Computer-Aided Program Development

Head:

Prof. Dr. Rudolf Berghammer
Tel.: (0431) 560486
E-Mail: rub@informatik.uni-kiel.d400.de

Location:

Preusserstraße 1–9
D–24105 Kiel

Secretary:

Claudia Herbers
Tel.: (0431) 560427
Fax: (0431) 566143
E-Mail: ch@informatik.uni-kiel.d400.de

Scientific Staff:

Dipl.-Inform. Ralf Behnke

Programmer:

Dipl.-Ing. (FH) Peter Schneider

Research Report

The central topics of research of the group are the foundations of programming languages and programming methodology and the use of mechanical support. This is done in intensive collaboration with many computer scientists all over the world. A formal collaboration is in each case given by the German-Brazilian project HORACE and the multi-national project ARTSWARE. Besides many groups in Germany, close non-formalized connections exist especially with computer scientists in the Netherlands (Eindhoven and Utrecht), Sweden (Chalmers University), Great Britain (Oxford), and Canada (Quebec).

Algebraic Calculi for Deriving Programs

In many engineering disciplines calculation is an indispensable tool in the construction process. Since some years computer scientists try to carry over this idea to programming and seek for algebraic calculi for developing computer programs. There are many design calculi for program construction. Whereas most of them involve a clear separation between the specification and implementation language, A. Tarski's component-free relational algebra offers a very elegant vehicle for both specifying problems and transforming such specifications "calculationally" into executable programs.

In the last two years one main point of interest of our group was the investigation of relational methods in computer science, especially in formal problem specification and algorithm derivation (and relational semantics, see below). In the paper [4] we concentrate mainly on the relational calculation of "classical" graph-theoretic algorithms like closure algorithms or algorithms for computing kernels. But relations can also be used in other problem domains. In [10] we present a method for specifying and implementing algorithms for the analysis of Petri nets which is formally grounded in relational algebra. Further examples can be found in [9]. In this article we concentrate mainly on prototyping relational specifications using certain higher-order functions on relations.

The power of the component-free relational approach can be increased by using sequential algebra, a slight generalization of relational algebra introduced by B. von Karger and C.A.R. Hoare in 1994/95. Thus, it is able to extend the scope of relational methods to reactive systems, where the semantics is not an input/output relation but the behaviour between initiation and termination is also important. In [3, 6] sequential algebra is successfully applied to the development of reactive systems from non-algorithmic temporal specifications by integrating both the process language CSP and linear temporal logic into this algebraic structure. A further example, a number-theoretic problem which is difficult to treat in the classical relational framework, can be found in [1].

One investigated aspect in the calculational derivation of programs from relational resp. sequential problem specifications was the introduction of recursion and loops using transformational techniques. Here a number of schemes for fixed point calculations have been regarded and formally verified. Another specific aspect was the optimization of the derived programs using certain algebraic constructions.

Semantics of Programming Languages

A precise description of a programming language is a prerequisite for its use in formal program development. Hence, semantics of programming languages and its application to programming has also been a research topic during the last two years. Here we considered several kinds of languages as well as semantics descriptions.

Firstly, we investigated sequential imperative languages and the axiomatic predicate transformer method. Using weak second-order logic, in [8] a purely syntactical but nevertheless handy description of E.W. Dijkstra's wp-function is presented. This formalization allows to prove a normal form theorem of wp from which many important properties (e.g., that wp behaves well on some of the logical operators) can be derived easily.

The papers [5] and [14] deal with denotational relational semantics of sequential languages. Using relations rather than functions in denotational semantics

one can avoid the complexity introduced by artificial bottom elements denoting undefinedness. As both artices demonstrate, relations are also natural candidates for modelling non-determinism. But perhaps the main reason for using relations is again that they can be calculated with so well.

In [5] various kinds of non-determinism are explored for sequential functional languages whereas [14] concentrates on a new approach to demonic nondeterminism of imperative languages and robust correctness which is similar to C.A.R. Hoare's chaos semantics. In the latter article also the correctness of unfold/fold for demonic non-determinism is shown and relationships to E.W. Dijkstra's wp-calculus and C. Morgan's specification statement are investigated.

For the semantical treatment of communicating sequential processes using sequential algebra, see the above section on algebraic calculi for deriving programs.

Computer-Aided program Development

This was the third main research topic of the group in the last two years. A workshop dedicated to it was held; see [11]. We investigated and used several systems, including the Larch Prover, PAMELA, PVS, RAP, and TIP. But mainly, we concentrated on two relation-algebraic systems, called RELVIEW and RALF.

RELVIEW is a relation-based computer system for visualization, analysis and manipulation of discrete structures. Written in the C programming language, it runs under X windows and makes full use of the graphical user interface. The first version of RELVIEW was written at the University of the German Forces Munich. In the last two years, the system was redesigned and extended by our group and now Kiel University is responsible for its further development. Currently RELVIEW is used in about 30 installations all over the world. At Kiel University, it is applied in education (i.e., in lectures and seminars, see [7]) as well as in science. The article [10] demonstrates an application in Petri net theory. A further application in relational semantics can be found in [11].

The RALF system is a relation-algebraic formula manipulation system and proof checker. It was also written at the University of the German Forces Munich. In [2] one finds the fundamental requirements and a description of the system's general structure. In the meantime, the system's language also comprises the operations of sequential algebra. Present extensions concern the incorporation of quantifiers, second-order constructs (like least upper and greatest lower bounds), and fixed point operators. They are done in close connection with our group (cf. the two visits of Claudia Hattensperger). During the last years, at Kiel University RALF was used for education only; see [7]. But scientific use is planned for the future, too.

Functional Programming and Reusability

A main task in software engineering is programming-in-the-large, i.e., the arrangement of modules into a system architecture. In conventional languages like C, Modula-2 or Ada, a system architecture is frozen. But in the language Standard ML, which module language supports a functional style for programming-in-the-large, so-called functors enable a flexible architecture to be designed.

Functors in Standard ML can be used not only for linking together structures but also for parameterizing algorithms to obtain algorithm schemas in a very comfortable way. These schemas (like depth-first-search, the Floyd-Warshall algorithm on semirings, and the matroid greedy algorithm) then can be reused to solve a whole class of related problems. How to parameterize algorithms using the Standard ML module language is demonstrated in [12, 13].

Bibliography

[1] Behnke R. and von Karger B. Spezifikation und Algorithmenentwicklung im sequentiellen Kalkül. In T. Margaria, editor, *Programmiersprachen und Grundlagen der Programmierung (proceedings)*, pages 85–90. Fak. für Mathematik und Informatik, MIP-9519, Universität Passau, 1995.

[2] Berghammer R. and Hattensperger C. Computer-aided manipulation of relational expressions and formulae using RALF. In Buth B. and Berghammer R., editors, *Systems for computer-aided specification, development and verification (proceedings)*, pages 62–78. Inst. für Informatik und Prakt. Mathematik, Report 9416, Universität Kiel, 1994.

[3] Berghammer R. and von Karger B. Formal derivation of CSP programs from temporal specifications. In Möller B., editor, *Mathematics of Program Construction (proceedings)*, volume 947 of *Lecture Notes in Computer Science*, pages 180–196. Springer-Verlag, 1995.

[4] Berghammer R. and von Karger B. Algorithms from relational specifications. In Schmidt G. Brink C. and Albrecht R., editors, *Relational Methods in Computer Science*, Supplementary Volume of Computing. Springer-Verlag, (to appear 1996).

[5] Berghammer R. and von Karger B. Relational semantics of functional programs. In Schmidt G. Brink C. and Albrecht R., editors, *Relational Methods in Computer Science*, Supplementary Volume of Computing. Springer-Verlag, (to appear 1996).

[6] Berghammer R. and von Karger B. Towards a design calculus for CSP. *Science of Computer Programming*, (to appear May 1996).

[7] Berghammer R., Buth B. and Buth K.-H. Rechnergestützte Programmentwicklung und -verifikation in der Lehre: Konzepte und Erfahrungen. In Spillner A. and Breymann U., editors, *Software Engineering im Unterricht der Hochschulen (proceedings)*, pages 11–21. Teubner-Verlag, 1995.

[8] Berghammer R., Elbl B. and Schmerl U. Formalizing Dijkstra's predicate transformer wp in weak second-order logic. *Theoretical Computer Science B*, 146:185–197, 1995.

[9] Berghammer R., Gritzner T. and Schmidt G. Prototyping relational specifications using higher-order objects. In Heering J., Meinke K., Möller B. and Nipkow T., editors, *Higher Order Algebra, Logic and Term Rewriting (proceedings)*, volume 816 of *Lecture Notes in Computer Science*, pages 56–75. Springer-Verlag, 1994.

[10] Berghammer R., von Karger B. and Ulke C. Relation-algebraic analysis of Petri nets with RELVIEW. In T. Margaria and B. Steffen, editors, *Tools and Applications for the construction and analysis of systems (proceedings)*, volume 1055 of *Lecture Notes in Computer Science*, pages 49–69. Springer-Verlag, 1996.

[11] Buth B. and Berghammer R. Systems for computer-aided specification, development and verification (proceedings). Technical Report 9416, Inst. für Informatik und Prakt. Mathematik, Universität Kiel, 1994.

[12] Berghammer R. Graphdurchlauf mit Vererbung und Synthese: Eine Fallstudie über die Parameterisierung von Programmen mittels ML-Funktoren. In Simon F., editor, *Deklarative Programmierung und Spezifikation (proceedings)*, pages 10–16. Inst. für Informatik und Prakt. Mathematik, Report 9412, Universität Kiel, 1994.

[13] Berghammer R. Wiederverwendung in der Programmiersprache ML mittels Signaturen, Strukturen und Funktoren. In Snelting G., editor, *Softwaretechnik '95 (proceedings)*, Sonderband Softwaretechnik-Trends, pages 76–85, 1995.

[14] Gritzner T. and Berghammer R. A relation algebraic model of robust correctness. *Theoretical Computer Science B*, 160, (to appear June 1996).

Visiting Scientists

– Dipl.-Math. Claudia Hattensperger, University of the German Forces Munich,
June 13 – June 10, 1994 and January 29 – February 4, 1996

- Prof. Dr. Bernhard Möller, Augsburg University,
  May 26 – May 29, 1994
- Prof. Dr. Walter Dosch, Augsburg University,
  December 8 – December 10, 1994
- Prof. Dr. Jules Desharnais, Laval University (Quebec, Canada),
  November 20 – November 26, 1995

Talks and Presentations

- Rudolf Berghammer, Graphdurchlauf mit Vererbung und Synthese – Eine Fallstudie über die Parameterisierung von Programmen mittels ML-Funktoren, Workshop "Deklarative Programmierung und Spezifikation", Bad Honneff, May 1994
- Rudolf Berghammer, Ein relationenalgebraisches Modell für robuste Korrektheit, Workshop "Domains", TU Darmstadt, June 1994
- Rudolf Berghammer, Rechnergestützte Manipulation relationaler Terme und Formeln mittels RALF, Workshop "Programmsysteme für rechnergestützte Programmentwicklung und -verifikation", Universität Kiel, July 1994
- Rudolf Berghammer, Beobachtungsräume und Sequentielle Algebren mit Anwendungen, Kolloquiumsvortrag, Institut für Mathematik, Universität Augsburg, February 1995
- Rudolf Berghammer, Formale Entwicklung von CSP-Programmen aus temporalen Spezifikationen, Workshop "Deklarative Konzepte für Sprachen und Rechner", Bad Honneff, May 1995
- Rudolf Berghammer, Wiederverwendung in der Programmiersprache ML mittels Signaturen, Strukturen und Funktoren, Softwaretechnik '95, Technische Universität Braunschweig, October 1995
- Ralf Behnke, Spezifikation und Algorithmenentwicklung im sequentiellen Kalkül, Kolloquium "Programmiersprachen und Grundlagen der Programmierung, Alt Reichenau (near Passau), October 1995
- Rudolf Berghammer, Programmieren von Algorithmenschemata in ML, Kolloquiumsvortrag, Fachbereich Mathematik (Arbeitskreis Informatik), Universität Gießen, February 1996
- Rudolf Berghammer, Relation-algebraic analysis of Petri nets with RELVIEW. Second Workshop "Tools and Applications for the Construction and Analysis of Systems (TACAS '96)", Universität Passau, March 1996

Other Avtivities

Prof. Rudolf Berghammer was a member of the program committees of the Third Maghrebian Conference on Software Engineering and Artificial Intelligence, Ra-

bat (Marocco), April 1994 and of the Fourth Maghrebian Conference on Software Engineering and Artificial Intelligence, Zeralda/Algiers (Algeria), April 1996. Furthermore, he is an observer of the IFIP Working Group 2.1 "Algorithmic Languages and Calculi".

Workshop Programmsysteme für rechnergestützte Programmentwicklung und -verifikation

Kiel, July 4 – July 5, 1994
Organizer: Prof. Dr. R. Berghammer

During the last years, several systems for computer-aided specification, development, and verification have been constructed by various groups with different background in Germany. Therefore, this workshop dedicated to this special field was held (for the proceedings, see [11]).. It was attended by approx. 40 participants from eight german universities, industry and even the Ministry of Economics. The workshop's intention was to bring together developers and users of software tools for program development aiming at an improved understanding of the systems and their backgrounds, a better coordination of the mutual benefits and needs, the utilization of experiences made, and closer cooperation for the future. The main part of the workshop consisted of a series of talks presenting the tools under various aspects. This also included some teaching experiences. Besides the talks the participants had the possibility to use and test the systems in a computer laboratory containing ten modern workstations. The last session of the workshop was devoted to a panel discussion, opened by two ten minutes talks. The first was given by a student of computer science who reported experiences gained while using the systems in a seminar; in the second talk the ideas, wishes and requirements of industry were presented.

It is planned that the 1994 Kiel workshop is the beginning of a biannual series of workshops on systems for computer-aided specification, development, and verification. The organizer of the next workshop from May 3 – 5, 1996, is Bremen University.