Educational Matters

The Educational Matters Column is running for three years already. I wish all its readers a peace- and successful year 1995. I wish me many contributions that it can go on. Please send everything interesting for the discussion of educational matters in Computer Science to

Prof. Dr. Hans-Jörg Kreowski
Fachbereich Mathematik und Informatik
Universität Bremen
Postfach 33 04 40
D-28334 Bremen
email:kreo@informatik.uni-bremen.de
Fax: ++49-421-218-4322

In this issue, you find my notes on a course I gave two years ago. I believe and hope that reports on teaching activities may be worthwhile parts of the column. Hence I would appreciate to get further contributions of this kind.

Conception of a course on Syntactic Methods in Picture Generation

Hans-Jörg Kreowski
Universität Bremen

The aim of this note is to sketch the idea and conception of an experimental course on *Syntactic Methods in Picture Generation* I held in the winter semester 1992/93. I hope what I experienced is worth to be communicated.

Background

The Computer Science studies in Bremen are similar to others in Germany with some specialities.

Students in the undergraduate phase must pass 17 mandatory courses among them 4 in Mathematics and 2 in Theoretical Computer Science. The aim at the latter is to introduce foundations of Computability and Complexity Theory as well as of Formal Language Theory.

Students in the graduate phase must pass 4 oral examinations (each on a subject covering about 2 courses), 4 additional courses, a project and a diploma thesis. About a quarter of the graduate studies is expected to belong to Theoretical Computer Science. In practice, an average student visits 2-3 courses chosen out of the subareas Algorithm and Complexity Theory, Formal Language Theory, Theory of Programming or a special field (that may be Foundations of Artificial Intelligence, Petri Nets, Term Rewriting and Theorem Proving, etc. depending on the actually offered courses). Therefore most of the courses in Theoretical Computer Science should be introductory and fundamental offering knowledge of and good insight into typical questions, topics and methods as well as the systematic and exact way of thinking and proving.

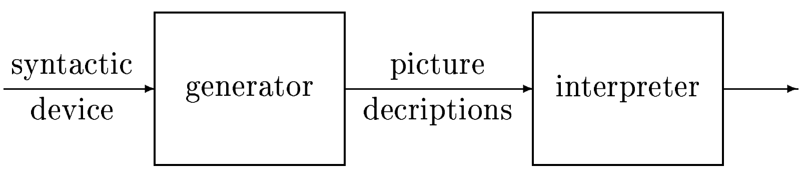
The projects are one of the specialities. They run usually for two years in groups of 15-30 students. The aim is usually to design and implement a software system for a particular purpose or application. Students work with great enthusiasm in the projects and appreciate their characteristics: team work, learning by doing, interdisciplinary, self organization. But teaching in projects is not free of trouble. A serious problem is caused by the start of projects at the beginning of the graduate phase when the participants do not know much about the principles of software technology and the particular purposes and topics addressed in the projects. Hence one must hurry to provide this knowledge so that the participants can start to think, discuss and work properly before their interest and enthusiasm are cooled down.

This was a major problem I faced when I suggested and initiated the project BIZARR (the shortcut of "Bilderzauber", i.e. magic of pictures) in winter semester 1992/93. BIZARR had an open planning: In the first half year, the about 30 participants were urged to study a variety of syntactic methods for picture generation including experimental implementations. After half a year, the participants were assumed to decide what they liked to do in the rest of project. But the course program of the semester did neither offer a course on Computer Graphics to support the picture processing aspect nor on Formal Language Theory to back up the syntactic methods part of the project. Hence I was more or less forced to come up with a new course to accompany the BIZARR project properly. And at the same time, the course should have an introductory character to serve the purposes of the curriculum in Theoretical Computer Science.

Koch tree: rules and a derivation

Concept

Syntactic approaches to picture processing are known for more than 20 years (cf., e.g., Feder 68, Pfaltz and Rosenfeld 69, Shaw 69, Fu 74, Gips 75, Stiny 75). In the last 10 years, the study of syntactic methods for picture processing has been intensified. The reasons may be that the demand of methods in Computer Graphics is quite high, the graphical surfaces are much improved so that rule-based and interactive methods has become more and more feasible and that fractal geometry as the mathematical counterpart of syntactic methods for picture processing has become extraordinarily popular. The cource on *Syntactic Methods for Picture Generation* reflects the historic development with an emphasis an generative methods by putting together more traditional approaches (like chain code picture languages and cellular automata, see, e.g., Freeman 74, Maurer, Rozenberg and Welzl 82, Dassow 89, Codd 68, Toffoli and Margolus 87) more recent approaches (like turtle-interpreted languages and collage grammars, see, e.g., Prusinkiewicz and Lindenmayer 90, Drewes, Habel, Kreowski and Taubenberger 93 and 94) and concepts from fractal geometry (like iterated function systems, see, e.g., Barnsley 88, Peitgen, Jürgens and Saupe 93). All the methods follow the same schema of picture generation: a syntactic device generates (or specifies) a set or stream of picture descriptions that can be interpreted as a set or stream of pictures.



In the cases of chain code picture languages and turtle-interpreted languages, picture descriptions are strings such that Chomsky grammars, L-systems and such can be used as syntactic devices and there is a good chance to apply concepts and results of Formal Languages Theory. The symbols are commands for some drawing device (a plotter in the first case and the turtle of the programming languages LOGO in the second case) such that pictures are drawn while the strings are read.In the other cases the seperation of generator and interpreter is less significant.

Contents

Besides the basic syntactic and semantic notions of the five approaches, some first steps of the theory were developed in each case discussing structural properties and decision problems.

A Kleene-like characterization of regular clain code picture languages and fixed-point theorems for iterated function systems and collage grammars belong to the structural results one can show in such a course. The characterization uses directly the Kleene characterization of regular string languages. The result for iterated function system (with contracting functions) is based on the convergence of Cauchy sequences and allows to recall some important results of Calculus and Topology. The result for collage grammars (that are hyperedge replacement grammars deriving geometric structures) adapts the fixed-point theorem for context-free string languages.

The membership problem for context-free clain code picture languages can be reduced to the emptiness problem of context-free string languages (using the fact that the intersection of the latter with regular sets are context-free). As another positive result, one can decide for a certain type of collage grammars whether a given region is not fully covered by any of the pictures generated by the input grammar. In all the approaches, it is easier to come up with undecidability results. For example, Post's Correspondence Problem can be reduced in a nice way to the question whether a linear turtle-interpreted language (or a linear clain code picture language likewise) contains only simple lines (or line drawings with branching or crossing).

It seems to difficult to compare to various approaches. The only substantial result I succeeded to present in the course was a translation of iterated function systems into collage grammars.

Sierpinski tetraeder: generated by a collage grammar

Educational Aspects

Syntactic methods in picture processing are comparatively easy to motivate because one can show many pictures illustrating potentials, applications and relations to other areas: modelling, visualization and animation of natural phenomena (growing of plants, texture of shells, clouds, etc.), image compression, tilings, fractals, (computer) artworks. (But one should be careful because pictures are often more convincing than is justified).

Although the subject of the course is somewhat exotic, there are various possibilities to connect it with traditional topics of Theoretical Computer Science and Mathematics. Therefore, systematic investigation and typical questions and methods can be combined with actual and attractive considerations. Moreover, the theory of syntactic picture generation is nowhere far developed so that one can point out many open ends and problems even in early stages of presentation. This can encourage students to try creative thinking and research.

From the reactions during the course and the smalltalk around it, I got the impressionen that most of the about 80 participants liked the course and tended to believe me that Theoretical Computer Science can be useful and interesting and may not necessarily be difficult to understand. But I must admit that the examinations on the matter indicated rather that students did not learn more or understand better than in other theoretical courses.

Some Outcome

The course worked well as the fundament for the BIZARR project. During the course, most of the presented methods were implemented experimentally. And after the course, the project participants decided to develop a computerized studio for syntactic films, named *BIZARR goes to Hollywood*. The key idea is that a derivation (in any of the syntactic methods for picture generation) does not only generate a single picture, but can be considered as a sequence of picture descriptions specifying a sequence of pictures. The main module of the system is a scene manager where one can put together various derivations in various methods including lights and movable cameras. The film shows the overlay of the sequences of pictures given by the derivations as seen by one of the cameras. There is a wireframe output and raytracer output (which needs days and weeks for few seconds of film). So far, the implemented methods are L-systems with turtle interpretation, iterated function systems, map-L-system, cellular automata and collage grammars. The system is implemented in C^{++} and runs on SUN workstations under UNIX/X-Windows. It will be presented at the annual international computer fair CEBIT in Hannover on March 8-15, 1995.

References

[1] M. Barnsley: Fractals Everywhere, Academic Press, Boston 1988.

[2] E.-F. Codd: Cellular Automata, Academic Press, 1968.

[3] J. Dassow: Graph-Thoeretic Properties and Chain Code Picture Languages, J. Inf. Process Cybern. EIK 25,8/9, 423-433, 1989.

[4] F. Drewes, A. Habel, H.-J. Kreowski, S. Taubenberger: Generating Self-affine Fractals by Collage Grammars. Developments in Languages Theory, World Scientific, Singapore, 278-289, 1994.

[5] J. Feder: Languages of encoded line patterns, Information and Control 13, 1968.

[6] H. Freeman: Computer processing of line-drawing images, Computer Surveys 6, 57-97, 1974.

[7] K.S. Fu: Syntactic Methods in Pattern Recognition, Academic Press, New York 1974.

[8] J. Gips: Shape Grammars and Their Uses, Articical Perception, Shape Generation and Computer Aesthetics, Birkhäuser Verlag, Basel/Stuttgart 1975.

[9] A. Habel: Hyperedge-replacement: grammars and languages, Lecture Notes in Computer Science 643, 1992.

[10] A. Habel, H.-J. Kreowski, S. Taubenberger: Collages and Patterns Generated by Hyperedge Replacement, Languages of Design 1, 125-145, 1993.

[11] B.B. Mandelbrot: The Fractal Geometry of Nature, W.H. Freeman and Company, New York 1983.

[12] H.A. Maurer, G. Rozenberg, E. Welzl: Using String Languages to Describe Picture Languages, Information and Contral 54, 155-185, 1982.

[13] yy H.-O. Peitgen, H. Jürgens, D. Saupe: Chaos and Fractals, Springer-Verlag, New York 1993.

[14] J.L. Pfaltz, A. Rosenfeld: Web grammars, in: Proc. of International Joint Conference on Artificial Intelligence, 1969.

[15] P. Prusinkiewicz, A. Lindenmayer: The Algorithmic Beauty of Plants, Springer, New York 1990.

[16] G. Stiny: Pictorial and Formal Aspects of Shape and Shape Grammars, Birkhäuser Verlag, Basel/Stuttgart 1975.

[17] T. Toffoli, N. Margolus: Cellular Automata Machines. MIT-Press, 1987.