Lecture Notes in Computer Science

Edited by G. Goos and J. Hartmanis

411

M. Nagl (Ed.)

Graph-Theoretic Concepts in Computer Science

15th International Workshop WG '89
Castle Rolduc, The Netherlands, June 14–16, 1989
Proceedings

Springer-Verlag
Berlin Heidelberg New York London Paris Tokyo Hong Kong

Editor
Manfred Nagl
Lehrstuhl für Informatik III
RWTH Aachen, Ahornstraße 55
D-5100 Aachen, FRG

CR Subject Classification (1987): B.6–7, C.1.2–3, C.2.1, C.2.4, D.2.2, D.3.3, E.1, F.1.2, F.2, G.2.1–2, H.2.1, H.3.2

ISBN 3-540-52292-1 Springer-Verlag Berlin Heidelberg New York
ISBN 0-387-52292-1 Springer-Verlag New York Berlin Heidelberg

Printing and binding: Druckhaus Beltz, Hemsbach/Bergstr.
2145/3140-543210 – Printed on acid-free paper

Preface

The 15th International Workshop on Graph-Theoretic Concepts in Computer Science (WG '89) was held at Castle Rolduc, Kerkrade, the Netherlands, near Aachen, Federal Republic of Germany, from June 14 to 16, 1989. It was organized by Lehrstuhl für Informatik III of Aachen University of Technology.

The successful tradition of the WG workshops which take place in alternate places in Central Europe every year is reflected by the list of predecessor workshops and their organizers:

Berlin	1975	U. Pape
Göttingen	1976	H. Noltemeier
Linz	1977	J. Mühlbacher
Castle Feuerstein near Erlangen	1978	M. Nagl & H. J. Schneider
Berlin	1979	U. Pape
Bad Honnef	1980	H. Noltemeier
Linz	1981	J. Mühlbacher
Neunkirchen near Erlangen	1982	H. J. Schneider & H. Göttler
Haus Ohrbeck near Osnabrück	1983	M. Nagl & J. Perl
Berlin	1984	U. Pape
Castle Schwanenberg near Würzburg	1985	H. Noltemeier
Monastery Bernried near München	1986	G. Tinhofer & G. Schmidt
Castle Banz near Bamberg	1987	H. Göttler & H. J. Schneider
Amsterdam	1988	J. van Leeuwen

The program committee of the WG '89 Workshop consisted of:

H. Bunke,	University of Berne, Switzerland
B. Courcelle,	University of Bordeaux, France
J. van Leeuwen,	University of Utrecht, The Netherlands
M. Nagl,	Aachen University of Technology, West Germany
H. Noltemeier,	University of Würzburg, West Germany
G. Schmidt,	University of German Forces, West Germany
H. J. Schneider,	University of Erlangen, West Germany

In the call for papers contributions were solicited describing original results in the study and application of graph-theoretic concepts in various fields of computer science. From 48 submissions the program committee selected 27 for presentation. The selection reflects several current research directions that are representative for the topic of the workshop. However, certainly not all aspects could be covered by a three-day workshop.

The present volume contains the revised versions of nearly all the papers presented at the workshop. The revisions are based on the comments and suggestions received by the authors from referees and/or participants of the workshop. Several papers are in the form of preliminary reports on ongoing research. It is expected that more elaborate versions of

these papers will eventually appear in scientific journals. The members of the program committee hope that this volume gives a good impression of activities in the topic of graph-theoretic concepts in computer science.

The workshop was attended by 50 participants from 10 countries (Canada, France, Israel, Italy, Poland, Spain, Switzerland, The Netherlands, United States of America, Federal Republic of Germany). The success of the workshop is due to the activeness of participants contributing to the presentations and discussions, and to the work done by referees and especially by the members of the program committee.

The editor is grateful for financial support from the Aachen University of Technology. He would like to thank the Rector, Prof. Dr. Habetha, for this valuable help. Furthermore, sponsorship by the European Association for Theoretical Computer Science (EATCS) and the Gesellschaft für Informatik (GI) is thankfully acknowledged. The friendly atmosphere of this workshop was in great part due to the attractive workshop location in Castle Rolduc. We would like to thank the personnel of this education center who looked after us so well. Last but not least the editor would like to say thanks to the members of the organization committee consisting of M. v.d. Beeck, U. Cordts, A. Fleck, M. Hirsch, Th. Janning, N. Kiesel, M. Lischewski, U. Schleef, A. Schürr, C. Weigmann, B. Westfechtel and some students of computer science at Aachen University of Technology.

Aachen, October 1989 *M. Nagl*

The following persons helped as referees to select the papers for presentation:

A. Arnold
L. Bayer
R. Berghammer
J. Bond
H. Bunke
B. Courcelle
S. Dulucq
J. Dvorak
J. Ebert
T. Glanser
E. Gmür
D. Janssens
X.-Y. Jiang
P. Kropf
M. Kunde
M. Ley
Th. Ludwig
K. Mehlhorn
M. Nagl
H. Noltemeier
W. Oberschelp
J. G. Penaud
J. Perl
G. Reich
W. Ries
A. Scheuing
I. Schiermeyer
J. Schmid
G. Schmidt
L. Schmitz
H. J. Schneider
J. Schreiber
A. Schürr
W. Thomas
G. Tinhofer
J. van Leeuwen
W. Vogler
F. Wagner
B. Walter
B. Westfechtel
P. Widmayer

Contents

Graph Embeddings

Algorithmic Graph Theory

Graphs and Data Structures

Graphs and Computational Geometry

Annexes

Foreword
The Aim of the WG Workshops

The aim of this workshop series is to contribute to integration in computer science. This striving for integration is achieved by applying graph–theoretic concepts. Thereby integration appears in two ways. First, graph–theoretic concepts are applied to various fields of specialization in computer science and thus commonalities between the fields are detected. Second, the workshops aims to combine theoretical aspects with practice and applications; this is achieved either by applying theoretical concepts to practice or by taking up problems from practice and trying to solve them theoretically.

This workshop series is thus a rarity in computer science, as it is neither purely theoretical nor practical or oriented to applications. It is also a vertical cut through the different fields of computer science in which graphs and graph–theoretic concepts can be applied. On the other hand the workshop has a tradition: this volume collects the papers of its fifteenth occurrence. That alone is remarkable in a field like computer science where new topics appear and disappear quite rapidly.

Looking into this volume and its predecessors (see the list on the last page of this volume) one can see that there are many disciplines of computer science where graph–theoretic concepts can be applied and have been applied. The volumes present applications in data structures, databases, information retrieval, software engineering, design of software systems, compiler construction, construction of intelligent software tools, design of inherently parallel hardware architectures as in VLSI layout, distributed systems, process control, concurrency, communication, graphics, computational geometry, computer–aided design, computer vision, and design of algorithms and investigation of complexity, for example in operations research and optimization and in pattern recognition. In some cases applications outside computer science in such fields as chemistry, biology, geology have also been presented.

Graphs are used in these disciplines for different purposes. They are used as an appropriate level of abstract representation at which certain problems can be studied and at which certain details of representation and implementation are suppressed. Thus for a given problem a suitable graph representation has to be found. As such a problem belongs to a certain application area or to a certain class of problems, the graph is a member of a certain class of graphs.

For this class of graphs certain structural properties and/or the complexity of corresponding algorithms can be investigated. An example of a structural property of graphs and a corresponding class of graphs is given by the bounded degree graphs. An example of a particular algorithm is the Fairley algorithm. In such cases a class of graphs is given by static properties, or by algorithms with a certain complexity. We call this branch of graph–theoretic concepts *structural graph theory*.

In many application fields the graphs are not static but rather change dynamically. In these applications not only has a suitable class of graphs to be detected but also the dynamic behavior of changes has to be formally specified. Changes may belong to structural properties (for example specifying a complex insertion in a database) or they may belong to some state of execution (such as firing a transition, or reevaluating attributes of entities). In the first case graph rewriting systems (graph grammars) are a suitable formal method to describe changes, and in the second case Petri nets and attributed grammars are well known methods. Let us call this branch of graph-theoretic concepts *conceptual graph modeling*.

Solving practical problems which are regarded as problems on graphs is dramatically facilitated if suitable means of implementation are available. In our case these means should help in implementing graph classes, algorithms on graphs, changes on graphs and so on. This help may be restricted to implementing a special graph class or it may extend to implementing arbitrary classes. In this branch we find hints, mechanisms, methods, notions, tools, or basic software systems which facilitate the implementation of a general graph solution or a special solution based on graphs. Let us call this branch of graph-theoretic concepts *graph implementation mechanisms*. In this category we find graph programming languages, special data structures for graphs, special nonstandard database systems for graphs and so on.

Summing up, the importance of graphs is that we can study structural properties, special algorithms, structural or value changes, and implementation mechanisms rather separately from the problem or the application area they originate from. This level of abstraction makes it easier to get structural results independent of application or to characterize different specific occurrences of a more general problem. To give an example, phenomena like parallelism or concurrency can be adequately studied and characterized independently of applications. This formulation is abstract relative to its representation in an application area and independent of its applications in the sense that the same phenomenon is detected in different application areas. Thus the formulation helps to detect similarities of problems and solutions, or to detect uniform mechanisms which solve classes of problems rather than single problems.

The importance of graphs is growing and we have accumulated a lot of knowledge on how to handle graphs in the last fifteen years in structural graph theory, in conceptual graph modeling and in graph implementation mechanisms. On the other hand it is increasingly obvious that specialized structures like trees, even with attributes, can only serve as an adequate underlying model for limited classes of problems. The growing importance of graphs is stressed in many conferences and journals. In this sense, this workshop is quite modern despite its long tradition.

M. Nagl