Second Annual Progress Report of the GETGRATS Research Network Contract ERB FMRXCT960061

Andrea Corradini
Dipartimento di Informatica, Pisa
andrea@di.unipi.it

November 2, 1998

Contents

1 Progress of the Network during the Second year

1.1 Comparison of accomplished work and objectives in the Project Programme

The research and organizational activities actually accomplished during the second year of the Network meet the objectives, schedule and milestones set out in the Project Programme (Annex I of the contract).

Significant progress has been made in the research activity in all the seven Focus Areas in which the research themes of the project are divided, as described in the next subsection and in the Appendix (Section 4).

Concerning schedule and milestones, the first GETGRATS Workshop was held as planned in the First Year Report in the period October 8-10, 1997, in Bordeaux; and the "European School on Graph Transformation" (ESoGT) was held in Bremen in the period March 2-7, 1998, and it hosted the Mid-Term Review of the Project.

During the Mid-Term Review the coordinator, in agreement with all the partners, asked the EC representatives to consider the possibility of extending the planned duration of the project for 12 months, till August 31, 2000, bringing the total duration of the project to 48 months. This request has been accepted by the EC. As a consequence the schedule of the various events has been adjusted in the following way:

- The final workshop of the network is now planned in Berlin in March 2000, in connection with the international federated conference ETAPS 2000.
- The five remaining Focus Area Meetings will be organized during the third and fourth year of the project.

1.2 Scientific Highlights

A quite detailed description of the results of the scientific research carried on during the second year of the project is presented in the Appendix (Section 4).

Significant contributions have been made to research themes of all the seven Fous Areas. Among them we recall the study of Pullback Rewriting as a decomposition mechanism; the extension of the single-pushout (SPO) approach to the algebraic transformation of partial many-sorted unary algebras; the use of (d)gs-categories for a categorical presentation of the double-pushout (DPO) approach; the characterization of context free languages using Monadic Second Order (MSO) Logic and of MSO tree transduction using attribute grammars; results about undecidability of termination of DPO graph rewriting; the introduction of the concept of tree-based picture generation; a theorem about the MS definability of the decomposition of certain classes of graphs; the study of the complexity of optimisation problems expressible in MSO logic; the analysis of variation of the DPO approach using non-injective morphisms; resuts about the generative power of context-free collage and chain-code grammars; the proof that bisimilarity for equational rooted graphs is decidable; the definition of the rewriting of term graphs modulo bisimilarity; the introduction of the concept of graph interpreted temporal logic; the use of the loose semantics of GTS for the compositional verification of specifications using the notion of view; the development of process semantics for embedding-based GTS; the study of the foundations of Local Action Systems (LAS); the categorical description of possibly cyclic term graph rewriting; the study of rational term rewriting; the analysis of the delimitation of the power of local computations for Graph Relabelling Systems; the introduction of Parallel High-Level Replacement systems; the generalization of the theory of parallelism of graph transformation sequences for non-injective rules; the extension of the notion of transformation unit with the inclusion of control mechanisms; the definition of a module concept for embedding-based GTS; the study of spatial and temporal refinement relations betwen typed GTS; the application of GTS to software architecture styles; the definition of an unfolding semantics for graph transformation in the DPO approach; and the development of a library for colimit computations in various categories based on an efficient algorithm for colimit construction

1.3 Principal Networking and Coordination Activities

Here is a short description of the main events organized within the network during the second year:

1. The First GETGRATS Workshop was held last October 8-10, 1997, in Bordeaux, organized by LaBRI. About 32 researchers from all sites of the network took part to the workshop, where 24 scientific presentations where given. The abstracts of the presentations have been collected in a volume which has been distributed to the participants.

2. The "European School on Graph Transformation" (ESoGT) was held in Bremen last March 2-7, 1998. This was intended to be the main training event of the network, and has been very successful. There were 76 participants, including the lecturers. Almost all lectures were held by members of the network. A copy of both the lecture notes and the first volume of the "Handbook of Graph Grammars and Computing by Graph Transformation" were distributed to all participants.

3. During the ESoGT the Mid-Term Review of the network took place. The judgements of both the EC representatives (Christiane Bernard

and Graham Wilkie) and of the external referee (Prof. José Manuel E. Valença) were very positive.

4. A tutorial with the title "Introduction to Graph Transformation" was held during the conference ETAPS'98 in Lisbon, on March 29, 1998. The tutorial was organized by members of the Berlin and Pisa sites of the network.

5. The coordinator and a network fellow funded by the project took part to the TMR conference in Graz, May 6-9, 1998. They presented the main aspects of the network in Poster Sessions and in brief presentations to the audience. They also participate actively to some of the Workshops which were organized during the conference.

1.4 Progress with the Appointment of Young Researchers

During the first half of the network duration (i.e., from September 1, 1996, to August 31, 1998) about 25 grants have been assigned, financing a total of about 134 months (71.5 months during the second year), out of a total of 234 for the whole duration of the project. This means that the appointment of young researchers is proceeding in line with the expectations.

The next Call for Application will be published during November 1998, just before the International Workshop on Theory and Applications of Graph Transformation, which will be held in Paderborn (Germany) next November 16-20, 1998. GETGRATS is a co-sponsor of this workshop.

1.5 Difficulties

No significant difficulties arose during the second year of the project. The difficulties mentioned in the First Year Progress Report were completely solved.

2 Information on each Participant

The following pages are copies of the form "Information on each Participant" filled in by each participant.

3 Joint Publications

Here is the list of joint publications resulting from the joint activities of the network. The participants involved are indicated in bold at the end of each bibliographic item, using the following achronyms:

N.	Achronym	Site	C.	Scienties-in-Charge
1	UNIPISA	Dipartimento di Informatica, Pisa	IT	U. Montanari
2	UIA	University of Antwerp	BE	D. Janssens
3	TUB	Technische Universität Berlin	DE	H. Ehrig
4	LaBRI	Laboratorie Bordelais de Recherche en Informatique	FR	M. Bauderon
5	UNIBREMEN	Universität Bremen	DE	H.J. Kreowski
6	RUL	University of Leiden	NL	G. Rozenberg
7	UNIROMA1	Università di Roma La Sapienza	IT	F. Parisi Presicce

3.1 Joint papers published or accepted for publication on a Journal

- Frank Drewes and Joost Engelfriet. Decidability of the finiteness of ranges of tree transductions. *Information and Computation*, 1998. Accepted, to appear. Also as Report 9/96, Univ. Bremen, 1996. **UNIBREMEN, RUL**.

- A. Corradini and F. Gadducci. An Algebraic Presentation of Term Graphs, via GS-Monoidal Categories. *Applied Categorical Structures*, 1998. To appear. **TUB, UNIPISA**

- R. Heckel, H. Ehrig, U. Wolter, and A. Corradini. Double-pullback transitions and coalgebraic loose semantics for graph transformation systems. *Applied Categorical Structures*, 1998. To appear. **TUB, UNIPISA**

3.2 Joint papers presented at Conferences

- A. Corradini and F. Gadducci. Rational term rewriting. In M. Nivat, editor, *Proceedings FoSSaCS'98*, volume 1378 of *LNCS*, pages 156–171. Springer Verlag, 1998.**TUB, UNIPISA**

- M Große–Rhode, F. Parisi Presicce, and M Simeoni. Spatial and temporal refinement of typed graph transformation systems. In L. Brim,

J. Gruska, and J. Zlatuška, editors, *Proc. of Mathematical Foundations of Computer Science 1998*, volume 1450 of *LNCS*, pages 553–561. Springer Verlag, 1998. **TUB, UNIROMA1**

- A. Corradini, M. Große-Rhode, and R. Heckel. An algebra of graph derivations using finite (co-)limit double theories. In *Proceedings WADT'98*, 1998. To appear. **TUB, UNIPISA**

- R. Bruni, F. Gadducci, and U. Montanari. Normal forms for partitions and relations. In *Proceedings WADT'98*, 1998. To appear. **TUB, UNIPISA**

3.3 Joint Technical Reports, Abstracts, Papers submitted for publication, Book chapters in preparation

Note: These publications are not counted for the Statistical Summary.

- P. Baldan, A. Corradini, H. Ehrig, M. Loewe, U. Montanari, and F. Rossi. Concurrent semantics of algebraic graph transformations. In G. Rozenberg, editor, *Handbook of Graph Grammars and Computing by Graph Transformation, Vol.3: Concurrency, Parallellism, and Distribution*, volume 3. World Scientific, 1998. In preparation. **TUB, UNIPISA**

- A. Corradini and F. Gadducci. Rewriting on cyclic structures. Technical Report TR-98-05, Dipartimento di Informatica, Pisa, 1998. Extended abstract to appear in Proceedings FICS'98. **TUB, UNIPISA**

- Hélène Jacquet and Renate Klempien-Hinrichs. Node replacement in hypergraphs: Translating NCE rewriting into the pullback approach. In *Preliminary Proc. Sixth International Workshop on Theory and Application of Graph Transformations (TAGT '98)*, 1998. To appear. **LaBRI, UNIBREMEN**

- Reiko Heckel, Berthold Hoffmann, Peter Knirsch, and Sabine Kuske. Simple modules for GRACE. In *Preliminary Proc. Sixth International Workshop on Theory and Application of Graph Transformations (TAGT '98)*, 1998. To appear. **TUB, UNIBREMEN**

- M. Große–Rhode, F. Parisi Presicce, and M. Simeoni. Refinements and modules for typed graph transformation systems. Submitted for publication, 1998. **TUB, UNIROMA1**

- Annegret Habel, Jürgen Müller, and Detlef Plump. Double-pushout graph transformation revisited. In *Preliminary Proc. Sixth Intl. Workshop on Theory and Application of Graph Transformations* (*TAGT '98*), to appear. **UIA, UNIBREMEN**

- A. Corradini and F. Gadducci. Functorial semantics for multialgebras. Submitted for publication, 1998. **TUB, UNIPISA**

4 Appendix - Scientific Contributions

This section describes the main scientific contributions of the research activity carried on during the second year of the GETGRATS project. It is divided in seven subsections, one for each of the Focus Areas into which the research themes of the network are divided (as listed in the Annex I of the contract), followed by a list of references.

4.1 (a) Foundation, Unification, Combination and Comparison

The study of pullback as a rewriting mechanism which can unify several well-known rewriting mechanisms such as Node-Label Controlled embedding or Neighbourhood-Controlled Embedding has been extended to that of pullback rewriting as a decomposition mechanism. Indeed, pullback rewriting can be used in a quite elegant way to rewrite several nodes in parallel and conversely to provide a decomposition of some graphs, generated by parallel rewriting, in a product of simpler graphs for some families of graphs generated used in interconnection networks: grids, Butterflies or Banyans. The paper [Kle98] written during the stay of Renate Klempien-Hinrichs in Bordeaux as a GETGRATS TMR fellow has been accepted for publication at FOSSACS'98. A joint paper with Hélène Jacquet on the description of NCE graph rewriting by the pullback approach and its application to node rewriting in hypergraphs will be presented at the TAGT'98 conference (Paderborn November 98) [JK98]. The technical report [MS96] has been accepted for publication in Theoretical Computer Science.

In [BMRV98] the single-pushout approach to algebraic transformation is extended to the algebraic transformation of partial many-sorted unary algebras. The main result is an algebraic characterization of the single-pushout transformation in the categories of all conformisms, all closed quomorphisms, and all closed-domain closed quomorphisms of unary partial algebras over a given signature, together with a corresponding operational characterization. Another important result is the definition of high-level replacement conditions for amalgamation, whereas the three categories are shown to satisfy all of the high-level replacement conditions for parallelism and amalgamation.

In [CGRH98], the development of an algebra of graph derivations as a descriptive model of graph transformation systems is discussed. For that purpose, a categorical three level approach for the construction of models of computations based on structured transition systems is used. Categorically the algebra of graph derivations can then be characterized as a double category with finite horizontal colimits. One of the main objectives of this paper is to show how algebraic techniques can be used for the development of this formal model, in particular to obtain a clear and well structured theory. Thus it may be seen as a case study in theory design and its support by algebraic development techniques.

The dynamic behavior of rule-based systems can be traditionally determined in two orthogonal ways. Either *operationally*, in the sense that a way of embedding a rule into a state is devised, stating explicitly how the result is built: This is the role played by (the application of) a substitution in term rewriting. Or *inductively*, showing how to build the class of all possible reductions from a set of basic ones: For term rewriting, this is the usual definition of the rewrite relation as the minimal closure of the rewrite rules. As far as *graph transformation* is concerned, the operational view is by far more popular: In [GH98], Gadducci and Heckel lay the basis for the orthogonal view. First, an inductive description for graphs as arrows of a freely generated *dgs-monoidal* category is provided. Then, 2-categorical techniques can be applied, already known for term and term graph rewriting [Pow90, CG97], recasting in this framework the usual description of graph transformation via double-pushout [CMR$^+$97].

In recent times there has been a growing interest towards algebraic structures which are able to express formalisms different from the well-known, tree-like representation of term. Many of the approaches adopted for such description reveal a common, specific interest towards their application in the "distributed and concurrent systems" field, but a complete comparison between them is very difficult because their presentation can be quite different. Paper [BGM98] presents a first step towards a unified view, which is able to recast all those formalisms into a more general one, where they can be easily compared. A general schema is introduced for describing a characteristic normal form for many interesting algebraic formalisms, and it is shown that those normal forms can be thought of as arrows of suitable concrete monoidal categories, whose operations preserve the normal form itself.

It is known that a language is context-free iff it is the set of borders of the trees of a recognizable set, where the border of a (labelled) tree is the word consisting of its leaf labels read from left to right. In [CL98] a generalization of this result in terms of planar graphs of bounded tree-width is given. Here, the border of a planar graph is the word of edge labels of a path which borders a face for some planar embedding. It is shown that a language is context-free iff it is the set of borders of the graphs of a set of (labelled) planar graphs of bounded tree-width which is definable by a formula of monadic second-order logic. In [Lap98] it is shown that for each k, there exists a monadic second-order transduction that associates with every graph of tree-width at most k one of its tree-decompositions of width at most k. Together with a known result by Courcelle it follows that every set of graphs of bounded tree-width is CMSO-definable if and only if it is recognizable.

In [BE98] and [EM98a] a comparison is made between MSO definable tree transductions, i.e., tree transductions that can be specified in monadic second order logic, and classical, implementation-oriented tree transduction formalisms: attribute grammars and functional programming. It is shown in [BE98] that MSO definable tree transductions correspond to "single use" attribute grammars, and in [EM98a] that they correspond to "finite copying" macro tree transducers. These results can be viewed as the generalization to tree translations of Büchi's classical result that MSO definable string languages correspond to finite automata.

Term graph rewriting is concerned with the representation of functional expressions as graphs, and the evaluation of these expressions by rule-based graph transformation. Representing expressions as graphs allows to share common subexpressions, improving the efficiency of term rewriting in space and time. A survey of (acyclic) term graph rewriting is given in [Plu98a], where emphasis is given to the relations between term and term graph rewriting. The paper focusses on soundness of term graph rewriting with respect to term rewriting, on completeness for proving validity of equations and for computing term normal forms, on termination and confluence, and on term graph narrowing. In [Plu98b] it is shown that termination is an undecidable property of graph rewriting systems (in the double-pushout approach). The proof is by a reduction of the Post Correspondence Problem. It is also argued that neither the halting problem for Turing machines nor the termination problem for string rewriting systems can be reduced to the termination problem for graph rewriting in a straightforward way.

In [DK98] an introduction to picture generation by collage grammars is given, including context-free, table-driven, and context-sensitive derivation modes. In [Dre98c] the concept of tree-based picture generation is introduced. This yields a unified framework for the generation of pictures that can be extended to other sorts of objects as well. In the paper it is shown that collage grammars, iterated function systems, chain-code grammars, and 0L-systems with turtle interpretation are equivalent to natural types of tree-based picture generators. A system called TREEBAG which implements the idea of tree-based object generation is presented in [Dre98b, Dre98d] (see also http://www.informatik.uni-bremen.de/˜drewes/treebag).

In [HK98], atom replacement in hypergraphs is introduced which combines and unifies the concepts of node replacement and hyperedge replacement. In [ET98], Ehrig and Taentzer give a short overview on the field of graph transformation and its relevance for computer science. Whereas the theory of graphs is a well-established field in Mathematics with important applications to Computer Science, especially concerning the area of graph algorithms, dynamic changes of graphs, called graph transformation, and the corresponding theory are discussed in [ET98]. Moreover, not only graphs but all kinds of graphical representation using icons, diagrams and visual representations of texts, programs and specifications are considered.

4.2 (b) Classification and Expressive Power

Certain graphs, such as cographs, have a unique hierarchical decomposition. It is natural to ask whether this decomposition can be defined (in a precise sense) by means of formulas of monadic second-order (MS) logic. For cographs this is possible, provided the graph is given with an auxiliary linear order. In [Cou98a], the unique decomposition of connected graphs as defined by Tutte is investigated in this perspective and it is proved that it is definable by MS formulas. This decomposition has two levels: every connected graph is a tree of "2-connected components", and every 2-connected graph is a tree of so-called 3-blocks. The proof uses 2-dags which are certain acyclic orientations of the considered graphs. Thus a unique decomposition theorem for 2-dags is obtained, an MS definability theorem of this decomposition, and, as a corollary, the MS definability of the Tutte decomposition.

By Kuratowski's Theorem, nonplanar graphs can be characterized by an existential MS formula. When the considered graph is planar, the nonvalidity of this formula gives no information on its planar drawings: it is of interest to define by MS formulas the combinatorial map (a finite logical structure) underlying a possible drawing. It is proved in [Cou98b] that a planar map of a linearly ordered planar graph is MS definable.

It is proved in [CMR98] that certain optimisation problems expressible in monadic second-order logic can be solved in polynomial time on certain graphs (like P4-sparse graphs, which generalize cographs) because these graphs have clique-width at most 4, and that a corresponding decomposition can be constructed in polynomial time.

Three variants of the classical double-pushout approach to graph transformation are investigated in [HMP98]. By employing injective matching and/or arbitrary (possibly non-injective) right-hand morphisms in rules, greater expressive power than in the classical approach is obtained. For each of the three variants it is clarified whether the well-known commutativity and parallelism results still hold, and modified results are provided where necessary.

Context-free collage and chain-code grammars are compared with respect to their generative power in [Kus98b]. It is shown that context-free collage grammars based on similarity transformations can be simulated by context-free chain-code grammars, but that, in general, the language classes generated by both types of grammars are incomparable.

In [EM98b] a characterization is given of the class of tree languages which can be generated by context-free hyperedge replacement graph grammars, in terms of macro tree transducers: it is the class of output tree languages of macro tree transducers which are finite copying and special in the parameters (that is, linear and nondeleting).

In [Man98] the concept of cooperation and distribution as known from the area of grammar systems is introduced to graph grammars, more precisely, to hyperedge replacement (HR) grammars. Two different derivation modes are considered, the so-called t-mode and the $=k$-mode. For the t-mode it is shown that the class of hypergraph languages obtained is equal to the class of ET0L HR languages which are a natural generalization of the ET0L languages to hypergraphs.

Attributed context-free hypergraph grammars (acfhg grammars) are introduced in [MV98]. Acfhg grammars and attribute grammars are closely related: an attribute grammar can be used to compute the attribute values of an acfhg grammar. Several known formalisms, such as compatible functions and attributed tree grammars can be embedded into acfhg grammars. An acfhg grammar can be used to define the semantics of a programming language P in such a way that non-context-free constraints of P are already checked in the syntactical phase, i.e., by the underlying graph grammar.

The following papers were described in the First Annual Report:

- [Dre98a], accepted for publication in Theory of Computing Systems;
- [EV98], appeared in J. of Comput. System Science.

4.3 (c) Analysis and Verification Techniques

[Sén98b] completely solves the so-called "equivalence problem for dpda's" (which has been open for over 30 years). This problem can be reformulated as follows: given two equational rooted, deterministic graphs G_1, G_2 , is it possible to decide whether they are bisimilar? This decidability proof can be seen as a *verification* technique for processes whose behavior can be described by an equational graph. As the bisimulation equivalence is usually considered as a useful notion in concurrency theory, this work can also be perceived as a contribution to Focus Area (e).

[Sén97], [Sén98a] extend the above work to the non-deterministic case, while [Sén98c] extends it to dpda's with outputs in an abelian group (equivalently, to graphs with labels belonging to an abelian group).

Two term graphs are said to be *bisimilar* if they represent the same term. In [AKP98], sufficient conditions for—and counterexamples to—confluence, confluence modulo bisimilarity and the Church-Rosser property modulo bisimilarity of term graph rewriting are established. Moreover, the paper addresses rewriting modulo bisimilarity, that is, rewriting of bisimilarity classes of term graphs.

System verification in the broadest sense deals with those semantic properties that can be decided or deduced by analyzing a syntactical description of the system. In [KV98] the semantic properties of redundancy and subsumption are carried over from rule-based systems to high-level replacement systems, which in turn generalize graph and hypergraph grammars. The main results are a characterization of subsumption and a sufficient condition for redundancy, which involves composite productions.

In [GHK97, GHK98] graph-interpreted temporal logic is introduced as a specification tool accompanying rule-based graph transformation. It specializes propositional temporal logic by interpretation on transition systems where the states are graphs, so-called graph transition systems. It is shown that graph transition systems have the same discriminating power as general transition systems. Thus any sound and complete deduction calculus for propositional temporal logic is also sound and complete for the graph-interpreted variant. Properties of the states are expressed by graphical constraints which interpret the propositional variables in the temporal formulas. For any interpretation a graph transition system is constructed which is typical and fully abstract. Typical here means that the constructed system satisfies a temporal formula if and only if the formula is true for all transition systems with this interpretation. By fully abstract it is meant that any two states of the system that cannot be distinguished by graphical constraints are equal. Thus, for a finite set of constraints we end up with a finite transition system which is suitable for model checking.

A loose semantics for graph transformation rules which has been developed recently is used in [Hec98] for the compositional verification of specifications. The main conceptual tool is the notion of *view*, that is, an incomplete specification describing only a certain aspect of the overall system. A view anticipates the (potential) behavior of the complete system by its loose semantics. This ensures that properties of the view are inherited by the complete system. By exploiting this fact, one may verify temporal properties by decomposing a specification into several views, analyzing them separately, and deriving the desired property from properties shown for the views.

4.4 (d) Abstract Semantics

The development of a process semantics for embedding-based GTS has been continued. The aim of this work is to investigate whether it is possible to build a complete semantic theory of GTS starting from the representation of system behaviour by a set of processes. This includes not only concurrency aspects, for which the use of processes is well known, but also abstraction mechanisms, semantic equivalence, system composition, expressiveness etc. It has been shown in earlier work (e.g. [JM96]) that such an approach is possible for ESM systems and actor systems.

In [Jan98] a very general type of GTS is considered, called Local Action Systems or LAS, where the embedding mechanism is based on a set of local actions, which are automorphisms on monoids of edge and node labels. Known models, such as ESM systems and Actor grammars, but also Petri nets and (in a somewhat weaker sense) NLC grammars can be obtained as special cases. A production in a LAS can be viewed as a set of synchronized local actions, equipped with a left-hand side which is used to control where this set of local actions can occur. On the other hand, a production is also a (marked) process, so that the composition of processes can be used to define inductively the process semantics of a system. Moreover, the construction introduced in earlier work about ESM systems for extracting a production from an arbitrary process can be generalized to LAS - an observation that forms the basis for ongoing work on modularity.

On a technical level, the new approach has the advantage that local actions occurring in a process can be grouped together into nonlocal operations on structures rather than on labels, and that these nonlocal operations again form a semiring. This enables one to give algebraic, and thus more reliable and readable, proofs for a number of relevant properties. Further ongoing research is concerned with, among others, the possibility of relaxing the requirements about the algebraic structure on labels and local operations, so that the formalism becomes easier to use for the modeling of concrete situations.

In [CG98c] the authors extend their previous work [CG97] on the categorical description of possibly cyclic term graph rewriting using suitable 2-categories to the rewriting of possibly cyclic term graphs. They show that this presentation is equivalent to the well-accepted operational definition proposed by Barendregt et. al., but for the case of *circular redexes*, for which they propose (and justify formally) a different treatment. The categorical framework is based on *traced gs-monoidal 2-categories*, and it allows to model also automatic garbage collection and rules for sharing/unsharing and folding/unfolding of structures. Furthermore, it can be used for defining various extensions of term graph rewriting, and for relating it to other rewriting formalisms.
The characterization of term graphs as arrows of a gs-monoidal category presented in [CG98a] (and on which the categorical description of [CG98c] is based) has been exploited also in [MR98b] and [MR98a], where graph grammars are used for modeling distributed systems which behave in a synchronized way.

In [CG98b] the rewriting of *rational terms* is considered. Rational terms (i.e., possibly infinite terms with finitely many subterms) can be represented in a finite way via μ-terms, that is, terms over a signature extended with self-instantiation operators. For example, $f^\omega = f(f(f(\ldots)))$ can be represented as $\mu_x.f(x)$. Now, if one reduces a μ-term t to s via a rewriting rule using standard notions of the theory of Term Rewriting Systems, how are the rational terms corresponding to t and to s related? The paper answers to this question in a satisfactory way, resorting to the definition of infinite parallel rewriting proposed in [Cor93]. It also provides a simple, algebraic description of μ-term rewriting through a variation of Meseguer's Rewriting Logic formalism.

Paper [BCM98a], whose contents was described in the First Year Report, has been published in the proceedings of ICALP'98.

4.5 (e) Concurrency Aspects

Various researchers involved in this Focus Area are working on contributions to be included in the third volume of the *Handbook of graph grammars and computing by graph transformation*. In particular, [LMS98] presents a general overview of graph relabelling systems and distributed computations; [BCE$^+$98] reviews the theory of concurrency for the DPO approach to graph rewriting; and [MPR98] describes some applications of graph rewriting to process calculi, constraint programming and coordination.

In the area of Graph Relabelling Systems, more recently we have been interested in the delimitation of the power of local computations on graphs having some initial knowledge. Using new proof techniques we have proved in [GMM98] that even knowing the number of vertices and/or the number of edges we cannot determine by using local computations whether a given graph belongs to a certain graph class like bipartite graphs, graphs having a cut vertex or a cut edge, graphs having a given graph as minor.
In [BM98a] we present some algorithms of enumeration encoded by graph relabelling systems working on $1-$graphs. We give two applications : minor searching and existence of normal forms.
The Kronecker product of a graph G by K_2 is a covering of G. By this way, we study the following general question : *given two connected graphs whose properties are known, what's happening to these properties when we do the Kronecker product of these two graphs?* We give in [BM98b] results concerning graph minors, planarity, cut vertex and cut edge.

In the area of ESM-Systems, the investigation of a process semantics for embedding-based GTS (see, e.g. [JM96]) has been continued and generalized to the framework described in [Jan98]. The approach is different from most treatments of true concurrency in that atomic events are associated to pairs of the causal relation, rather than to nodes of the structures used. Currently the notion of a *marked process* is investigated; this notion extends the traditional notion of a process by including not only an initial structure and a combination of events, but also information about the interaction of a process with its environment. A new subject considered is the introduction of timing information into the process semantics. In cooperation with J. Müller it has been shown that instantaneous cuts (cuts correspondig to a point in time) can be elegantly characterized, and that as a result Timed Petri nets can be modeled without introducing ad-hoc new constructs into the formalism.

In [TK97], the theory of Distributed Graph Transformation is extended to attributed graphs. Among several attribution concepts for graph transformation in the literature the description of data and data operations by partial algebras is chosen leading to a powerful and flexible attribution of graph transformation.

In [Tae97], Parallel High-level Replacement systems are introduced. High-level replacement systems are an axiomatic categorical framework based on double-pushouts in order to unify replacement systems like grammars for different kinds of graphs and relational structures or other types of structures like algebraic specifications. Parallel high-level replacement systems are introduced to formalize parallel rewriting of these high-level structures. On one hand this concept generalizes and extends parallel graph grammars presented so far in the algebraic approach by allowing other structures than graphs, on the other hand the kinds of replacement introduced for high-level replacement systems are extended by different types of parallel replacement which are compared to each other in different parallel replacement theorems. An abstract version of a window-based graph editor and movement of objects in configuration spaces are presented as examples of parallel high-level replacement systems.

In [Mül97b], the theory of parallelism of graph transformation sequences is generalized. If consecutive rule applications are sequentially independent, they may be interchanged by the SHIFT operator. This result is well-known for injective rules. In [Mül97b], it is generalized to non-injective rules. In [Mül97a], a non-categorical characterization of sequential independence is given for the case of non-injective rules. This characterization is not only shown for labeled, directed graphs, but also for the more general notion of graph structures.

4.6 (f) Modularity Aspects

GRACE is a graph transformation language for specifying and programming graph-centered environments that is being developed by researchers from Aachen, Berlin, Bremen, Erlangen, Hildesheim, München, and Paderborn. The basic concept of GRACE is the transformation unit, a structuring principle for composing large graph transformation systems from small pieces. Each transformation unit defines *one* graph transformation (a binary relation on graphs). [HHKK98] extends the notion of a transformation unit to a simple module concept, which defines *a set* of graph transformations.

Each transformation unit contains a control condition that allows to regulate its graph transformation process. In [Kus98a] a selection of control conditions for transformation units is discussed including regular expressions, priorities and the conditions “once” and “as long as possible”. It is shown how some classes of control conditions can be expressed by others without changing the semantics of the corresponding transformation unit. Moreover, three properties of control conditions are studied: minimality, invertibility and continuity.

As a result of the cooperation with M. Simeoni (from Univ. Roma La Sapienza), a module concept for embedding-based GTS is being developed in Antwerp: such a module consists of a set of productions representing its external behaviour- this set of productions is exported - and an implementation of that set of productions. This implementation in turn consists of two sets of productions: on the one hand, *interface* productions providing a translation of graphs over the external alphabet into graphs over an internal alphabet, and on the other hand a *body* of productions over the internal alphabet. The relation between the exported productions and the implementation is based on the constructions of [JM96] and [Jan98], which allow the extraction of productions from processes. The module concept allows one to handle both data and control abstraction. The approach is in the line of the one based Transformation Units developed by Kreowski et al, but it differs from it in that the external effect of a unit is described by a set of productions rather than by a relation over graphs.

The idea of a combined reference model- and view-based specification approach has been proposed recently in the software engineering community. In [EHTE97] and [EEHT97], a specification technique based on graph transformations is presented which supports such a development approach. The use of graphs and graph transformations supports an intuitive understanding and an integration of static and dynamic aspects on a well-defined semantical base. On this background, formal notions of view and view relation are developed and the behavior of views is described by a loose semantics. We define a construction for automatic view integration which assumes that the dependencies between different views are described by a reference model. The views and the reference model are kept consistent manually, which is the task of a model manager. All concepts and results are illustrated at the well-known example of a banking system. In case of more than two views more general scenarios are developed and discussed in [EEHT97].

Continuing the collaboration started during M. Große-Rhode's stay in Rome, spatial and temporal refinement relations between typed graph transformation systems have been introduced in [GPS98b]. In a spatial refinement each rule is refined by an amalgamation of rules while in a temporal refinement it is refined by a sequence of rules: in both cases, the refinement relation supports the modeling of implementation. In [GPS98a], the properties of spatial and temporal refinements are further investigated. Furthermore, they are employed for the development of a module concept for Typed Graph Transformation Systems (TGTS). Finally, as a first step towards an algebra of modules of TGTS, the operations union and composition of modules are introduced.

The description of a software architecture style must include the structural model of the components and their interactions, the laws governing the dynamic changes in the architecture, and the communication pattern. In [HIM98, HIM99] a system is represented as a graph where hyperedges are components and nodes are ports of communication. The construction and dynamic evolution of the style is represented as context-free productions and graph rewriting. To model the evolution of the system it is proposed to use techniques of constraint solving. From this approach one obtains not only an intuitive way to model systems with nice characteristics for the description of dynamic architectures and reconfiguration, but also a unique language to describe the style, model the evolution of the system, and prove properties.

4.7 (g) Morphisms, Transformations and Operations

An unfolding semantics for graph transformation systems in the double-pushout (DPO) approach is proposed in [BCM98c]. Mimicking Winskel's construction for Petri nets, a graph grammar is unfolded into an acyclic branching structure, that is itself a (nondeterministic occurrence) graph grammar describing all the possible behaviours of the original grammar. This unfolding constructions is a first step towards a categorical semantics relating grammars to events structures via a pair of adjoint functors.

Paper [BCM98b], whose contents was described in the First Year Report, has been published in the proceedings of FoSSaCS'98.

For the treatment of structuring mechanisms categorical constructions can be nicely used. One most important concept in category theory is the colimit construction. In applications related to computer science, the colimit separates elements in different components and identifies elements which are connected via their relations. The colimit construction is not only relevant for scheme transformation. Recently, current object oriented modeling methods like the UML, integrate structural and behavioral aspects with extensions of state machine formalisms. Category theory and specially the colimit construction, is suitable for both aspects, providing a unifying formal framework for object oriented design. Other application areas are Structuring and Refinement of formal specifications based on colimits in the category of signatures, Operational Semantics of functional logic programming languages based on the category of jungles, and Algebraic Development Techniques and their extensions (e.g. State charts, Graph Grammars, Algebraic High-Level Petri Nets, Action Nets or Dynamic Abstract Data Types). These applications can be based on colimit computation in two categories, namely signatures and graph signatures. In [Wol98], a new constructive proof of cocompleteness for sets, signatures and graph structures with partial morphisms is given. This proof establishes an (almost) linear complexity of the colimit constructions in all these categories. Moreover, a library for colimit computations in these categories implemented in Eiffel, Java and C++ is provided. The algorithms are directly derived from the constructed proofs mentioned above. Additionally, the application of the colimit library in different areas both in the context of specification languages and graph transformation is outlined.

References

[AKP98] Zena M. Ariola, Jan Willem Klop, and Detlef Plump. Bisimilarity in term graph rewriting. *Information and Computation*, 1998. To appear.

[BCE+98] P. Baldan, A. Corradini, H. Ehrig, M. Loewe, U. Montanari, and F. Rossi. Concurrent semantics of algebraic graph transformations. In G. Rozenberg, editor, *Handbook of Graph Grammars and Computing by Graph Transformation, Vol. III: Concurrency, Parallellism, and Distribution*. World Scientific, 1998. In preparation.

[BCM98a] P. Baldan, A. Corradini, and U. Montanari. Concatenable graph processes: relating processes and derivation traces. In *Proceedings ICALP'98*, volume 1443 of *LNCS*. Springer Verlag, 1998.

[BCM98b] P. Baldan, A. Corradini, and U. Montanari. An event structure semantics for P/T contextual nets: Asymmetric event structures. In M. Nivat, editor, *Proceedings FoSSaCS'98*, volume 1378 of *LNCS*, pages 63–80. Springer Verlag, 1998.

[BCM98c] P. Baldan, A. Corradini, and U. Montanari. An unfolding semantics for DPO graph transformation systems. In *Preliminary Proc. Sixth International Workshop on Theory and Application of Graph Transformations (TAGT '98)*, 1998. To appear.

[BE98] R. Bloem and J. Engelfriet. A comparison of tree transductions defined by monadic second order logic and by attribute grammars. Technical Report 98-02, Leiden University, January 1998.

[BGM98] R. Bruni, F. Gadducci, and U. Montanari. Normal forms for partitions and relations. In *Proceedings WADT'98*, 1998. To appear.

[BM98a] A. Bottreau and Y. Métivier. Minors searching, normal forms of graph rewriting : two applications based on enumeration by graph relabelling. In *Foundations of System Specification and Computation Structures*, pages 110–124. Lecture Notes in Comput. Sci 1378, 1998.

[BM98b] A. Bottreau and Y. Métivier. Some remarks on the Kronecker product of graphs. *Information Processing letters*, 1998. To appear.

[BMRV98] Peter Burmeister, Miquel Monserrat, Francesc Rosselló, and Gabriel Valiente. Algebraic transformation of unary partial algebras II: Single-pushout approach. *Theoretical Computer Science*, 1998. To appear.

[CG97] A. Corradini and F. Gadducci. A 2-categorical presentation of term graph rewriting. In *Proceedings CTCS'97*, volume 1290 of *LNCS*. Springer Verlag, 1997.

[CG98a] A. Corradini and F. Gadducci. An algebraic presentation of term graphs, via gs-monoidal categories. *Applied Categorical Structures*, 1998. To appear.

[CG98b] A. Corradini and F. Gadducci. Rational term rewriting. In M. Nivat, editor, *Proceedings FoSSaCS'98*, volume 1378 of *LNCS*, pages 156–171. Springer Verlag, 1998.

[CG98c] A. Corradini and F. Gadducci. Rewriting on ciclyc structures. Technical Report TR-98-05, Dipartimento di Informatica, Pisa, 1998. Extended abstract to appear in Proceedings FICS'98.

[CGRH98] A. Corradini, M. Große-Rhode, and R. Heckel. An algebra of graph derivations using finite (co-)limit double theories. In *Proceedings WADT'98*, 1998. To appear.

[CL98] B. Courcelle and D. Lapoire. Facial circuits of planar graphs and context-free languages. In L. Brim, J. Gruska, and J. Zlatuška, editors, *Proc. of Mathematical Foundations of Computer Science 1998*, volume 1450 of *LNCS*. Springer Verlag, 1998.

[CMR+97] A. Corradini, U. Montanari, F. Rossi, H. Ehrig, R. Heckel, and M. Löwe. Algebraic Approaches to Graph Transformation I: Basic Concepts and Double Pushout Approach. In G. Rozenberg, editor, *Handbook of Graph Grammars and Computing by Graph Transformation.* World Scientific, 1997.

[CMR98] B. Courcelle, J. Makowsky, U. Rotics. Linear time solvable optimization problems on graphs of bounded clique-width. Technion, Haifa, Israel. To appear in the proceedings of *WG'98.*

[Cor93] A. Corradini. Term rewriting in CT_Σ. In *Proceedings CAAP '93*, volume 668 of *LNCS*, pages 468–484. Springer Verlag, 1993.

[Cou98a] B. Courcelle. The monadic second-order logic of graphs XI: Hierarchical decompositions of connected graphs, October 1997, revised April 1998, to appear in *Theoret. Comput. Sci.*

[Cou98b] B. Courcelle. The monadic second-order logic of graphs XII: Planar graphs and planar maps, May 1998, submitted for publication.

[DK98] Frank Drewes and Hans-Jörg Kreowski. Picture generation by collage grammars. In H. Ehrig, G. Engels, H.-J. Kreowski, and G. Rozenberg, editors, *Handbook of Graph Grammars and Computing by Graph Transformation*, volume 2. World Scientific, 1998. To appear.

[Dre98a] F. Drewes. A characterization of the sets of hypertrees generated by hyperedge-replacement graph grammars. *Theory of Computing Systems*, 1998. To appear.

[Dre98b] Frank Drewes. TREEBAG—a tree-based generator for objects of various types. Report 1/98, Univ. Bremen, 1998.

[Dre98c] Frank Drewes. Tree-based picture generation. Report 7/98, Univ. Bremen, 1998.

[Dre98d] Frank Drewes. TREEBAG – Baum-basierte Generierung und Transformation von Objekten. In *Proc. Informatik '98*, Informatik Aktuell, 1998. In German; to appear.

[EEHT97] G. Engels, H. Ehrig, R. Heckel, and G. Taentzer. A combined reference model- and view-based approach to system specification. *International Journal of Software Engineering and Knowledge Engineering*, 7(4):457–477, 1997.

[EHTE97] G. Engels, R. Heckel, G. Taentzer, and H. Ehrig. A view-oriented approach to system modelling using graph transformation. In LNCS 1301, editor, *Proc. of ESEC/FSE'97, Zürich*, pages 327–343. Springer Verlag, 1997.

[EM98a] J. Engelfriet and S. Maneth. Macro tree transducers, attribute grammars, and MSO definable tree translations. Technical Report 98-09, Leiden University, August 1998.

[EM98b] J. Engelfriet and S. Maneth. Tree languages generated by context-free graph grammars. In *Preliminary Proc. Sixth Intl. Workshop on Theory and Application of Graph Transformations* (*TAGT '98*), to appear.

[ET98] H. Ehrig and G. Taentzer. Graphical representation and graph transformation. Accepted for Computer Surveys., 1998.

[EV98] J. Engelfriet and H. Vogler. The equivalence of bottom-up and top-down tree-to-graph transducers. *J. of Comput. System Sci.*, 56:332–356, 1998.

[GH98] F. Gadducci and R. Heckel. An inductive view of graph transformation. In F. Parisi-Presicce, editor, *Recent Trends in Algebraic*

Development Techniques, volume 1376 of *LNCS*, pages 219–233. Springer Verlag, 1998.

[GHK97] F. Gadducci, R. Heckel, and M. Koch. Combining graph transformations with temporal logic. In *First TMR GETGRATS Workshop*, October 1997, Bordeaux, 1997.

[GHK98] F. Gadducci, R. Heckel, and M. Koch. A fully abstract model for graph-interpreted temporal logic. Submitted for publication, 1998.

[GMM98] E. Godard, Y. Métivier and A. Muscholl. The power of local computations in graphs with initial knowledge. In *Theory and application of graph transformations*. Lecture Notes in Comput. Sci, 1998. to appear.

[GPS98a] M. Große–Rhode, F. Parisi Presicce, and M. Simeoni. Refinements and modules for typed graph transformation systems. Submitted for publication, 1998.

[GPS98b] M. Große–Rhode, F. Parisi Presicce, and M. Simeoni. Spatial and temporal refinement of typed graph transformation systems. In L. Brim, J. Gruska, and J. Zlatuška, editors, *Proc. of Mathematical Foundations of Computer Science 1998*, volume 1450 of *LNCS*, pages 553–561. Springer Verlag, 1998.

[Hec98] R. Heckel. Compositional verification of reactive systems specified by graph transformation. In E. Astesiano, editor, *Proceedings FASE'98*, volume 1382 of *LNCS*, pages 138–153. Springer Verlag, 1998.

[HHKK98] Reiko Heckel, Berthold Hoffmann, Peter Knirsch, and Sabine Kuske. Simple modules for GRACE. In *Preliminary Proc. Sixth International Workshop on Theory and Application of Graph Transformations (TAGT '98)*, 1998. To appear.

[HIM98] D. Hirsch, P. Inverardi, and U. Montanari. Graph grammars and constraint solving for software architecture styles. In *Proc. ISAW'98*, 1998. To appear.

[HIM99] D. Hirsch, P. Inverardi, and U. Montanari. Modeling software architectures and styles with graph grammars and constraint solving. In *Proc. WICSA'99*, 1999. To appear.

[HK98] Annegret Habel and Renate Klempien-Hinrichs. Atom replacement in hypergraphs. In *Preliminary Proc. Sixth International Workshop on Theory and Application of Graph Transformations (TAGT '98)*, 1998. To appear.

[HMP98] Annegret Habel, Jürgen Müller, and Detlef Plump. Double-pushout graph transformation revisited. In *Preliminary Proc. Sixth Intl. Workshop on Theory and Application of Graph Transformations* (*TAGT '98*), to appear.

[Jan98] D. Janssens. Actor Grammars and Local Actions. In G. Rozenberg, editor, *Handbook of Graph Grammars and Computing by Graph Transformation*, volume 3, chapter 3. World Scientific, 1998.

[JK98] Hélène Jacquet and Renate Klempien-Hinrichs. Node replacement in hypergraphs: Translating NCE rewriting into the pullback approach. In *Preliminary Proc. Sixth International Workshop on Theory and Application of Graph Transformations (TAGT '98)*, 1998. To appear.

[JM96] D. Janssens and T. Mens. Abstract semantics for ESM systems. *Fundamenta Informaticae*, 26:315–339, 1996.

[Kle98] Renate Klempien-Hinrichs. Net refinement by pullback rewriting. In M. Nivat, editor, *Proc. Foundations of Software Science and Computation Structures*, volume 1378 of *Lecture Notes in Computer Science*, pages 189–202. Springer, 1998.

[Kus98a] Sabine Kuske. More about control conditions for transformation units. In *Preliminary Proc. Sixth International Workshop on Theory and Application of Graph Transformations (TAGT '98)*, 1998. To appear.

[Kus98b] Sabine Kuske. Some remarks on the generative power of collage grammars and chain-code grammars. In *Preliminary Proc. Sixth International Workshop on Theory and Application of Graph Transformations (TAGT '98)*, 1998. To appear.

[KV98] Hans-Jörg Kreowski and Gabriel Valiente. Redundancy and subsumption in high-level replacement systems. In Gregor Engels, editor, *Preliminary Proc. Sixth International Workshop on Theory and Application of Graph Transformations (TAGT '98)*, 1998. To appear.

[Lap98] Denis Lapoire. Recognizability equals monadic second-order definability, for sets of graphs of bounded tree-width. In *Proc. STACS '98*, volume 1373 of *Lecture Notes in Computer Science*, pages 618–628, 1998.

[LMS98] I. Litovsky, Y. Métivier and E. Sopena. Graph relabelling systems and distributed algorithms. In *Handbook of graph gram-*

mars and computing by graph transformation Vol. III : Concurrency, parallelism and distribution. World Scientific, 1998. To appear.

[Man98] S. Maneth. Cooperating distributed hyperedge replacement grammars. In A. Kelemenová, editor, *Proc. MFCS'98 Satellite Workshop on Grammar Systems*, pages 149–164. Silesian University, 1998. (also to appear in *Grammars*).

[MPR98] U. Montanari, M. Pistore, and F. Rossi. Modeling concurrent, mobile and coordinated systems via graph transformations. In G. Rozenberg, editor, *Handbook of Graph Grammars and Computing by Graph Transformation, Vol. III: Concurrency, Parallellism, and Distribution*. World Scientific, 1998. In preparation.

[MR98a] U. Montanari and F. Rossi. Graph rewriting, constraint solving and tiles for coordinating distributed systems. *Theoretical Computer Science*, 1998. To appear.

[MR98b] U. Montanari and F. Rossi. Modeling process coordination via tiles, graphs and constraints. In *Proc. 3rd Biennial World Conference on Integrated Design and Process Technology*, volume 4, pages 1–8, 1998.

[MS96] M. Mosbah and N. Saheb. Non uniform random spanning trees on weighted graphs. Technical Report 1143–96, LaBRI, University of Bordeaux, 1996. To appear in Theoret. Comput. Science.

[Mül97a] Jürgen Müller. A Non-categorical Characterization of Sequential Independence for Algebraic Graph Rewriting and some Applications. Technical Report 97-18, Technical University of Berlin, 1997.

[Mül97b] Jürgen Müller. Shifting Derivations of non-injective Rules in the Algebraic Graph Rewriting Approaches. Technical Report 97-16, Technical University of Berlin, 1997.

[MV98] S. Maneth and H. Vogler. Attributed context-free hypergraph grammars. (To appear in *Journal of Automata, Languages and Combinatorics*), 1998.

[Plu98a] Detlef Plump. Term graph rewriting. In H. Ehrig, G. Engels, H.-J. Kreowski, and G. Rozenberg, editors, *Handbook of Graph Grammars and Computing by Graph Transformation*, volume 2. World Scientific, 1998. To appear.

[Plu98b] Detlef Plump. Termination of graph rewriting is undecidable. *Fundamenta Informaticae*, 33(2):201–209, 1998.

[Pow90] J. Power. A 2-categorical parsing theorem. *Journal of Algebra*, 129:439–445, 1990.

[Sén97] G. Sénizergues. $\Gamma(A) \sim \Gamma(\mathrm{B})$? Technical report, nr 1183-97, LaBRI, Université Bordeaux I, can be accessed at URL, http://www.labri.u-bordeaux.fr/~ges/, 1997.

[Sén98a] G. Sénizergues. Decidability of bisimulation equivalence for equational graphs of finite out-degree. In *Proceedings FOCS 98*. IEEE, 1998. To appear.

[Sén98b] G. Sénizergues. L(A) = L(B)? Technical report, LaBRI, Université Bordeaux I, 1998. Corrected and extended version of report 1161-97; can be accessed at URL, http://www.labri.u-bordeaux.fr/~ges/; pages 1-166; submitted for publication in TCS.

[Sén98c] G. Sénizergues. The equivalence problem for deterministic push-down transducers into abelian groups. In *Proceedings MFCS 98*, pages 305–315. Springer, LNCS 1450, 1998.

[Tae97] G. Taentzer. Parallel high-level replacement systems. *TCS*, 186, November 1997.

[TK97] G. Taentzer and M. Koch. Distributing attributed graph transformation. In *First European Workshop on "General Theory of Graph Transformation Systems", Bordeaux, Oct. 8-10,1997*, 1997.

[Wol98] D. Wolz. *Colimit Library for Graph Transformations and Algebraic Development Techniques*. PhD thesis, Technical University of Berlin, Dep. of Comp. Sci., 1998.