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

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

October 24, 1997

Contents

1 Progress of the Network during the first year

1.1 Comparison of accomplished work and objectives in the Project Programme

The research and organizational activities actually accomplished during the first year of the Network perfectly 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, two Focus Area Meetings took place as planned, and they are described in Section 1.3. For each Focus Area Meeting a corresponding Report has been prepared and distributed to partners. Such reports togheter with other project documents are available by request.

Furhtermore, the dates of the major events of the second year of the Project are now fixed. They are slightly different from what planned in Section B.3 of the Project Programme: in fact, the first Workshop has been anticipated at the beginning of the second year, namely during October 1997 in Bordeaux; and the "European School on Graph Transformation" (ESoGT) is planned in Bremen in the period March 2-7, 1998, and it will host the Mid-Term Review of the Project.

1.2 Scientific Highlights

A quite detailed description of the results of the scientific research carried on during the first 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 results in the theory of Term Graph Rewriting (concerning confluence and termination, adequacy with respect to rational term rewriting, and categorical characterizations) and of Term Graph Narrowing; results about decidability of finiteness of range of tree transductions, and about generating power of various classes of transducers; the development of a new approach to graph rewriting based on pullbacks, which is a categorical variant of the set-theoretical node replacement approaches; the definition of a coalgebraic loose semantics for GTS, with the possiblity of specifying behavioural constraints; results about the expressiveness of properties definable in Monadic Second Order Logic and their implementability by attribute grammars; about confluence properties of collage grammars; contributions to the process semantics for GTS based on node rewriting (including actor grammars) and on the Double-Pushout approach (with the original definition of concatenable graph processes); results in the theory of 2-structures, of graph relabelling systems, of ESM systems, and of Distributed Graph Transformations; various proposals for the modular development of rule-based graph transformation systems; the definition of a variant of Event Structures suitable to model contextual rewriting; and the use of the Tile Model to represent a simple kind of graph rewriting with synchronization.

The efforts of the participants in training activities is witnessed by a number of publications which are either tutorial introductions to the various approaches to graph transformation systems, or surveys of results in specific research areas. Among them we mention [Roz97], the first of a series of three Handbooks, which is completely authored by researchers belonging to GETGRATS teams.

1.3 Principal Networking and Coordination Activities

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

1. Team Leader Meeting in Dagstuhl, September 12, 1996. This has been the first organizational meeting of the Site Coordinators. We discussed the Planning of GETGRATS Workshops, School, and Focus Area Meetings, the current status of GETGRATS Fellowships, and a number of administrative issues concerning the start of the Project.

2. Focus Area Meeting on "Concurrency Aspects" in Berlin, March 6–7, 1997. This has been the first scientific meeting of the project. It consisted of a presentation of the scientific activity of the various teams on topics related to Concurrency Aspects of Graph Transformation Systems; of a Team Leader Meeting where planning issues and the current status of the GETGRATS fellowships have been discussed; of a general discussion about "Concurrency in Graph Transformation Systems"; and of a number of smaller meetings where partners worked on topics of common interests. There were about 25 participants.

3. Focus Area Meeting on "Modularity Aspects" in Tarquinia (IT), June 7–8, 1997. There were presentations of the scientific activity of the various teams on topics related to Modularity Aspects of Graph Transformation Systems; and a general discussion about "Modularity in Graph Transformation Systems". During this discussion it has been decided to collect a number of Case Studies in a document, and to invite everybody to test their own approach to structuring and modularity in graph transformation systems with them. There were about 15 participants.

Related to the above events, the following documents have been produced, which can be obtained from the Network Coordinator by request and will be presented anyway at the Mid-Term Review of the Project: (1) **Minutes of**

Team Leader Meeting in Dagstuhl, (2) **Report on the Focus Area Meeting on "Concurrency"**, (3) **Report on the Focus Area Meeting on "Modularity"**, (4) **Case studies for modularity**.

Here is a list of other scientific or administrative activities performed during the first year:

1. A home page has been established on Internet describing the Project at URL ⟨http://www.di.unipi.it/∼andrea/getgrats⟩. Some partners established local home pages (for example UNIBREMEN ⟨http://www.-informatik.uni-bremen.de/grp/ag-ti/getgrats⟩ and UIA ⟨http://win-www.uia.ac.be/u/dmjans/getgrats.html⟩)

2. A scientific meeting called "GRACE Day" took place in Bremen on January 24, 1997, about Modularization of Graph Transformation Systems, with about 15 participants.

3. The "Graph Grammars and Graph Transformations Day" took place in Leiden on June 18, 1997. The aim of this event was to present to young researchers and Ph.D. students a number of research lines in this area. The lectures, given in a tutorial style, covered a number of typical research areas ranging from very theoretical to applied. Including the speakers, there were about 20 participants.

4. Concerning the "European School on Graph Transformation" (ES-oGT) planned in Bremen next March 2-7, 1998, a TMR Summer School proposal has been submitted to the EC in order to provide funding to participants external to the Network. The proposal has been accepted. A Call for Participation was distributed over several newsletters and mailing lists.

5. A mailing list was set up to be used to announce events of relevance for the GETGRATS project. The mailing list is mantainted by the Network Coordinator, and the address is `getgrats@di.unipi.it`.

6. GETGRATS joined the EU initiative of publishing a brochure describing all TMR networks. We provided in due time the requested material, including textual information and pictures.

7. A document containing a short description of the structure and objectives of the Network (as well as of the ESPRIT Working Group APPLIGRAPH – Application of Graph Transformation) has been prepared by the Network Coordinator jointly with H.J. Kreowski, Coordinator of APPLIGRAPH, to be published in the Software Specification Column of the Bulletin of the EATCS, the European Association of Theoretical Computer Science.

1.4 Progress with the Appointment of Young Researchers

For the appointment of young researchers, the following procedure has been followed. Firstly, an Internal Call for Application has been published on May 10, 1996 (with deadline June 10, 1996) limited to young researchers of GETGRATS sites. Next, an Open Call for Application was published on July 5, 1996, with open deadline. This second CfA was published to various mailing lists. The vacancies have been published on the CORDIS server as well. This first round of CfA's was intended to cover the first half of the Project. A second round started just recently to cover the rest of the project.

In response to the CfA's, we received a total of 22 applications (some of them were presented by the same person for different sites), evenly divided for postdocs and doctoral students. A total of 14 grants have been assigned for the first part of the project. Out of these, 10 have started during the first year of the project.

1.5 Difficulties

(1) There has been a reduced use of the GETGRATS resources in Pisa during the first year, due to the simultaneous leave of Prof. U. Montanari and of Dr. A. Corradini. This is one main reason why no grants were assigned during the first year. Two grants just started at the beginning of the second year.

(2) D. Janssens reported about some problems with the administration of the University of Antwerpen, because the netto amount of the grant that can be offered is either 1500 or 1900 ECU/month, with a corresponding gross amount of 2350 or 3500 ECU/month. After a short discussion during the Team Leader Meeting in Berlin, it was agreed that the lower amount should be offered, charging 2350 ECU/month to GETGRATS, even if this was slightly more than what planned. D. Janssens informed that since 1500 ECU/month is a quite low salary taking into account the cost of life in Belgium, difficulties may arise in finding candidates for the grants offered by the UIA. At the moment one grant has been assigned by Antwerpen to a young researcher who started his activity at the beginninig of October 1997.

2 Factual Information on each Participant

The following pages are copies of the form "Factual Information on each Participant" filled in by each participant. The Coordinator added to such forms the data required by the new instructions received from the EC during the report preparation, namely: 1) the scientific speciality of the participant, and 2) the number of individual researchers contributing to the joint activities of the network.

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 (via the achronyms used in the contract).

3.1 Joint papers published or accepted for publication on a Journal

- R. Heckel, A. Corradini, H. Ehrig, M. Löwe. Horizontal and Vertical Structuring of Graph Transformation Systems. *Mathematical Structures in Computer Science*, 6:613–648, 1996. **TUB.KST, UPSA.DI.**

- G. Engels, H. Ehrig, R. Heckel, and G. Taentzer. A combined reference model- and view-based approach to system specification. *Int. Journal of Software and Knowledge Engeneering*, 1997. Accepted, to appear. **TUB.KST, RUL.I.**

3.2 Joint papers presented at a Conference or chapters of books

- A. Corradini and F. Gadducci. A 2-categorical presentation of term graph rewriting. In *Proceedings CTCS'97*, volume 1290 of *LNCS*. Springer Verlag, 1997. **TUB.KST, UPSA.DI**.

- Frank Drewes, Hans-Jörg Kreowski, and Denis Lapoire. Criteria to disprove context-freeness of collage languages. In B.S. Chlebus and L. Czaja, editors, *Proc. Fundamentals of Computation Theory XI*, volume 1279 of *Lecture Notes in Computer Science*, 1997. **UBRM.TCS, UBOD1.LABRI**.

- G. Engels, R. Heckel, G. Taentzer, and H. Ehrig. A view-oriented approach to system modelling based on graph transformation. In *Proc. of ESEC/FSE'97, Zürich*, volume 1301 of *LNCS*. Springer Verlag, 1997. **TUB.KST, RUL.I**.

- R. Heckel, H. Ehrig, U. Wolter, and A. Corradini. Integrating the specification techniques of graph transformation and temporal logic. In *Proc. of MFCS'97, Bratislava*, volume 1295 of *LNCS*. Springer Verlag, 1997. **TUB.KST, UPSA.DI**.

- 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, Volume 1: Foundations*. World Scientific, 1997. **TUB.KST, UPSA.DI**.

- H. Ehrig, R. Heckel, M. Korff, M. Löwe, L. Ribeiro, A. Wagner, and A. Corradini. Algebraic approaches to graph transformation II: Single pushout approach and comparison with double pushout approach. In G. Rozenberg, editor, *The Handbook of Graph Grammars and Computing by Graph Transformations, Volume 1: Foundations.* World Scientific, 1996. **TUB.KST, UPSA.DI**.

3.3 Joint Technical Reports or papers submitted for publication

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

- Marc Andries, Gregor Engels, Annegret Habel, Berthold Hoffmann, Hans-Jörg Kreowski, Sabine Kuske, Detlef Plump, Andy Schürr, and Gabriele Taentzer. Graph transformation for specification and programming. Report 7/96, Univ. Bremen, 1996. **TUB.KST, UBRM.TCS, RUL.I**.

- A. Corradini and F. Drewes. (Cyclic) term graph rewriting is adequate for rational parallel term rewriting. Technical Report TR-97-14, Dipartimento di Informatica, Pisa, 1997. **UBRM.TCS, UPSA.DI**.

- A. Corradini and F. Gadducci. An algebraic presentation of term graphs, via ps-monoidal categories. Submitted for publication in *Applied Categorical Structures*, 1997. **TUB.KST, UPSA.DI.**

- Frank Drewes and Joost Engelfriet. Decidability of the finiteness of ranges of tree transductions. Report 9/96, Univ. Bremen, 1996. **UBRM.TCS, RUL.I**.

- R. Heckel, H. Ehrig, U. Wolter, and A. Corradini. Double-pullback transitions and coalgebraic loose semantics for graph transformation systems. *Applied Categorical Structures*, 1997. Submitted. **TUB.KST, UPSA.DI.**

- R. Heckel, H. Ehrig, U. Wolter, and A. Corradini. Loose semantics and constraints for graph transformation systems. Technical Report 97-07, TU Berlin, 1997. **TUB.KST, UPSA.DI.**

4 Appendix - Scientific Contributions

This section describes the main scientific contributions of the research activity carried on during the first 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

In [CMR+97] and [EHK+97] a survey of the algebraic approaches to graph transformation is presented. The algebraic approaches to graph transformation are based on the concept of gluing of graphs, modelled by pushouts in suitable categories of graphs and graph morphisms. This allows one not only to give an explicit algebraic or set theoretical description of the constructions, but also to use concepts and results from category theory in order to build up a rich theory and to give elegant proofs even in complex situations.

In [CMR+97] we start with an overwiev of the basic notions common to the two algebraic approaches, the *double-pushout (DPO) approach* and the *single-pushout (SPO) approach*; next we present the classical theory and some recent development of the double-pushout approach. [EHK+97] is devoted instead to the single-pushout approach and to some of its main results. Application conditions for graph productions and the transformation of more general structures than graphs are studied in the SPO-approach (similar generalizations have been or could be studied also in the DPO-approach). Finally, a detailed comparison of the DPO- and the SPO-approach is presented, especially concerning the solutions to the problems discussed for both approaches in [CMR+97].

In [HEWC97b] an integration of graph grammars with different kinds of behavioural constraints, in particular with temporal logic constraints is presented. Since the usual algebraic semantics of graph transformation systems is not able to express constrained behaviour we introduce - in analogy to other approaches - a coalgebraic semantics which associates with each system a category of models (deterministic transition systems). Such category has a final object, which includes all finite and infinite transition sequences. The coalgebraic framework makes it possible to introduce a general notion of 'logic of behavioural constraints'. Instances include, for example, graphical consistency constraints and temporal logic constraints. We show that the considered semantics can be restricted to a final coalgebra semantics for systems with behavioural constraints.

In [CD97] it is shown that the rewriting of cyclic term graphs (which denote rational, possibly infinite terms) is adequate with respect to rational parallel term rewriting. The approach constrasts with the "classical" one in that a single graph reduction corresponds to a single (but possibly infinite) parallel term rewrite step whereas, classically, a graph reduction may correspond to an infinite sequence of term rewrite steps.

An introduction to the theory of hyperedge-replacement as a context-free method of graph generation was given in [DHK97]. The emphasis is put there on structural properties as well as on decidability results.

In [DE96] the finiteness of ranges of a large class of tree transductions is proved to be decidable. It is shown that this yields a generalization of known decidability results concerning boundedness problems for functions on graphs and other domains. The functions that can be dealt with are definable by recursion on the derivation tree of a graph. This does not require a particular notion of graph rewriting.

Basic properties of the equational and recognizable subsets of general algebras are reviewed in [Cou96a]; these sets can be seen as generalizations of the context-free and regular languages respectively. This approach, based on Universal Algebra, facilitates the development of the theory of formal languages so as to include the description of sets of finite trees, finite graphs, finite hypergraphs, tuples of words, partially commutative words (also called traces) and other similar finite objects. Descriptive complexity is concerned with characterization of complexity classes by logical formulas. Of special interest are formulas starting with existential set quantifications, and followed by first order quantifications. They are reviewed in [Cou97b] the main basic graph properties in the perspective of expression by these formulas. In particular, it considers constructive expression of planarity where the formula describes a planar embedding. This should be contrasted with the classical negative characterization by Kuratowski's theorem, which says nothing of the planar embedding of a graph recognized as planar. [Cou97a] is a chapter of a tutorial book on graph grammars, which presents the universal algebra tools behind context-free graph grammars, of HR or VR type, the algebraic notion of recognizability of a set of graphs, the description of sets of graphs and of graph transformations by monadic second order formulas. It surveys the known closure results and comparison results between these various tools.

Pullback rewriting was introduced a few years ago as a categorical framework to describe node rewriting in graphs. It has since appeared as a fairly powerful mechanism to describe a large number of graph rewriting systems. Bauderon and Jacquet have shown in [BJ96] how this could be used to provide a uniform rewriting mechanism for a wide class of graphs which are uniformly described under the name of structured graphs. Pullback rewriting was then used by Klempien-Hinrichs in [KH97] to describe refinement in Petri net. The complexity of the graphs generated by pullback rewriting is such that visualization tools quickly become mandatory. Unfortunately, the ones which are available in the graph rewriting community do not cope with the kind of graphs and graph rewriting we need. Cristiano Corsani has been working for six month with the Calico team in LaBRI, which is developing software to visualize complex mathematical objets. New tools should therefore be soon available. In [Ly97], it is shown that the hierarchy of formulas defining graphs up to isomorphism in weak monadic second order logic is infinite. The Ph.D. thesis [Pel97] provides a detailed structure of the automorphism groups of various kinds of infinite graphs. In order of complexity: context-free, equational and automatic.

In [BE97b, BE97a] it is shown that monadic second order logical (MSO) properties of the nodes of trees can be implemented by attribute grammars with boolean values only, whereas MSO relations between the nodes of trees can be implemented by tree-walking automata with MSO tests. These characterizations can be used to implement MSO definable tree transductions by attribute grammars.

Equivalence classes of group labeled 2-structures (also called switching classes of gain graphs) have been investigated. Results on sizes of switching classes are given in [HH97]. Also, the complexity of several decision problems concerning gain graphs, and their relationship with the analogous problems for ordinary graphs, was investigated. The investigation of the membership problem for gain graphs, i.e., the question whether two gain graphs belong to the same switching class, has been initiated.

4.2 (b) Classification and Expressive Power

In [Cou96b] some answers are given to the following general question: for which classes of graphs C is it possible to specify a linear ordering of the set of vertices of each graph of C by a fixed monadic second-order formula? It is also investigated whether recognizability and monadic second-order definability are the same for subclasses of C, and whether the hierarchical algebraic structure of a graph from C can be defined in (an extension of) monadic second-order logic. The main result of [CS96] is described by the title. Auxiliary results concern HR graph grammars which, considered as systems of equations, have a unique solution.

Sets of random walks on graphs can be formally described by regular expressions over an alphabet [MS97c, MS97b, MS96]. This provides a powerful technique to evaluate various parameters related to random walks on graphs, such as the cover time and the hitting time. An algorithm for generating random spanning trees in a weighted graph is obtained by considering the spanning tree defined by the edges corresponding to first visits to the vertices.

Most distributed algorithms can be modeled as graph model checking or rewriting algorithms. On the other hand, fixpoint calculi techniques, which work well for tree-like structures, seem applicable to more complex structures such as graphs. In such a setting, the general properties of fixpoint calculi for complete partial orders were investigated in [Jan97a, JW96, Jan97b].

In particular, an automata-theoretic characterization was obtained. These results may lead to new insights in the various possible notions of automata or grammars over graphs.

In [DKL97] some criteria are presented that can be used to disprove context-freeness of collage languages. Such criteria are of interest because the usual tools like pumping lemmata fail in the case of collages. Roughly speaking, it is shown that the number of parts of collages in a context-free collage language grows at most linearly, and that the volume of parts grows exponentially.

In [Dre97a, Dre97b] the structure of hyperedge-replacement graph languages consisting of (hyper)trees is investigated. It is shown that these languages can be generated by productions whose right-hand sides are themselves hypertrees. However, for this, the sets of derivation trees have to be produced by a finite-copying top-down tree transducer. Thus, finite copying compensates the loss in power that results from allowing only hypertrees as right-hand sides.

Survey papers on the expressive power of context-free graph grammars have appeared in two handbooks (see [ER97, Eng97]). It is shown in [EV96] that bottom-up and top-down tree-to-graph transducers define the same class of tree transductions (as opposed to the classical bottom-up and top-down tree transducers). In [BE97c] a new notion of treewidth is studied, closely related with apex graph grammars.

4.3 (c) Analysis and Verification Techniques

Constraints for graph transformation are introduced in [HEWC97a] and [HEWC97c] in order to restrict the loose semantics of graph transformation systems presented in [HEWC97b]. A coalgebraic framework makes it possible to define a general notion of logic of behavioural constraints. Instances include, for example, start graphs, application and consistency conditions, and temporal logic constraints. It is shown that the considered semantics can be restricted to a final coalgebra semantics for systems with behavioural constraints.

In [Mül96], a necessary and sufficient condition for termination of single-pushout graph rewriting systems is established. Termination is equivalent to the consumptiveness and the finiteness of all 'forward closures' being certain minimal derivations in which each step depends on previous steps. This characterization is the same as for the double-pushout graph rewriting approach, but it cannot be obtained by only translating the double-pushout rules to rules in the single-pushout approach. The translation of the rules does not reflect the application condition of rules in the double-pushout approach. The absence of application conditions in the single-pushout approach allows to show the compatibility of (in)dependence and instance of derivations.

In [AKP97] a survey of confluence properties of (acyclic) term graph rewriting is presented. Results and counterexamples are given for plain applications of rewrite rules, extensions with the operations of collapsing and copying, and with both operations together. Collapsing and copying together constitute bisimilarity of term graphs.

In [HP96] term graph narrowing, combining term graph rewriting with first-order term unification, is introduced as an approach for solving equations by transformations on term graphs. The main result is that this mechanism is complete for all term rewriting systems over which term graph rewriting is normalizing and confluent. This includes, in particular, all convergent term rewriting systems.

Computations by term graph rewriting terminate more often than computations by the corresponding term rewrite systems. For this reason, in [Plu97] *simplification orders* on term graphs are introduced and shown to be well-founded. Thus, these orders can be used in termination proofs. A *recursive path order* on term graphs is defined by analogy with the well-known order on terms, and is shown to be a simplification order.

4.4 (d) Abstract Semantics

The theory of algebraic graph transformation systems has been extended in [HEWC97a] by a loose semantics. For this purpose, graph transitions are introduced as a loose interpretation of graph productions. They are defined using a double pullback construction in contrast to classical graph derivations based on double-pushouts. Two characterization results relate graph transitions to the classical double-pushout derivations and to amalgamated derivations, respectively. Moreover, a loose semantics for graph transformation systems is defined, which associates with each system a category of models (deterministic transition systems) defined as coalgebras over a suitable functor. Such category has a final object, which includes all finite and infinite transition sequences. Building on the reported results the aim of [HEWC97b] is an extension of the theory of graph transformation systems in order to make them suitable for the specification of reactive systems.

A new categorical description for term graph rewriting is proposed in [CG97a]. It is already known that term rewriting systems can be faithfully described by a Cartesian 2-category, where horizontal arrows represent terms, and cells represent rewriting sequences. A similar, original 2-categorical presentation is presented for term graph rewriting. In [CG97b] the main result is a categorical characterization of term graphs that parallels the well-known characterization of terms as arrows of the algebraic theory of a given signature. In particular, it is shown that term graphs over a signature Σ are one-to-one with the arrows of the free gs-monodial category generated by Σ. Such a category satisfies all axioms for Cartesian categories but for the naturality of two transformations providing in this way an abstract and clear relationship between terms and term graphs.

For node rewriting GTS, the use of sets or categories of processes as a semantics has been further investigated, along the lines of [JM96]. Recently an elegant characterization has been given for a large class of such GTS that allows the construction of processes. ESM systems and actor grammars are special cases of this abstract framework. The expressivity of the basic process semantis, as well as the more abstract versions that can be derived from it, is being investigated in the light of a number of test-cases, such as the well known AB protocol [JV97]; it is shown that basic safety properties can be expressed and proved in terms of processes.

Work on the use of node rewriting GTS as a semantics for actor languages has resumed; the ideas about actor grammars of, e.g. [JR89] have been demonstrated for an actor language with a standard syntax. On the one hand this means that the notions about process semantics developed for GTS become available for the analysis of actor systems, and on the other hand it opens the perspective of developing an interpreter for actor languages using the GTS as an intermediate language in the implementation.

In [BCM97a] an original notion of *concatenable graph processes* is proposed for typed graph grammars using the double-pushout approach. These structures allow the authors to relate in a completely satisfactory way the semantics based on derivation traces proposed in [CEL$^+$94] and the notion of graph processes introduced in [CMR96], showig their equivalence as truly concurrent semantics for graph grammars.

4.5 (e) Concurrency Aspects

In this Focus Area several approaches to specifying concurrent and distributed systems by graph transformation are used: *graph relabelling systems*, *dynamic labeled 2-structures*, *ESM-systems* and *distributed graph transformation.*

Graph relabelling systems and distributed computations

Local computations on graphs, as given by graph rewriting systems (with priorities and/or forbidden contexts) are a powerful model for local computations that can be executed in parallel. Rewriting systems provide a general tool for encoding distributed algorithms and for proving their correctness. We consider a network of processors with arbitrary topology. It is represented as a connected, undirected graph where vertices denote processors, and edges denote direct communication links. An algorithm is encoded by means of local relabellings. Labels attached to vertices and edges are modified locally, that is, on a subgraph of fixed radius k of the given graph, according to certain rules depending on the sub-graph only (such local computations are called $k-$local). The relabelling is performed until no more transformation is possible (i.e., until a "normal form" is obtained). Two sequential relabelling steps are said to be independent if they are applied on disjoint subgraphs. In this case they may be applied in any order or even concurrently; [MS97a] presents a general overview.

More recently, we have been interested in the existence of rewriting systems having the property of local detection of the global termination. In the terminology of distributed computing, we say that a distributed algorithm terminates whenever it reaches a terminal configuration, i.e. a configuration in which no steps of the algorithm can be applied anymore. We say that a local computation system allows the local detection of (global) termination if for any computation there is a vertex for which its neighborhood of a given radius r determines whether or not a normal form has been reached (we say that global termination is $r-$locally detected).

Our aim in [MM97] is twofold. First, we present a formalization of the local detection problem and we introduce some new methods for obtaining impossibility results. More precisely, we extend the notion of coverings, which is known from algebraic topology and has been already used in distributed computing for negative results, to quasi-coverings. Quasi-coverings capture some topologies, which fail to cope with the classical coverings. We show in this paper that one cannot detect locally the global termination for uniformly labelled graphs belonging to certain families of connected graphs $\mathcal{C}$. More specifically, it suffices for our results that $\mathcal{C}$ contains some graph G and a nontrivial $k-$covering of G, (or, some graph G and quasi-coverings of G of arbitrary large size). Our second aim is to discuss some implications of our results for the borderline between positive and negative results. More precisely, we are interested in the question whether certain additional knowledge about a network, which is used in specific graph rewriting systems, is really necessary. For example, we show that election in T-prime graphs is possible (if and) only if the size of the graph is provided. Our results are obtained for anonymous networks. (Clearly, if a leader is available, then the termination can be easily detected locally using standard methods).

It has been proved that we cannot decide whether or not a fixed graph is included as a minor in a given graph by means of local computations. In [BM97] 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.

Dynamic Labeled 2-Structures

Dynamic labeled 2-structures form a network-oriented model of concurrency and distribution. One of the central problems is to determine whether global states with certain properties can occur during the evolution of the system. This problem has been investigated in terms of patterns and configurations. A number of results concerning the relationship between patternes and configurations have been obtained.

ESM-Systems

For embedding-based GTS the work on concurrency aspects is aimed at exploring the notion of a process, as used in, e.g, [JM96]. Based on a case study [JV97] it has been demonstrated that, for a system composed of various sequential components, each concurrent history of the system, expressed by a process, can be obtained by the composition of the sequential histories of the components. The construction does not require a fixed number or configuration of the components; i.e., the components may be mobile agents.

The possibility to characterize such sequential histories, and to compose them, is an important advantage when describing histories of actor systems by processes, as the main structuring mechanism of those systems (actors) correspond to sequential components, which are organized in a dynamically evolving network.

Distributed Graph Transformation

The new approach of distributed graphs and graph transformation as presented in [Tae96] and [Tae97] allows to use structured graph transformation on two abstraction levels, the network and the local level. The network level contains the description of the topological structures of a system. The local level covers the description of states and their transitions in local systems. Local state transitions may depend on others using suitable synchronization mechanisms. The main distribution concepts of categorical graph grammars presented by Schneider are combined with the algebraic approach to distributed graph transformation introduced by Ehrig et.al.

Modeling of distributed systems by this new kind of distributed graph transformation offers a clear and elegant description of dynamic networks, distributed actions as well as communication and synchronization using a graphical notation. Moreover, distributed graph transformation offers the possibility to describe splitting and joining of local graphs as well as parallel transformations in local systems.

The formalization of distributed graph transformation is done by means of category theory. A distributed graph is formalized by a diagram in the category GRAPH of graphs and total graph morphisms. A distributed transformation step is characterized by a double-pushout in the category DISTR(GRAPH) of distributed graphs and distributed graph morphisms. A pushout over distributed graph morphisms cannot always be constructed componentwise in each local part. But especially the componentwise construction of a distributed transformation reflects distributed computations best. Thus, we present the necessary conditions for componentwise construction of pushouts in DISTR(GRAPH) and show that they are also sufficient. These conditions are summarized in the distributed gluing condition. Moreover, these conditions are needed for componentwise construction of pushout complements which are used to characterize the first step of a distributed graph transformation.

The approach presented in [MR96] is based on the fact that graphs and graph grammars can be fritfully used to model distributed systems which behave in a synchronized way. In fact, graphs can represent networks of processes and suitable graph rewriting rules, based on the combination of simple context-free productions via a synchronization mechanisms, can represent the dynamics of such systems. To cope with the problem of efficiently combining productions together, we can then use well-known constraint propagation techniques, which can help reduce the possible choices for each process and thus simplify the combination problem. This same approach can also be modelled via the tile model, in which the considered notion of graph rewriting can be cast [MR97].

In [Koc97], distributed graph transformation is extended to allow application conditions for distributed rules. This extended formalism is applied to model a distributed database transaction model.

4.6 (f) Modularity Aspects

In [AEH$^+$96] recent developments concerning the application of graph transformation as a rule-based framework for the specification and development of systems, languages, and tools are surveyed. Central concepts discussed are approach independence and modularity, in order to allow for the structured specification of systems without requiring a particular graph transformation approach.

In [KKS97] a new concept for the modular development of rule-based graph transformation systems, called nested graph transformation unit, is presented. It consists of a (possibly cyclic) network of local graph transformation units that reflects the import structure of the system. It is shown that nested graph transformation units have a reasonable operational semantics, which is obtained by iterating the so-called interleaving operator.

A module is basically given by export and import interfaces, and a body that implements the features offered by the export interface in terms of the features required in the import interface. In [GRPS97a, GRPS97b] two kinds of refinement relations between GTS are investigated, modelling the notions of implementation and abstractions which relate the export interface to the body and the corresponding derivations. Pushouts of refinements are shown to exist, thus allowing the definition of operations on modules, such as composition, with a predictable compositional semantics.

The work on modularity for embedding-based GTS focuses on the possibility to build networks of GTS, connected by interfaces. The role of these interfaces is essentially to change the representation of data, and thereby to call the modules (the GTS in the network), since the behaviour of the GTS considered here is entirely data-driven. By associating the whole network with a new GTS, one may view the building of the network as a composition operation, building the new GTS from modules and interfaces. Evidently the problem of providing such a construction is closely related to the development of suitable abstract semantics, since it must be possible to obtain the semantics of the new GTS from the semantics of the components, rather than from the components itself. Preliminary results indicate that this approach is possible for the process semantics of, e.g. [JM96].

Engels, Wagner, and Busatto started developing a module concept, called *instance level packages*. In contrast to other modularization concepts which are currently under development, instance level packages give additional structure to the graph and not to the set of rules. This additional structure is used to restrict the applicability of the rules.

4.7 (g) Morphisms, Transformations and Operations

In [BCM97b] the authors propose an event based semantics for contextual nets, i.e., an extension of Place/Transition Petri nets where transitions can have not only preconditions and postconditions, but also context conditions, modelling resources that can be read without being consumed. The result is a generalization of Winskel's work on safe nets: the truly concurrent event based semantics is given at categorical level via a chain of coreflections leading from the category WS-CN of weakly safe contextual nets to the category DOM of finitary prime algebraic domains (equivalent to the category PES of prime event structures).

A fundamental role is played by the notion of Asymmetric Event Structure that generalizes Winskel's prime event structures, following an idea similar to that of "possible flow" introduced by Pinna and Poigné. Asymmetric event structures still have the usual causal relation of traditional prime event structures, but replace the symmetric conflict with a relation modelling asymmetric conflict or weak causality. Such relation allows one to represent the new kind of dependency between events arising in contextual nets, as well as the usual symmetric conflict. Moreover it is used in a non-trivial way in the definition of the ordering of configurations, which is different than the usual set inclusion.

These notions and result about contextual nets can be seen as a first step towards the definition of an unfolding semantics for graph grammars. We think that the work on contextual nets could be a guide for the definition of the notions of non-deterministic occurrence graph grammar and graph grammar unfolding that are still lacking or not consolidated.

Another active topic of research in this Focus Area is the Tile Model [GM97]. Tiles are much like SOS inference rules, but they can be composed horizontally and vertically to build larger proof steps. They generalize Kim Larsen and Liu Xinxin context systems since they allow for more general rule formats. The tile model also extends rewriting logic by José Meseguer (in the nonconditional case), since it takes into account rewritings with side effects and rewriting syncronization, and can be naturally equipped with observational equivalences and congruences.

Tile systems can be seen as double categories and tiles themselves as double cells. However, the tile model can be also presented in a purely logical form, tiles being just special sequents subject to certain inference rules. Besides ordinary process algebras, tiles model adequately calculi with name bindings (e.g. the pi–calculus) and coordination models equipped with flexible synchronization primitives. They have been used to model a graph rewriting formalism aimed at describing the concurrent, synchronized evolution of local processes in [MR97].

References

[AEH+96] Marc Andries, Gregor Engels, Annegret Habel, Berthold Hoffmann, Hans-Jörg Kreowski, Sabine Kuske, Detlef Plump, Andy Schürr, and Gabriele Taentzer. Graph transformation for specification and programming. Report 7/96, Univ. Bremen, 1996.

[AKP97] Zena M. Ariola, Jan Willem Klop, and Detlef Plump. Confluent rewriting of bisimilar term graphs. In *Proc. Fourth Workshop on Expressiveness in Concurrency*, volume 7 of *Electronic Notes in Theoretical Computer Science*. Elsevier, 1997. URL: `http://www.elsevier.nl/locate/entcs/volume7.html`.

[BCM97a] P. Baldan, A. Corradini, and U. Montanari. Concatenable Graph Processes: Relating Processes and Derivation traces. 1997.

[BCM97b] P. Baldan, A. Corradini, and U. Montanari. An Event Structure Semantics for P/T Contextual Nets: Asymmetric Event Structures. Submitted for Publication, 1997.

[BE97a] R. Bloem and J. Engelfriet. Characterization of properties and relations defined in monadic second order logic on the nodes of trees. Tech.Report 97–03, Leiden University, August 1997.

[BE97b] R. Bloem and J. Engelfriet. Monadic second order logic and node relations on graphs and trees. In J. Mycielski, G. Rozenberg, and A. Salomaa, editors, *Structures in Logic and Computer Science*, volume 1261 of *Lecture Notes in Computer Science*, pages 144–161. Springer-Verlag, 1997.

[BE97c] H. L. Bodlaender and J. Engelfriet. Domino treewidth. *J. of Algorithms*, 24:94–123, 1997.

[BJ96] M. Bauderon and H. Jacquet. Categorical product as a generic graph rewriting mechanism. Technical Report 1166–97, LaBRI, University of Bordeaux, 1996. Submitted to J.A.C.T.S.

[BM97] A. Bottreau and Y. Métivier. Normal forms of graph rewriting, minor searching : two applications of enumeration. Technical report, LaBRI, 1997. submitted.

[CD97] A. Corradini and F. Drewes. (Cyclic) term graph rewriting is adequate for rational parallel term rewriting. Technical Report TR-97-14, Dipartimento di Informatica, Pisa, 1997.

[CEL+94] A. Corradini, H. Ehrig, M. Löwe, U. Montanari, and F. Rossi. Abstract Graph Derivations in the Double-Pushout Approach. In H.-J. Schneider and H. Ehrig, editors, *Proceedings of the Dagstuhl Seminar 9301 on Graph Transformations in Computer Science*, volume 776 of *LNCS*, pages 86–103. Springer Verlag, 1994.

[CG97a] A. Corradini and F. Gadducci. A 2-categorical presentation of term graph rewriting. In *Proceedings CTCS'97*, volume 1290 of *LNCS*. Springer Verlag, 1997.

[CG97b] A. Corradini and F. Gadducci. An algebraic presentation of term graphs, via ps-monoidal categories. Submitted for publication. Available at `http://www.di.unipi.it/~gadducci/papers/aptg.ps`, 1997.

[CMR96] A. Corradini, U. Montanari, and F. Rossi. Graph processes. *Fundamenta Informaticae*, 26:241–265, 1996.

[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, Volume 1: Foundations.* World Scientific, 1997.

[Cou96a] B. Courcelle. Basic notions of universal algebra for language theory and graph grammars. *Theoretical Computer Science*, 163:1–54, 1996.

[Cou96b] B. Courcelle. The monadic second-order logic of graphs X: Linear orders. *Theoretical Computer Science*, 160:87–143, 1996.

[Cou97a] B. Courcelle. The expression of graph properties and graph transformations in monadic second-order logic. In G. Rozenberg, editor, *Handbook of Graph Grammars and Computing by Graph Transformations. Vol. I: Foundations*, chapter 5, pages 313–400. World Scientific, 1997.

[Cou97b] B. Courcelle. On the expression of graph properties in some fragments of monadic second-order logic. In N. Immerman and P.Kolaitis, editors, *Descriptive complexity and finite models*, volume 31, pages 33–62. DIMACS Series in Discrete Mathematics and Theoretical Computer Sciences, June 1997.

[CS96] B. Courcelle and G. Sénizergues. The obstructions of a minor-closed set of graphs defined by a hyperedge-replacement graph grammar can be constructed. *Discrete Mathematics*, 1996. In press.

[DE96] Frank Drewes and Joost Engelfriet. Decidability of the finiteness of ranges of tree transductions. Report 9/96, Univ. Bremen, 1996.

[DHK97] Frank Drewes, Annegret Habel, and Hans-Jörg Kreowski. Hyperedge replacement graph grammars. In G. Rozenberg, editor, *Handbook of Graph Grammars and Computing by Graph Transformations. Vol. I: Foundations*, chapter 2, pages 95–162. World Scientific, 1997.

[DKL97] Frank Drewes, Hans-Jörg Kreowski, and Denis Lapoire. Criteria to disprove context-freeness of collage languages. In B.S. Chlebus and L. Czaja, editors, *Proc. Fundamentals of Computation Theory XI*, volume 1279 of *Lecture Notes in Computer Science*, 1997.

[Dre97a] Frank Drewes. A characterization of the sets of hypertrees generated by hyperedge-replacement graph grammars. Report 3/97, Univ. Bremen, 1997.

[Dre97b] Frank Drewes. On the generation of trees by hyperedge replacement. In I. Prívara and P. Ružička, editors, *Proc. Mathematical Foundations of Computer Science 1997*, volume 1295 of *Lecture Notes in Computer Science*, pages 229–238, 1997.

[EHK+97] H. Ehrig, R. Heckel, M. Korff, M. Löwe, L. Ribeiro, A. Wagner, and A. Corradini. Algebraic approaches to graph transformation II: Single pushout approach and comparison with double pushout approach. In G. Rozenberg, editor, *The Handbook of*

Graph Grammars and Computing by Graph Transformations, Volume 1: Foundations. World Scientific, 1997.

[Eng97] J. Engelfriet. Context-free graph grammars. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages*, volume 3: Beyond Words. Springer-Verlag, 1997.

[ER97] J. Engelfriet and G. Rozenberg. Node replacement graph grammars. In G.Rozenberg, editor, *Handbook of Graph Grammars and Computing by Graph Transformation*, volume 1: Foundations. World Scientific, 1997.

[EV96] J. Engelfriet and H. Vogler. The equivalence of bottom-up and top-down tree-to-graph transducers. Technical Report TUD/FI96/17, Technical University of Dresden, October 1996.

[GM97] F. Gadducci and U. Montanari. The Tile Model. In G. Plotkin, C. Stirling, and M. Tofte, editors, *Proof, Language and Interaction: Essays in Honour of Robin Milner*. MIT Press, 1997. To appear.

[GRPS97a] Martin Grosse-Rhode, Francesco Parisi Presicce, and Marta Simeoni. A simple notion of refinement in graph transformation systems. Report 10/97, Univ. Roma 'La Sapienza', 1997.

[GRPS97b] Martin Grosse-Rhode, Francesco Parisi Presicce, and Marta Simeoni. Spatial and temporal refinement of typed graph transformation systems. Report 11/97, Univ. Roma 'La Sapienza', 1997.

[HEWC97a] R. Heckel, H. Ehrig, U. Wolter, and A. Corradini. Double-pullback transitions and coalgebraic loose semantics for graph transformation systems. *Applied Categorical Structures*, 1997. Submitted.

[HEWC97b] R. Heckel, H. Ehrig, U. Wolter, and A. Corradini. Integrating the specification techniques of graph transformation and temporal logic. In *Proc. of MFCS'97, Bratislava*, volume 1295 of *LNCS*. Springer Verlag, 1997.

[HEWC97c] R. Heckel, H. Ehrig, U. Wolter, and A. Corradini. Loose semantics and constraints for graph transformation systems. Technical Report 97-07, TU Berlin, 1997. `http://www.cs.tu-berlin.de/cs/ifb/TechnBerichteListe.html`.

[HH97] J. Hage and T. Harju. The size of switching classes with skew gains. Tech.Report 97–02, Leiden University, Department of Computer Science, 1997.

[HP96] Annegret Habel and Detlef Plump. Term graph narrowing. *Mathematical Structures in Computer Science*, 6:649–676, 1996.

[Jan97a] D. Janin. Automata, tableaus and a reduction theorem for fixpoint calculi in arbitrary complete lattices. In *IEEE Symp. on Logic in Computer Science*, 1997.

[Jan97b] D. Janin. When the satisfiability problem for fixpoint expressions in complete ordered sets is simple. In *CSL'97*, 1997. Submitted to the proceedings.

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

[JR89] D. Janssens and G. Rozenberg. Actor Grammars. *Mathematical Systems Theory*, 22:75–107, 1989.

[JV97] D. Janssens and N. Verlinden. Reasoning on graph-rewriting processes. Technical Report 97–07, University of Antwerp - UIA, dept. of Math. and Comp. Sci., 1997.

[JW96] D. Janin and I. Walukiewicz. On the expressive completeness of the modal mu-calculus w.r.t. monadic second order logic. In *Proc. CONCUR'96*, volume 1119 of *Lecture Notes in Computer Science*, 1996.

[KH97] R. Klempien-Hinrichs. Net refinement by pullback rewriting. Technical report, LaBRI, University of Bordeaux, 1997.

[KKS97] Hans-Jörg Kreowski, Sabine Kuske, and Andy Schürr. Nested graph transformation units. *International Journal of Software Engineering and Knowledge Engineering*, 1997. to appear.

[Koc97] M. Koch. Bedingte verteilte Graphtransformation und ihre Anwendung auf verteilte Transaktionen. Technical Report 97-11, Technical University of Berlin, 1997.

[Ly97] O. Ly. On the logical hierarchy of graphs. Technical report, Colloqium New trends in Semantics, (satellite workshop of ICALP97), 1997.

[MM97] Y. Métivier and A. Muscholl. About the local detection of termination of local computations. In *The 4th International colloquium on structural information and communication complexity*. Carleton University Press, 1997.

[MR96] U. Montanari and F. Rossi. Graph rewriting and constraint solving for modelling distributed systems with synchronization. In *Coordination Languages and Models*, volume 1061 of *LNCS*, pages 12–27. Springer Verlag, 1996.

[MR97] U. Montanari and F. Rossi. Graph rewriting, constraint solving and tiles for coordinating distributed systems. Submitted for publication, 1997.

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

[MS97a] Y. Métivier and E. Sopena. Graph relabelling systems : a general overview. *Computers and artificial intelligence*, 16:167–185, 1997.

[MS97b] M. Mosbah and N. Saheb. Formal rational fractions and random walks on cycle graphs. In *Formal Power Series and Algebraic Combinatorics, 9th Conference*, Vienna, Austria, July 1997. Technical Report 1147–96, University of Bordeaux.

[MS97c] M. Mosbah and N. Saheb. A syntactic approach to random walks on graphs. In *23rd International Workshop, WG'97*, Berlin, Germany, June 1997. Lect. Notes. Comput. Sci., to appear. Technical Report 1163–97, University of Bordeaux.

[Mül96] J. Müller. On termination of single-pushout graph rewriting. Technical Report 96-38, TU Berlin, 1996.

[Pel97] L. Pelecq. Isomorphismes et automorphismes des graphes contextfr-free, équationnels et automatiques. *Thèse de doctorat,université Bordeaux 1*, pages 1–127, 1997.

[Plu97] Detlef Plump. Simplification orders for term graph rewriting. In I. Prívara and P. Ružička, editors, *Proc. Mathematical Foundations of Computer Science 1997*, volume 1295 of *Lecture Notes in Computer Science*, pages 458–467, 1997.

[Roz97] G. Rozenberg, editor. *Handbook of Graph Grammars and Computing by Graph Transformation*, volume 1: Foundations. World Scientific, 1997.

[Tae96] G. Taentzer. *Parallel and Distributed Graph Transformation: Formal Description and Application to Communication-Based Systems.* PhD thesis, TU Berlin, 1996. Shaker Verlag.

[Tae97] G. Taentzer. Distributed graphs and graph transformation. *Applied categorical Structures*, 1997. Submitted.