• Konstruktionsbegriffe für die 3D-Raumgeometrie

Andreas Goebel und Reinhard Oldenburg, Frankfurt

Die Entwicklung von Programmen für die dynamische Raumgeometrie erfordert einen präzise definierten Begriff der geometrischen Konstruktion im Raum. Bei der Wahl eines solchen Konstruktionsbegriffs spielen technische, mathematische und didaktische Fragen eine Rolle. Einige Möglichkeiten sollen vorgestellt und didaktisch bewertet werden.

1 Einleitung

Dynamische Geometriesysteme erlauben dem Benutzer, geometrische Konstruktionen zu erstellen und die von ihnen bestimmten Figuren in parametrischer Abhängigkeit (dynamisch) von der Konfiguration der Ausgangsobjekte zu untersuchen. Die geometrischen Konstruktionen gewannen dadurch wieder an Bedeutung. Allerdings haben nicht alle Schüler diese Wendung verinnerlicht und so wurde schon kurz nach der Einführung von DGS beobachtet, dass Schüler Probleme haben, korrekte Konstruktionen aufzustellen. Diese Probleme können bei bestimmten Zielen vermieden werden, indem fertige Konstruktionen vorgegeben werden, aber gerade bei der Arbeit an offenen Problemstellungen ist es nach wie vor sinnvoll, wenn die Schüler selbst Ideen durch Konstruieren ausprobieren können.

Mit der Entwicklung von dynamischen Geometriessystemen für die Raumgeometrie stellt sich die Frage nach den Konstruktionen erneut: Beobachtungen von Schülern und Lehramtsstudenten zeigen, dass keineswegs intuitiv ist, wie man eine Konstruktion erstellt. Dies gilt auch dann, wenn das zu konstruierende Objekt in allen Details bekannt ist. So hat Hattermann Studenten Würfel konstruieren lassen, wofür diese recht lange brauchten und teilweise nicht zum Erfolg kamen. Da in seiner Studie die Studenten keine Einführung in Raumgeometrie bekommen haben, gibt es viel mehr erklärende Ursachen als nur die Probleme mit der Erstellung der Konstruktion. Aber auch in eigenen Versuchen zeigte sich, dass das Finden von Konstruktionen eine anspruchsvolle Tätigkeit ist. Die meisten Konstruktionsaufgaben sind für die meisten Probanden Problemlöseaufgaben. Die Erfahrung zeigt, dass Schüler und Studenten ihre Kompetenzen in diesem Bereich verhältnismäßig schnell ausbauen können. Dynamische Raumgeometrie kann also in der gegenwärtig verfügbaren Form durchaus erfolgreich eingesetzt werden und wird dies ja auch schon an vielen Orten.

Trotzdem bleibt die Beobachtung bestehen, dass Schüler und Studenten häufig nicht Dinge tun können, die sie tun wollen. 3D-Konstruktionsprogramme sind sperriger als mehr am Zeichnen orientierte Programme wie Google Sketchup. Von der mentalen Vorstellung eines Körpers und der Bewusstwerdung seiner definierenden Eigenschaften bis zu einer Konstruktion ist es ein weiter Weg. Zwei didaktische Positionen dazu:

Position 1: Die Probleme sind nicht artifiziell, sondern sie sind der Konstruktion im Raum immanent. Die Schüler sollen diese Probleme lösen als wertvollen Teil ihres Lernprozesses

Position 2: Die Software ist nicht perfekt, man kann es den Schülern leichter machen, Körper zu konstruieren

Welche Position ist richtig? Vielleicht beide und in diesem Aufsatz soll die Frage aufgeworfen werden, ob es möglich ist, den zugrunde gelegten Konstruktionsbegriff im Sinne von Position 2 so zu modifizieren, dass die Zugänglichkeit erhöht wird, ohne (Position 1!) die Erstellung der geometrischen Konstruktionen zu trivialisieren. Geometrische Konstruktionsaufgaben sind Problemlöseaufgaben und diese können nur bildend wirken, wenn sie nicht zu einfach sind. Andererseits kommt es auf die richtige Balance an zwischen der Problemschwierigkeit und dem Motivationseffekt, d.h. die Schwierigkeit sollte so eingestellt werden, dass das Konstruieren weder demotiviert noch so einfach ist, dass kein Problemlöseprozess mehr angeregt wird.

2 Konstruktionen

Geometrische Konstruktionen sind Algorithmen, die die Erzeugung einer Konfiguration geometrischer Objekte aus bestimmten Anfangsdaten auf elementare Schritte zurückführen. In der Regel löst eine Konstruktion ein bestimmtes Problem, d.h. die konstruierten Objekte stehen untereinander in einer bestimmten Relation. Eine Konstruktion bestimmt eine Folge $A_1, A_2, A_3, \ldots, A_n$ von geometrischen Objekten. Dabei sind die ersten Objekte als Ausgangsdaten gegeben (Startkonfiguration). Die Konstruktion bestimmt die folgenden Objekte aus den vorhergehenden jeweils durch einen elementaren Konstruktionsschritt, der eine funktionale Abhängigkeit darstellt: $A_k = f_k(A_1, A_2, \ldots, A_{n-1})$. Was ein elementarer Konstruktionsschritt ist, also welche Funktionen hier zugelassen sind, hängt einerseits von den zugelassenen Konstruktionswerkzeugen ab, ist andererseits eine Frage der Bequemlichkeit. Wie auch in anderen Bereichen der Algorithmik werden häufig gebrauchte Teilalgorithmen in eigene Modulen gekapselt (Standardkonstruktionen nach Holland;

Makros in DGS), die dann als ein Konstruktionsschritt erscheinen. Auch viele in DGS fest eingebaute Konstruktionsschritte könnten auf andere, einfachere zurückgeführt werden. Eine Teilmenge der elementaren Konstruktionsschritte, die nicht weiter verkleinert werden kann, ohne die Konstruktionsmöglichkeiten einzuschränken, soll essentiell heißen.

3 Der Konstruktionsbegriff in 3D-DGS

Bisher gibt es mit Archimedes Geo 3D und Cabri 3D zwei etablierte dynamische Raumgeometriesysteme. Die jeweiligen Konstruktionsbegriffe sind nicht identisch, aber trotzdem soll hier versucht erden, eine vereinheitlichende Darstellung zu geben.

Eine geometrische Konstruktion im Sinne der dynamischen Raumgeometrie eine Spezialisierung des Konstruktionsbegriffs aus dem vorhergehenden Abschnitt. Hier folgt eine Auflistung von essentiellen elementaren Konstruktionsschritten. Dabei beschränken wir uns auf Konstruktionen der synthetischen Geometrie und beschränken den Objektvorrat auf Punkt, Gerade, Ebene und Kugel.

- ▷ Schnittpunkt Ebene-Gerade
- ▷ Schnittpunkt Kugel-Gerade
- ▷ Punkt auf Gerade, Kreis, Ebene
- ▷ Gerade durch zwei Punkte
- ▷ Lot zu Ebene durch Punkt
- ▷ Parallele Gerade zu anderer Gerade durch einen bestimmten Punkt
- ▷ Kugel aus Mittel- und Randpunkt
- ▷ u.s.w.

Das Problem in dieser Aufzählung ist das unscheinbare „u.s.w.“: Es würde den hier zur Verfügung stehenden Platz sprengen, alle Konstruktionsmöglichkeiten im Raum aufzuzählen. Lernende müssen sich dieser Vielfalt bewusst lernen, sie müssen herausfinden, wie man all diese Operationen mit dem jeweiligen Werkzeug ausführt — und dann haben sie immer noch nicht das Problem gelöst, zu einer Konstruktionssequenz zu kommen.

4 Schwierigkeiten mit halbfreien Elementen in Raumgeometrischen Konstruktionen

In der dynamischen Raumgeometrie gibt es (zumindest potentiell — die real existierenden Systeme realisieren längt nicht alle Fälle) mehr halbfreie Objekte als in der 2D-DG. Dies sind Objekte, deren Lage durch die Konstruktion nicht eindeutig bestimmt wird. In zweidimensionalen Geometriesystemen sind dies vor allem: Punkt auf Gerade und Punkt auf Kreis. Im Prinzip könnte es auch Orthogonale, Parallele (jeweils ohne die Eindeutigkeit herstellende weitere Forderung durch einen bestimmten Punkt zu gehen), sowie Gerade durch einen Punkt geben. Diese halbfreien geraden wären dann noch verschiebbar (in den ersten beiden Fällen) bzw drehbar (im letzen Fall).

In 3D-DGS gibt es potentiell sehr viele halbfreie Objekte, eine Auswahl:

- ▷ Punkte auf Geraden und Kreisen, aber auch auf Kugel und Ebenen
- ▷ Geraden in Ebenen
- ▷ Kreise auf Kugeln
- ▷ Geraden g, die durch einen Punkt P auf einer Geraden h laufen und zu h orthogonal sind (diese Geraden können noch um P in der Lotebene auf h gedreht werden)
- ▷ Geraden g, die durch einen Punkt P verlaufen und zu einer Ebene E parallel sind (diese Geraden können noch in der Parallelebenen zu E durch P gedreht werden) Noch größer wird die Zahl der halbfreien Objekte, wenn man metrische Eigenschaften hinzunimmt (z.B. Strecken fester Länge).

Die Implementierung halbfreier Elemente ist nicht einfach und es können merkwürdige Effekte resultieren (Wird ein Bestimmungsstück ein halbfreien Objektes gezogen, definiert die Konstruktion nicht eindeutig, wo das halbfreie Objekte hingeschoben werden soll. Wenn das DGS diese Freiheit stets nach einer bestimmten Strategie füllt, resultieren Zugfiguren, die mehr Relationen invariant lassen, als aus der Konstruktion logisch folgen.). Es ist daher nicht verwunderlich, dass die beiden bisher existierenden 3D-DGS nicht alle denkbaren halbfreien Objekte implementieren. Halbfreie Elemente könnten durch hinzufügen einer (oder mehrerer) weiteren Relation eindeutig spezifiziert werden. Diese Umwandlung ist in 2D-Systemen oft nicht vorgesehen, weil sie dort nur wenige Fälle betrifft: Es könnte sinnvoll sein, einen Punkt, der an eine Gerade gebunden ist, an eine weitere Gerade zu binden, also zum Schnittpunkt der beiden Geraden zu machen.

Jedes sachlogisch mögliche halbfreie Objekt entspricht einem durch das Wissen des Lernenden unterbestimmten Konstruktionsschritt. Deshalb kann man folgende These formulieren:

These 1. *Ein Teil der Probleme von Anfängern mit der 3D-DG resultieren daraus, dass zur jeweiligen Situation passende halbfreie Objekte nicht zur Verfügung stehen oder dass halbfreie Objekte nicht weiter spezifiziert werden können.*

Diese These stützt sich sowohl auf die Beobachtung von Studierenden als auch auf die Introspektion (sic!) der Autoren, sie wurde aber a-posteriori wegen ihrer erklärenden Kraft aufgestellt und nicht durch Interviews abgesichert. Man kann sich aber leicht davon überzeugen, dass es beim raumgeometrischen Konstruieren häufig zu

Situationen der folgenden Art kommt: Man möchte z.B. eine Strecke s konstruieren, kennt schon einen Endpunkt A und weiß dass sie senkrecht auf einer bereits existierenden Geraden g stehen muss. Damit kann die Strecke aber noch nicht eindeutig konstruiert werden. Wenn es kein Werkzeug für solche halbfreien Objekte gibt, kann der Benutzer seine Idee nicht weiter verfolgen, es sei denn, er verfügt über folgend Strategie: Man erzeugt die Lot-Ebene E zu g durch P. Der Endpunkt der gesuchten Strecke S muss in E liegen. Man erzeugt also einen halbfreien Punkt Q auf E und eine halbfreie Strecke $S = PQ$. Die Hilfsebene E kann danach verborgen werden. Neulinge in der 3D-DG verfügen noch nicht über solche Strategien. Zudem haben sie folgenden Nachteil: Wenn man sich im weiteren Verlauf der Konstruktion klar macht, welche Länge die Strecke haben soll, kann dies nicht mehr mit $s = PQ$ umgesetzt werden, denn dazu müsste Q umdefiniert werden (vom Punkt in der Ebene E zum Punkt auf einem Kreis in E um P mit dem passenden Radius). Dieses komplexe und evtl. mit dem jeweiligen 3D-DGS nicht durchführbare Verfahren ist fehleranfällig. Es wäre unnötig, wenn halbfreie Objekte durch hinzufügen von Eigenschaften leicht ausspezifiziert werden könnten.

5 Geometrie ohne Konstruktionen?

Geometriesysteme wie Geometry Expressions oder FeliX benötigen keine Konstruktionsbeschreibung: Der Benutzer spezifiziert nur, welche Objekte es gibt und welche Relationen zwischen diesen gelten sollen. Bei FeliX ist die Reihenfolge dieser Vorgänge beliebig (bis auf die triviale Bedingung, dass eine Relation nur Objekte betreffen kann, die bereits existieren). Es gibt keine prinzipiellen Probleme bei der Übertragung dieses Ansatzes auf die Raumgeometrie. Allerdings arbeitet zB FeliX mit Algorithmen, deren Laufzeit exponentiell mit der Zahl der Variablen steigt, und Raumgeometrieprobleme führen zu noch mehr Variablen und verschärfen daher die Performance-Problematik.

Neben diesem technischen Argument sind es vor allem didaktische Überlegungen, die diesen Weg bei geometrischer (und nicht algebraischer!) Zielsetzung als problematisch erweisen: „Konstruktionen“ werden mit solchen Systemen trivialisiert und damit wird dieses Übungsfeld des Problemlösens wertlos.

Neben dem Aspekt des Problemlösens liegt die didaktische Bedeutung des Konstruierens u.E. auch wesentlich darin, dass hier Handlungsabläufe geplant und organisiert werden. Dieser Aspekt des Konstruktionsbegriffs steht Inhalten der Informatik nahe. Die exemplarische Bedeutung erstreckt sich auf viele Stellen, wo in unserer Gesellschaft organisiert und geplant wird. Es können grundlegende Begriffe wie etwa Abhängigkeit und Unabhängigkeit von Teilschritten eingeübt werden.

6 Variante 1: Konstruktion = Reihenfolge von Objekten

Die oben dargestellten Schwierigkeiten mit halbfreien Objekten können ausgeräumt werden, wenn der Konstruktionsbegriff so geändert wird, dass es keine prinzipielle Unterscheidung mehr zwischen freien, halbfreien und eindeutig konstruierten Objekten gibt. Ein solcher Konstruktionsbegriff soll hier dargestellt werden. Dabei wird eine Konstruktion verstanden als eine Tabelle der folgenden Art:

Name	Art	Relationen
P	Punkt	—
Q	Punkt	Abstand zu $Q = 5$
T	Punkt	Koordinaten=$(1,2,3)$
g	Gerade	Geht durch P und Q
s	Strecke	Anfangspunkt P; senkrecht zu g
E	Ebene	Parallel zu s; parallel zu g

Der zeilenweise Aufbau gibt die Schritte der Konstruktion wider. Jedem Objekt können Relationen auferlegt werden, die es erfüllen soll. Vorteil dieses Ansatzes ist, dass die Reihenfolge, in der die Relationen eingetragen werden, offen ist. Man braucht nicht gleich bei der Erzeugung eines Objektes alle Eigenschaften kennen, sondern es können schrittweise Relationen hinzugefügt werden, bis das Objekt so festgelegt ist, wie es der Intention des Benutzers entspricht. Entscheidend dafür, dass es sich trotz dieser Freiheit um einen effektiv ausführbaren Konstruktionsbegriff handelt ist die Einhaltung der folgenden Regel: Die Relationen, die zu jedem Objekt angegeben werden, dürfen das Objekt nur durch Verweis auf andere Objekte spezifizieren, die in der Tabelle vorher spezifiziert worden sind, also oberhalb stehen. Es wäre also verboten, z.B. zu g die Relation „orthogonal zu E“ hinzuzufügen. Durch diese Vorschrift bleibt die Konstruktion leicht berechenbar. Wird beispielsweise mit der Maus Q an eine neue Position gezogen, kann folgendermaßen gerechnet werden: Als erstes wird versucht, dem Zugziel für Q unter Beachtung der Relationen an Q so nahe zu kommen wir möglich. Dann werden nacheinander alle folgenden Objekte neu berechnet werden. Wenn man das Objekt in der i-ten Zeile neu berechnet, sind die Objekte in den Zeile $1,\ldots,i-1$ bereits aktualisiert, ihre neuen Koordinaten liegen also fest. Im i-ten Schritt ist daher nur das verhältnismäßig kleine Problem zu lösen, die an das i-te Objekte gestellten Relationen zu erfüllen bzw. deren Nichterfüllbarkeit nachzuweisen. Beides ist algebraisch leicht möglich.

Dieser Konstruktionsbegriff ist implementierbar (eine *proof-of-concept*-Implementierung existiert, siehe Screenshot) und effektiv ausführbar, denn der Zeitbedarf steigt nur linear mit der Zahl der Objekte solange die Zahl der Relationen für jedes einzelne Objekt beschränkt ist.

Mit einem 3D-DGS, das auf diesem Konstruktionsbegriff aufbaut, kann effektiv gearbeitet werden. Allerdings erfordert es durchaus ein planvolles Vorgehen: Wenn man etwa zwei Punkte erzeugt und s als die Strecke zwischen ihnen definiert, dann ist es sinnlos, die Länge von s durch eine Relation an s zu spezifizieren, man würde dann nichterfüllbare Relationen postulieren, weil die Lage von s ja schon durch die Endpunkte eindeutig gegeben ist. Stattdessen könnte man eine Abstandsbedingung an die beiden Punkte stellen oder ganz auf die Eckpunkt verzichten. Allerdings zeigt das Beispiel auch eine problematische Eigenschaft: Während es naheliegend ist, Streckenlängen als Eigenschaften von Strecken zu sehen, muss man hier — bei dieser Art der Konstruktion — die Streckenlänge als Eigenschaft ihrer Endpunkte sehen. Um das zu umgehen, müsste man eine freie Strecke konstruieren, deren Länge und Anfangspunkt festlegen, und dann den zweiten Punkt als ihr Ende definieren.

Man sieht: Konstruieren in diesem Sinne ist durchaus kognitiv anspruchsvoll, denn man muss eine passende Reihenfolge der Objekte finden. Da dabei dem Benutzer Fehler unterlaufen können wäre es wünschenswert, wenn ein solches System die Möglichkeit böte, die Reihenfolge umzustellen. Dazu müssen Relationen von einem Objekt auf ein anderes abgewälzt werden. Das algorithmisch umzusetzen ist möglich, aber aufwändig. Es wird hier nur an einem Beispiel illustriert:

Name	Art	Relationen
P	Punkt	—
g	Gerade	Geht durch P
s	Strecke	senkrecht zu g

Um g an s vorbei nach hinten zu schieben, muss die Orthogonalitätsrelation von s auf g „umgebucht" werden. Das Ergebnis ist:

Name	Art	Relationen
P	Punkt	—
s	Strecke	—
g	Gerade	Geht durch P; senkrecht zu s

Die Relationen, die in diesem Konstruktionsbegriff möglich sind, unterliegen gewissen Einschränkungen, denn n-stellige Relationen sollten bei Vorgabe von $n-1$ Argumenten allein durch Variation des n-ten Argumentes erfüllt werden können. Dies ist beispielsweise bei der 3-stelligen Relation Mittelpunkt der Fall, nicht aber bei der dreistelligen Relation auf Geraden „paarweise orthogonal". Diese Relation kann also nicht (als eine Relation) verwendet werden, sondern muss aus zweistelligen Orthogonalitätsrelationen zusammen gebaut werden.

Es ist möglich, den Konstruktionsbegriff von Archimedes Geo3D so weiter zu entwickeln, dass er sich diesem neuen Konstruktionsbegriff annähert. Dazu ist vor allem die Flexibilität des Umdefinierens zu erhöhen. In umgekehrter Richtung erscheint es denkbar, einen Algorithmus zu finden, der eine Konstruktion in diesem neuen Sinne in eine Konstruktion im alten Stil überführen kann, sofern die verwendeten Relationen mit den Werkzeugen des klassischen 3D-DGS konstruierbar sind.

Bei der didaktischen Bewertung dieses Konstruktionsbegriffs sollte aber nicht vergessen werden, dass die einzelnen „elementaren" Schritte recht komplex werden können. Zwar ist auch die Arbeit der Konstruktionswerkzeuge in einem 3D-DGS wie Archimedes Geo3D von abstrakter Natur (denn anders als die Zirkel der ebenen Geometrie lassen sich Kugelzirkel nicht materiell umsetzen), aber die Kombination von Relationen ist eine logisch-abstrakte Operation und praktisch nur algebraisch durchführbar.

7 Variante 2: Konstruktion = Reihenfolge von Relationen

Die klassischen Konstruktionsbegriffe ebenso wie der im letzen Abschnitt diskutierte stellen ganz grob gesprochen vor allem eine Konstruktionsreihenfolge her, d.h. die Objekte haben eine bestimmte Reihenfolge. Dies mag auf den ersten Blick für eine Konstruktion geradezu wesensnotwendig sein, auch wenn man sich klar machen kann, dass es bei den meisten Konstruktionen unabhängige Teilschritte gibt, deren Reihenfolge umgestellt werden kann.

Bei Konstruktionen mit Baukästen u.ä. gilt diese scheinbar natürliche Regel aber nicht: Die Objekte werden nicht der Reihe nach erzeugt, sondern sie sind gleich alle da (mehr oder wenig ordentlich im Baukasten, der letztlich eine Menge von Objekten ist). Beim Konstruktionsprozess klickt (im Sinne von Lego, nicht von Mausinteraktion) man die Bausteine zusammen und das bedeutet beispielsweise, dass man eine Inzidenzrelation herstellt. Beim Bauen mit Bausteinen haben also nicht die Objekte, sondern die Relationen eine Reihenfolge! Mit dem hier vorzustellenden Konstruktionsbegriff soll versucht werden, dies zu modellieren.

Eine Konstruktion in diesem Sinne besteht also in einer Menge (ungeordnet) von Objekten O sowie einer geordneten Liste von Relationen R.

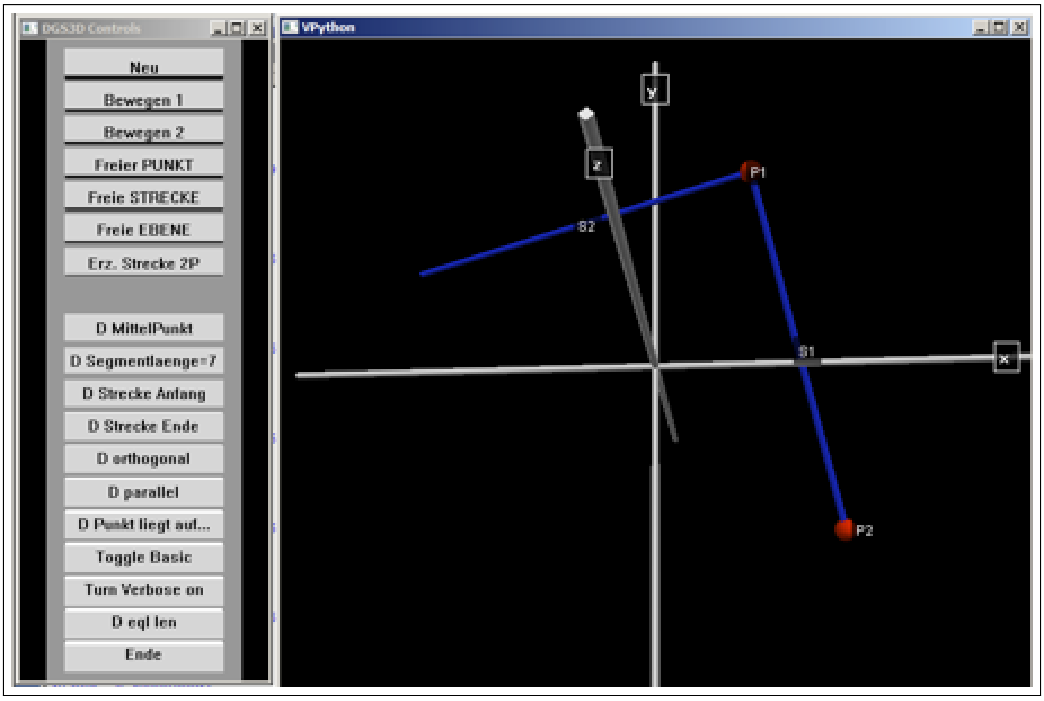

Abbildung 5.1: Experimentelle Realisierung der Variante 1. Hier ist $S2$ eine freie Strecke, die auf $S1$ senkrecht steht und die gleiche Länge hat. Sie ist aber noch um $S1$ drehbar.

Jedes Objekt a aus O gehört zu einem bestimmten Typ (Punkt, Gerade, etc.) und besitzt entsprechend eine Menge von Variablen V(a). Bei der ersten Variante des Konstruktionsbegriffs wurden in einem Schritt alle Variablen eines Objektes bestimmt. Das ist jetzt nicht mehr möglich. Typischerweise kommen die Variablen eines Objektes in verschiedenen Relationen vor und wenn die Konstruktionsberechnung Relation-nach-Relation durchgeführt wird, braucht man eine Strategie, wann man sie ändert. Ein erster Algorithmus zur Berechnung einer Konstruktion: U die Menge aller Variablen aller Objekte. Durch die Iteration über die Relationen hinweg soll U stets die Variablen enthalten, deren Werte noch nicht neu berechnet wurden. Für jede Relation $r \in R$ bildet man dann den Spezialfall r', in dem die bereits bestimmten Variablen eingesetzt werden, dann bildet man die Variablen $W = U \cap V(r')$, die evtl. aus r' bestimmt werden könnten. Dann sucht man eine möglichst kleine Teilmenge von Variablen aus W, die genügt, r' zu erfüllen. Diese werden dann neu berechnet, aus U entfernt und die nächste Relation wird erfüllt.

Die einzelnen Schritte dieses Algorithmus sind recht aufwändig, aber durchführbar und das wurde in einer Testimplementierung bestätigt. Die Arbeit mit einem darauf basierenden DGS gestaltet sich recht komfortabel, weil die Reihenfolge der Relationen beliebig umgestellt werden kann. Allerdings macht sich auch in der praktischen Arbeit sehr bald folgendes Problem bemerkbar: Der Algorithmus schafft es nicht, alle Relationen zu erfüllen, obwohl dies möglich wäre. Ursache dafür ist in der Regel, dass für einige Relationen zu viele Variable benötigt werden. Das hängt davon ab, durch welche Variablen die Objekte beschrieben werden. Strecken können beispielsweise durch die kartesischen Koordinaten ihrer beiden Endpunkte beschreiben werden oder durch Anfangspunkt, Länge und Richtungswinkel. Wenn bereits eine Relation erfüllt wurde, die den Anfangspunkt festlegt, erfasst eine Relation, die die Länge vorschreibt, in der ersten Beschreibungsform alle kartesischen Variablen des Endpunktes. Diese werden also (wenn auch teilweise willkürlich) neu festgelegt. Für eine spätere Relation, die die Strecke als orthogonal zu einer anderen bestimmt, bleibt dann keine Freiheit mehr über. Bei der zweiten Beschreibung tritt dieses Problem nicht auf. Nun kann man vermuten, dass die Beschreibung durch intrinsische geometrische Größen wie die Länge ohnehin besser ist. Da es aber auch Relationen gibt, die sich kartesische einfacher ausdrücken lassen, bleibt nur die aufwändige Lösung, die Beschreibung den gestellten Relationen anzupassen. Eine Lösung, die aber auch nicht alle Konfliktfälle entschärfen kann. Eine systematische Lösung könnte in der sukzessiven adhoc Einführung neuer lokaler Koordinatensysteme liegen, aber ein solches Vorhaben ist umfangreich und müsste stabil mit ausgearteten Situationen umgehen können.

Nachteilig an diesem Algorithmus ist vor allem aber auch, dass sein Verhalten nicht allein auf geometrischer Ebene verstanden werden kann. Es ist daher zu vermuten, dass Lernende das Verhalten eines entsprechenden DGS als willkürlich empfinden werden. Dies steht im Kontrast zu der eingangs gemachten Analogie zum Bauen mit einem Baukasten. Es ist offen, ob hier ein Ausgleich erfolgen kann. Die eingangs zitierte Metapher des Baukastens legt aber nahe, dass eine Lösung der technischen Probleme didaktisch reizvoll und lohnend sein könnte.

8 Fazit

Ob 3D-DGS auf Basis der hier beschriebenen Konstruktionsbegriffe geeignet sind, den Schülern selbsttätiges raumgeometrisches Tun zu ermöglichen, kann derzeit noch nicht beurteilt werden. Dazu bräuchte man einerseits eine empirische Vergleichsstudie, andererseits klare normative Zielvorstellungen für den Unterricht in Raumgeometrie. Das Erste ist ohne das Zweite nicht sinnvoll, und das Zweite ist gegenwärtig didaktisch „out“ und nicht drittmittelfähig.