1 Komplexität der Geographie

Volker Diekert und Ulrich Hertrampf

Institut für Formale Methoden der Informatik, Universität Stuttgart
Universitätsstr. 38, D-70569 Stuttgart

Zusammenfassung. Das allgemein als Prototyp eines PSPACE-vollständigen Spiels gesehene Geographiespiel wird bezüglich seiner Komplexität genauer untersucht. Das Interesse der theoretischen Informatik an diesem Spiel wurde sehr durch die Darstellung in dem Lehrbuch von Papadimitriou [Pap94] gefördert. Allerdings bestimmt dieses Lehrbuch nicht die Komplexität des Standardspiels sondern verwendet eine Verallgemeinerung. Die Aussage in dem Lehrbuch bleibt damit etwas unbefriedigend und hinter den Möglichkeiten. Wir zeigen hier, dass die komplexitätstheoretische Charakterisierung schon für die Standardvariante des Spiels gilt.

In der konkreten Version, die man tatsächlich als Gesellschaftsspiel verwenden kann, gibt es unserem Alphabet entsprechend nur 26 Buchstaben. Es ergibt sich, dass man bei konstanter Buchstabenzahl optimale Spielstrategien in polynomieller Zeit berechnen kann.

In der Praxis muss man allerdings wiederum anders vorgehen, um verwertbare Ergebnisse zu erhalten. Wir beschreiben das anhand eines Spiels mit den 305 Millionenstädten dieser Welt.

1.1 Das Spiel Geographie

Geographie ist ein Gesellschaftsspiel, welches man gut mit einer Gruppe von Schülern im Unterricht spielen kann, um z. B. deutsche (oder andere) Städtenamen zu behandeln. Man spielt mit einer kleinen Gruppe von beispielsweise 10 Leuten. In einer gewissen zyklischen Reihenfolge müssen die Teilnehmer nacheinander Städte nennen. Der Rest der Klasse fungiert zusammen mit dem Lehrer als Schiedsrichter, ob der Name wirklich eine Stadt bezeichnet. So sind alle involviert. Eine Regel ist, dass jede Stadt nur einmal genannt werden darf. Außerdem muss der erste Buchstabe stets der letzte Buchstabe der Vorgängerstadt sein. Beginnt also der erste Spieler mit Oldenburg, so darf mit Gelsenkirchen fortgesetzt werden. Dann findet man vielleicht Nürnberg und daraufhin Göttingen. Gelsenkirchen war nicht mehr möglich, da es schon genannt wurde. Auf Göttingen darf man auch nicht mehr Nürnberg nennen, aber Neumünster würde passen. Jetzt kann man nach Regensburg wechseln und so fort. Zu Anfang ist es sehr einfach passende Städtenamen zu finden, aber erfahrungsgemäß werden Schüler rasch Probleme bekommen, einen freien Namen zu finden. Man legt dann eine Zeit fest, etwa 30 Sekunden, und wer es nicht schafft innerhalb dieser Zeit fortzufahren, scheidet aus. Die letzte Person, die übrig bleibt, gewinnt das Spiel.

Im Folgenden betrachten wir dieses Spiel stets nur für zwei Spieler, die abwechselnd jeweils einen Namen aus einem vorher festgelegten Vorrat von Städtenamen nennen. Ob wir ein interessantes Spiel erhalten hängt entscheidend vom vereinbarten Vorrat ab. Alle deutschen Städte sind vielleicht zu viel und mit etwas Training kann man lange Ketten bilden. Was passiert, wenn die Liste zu kurz ist? Als Beispiel nehmen wir die folgenden 18 Städte:

Berlin, Hoffenheim, Hamburg, München, Leverkusen, Wolfsburg, Gelsenkirchen, Stuttgart, Dortmund, Bremen, Köln, Hannover, Frankfurt, Bielefeld, Bochum, Cottbus, Karlsruhe, Mönchengladbach.

Sollte der Start auf Berlin, München, Leverkusen, Stuttgart, Dortmund, Gelsenkirchen, Bremen, Köln, Hannover, Frankfurt oder Karlsruhe festgelegt sein, hätte der Spieler, der nun die nächste Stadt nennen soll, direkt verloren, da er keine Stadt mit dem Anfangsbuchstaben *N, T, D, R* oder *E* finden kann. (Im Fall von Dortmund wäre die einzige passende Stadt wieder Dortmund selbst, die aber schon als Anfangsstadt verbraucht wäre!)

In den anderen Fällen (es bleiben ja nur noch Hoffenheim, Hamburg, Wolfsburg, Bielefeld, Bochum, Cottbus und Mönchengladbach) kann der erste Spieler sofort gewinnen, indem er München, Gelsenkirchen, Dortmund, Stuttgart oder Hannover wählt.

Bei realistischeren Spielen muss man also mehr Städtenamen im Vorrat haben, um zu echten Herausforderungen zu gelangen. Gelegentlich hört man, dass *Geography* in amerikanischen Schulen gespielt wird, um die Hauptstädte der 51 amerikanischen Bundesstaaten zu lernen. Dies ist wenig glaubhaft, denn die Liste der Städte sieht wie folgt aus:

Montgomery, St. Paul, Montpelier, Juneau, Jackson, Richmond, Phoenix, Jefferson City, Olympia, Little Rock, Helena, Charleston, Sacramento, Lincoln, Madison, Denver, Carson City, Cheyenne, Hartford, Concord, Dover, Trenton, Washington, Tallahassee, Santa Fe, Atlanta, Albany, Honolulu, Raleigh, Boise, Bismarck, Springfield, Columbus, Indianapolis, Oklahoma City, Des Moines, Salem, Topeka, Harrisburg, Frankfort, Providence, Baton Rouge, Columbia, Augusta, Pierre, Annapolis, Nashville, Boston, Austin, Lansing, Salt Lake City.

Man kann sich hier leicht alle relevanten Übergänge zeichnen, zum Beispiel ist dann leicht zu sehen, dass man von Denver zu Richmond oder Raleigh gehen kann.

Tatsächlich bekommt man hier überhaupt kein interessantes Spiel. Man sieht sofort, dass es eine Reihe von Städten gibt, von denen man nicht weiter kann. (Albany, Cheyenne, Carson City, usw., insgesamt 21 Städte.)

Das bedeutet wiederum, dass man mit allen Städten, die auf einen der Buchstaben *A, B, C, H, J, L, M, N, O, P, S, T* enden, sofort verliert, die folgenden Städte darf man in einer Gewinnstrategie also auf keinen Fall verwenden: Olympia, Helena, Atlanta, Augusta usw., insgesamt 23 Städte.

Es bleiben uns nur noch 7 Städte, mit denen man weder sofort gewinnt, noch sofort verliert, nämlich Richmond, Concord, Hartford, Springfield, Denver, Dover und Montpelier. Man kann leicht ermitteln, dass man durch die Ansage der auf den Buchstaben *R* endenden Städte (Montpelier, Denver, Dover) gewinnt, während man durch die Ansage der auf *D* endenden Städte verliert.

Auch für amerikanische Schüler wird der Reiz an diesem Graphen schnell verloren gehen.

Interessanter ist die Situation, wenn man europäische Hauptstädte verwendet. In Bild 1.1 sind die Hauptstädte in ihrer ungefähren geographischen Lage zueinander abgebildet. Die Abbildung mag dem Leser als Trainings-und Spielvorlage dienen.

Dagegen sind die 305 Millionenstädte, die man etwa bei Wikipedia.de im Februar 2009 gefunden hat, wirklich spannend. Die Liste beginnt in der Reihenfolge der geschätzten Einwohnerzahlen mit Mumbai, Karatschi, Delhi, Moskau, Seoul, Istanbul, Sao Paulo, Shanghai, Lagos, Mexiko-Stadt,...

Von Streitfragen zur Rechtschreibung abgesehen, lassen sich jetzt relativ lange Ketten bilden und man ist überrascht, welche Städte genannt werden können. Es finden sich Städte zu jedem Anfangsbuchstaben, insbesondere Städtenamen, die mit *X* beginnen, wie Xiang oder mit *Y* wie Yokohama.

Es sind so viele Städte, dass der vollständige Graph sehr unübersichtlich ist. Wir können aber eine für das Spiel gleichwertige Tabelle angeben, in der wir eintragen, wieviele Millionenstädte es für ein gegebenes Paar von Anfangs- und Endbuchstaben gibt. Der Eintrag 2 der Zeile *B* in den Spalten *D* und *N* bedeutet also, dass es laut Wikipedia zwei Millionenstädte gibt, die mit *B* beginnen und mit *D* enden sowie zwei Millionenstädte, die mit *B* beginnen und mit *N* enden. Eine dieser vier Städte sollte dem Leser sofort einfallen. Alle vier zu kennen, ist weniger einfach. Die Spielmatrix der Anfangssituation ist in Abb. 1.2 dargestellt.

Man kann jetzt schon erahnen, dass es nicht unbedingt einfach ist, für alle möglichen Startsituationen herauszufinden, welcher der beiden Spieler gewinnen kann. Klar sollte jedoch sein, dass immer genau einer der beiden Spieler eine Gewinnstrategie hat und dass man prinzipiell ausrechnen kann, wer es ist. Dies ist ein

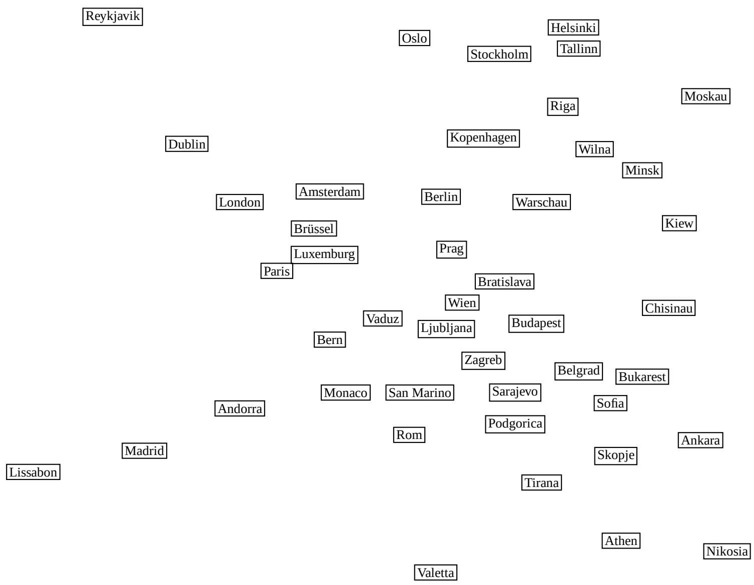


Abbildung 1.1: Hauptstädte Europas

typischer Induktionsbeweis. Ist die Liste der Städte leer, so verliert der anziehende Spieler. Sind jetzt noch n Städte in der Liste und ist $n > 0$, so kann man rekursiv für jeden möglichen Zug ausrechnen, ob er auf eine persönliche Gewinnstellung führt. Genau dann, wenn ein solcher Zug existiert, hat der anziehende Spieler eine Gewinnstrategie.

1.2 Komplexität von Geographie

Der populäre Beweis zur Komplexität des Graphen-Geographie-Spiels geht von einer verallgemeinerten Version des Spiels aus, wo ein beliebiger gerichteter Graph zugrundeliegt. Ein Anfangsknoten ist spezifiziert und die Spieler müssen abwechselnd vom aktuellen Knoten eine ausgehende Kante zu einem bisher nicht besuchten Knoten wählen. Wer diesen Zug nicht ausführen kann, da alle Kanten zu schon besuchten Knoten führen, hat verloren. Wir nennen dieses Spiel jetzt Graphen-Geographie. Diese Verallgemeinerung des ursprünglichen Geographie-Spiels wurde von Richard Karp vorgeschlagen und in dieser Form von Schaefer [Sch78] als

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	6	0	0	3	1	0	0	0	1	0	0	0	0	3	2	0	0	2	0	0	0	0	0	0	1	0
B	5	0	0	2	3	0	1	0	1	0	1	1	2	2	1	0	0	0	1	4	3	0	0	0	0	0
C	4	0	0	0	0	0	2	0	2	0	0	0	0	2	1	0	0	0	2	0	1	0	1	0	1	1
D	3	0	0	0	0	0	0	0	2	0	2	0	1	3	0	0	0	1	2	0	1	0	0	0	1	0
E	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0
F	2	0	0	2	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0
G	3	0	0	1	0	0	2	1	0	0	0	1	0	0	0	1	0	0	1	0	2	0	0	0	0	0
H	3	0	0	2	1	0	2	0	2	0	0	0	0	4	0	0	0	0	0	1	1	0	0	0	0	0
I	0	0	0	0	1	0	0	0	0	0	0	1	0	3	0	0	0	1	0	0	0	0	0	0	0	0
J	1	0	0	0	0	0	2	0	0	0	0	0	0	2	0	0	0	2	0	0	0	0	0	0	0	0
K	4	0	0	0	1	0	2	1	4	0	0	1	1	1	3	0	0	2	0	1	0	0	1	0	0	0
L	5	0	0	0	1	0	1	0	1	0	0	0	0	2	0	0	0	0	2	0	2	0	0	0	0	0
M	3	0	0	3	1	0	0	1	2	0	1	2	0	4	3	0	0	1	1	1	2	0	0	0	2	0
N	1	0	0	1	0	0	2	0	1	0	3	2	0	0	0	0	0	1	0	1	0	0	0	0	0	0
O	2	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	1	0	0	1	0	0	0	0	0
P	4	0	0	1	3	0	4	2	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0
Q	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	1	0
R	0	0	0	1	1	0	1	0	1	0	0	0	1	2	0	0	0	0	0	2	0	0	0	0	0	0
S	7	0	0	0	1	0	4	0	2	0	0	1	1	2	6	0	0	1	0	1	2	0	0	0	1	1
T	1	0	0	0	1	0	2	1	0	0	1	0	0	4	3	0	0	0	3	1	0	0	0	0	0	0
U	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
V	2	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0
W	0	0	0	1	0	0	0	0	1	0	0	0	0	2	0	0	0	0	0	0	1	0	0	0	0	0
X	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0
Y	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Z	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

Abbildung 1.2: Spielmatrix für das Millionenspiel

PSPACE-vollständig nachgewiesen, siehe auch [Pap94].

Ein Nachweis, dass Graphen-Geographie komplexitätstheoretisch schwierig ist, erlaubt a priori keinen Rückschluss auf die Schwierigkeit des Geographie-Spiels. Allerdings kann jeder Algorithmus für Graphen-Geographie auch für das eigentliche Geographie-Spiel benutzt werden. Wir zitieren zunächst das bekannte Resultat.

Satz 1
Graphen-Geographie ist PSPACE-vollständig.

Der Satz bedeutet, dass man Graphen-Geographie mit polynomiellem Speicherplatz lösen kann und dass dieses Spiel in der Klasse der in ploynomiellem Platz lösbaren Probleme, PSPACE, schwierig ist. Nehmen wir an, n ist ungerade und ich bin ein Spieler, der wissen möchte, ob es für mich einen Gewinnknoten gibt. Dies gilt genau dann, wenn es (für mich) einen Knoten v_1 gibt und für alle möglichen Folgeknoten v_2 (die möglichen Züge des Gegners) einen für mich möglichen Knoten v_3 gibt und für alle möglichen Folgeknoten v_4 einen für mich möglichen Knoten v_5 gibt u.s.w., bis am Ende ein Knoten v_n existiert und sich dann eine einfache logische Formel für die Knoten v_1 bis v_n zu *wahr* auswertet. Standardmetho-

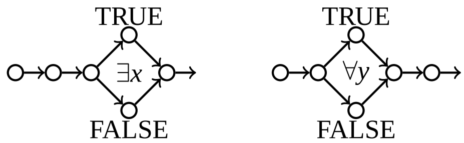


Abbildung 1.3: Links ein Existenzquantor, rechts ein Allquantor

den der Informatik zeigen, wie man diesen Ansatz in eine Tiefensuche übersetzt. Schwerer ist die *Schwierigkeit* dieses Problems zu zeigen. Der Beweis besteht darin, das bekannterweise PSPACE-vollständige Problem der wahren quantifizierten Booleschen Formeln (manchmal QSAT oder auch QBF genannt) auf Graphen-Geographie zu reduzieren. Eine quantifizierte Boolesche Formel sieht wie folgt aus:

$$\Phi = \exists x_1 \forall x_2 \cdots \exists x_{k-1} \forall x_k \; \phi(x_1, \ldots, x_k)$$

Hierbei ist $\phi(x_1, \ldots, x_k)$ eine aussagenlogische Formel in den Booleschen Variablen $x_1, \ldots, x_k$ und da alle Variablen gebunden sind, wertet sie sich zu einem Wahrheitswert aus. Diese Formel wird in ein Graphen-Geographie übersetzt, das genau dann zu gewinnen ist, wenn Φ *wahr* ist. Wir verwenden bei der Übersetzung die in Abb. 1.3 dargestellten beiden Teilgraphen zur Umsetzung eines Existenzquantors (links) bzw. eines Allquantors (rechts).

Diese Bausteine werden so zusammengesetzt, dass immer am linken Knoten Spieler 1 an der Reihe ist. Bei einem Existenzquantor führt das dazu, dass Spieler 1 die entsprechende Variable mit einem Wert belegen kann (er wählt den oberen Knoten für TRUE und den unteren für FALSE), entsprechend bei einem Allquantor Spieler 2.

An die Quantorenkette schließt sich dann ein Aufzweigungsbaustein an, der so gebaut ist wie in Abb. 1.4 gezeigt ist (auch hier ist Spieler 1 am Anfang – links – an der Reihe).

Die Aufzweigung hat genau so viele Zweige wie es Klauseln gibt. Diese führen zu den sogenannten Klauselknoten. Das bedeutet, Spieler 2 kann die Klausel wählen (er "behauptet" ja, es gäbe eine nicht erfüllte Klausel), anschließend muss dann Spieler 1 beweisen, dass diese Klausel doch erfüllt ist, d.h. er muss ein Literal der Klausel finden, das den Wahrheitswert TRUE hat. Um ihm hier noch genau diesen einen Zug zu gestatten, werden die Klauselknoten von Klauseln, in denen x_i vorkommt, mit dem FALSE Knoten des entsprechenden Auswahlbausteins verbunden (wenn diese Variable bei der Auswahl den Wert TRUE bekam, ist der FALSE Knoten als einziger Knoten dieses Bausteins noch nicht besucht

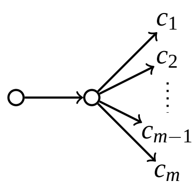


Abbildung 1.4: Teilgraph zur Auswahl einer von *m* Klauseln

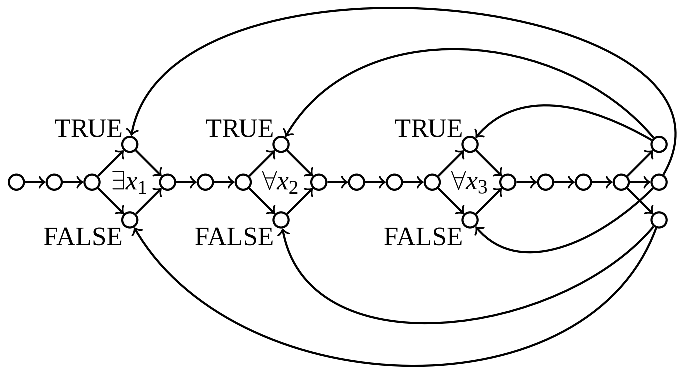


Abbildung 1.5: $\exists x_1 \forall x_2 \forall x_3 [(\neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_3) \wedge (x_1 \vee x_2)]$

worden), und analog Klauseln in denen $\neg x_i$ vorkommt, mit dem TRUE Knoten dieses Auswahlbausteins. Abb. 1.5 illustriert den Beweis an einem Beispiel.

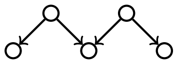
Dieses Prinzip kann man für jede beliebige QBF-Instanz verwenden, um einen Graph zu konstruieren, der als Graphen-Geographiespiel genau dann eine Gewinnstrategie für Spieler 1 beinhaltet, wenn die QBF-Instanz eine wahre Formel ist. Damit ist der Beweis der PSPACE-Vollständigkeit für Graphen-Geographie erbracht.

Das Graphen-Geographiespiel ist allerdings allgemeiner als das uns eigentlich interessierende Geographiespiel in seiner Originalform. Das liegt daran, dass im Originalspiel Kanten von A nach B und C automatisch erzwingen, dass jeder Knoten, der eine Kante zu B hat, auch eine solche zu C aufweist (denn es bedeutet ja, dass die zu B und C gehörenden Städte denselben Anfangsbuchstaben haben). Wie man im oben angegebenen Beispiel sofort sieht, ist diese Bedingung in den hier konstruierten Graphen immer verletzt – und das nicht zufällig, denn die unterschiedlichen Rollen der TRUE und FALSE Varianten in den Auswahlteilgraphen ist ja ein entscheidendes Detail des angegebenen Beweises.

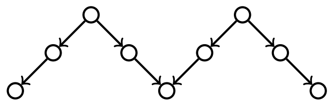
Dennoch können wir die PSPACE-Vollständigkeit auch für das Geographiespiel nachweisen, indem wir Graphen-Geographie auf Geographie reduzieren.

Satz 2
Geographie ist PSPACE-vollständig.

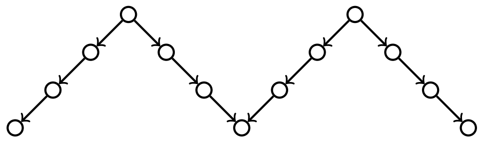
Die Idee des Beweises wurde in einem Seminar *Spiele in der Informatik* an der Universität Stuttgart von dem Teilnehmer Ivan Bogicevic erarbeitet: Da das Problem offenbar darin besteht, dass es Knoten gibt, deren Nachfolgermengen sich schneiden, aber nicht gleich sind, müssen wir diese Möglichkeit beseitigen, ohne das Spiel sonst entscheidend zu verändern. Wir geben ein Beispiel, das das typische Problem zeigt:



Dies kann keine Situation in einem Geographie-Spiel sein, denn die drei unteren Knoten würden Städten mit dem selben Anfangsbuchstaben entsprechen und von den beiden oberen Knoten müssten Kanten zu allen unteren gezogen werden. Wir würden das Problem gerne lösen, indem wir jede der Kanten durch zwei Kanten ersetzen – mit einem Knoten, der nur für diesen Zweck zu dem Graphen hinzugenommen wird, wie im folgenden Bild veranschaulicht:



Nun haben die Originalknoten niemals sich schneidende Nachfolgermengen, da die Nachfolger immer eigens zu den jeweiligen Kanten gebildete Knoten sind. Die neuen Knoten haben jeweils nur einen Nachfolgeknoten, so dass bei nichtleerem Schnitt die Gleichheit folgt. Sich schneidende Nachfolgermengen zwischen Originalknoten und neuen Knoten kann es gar nicht geben. Somit ist das Problem eigentlich gelöst, aber nach den Spielregeln ist das Spiel nun entscheidend verändert: Der zweite Spieler kommt generell nur bei den neuen Knoten an die Reihe und hat daher nie eine Auswahl; der erste Spieler bestimmt jeden echten Zug. Wir müssen also noch erreichen, dass die Entscheidungen wieder so verteilt werden, wie sie ursprünglich waren. Das erreichen wir, indem wir tatsächlich nicht nur einen, sondern zwei neue Knoten auf jede Kante platzieren. Damit erhalten wir im obigen Beispiel das folgende Bild:



Jetzt haben wir die gewünschte Eigenschaft und außerdem dieselbe Spiellogik wie zuvor, da jede Kante nunmehr drei Kanten entspricht und somit an den Originalknoten dieselbe Alternierung der Spielmöglichkeiten entsteht wie im Originalgraphen. Damit ist die PSPACE-Schwierigkeit auch von dem ursprünglichen Geographie-Spiel bewiesen. Die Reduktion ist so einfach, dass Graphen-Geographie und Geographie im Wesentlichen als gleich schwierig angesehen werden können.

1.3 Geographie mit festem Alphabet

Nun wollen wir den Grad der Abstraktion, der im vorigen Abschnitt zu dieser recht hohen Komplexitätseinstufung geführt hat, wieder ein wenig einschränken und uns dabei einem durchführbaren Spiel weiter annähern: In der Realität spielt man das Geographiespiel mit einem festen zugrundeliegenden Alphabet, über dem die Städtenamen gebildet sind. Auch in diesem Fall kann man (wie schon im Beispiel der Millionenstädte geschehen) die Eingabe als Matrix mit natürlichen Zahlen als Einträgen darstellen.

Wie bei den Millionenstädten bilden wir eine $k \times k$ Matrix und in die Position x, y tragen wir einfach die Zahl der Städte ein, die mit dem Buchstaben x beginnen und auf y enden. Umgekehrt entspricht jede solche Matrix einem (abstrakten) Geographie-Spiel. Nun kann man das Spiel so spielen, dass der jeweilige Spieler sich in einer Zeile der Matrix befindet und eine Zahl der Zeile um eins verringern muss. Die Nummer der Spalte, in der er das tut, besagt wiederum, in welcher Zeile der Gegenspieler weiterspielen muss.

Hat man also k Buchstaben zur Verfügung und gibt es höchstens n Städte über diesem Vorrat an Buchstaben, so zeigt eine einfache kombinatorische Überlegung, dass man mit diesen Parametern genau $\binom{n+k^2}{k^2}$ verschiedene Matrizen erzeugen kann.

Wir müssen uns noch merken, in welcher Zeile gerade gespielt werden soll, aber wenn k eine Konstante ist, gibt es bis auf einen konstanten Faktor nur n^{k^2}

viele Spielsituationen. Für realistische Spiele wird es viel weniger Möglichkeiten geben, die man wirklich betrachten muss. Aber um unser Polynomialzeitresultat zu erhalten, reicht diese grobe Abschätzung bereits aus. Da jede Spielsituation leicht bewertet werden kann, wenn alle Spielsituationen mit weniger Städten schon bewertet sind (bewertet heißt hier: es ist bekannt, ob der Spieler eine Gewinnstrategie hat), kann ein Algorithmus nach dem Prinzip des dynamischen Programmierens offenbar die Anfangssituation bewerten.

Der Algorithmus verläuft wie folgt: Eingabe sei eine Anfangssituation mit n Städten und eine Zahl aus dem Bereich $1,\dots,k$, die festlegt, in welcher Zeile der anfangende Spieler beginnen muss. Wir legen eine Tafel mit allen Spielsituationen an, die bei maximal n Städten und k Buchstaben möglich sind. Jetzt gehen wir in einer Schleife durch die Städteanzahlen (von 0 bis n) und untersuchen alle Situationen mit der gegebenen Städteanzahl wie folgt: Gibt es in der Spielzeile (d.h. Zeile in der der Spieler am Zug spielen muss) einen Zug, der zu einer Verliersituation führt (das kann immer in der Tafel nachgeschaut werden, da die Städteanzahl dann um eins geringer ist), so ist die untersuchte Spielsituation eine Gewinnersituation und wird mit einer 1 gekennzeichnet. Andernfalls ist sie eine Verlierersituation und wird mit einer 0 gekennzeichnet.

Dieser Algorithmus braucht zur Untersuchung jeder Spielsituation maximal k Einträge in der Tafel der schon errechneten Ergebnisse anzusehen, das geht je nach Datenstruktur in konstanter oder maximal logarithmischer Zeit. Da wir nur polynomiell viele Einträge bilden müssen, haben wir insgesamt einen Polynomialzeitalgorithmus. Das Endergebnis finden wir, indem wir einfach die Anfangssituation in unserer Tafel prüfen.

Wir haben damit bewiesen:

Satz 3
Geographie mit festem Alphabet ist in polynomieller Zeit lösbar.

Bei sehr wenigen Buchstaben wird das Spiel sehr einfach. Im Extremfall von nur einem Buchstaben ist das trivial. Denn dann haben alle Städte denselben Anfangs- und Endbuchstaben, etwa Atlanta, Augusta, Athena, Apolda, Alma Ata. In solchen Fällen ist klar, dass es nur darauf ankommt, ob die Städtezahl gerade (Verlierersituation) oder ungerade (Gewinnersituation) ist.

Wie sieht es bei zwei Buchstaben aus? Die Eingabe besteht nun aus einer 2 mal 2 Matrix, etwa $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$ Nehmen wir o.E. an, dass der erste Spieler in Zeile 1 beginnt. Wir behaupten, dass dieser eine Gewinnstrategie genau dann hat, wenn

i) a ungerade ist, oder

ii) sowohl $b > c$, als auch d gerade.

Zum Beweis machen wir eine Induktion über die Größe von $a+b+c+d$, beginnend mit dem Induktionsanfang bei $a+b+c+d=0$: Hier hat sicher der erste Spieler verloren, wie von dem angegebenen Kriterium vorausgesagt.

Nun sei also $a+b+c+d>0$. Wir untersuchen fünf Fälle:

1) a ungerade, $b>c$, d gerade.

2) a ungerade und $b\leq c$ oder a und d ungerade.

3) a gerade, d gerade, $b>c$.

4) a gerade, d gerade, $b\leq c$.

5) a gerade, d ungerade.

Wir müssen zeigen, dass in den ersten drei Fällen Spieler 1 und in den letzten beiden Fällen Spieler 2 gewinnt.

Im Fall 1) nimmt Spieler 1 den Übergang zu Zeile 2. Die Rollen der Buchstaben werden vertauscht und Spieler 2 hat nun die Matrix $\left(\begin{smallmatrix} d & c \\ b-1 & a \end{smallmatrix}\right)$ vor sich, die nach Induktionsvoraussetzung keine Gewinnersituation ist.

Im Fall 2) spielt Spieler 1 die erste Spalte, d.h. Spieler 2 muss nun die Matrix $\left(\begin{smallmatrix} a-1 & b \\ c & d \end{smallmatrix}\right)$ spielen. Da $a-1$ gerade ist und die Bedingung "$b>c$ und d gerade" nicht gilt, ist das eine Verlierersituation für Spieler 2.

Im Fall 3) wechselt Spieler 1 zu Zeile 2, d.h. Spieler 2 ist mit der Situation $\left(\begin{smallmatrix} d & c \\ b-1 & a \end{smallmatrix}\right)$ an der Reihe. Da d und a gerade sind und $c\leq b-1$ gilt, ist das eine Verlierersituation. Spieler 1 gewinnt also.

Im Fall 4) hat Spieler 1 zwei Möglichkeiten: Er kann a verringern, dann ist aber Spieler 2 mit der Situation $\left(\begin{smallmatrix} a-1 & b \\ c & d \end{smallmatrix}\right)$ und ungeradem $a-1$ offenbar in einer Gewinnsituation. Die zweite Möglichkeit für Spieler 1 ist ein Wechsel in Zeile 2, dann kommt Spieler 2 mit der Matrix $\left(\begin{smallmatrix} d & c \\ b-1 & a \end{smallmatrix}\right)$ an die Reihe. Weil $c>b-1$ gilt, ist das aber wieder eine Gewinnsituation für Spieler 2. In jedem Fall verliert Spieler 1.

Im Fall 5) kann Spieler 1 entweder in Zeile 1 bleiben und wieder kommt Spieler 2 mit der Situation $\left(\begin{smallmatrix} a-1 & b \\ c & d \end{smallmatrix}\right)$ und ungeradem $a-1$ an die Reihe und gewinnt. Oder Spieler 1 wechselt die Zeile und Spieler 2 kommt mit $\left(\begin{smallmatrix} d & c \\ b-1 & a \end{smallmatrix}\right)$ und ungeradem d wieder zum Sieg. In jedem Fall verliert Spieler 1.

Damit haben wir gezeigt, dass das angegebene leicht zu prüfende Kriterium tatsächlich die Frage beantwortet, welcher der beiden Spieler beim Geographiespiel mit 2 Buchstaben eine Gewinnstrategie hat. Ein ähnlicher, nur mit wesentlich mehr Fallunterscheidungen arbeitender Beweis zeigt, dass es ein vergleichbares Kriterium auch für den Fall von drei Buchstaben gibt. Es gilt also:

Satz 4
Geographie mit maximal drei Anfangs- bzw. Endbuchstaben ist in logarithmischem Platz lösbar.

1.4 P-vollständige Varianten

Es ist nicht unwahrscheinlich, dass man ein ähnliches Kriterium wie das für zwei Buchstaben gezeigte bzw. das für drei Buchstaben existierende auch für vier, fünf oder mehr Buchstaben erhalten kann, solange diese Zahl klein ist. Es ist jedoch unbekannt, ob es eine Buchstabenzahl gibt, ab der das Problem P-vollständig ist. Möglicherweise reicht eine Art *kritische Masse* an Buchstaben, um ein P-schwieriges Problem wie das Auswertungsproblem für Schaltkreise auf das Geographiespiel zu reduzieren.

Es gibt in der Tat natürliche Varianten, in denen P-Vollständigkeit gezeigt werden kann. Betrachten wir ein Geographie-Spiel, welches einen azyklischen gerichteten Graphen definiert. Dies bedeutet, dass die Regel, einmal genannte Städte zu verbieten, keinen Einfluss mehr auf den Spielverlauf hat, da es sowieso keine gerichteten Kreise gibt. Jetzt kann man von den *Senken* her die jeweiligen Verlust- und Gewinnknoten berechnen. Dies liefert einen Polynomialzeitalgorithmus, um das Spiel komplett zu analysieren. Es ist auch einfach, das Auswertungsproblem für Schaltkreise durch ein azyklisches Geographie-Spiel auszudrücken. Wir erhalten damit:

Satz 5
Azyklische Geographie ist P-vollständig.

1.5 Millionenspiel

Die bisherigen Betrachtungen suggerieren, dass man zur Lösung des Geographie-Spiels mit den 305 Millionenstädten bei 26 Buchstaben bis zu $305^{26\cdot 26}$ Möglichkeiten betrachten muss. Dies sind weit mehr als es Elektronen im Weltall gibt, die im Augenblick auf unter 10^{80} geschätzt werden. Der oben beschriebene effiziente (d.h. in polynomieller Zeit laufende) Algorithmus nach dem Prinzip des dynamsichen Programmierens scheitert hier schon an dem immensen Bedarf an Speicherplatz. Ironischerweise erscheint es daher sinnvoller, jetzt doch wieder auf den zu Ende des Abschnitts 1.1 skizzierten rekursiven Algorithmus zurückzugreifen. Allerdings scheitert auch dieser bei dem naiven Versuch, ihn auf die gegebene 26 mal 26 Matrix anzuwenden, an der kombinatorischen Explosion. Aber mit einigen Vorüberlegungen kommt man doch noch zum Ziel.

Man kann nämlich die Ausgangsmatrix durch folgende Überlegung vereinfachen: Wenn wir in den Eintrag der Zeile A und Spalte D bzw. den der Zeile D und Spalte A schauen, so stellen wir fest, dass dort jeweils die Zahl 3 steht. Das bedeutet, dass es drei Städte gibt, die mit A beginnen und mit D enden, aber auch drei Städte, die mit D beginnen und mit A enden. In diesem Fall können wir alle 6 Städte aus dem Spiel entfernen, ohne die Aufteilung in Gewinner- und Verlierersituation dadurch zu verändern. Generell kann man nach diesem Prinzip bei zwei positiven Einträgen in der i-ten Zeile und j-ten Spalte bzw. j-ten Zeile und i-ten Spalte den größeren Eintrag durch die Differenz der beiden Einträge ersetzen und den kleineren durch Null. Ein Beweis der Gleichwertigkeit lässt sich induktiv leicht führen. Im Fall $i = j$ entspricht das der Reduktion modulo 2, d.h. ungerade Zahlen in der Hauptdiagonalen werden zu 1, gerade zu 0.

Wenn wir alle solchen Fälle entsprechend ersetzt haben, sagen wir, die Spielmatrix ist *in reduzierter Form*. Wir benutzen nun folgenden Algorithmus:

1) Gegeben sei unsere Ausgangsmatrix und eine Startstadt, d.h. in Wahrheit ein Paar von Buchstaben (Anfangs- und Endbuchstabe der Startstadt). Man beachte: Es ist egal, ob man mit der südkoreanischen Stadt Busan oder mit der deutschen Hauptstadt Berlin startet. Im ersten Schritt entfernen wir die Stadt aus der Matrix, indem wir den entsprechenden Eintrag um 1 verringern.

2) Jetzt reduzieren wir die aktuelle Matrix wie oben beschrieben.

3) Nun rufen wir eine rekursive Prozedur auf, deren Parameter die Angabe der zum Endbuchstaben der Startstadt gehörenden Zeile ist. Die rekursive Prozedur probiert jede Möglichkeit, d.h. jeder Eintrag in der gegebenen Zeile mit Wert größer als 0 wird geprüft. Die Prüfung geschieht, indem der Eintrag verringert wird und rekursiv die Prozedur wieder mit der Spalte des verringerten Eintrags als Parameter aufgerufen wird. Ist unter allen diesen rekursiven Aufrufen mindestens einmal die Antwort "Verlierer", dann gibt die Prozedur "Gewinner" zurück, andernfalls "Verlierer".

Diesen Algorithmus führen wir für jede vorkommende Kombination von Anfangs- und Endbuchstaben durch. Das Ergebnis ist in Abbildung 1.6 dargestellt. Man sieht, dass zum Beispiel alle auf U endenden Städte dazu führen, dass der beginnende Spieler (der nun also eine der drei mit U beginnenden Städte nennen muss) in jedem Fall verlieren wird. Auch bei Startstadt London hat der anfangende Spieler keine Chance. Dagegen sollte man bei jeder mit N beginnenden Startstadt (z.B. New York) versuchen, der anfangende Spieler zu sein – dieser hat immer eine Gewinnstrategie. Von den deutschen Millionenstädten ist Hamburg eine Siegerstadt.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	+			+	+				+					−	+			−							−	
B	+			+	+		+		+		+	+	+	−	+				+	+	−					
C	+						+		+					−	+				+		−		+		−	+
D	+								+		+		+	−				−	+		−				−	
E														−					+							
F	+			+										−							−					
G	+			+			+	+				+				+			+		−					
H	+			+	+		+		+					−						+	−					
I					+							+		−				−								
J	+						+							−				−								
K	+				−		+	+	+			+	+	−	+			−		+			+			
L	+				+		+		+					−					+		−					
M	+			+	+			+	+		+	+		−	+			−	+	+	−				−	
N	+			+			+		+		+	+						+		+						
O	+										+			−				−			−					
P	+			+	+		+	+										−	+	+				+		
Q													+		+										−	
R				+	+		+		+				+	+						+						
S	+				+		+		+			+	+	−	+			−		+	−				−	+
T	+				+		+	+			+			−	+				+	+						
U	+								+									+								
V	+								+				+					−								
W				+					+					−							−					
X							+							−							−					
Y	+				+																					
Z																					−					

Abbildung 1.6: Ergebnis beim Millionenspiel

Auf der Internetseite der Autoren findet sich der Quelltext zum rekursiven Programm, welches zur Berechnung der obigen Tabelle benutzt wurde. In Vorbereitung ist auch eine interaktive Möglichkeit, gegen dieses Programm und verschiedene Listen von Städtenamen on-line anzutreten.

Literaturverzeichnis

[Pap94] PAPADIMITRIOU, CHRISTOS: *Computational Complexity*. Addison-Wesley, 1994.

[Sch78] SCHAEFER, THOMAS J.: *On the complexity of some two-person perfect-information games*. Journal of Computer and System Sciences, 16:185–225, 1978.