

AUTOMATISCHES ZEICHNEN VON DIAGRAMMEN

ABHANDLUNG

zur Erlangung

des Titels eines Doktors der Mathematik

der

EIDGENÖSSISCHEN TECHNISCHEN HOCHSCHULE ZÜRICH

vorgelegt von

KARL JOSEF BUCHER

dipl. Math. ETH

geboren am 1. September 1948

von Werthenstein /LU

angenommen auf Antrag von

Prof.Dr. P. Laeuchli, Referent

Prof.Dr. J. Nievergelt, Korreferent

1977

Abstract

Several techniques for automatic drawing of a wide class of diagrams are presented and evaluated. The line drawings that can be processed are recursively structured: subdiagrams consist of several vertically connected parallel sections. Each section consists of subdiagrams and atoms that are horizontally connected. Connections need not to be visible in the drawing. Atoms are arbitrary figures and may contain text. Syntax diagrams are an example of the above class.

The plotter programs must satisfy the following demands:

- Comfortable input and modification: short, readable, easy and safe commands expressed in terms of high-level pictorial properties, rather than in terms of coordinates.
- Flexible output: aesthetic, useful diagrams.

A graph rewriting system controlled by a sequential notation generates the diagram graphs. The two-dimensional coordinate assignment problem is separated into two different one-dimensional problems. Algorithms for solving these problems and a language for describing the input and output of each specific plotter program are presented. A meta program can then automatically transform the above description into the plotter program.

Zusammenfassung

Die vorliegende Arbeit befasst sich mit den Techniken, die dazu dienen koennen, das Zeichnen einer grossen Klasse von Diagrammen zu automatisieren. Bei der betrachteten Diagrammklasse handelt es sich um Strichzeichnungen, denen ein Graph zugrunde liegt, dessen Kanten mit Merkmalen versehen sind. Diese Merkmale sollen atomische Figuren, z.B. Kaestchen mit Textinhalt sein. Weiter soll es sich um "lesbare" Diagramme handeln. D.h. der Betrachter interessiert sich fuer die einzelnen Wege durch den Graphen. Wegen der verlangten Lesbarkeit der Diagramme ist es sinnvoll, sich auf Zeichnungen einzuschraenken, deren Layout horizontal orientiert ist. Dies soll bedeuten, dass diverse Unterdiagramme an horizontale Geraden (Ordner) gebunden sind. Das gemeinsame umschriebene Rechteck (mit horizontalen und vertikalen Kanten) dieser Unterdiagramme darf sich nicht mit einem analogen Rechteck eines andern Ordners ueberschneiden. Derartige Diagramme sollen also bei geeigneter Eingabe von einem Computerprogramm gezeichnet werden koennen. Dabei sei noch vorausgesetzt, dass jeder Anwender mit seinem Programm nur Zeichnungen aus einer engeren Teilklasse -aus dieser jedoch eine grosse Anzahl- darstellen will. Ein Beispiel fuer eine derartige Klasse bilden die Syntaxdiagramme, wie sie zur Illustration der Grammatik von formalen Sprachen verwendet werden. An das Darstellungsprogramm fuer eine spezielle Diagrammklasse seien folgende Forderungen gestellt:

- bequeme und sichere Eingabe:

D.h. die Eingabe soll moeglichst kurz, gut lesbar, leicht erlernbar und wenig fehleranfaellig sein.

Die Angabe von Koordinaten in irgend einer Form, wie dies praktisch alle (interaktiven) Graphenzeichnungssysteme verlangen, kommt somit sicher nicht in Frage.

Fuer die erwaehnten Syntaxdiagramme beispielsweise soll eine erweiterte Backus-Naur-Form zur Eingabe verwendet werden koennen.

- Modifikationsoperationen von hoechstmoeglichster Stufe:

D.h. einzelne Teile eines Diagramms sollen veraendert werden koennen, ohne dass auf dem Niveau der Geometrie ins Bild eingegriffen werden muss. Muessen Diagramme in aequivalente bezueglich einer Klasseneinteilung transformiert werden, so sollen die Modifikationsoperationen in dem Sinne sicher sein, dass ihre Anwendung immer die Aequivalenz garantiert.

- hinreichende Flexibilitaet bei der Ausgabe:

D.h. die Ausgabe soll sich so gestalten lassen, dass elementare praktische und aesthetische Ansprueche an die Diagramme erfuehrt werden koennen, wie etwa Eingepasstsein auf eine vorgegebene Seitenbreite, ausgewogene Platzausnutzung oder Lesbarkeit der Textinformation.

Das Ziel dieser Arbeit besteht darin, Prinzipien aufzustellen, nach welchen die verlangten Darstellungsprogramme

aufzubauen sind. Daneben werden diese Prinzipien praktisch erprobt.

Der Aufbau eines Darstellungsprogramms wird nun folgendermassen gegliedert:

- 1) Bestimmung der Form einer Eingabe und von Modifikationen, welche die obigen Forderungen erfuehlt.
- 2) Zusammensetzen eines abstrakten Diagramms (nur Kombinatorik, keine Geometrie).
- 3) Bestimmung der Koordinaten.
- 4) Ausgabe auf automatischen Zeichner oder Zeilendrucker.

Interessante Probleme ergeben hierbei die Punkte 1) und 3).

Fuer die Eingabe und den Diagrammaufbau wird ein linguistischer Ansatz gewaehlt: mit einer seriellen Notation wird ein Grapheneinsetzungssystem gesteuert. Jede Form eines einzusetzenden Untergraphen entspricht dabei einer syntaktischen Klasse der grammatikalischen Gliederung der Bilder.

Das zweidimensionale Problem der Koordinatenzuteilung wird durch die eingangs erwaehte Horizontalorientierung in zwei eindimensionale Probleme separiert. Diese Einschraenkung bewirkt eine starke Vereinfachung der Hoeheneinteilungsaufgabe. Fuer die Breiteneinteilung werden Algorithmen angegeben, welche eine ausgeglichene Platzeinteilung gewaehrleisten. Die systematische Loesung der oben aufgezaehten Aufgaben fuehrt zu einem Konzept, mit welchem man die Generierung der Plotprogramme fuer konkrete Diagrammklassen automatisieren kann.

Kern dieses Konzepts ist eine Sprache, welche Syntax und Semantik der Eingabe eines Plotprogramms beschreibt, samt einer Vorschrift, nach welcher diese Beschreibung in ein Plotprogramm zu uebersetzen ist. Diese Vorschrift wird selbst mit einem (Meta-) Programm realisiert. Auf diese Weise kann mit Hilfe des Metaprogramms die Beschreibung der Ein- und Ausgabe eines Plotprogramms automatisch in dieses Plotprogramm umgesetzt werden. Die Systematisierung der Plotprogrammgenerierung wird anhand des Beispiels der Syntaxdiagramme ueber mehrere Stufen entwickelt und auch mit Darstellungsprogrammen erprobt.