiziert, es spielt also keine Rolle, dass sich in Abb. 5.1 alle Strecken im selben Schnittpunkt treffen. Mit dicken Strichen wird jeweils das Streckenpaar hervorgehoben, dessen Kreuzung eliminiert wird – dadurch entsteht aus 5.1 die Konfiguration 5.2. Man prüfe, dass die gewählten Streckentausch-Operationen in Abb. 5.6 dieselbe

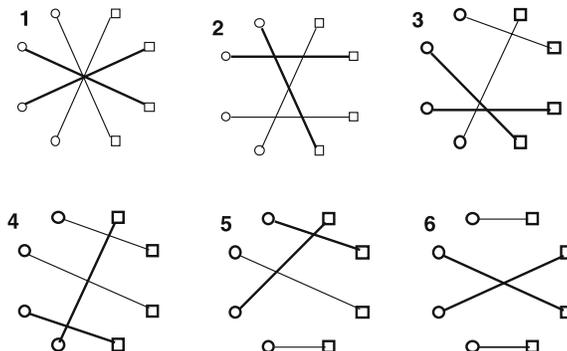


Abb. 5 Die Elimination von Kreuzungen kann zum paradoxen Ablauf führen, dass eine eliminierte Kreuzung wieder eingeführt wird

Kreuzung hervorbringen, die wir in 5.1 eliminiert haben.

Beim Problem der Punktepaarung in der Ebene lässt sich eine monoton veränderliche Grösse wie folgt finden. Die Elimination einer Kreuzung durch Streckentausch verkürzt die Länge der beteiligten Strecken. Dies sieht man „von Auge“ anhand von Beispielen und beweist es mit Hilfe der Dreiecksungleichung. Die Summe der Längen aller Paarungsstrecken nimmt daher während des Ablaufs von Dijkstras Algorithmus monoton ab und hat eine untere Schranke, zum Beispiel Null.

Eine Folge monoton abnehmender positiver Zahlen kann unendlich lang sein, also genügen die zwei Bedingungen „monoton abnehmend“ und „nach unten beschränkt“ noch nicht für einen Terminierungsbeweis. Es braucht zusätzlich die Feststellung, dass die Abnahme pro Schritt nicht beliebig klein werden darf. Dies trifft bei einem Streckentausch offensichtlich zu, denn die Anzahl der möglichen Streckenlängen ist endlich. Also muss der gierige Algorithmus terminieren und liefert damit eine kreuzungsfreie Paarung.

Ausblick

Allgemeine Aussagen, die streng beweisbar sind, erfordern ein Verständnis für Abstraktion, das uns nicht in die Wiege gelegt worden ist. Die dazu notwendige formale Logik muss man sich mühsam erarbeiten. Anschauliche Beispiele hingegen, die eine allgemeine Aussage illustrieren, ermöglichen es dem Lernenden, dank seiner Intuition einen leichteren und oft eindrucklicheren Zugang zur Lösung von Problemen zu finden. Damit sagen wir nicht, formale Beweise hätten im Unterricht keinen Platz. Wir vertreten allerdings die Ansicht,

dass auf der allgemein bildenden Stufe gut gewählte Beispiele beim Schüler oft einen tieferen Eindruck hinterlassen als formale Beweise.

Wie und wo findet man kurze, prägnante, lehrreiche Beispiele? In Bereichen der Informatik, die stark mathematisch/algorithmisch geprägt sind, ist dies leicht. Eine unerschöpfliche Quelle von eleganten Gedankengängen sind Lehrbücher, vor allem aber auch die Literatur der mathematischen Unterhaltungen und Denksportaufgaben. Viele solcher Beispiele gehören zum Allgemeingut. Meistens sind deren Erfinder unbekannt. Dies gilt auch für die im Artikel erwähnten Beispiele, mit Ausnahme der Aufgabe der Punktepaare in der Ebene, die aus [1] stammt.

Anregungen zu Beispielen aus einem breiten Spektrum der Informatik findet man in [2]. Hier werden in 61 kurzen Kapiteln Standardthemen der Informatik beispielhaft illustriert. Beispiele sollen nicht nur gefunden, sondern auch erfunden werden. Uns ist allerdings keine Literatur bekannt, die das Erfinden von Aufgaben behandelt. Es gibt hingegen Abhandlungen zur Technik des Lösens von Denkaufgaben. Ein berühmtes Buch zu diesem Thema, das den Problemlöser durch Fragestellungen wie „Kennst Du eine verwandte Aufgabe?“ auf den richtigen Weg führt, ist [3].

Dieselben Leitgedanken, die Lösungswege aufzeigen, können auch dazu dienen, Varianten oder neue Aufgaben zu erfinden.

Literatur

1. Dijkstra E.W.: Reasoning about programs. Distinguished Lecture Series 1990, Univ. Video Comm.
2. Dewdney A.K.: The Turing Omnibus, 61 Excursions in Computer Science. Computer Science Press 1989
3. Polya G.: Schule des Denkens – Vom Lösen mathematischer Probleme. Bern: A. Francke Verlag 1949