Rewriting nested graphs, through term graphs

Roberto Bruni[1], Andrea Corradini[1], Fabio Gadducci[1],
Alberto Lluch Lafuente[2], and Ugo Montanari[1]

[1] Dipartimento di Informatica, Università di Pisa, Italy
[2] IMT Institute for Advanced Studies, Lucca, Italy

Abstract. We present an algebra for graphs with nesting and restriction features as a handy linear syntax for denoting a class of hierarchical graphs. We discuss how such graphs can be encoded into term graphs, showing that through this translation we can borrow definitions and results from the established theory of term graph rewriting.

The use of graphs or diagrams of various kinds is pervasive in Computer Science, as they come handy for describing in a two-dimensional space either the logical or the topological structure of systems, models, states, behaviours, computations, and other entities of interest; examples include the graphical presentation of entity-relationship diagrams, of finite state automata, of static and behavioural UML diagrams, of computational formalisms like Petri nets, and so on.

The advantage of using graphs or diagrams, rather than a linear syntax based on terms or strings, lies in the fact that graphs can easily convey the attention to relevant topological features of the systems or models they describe, such as nesting, hierarchies, sharing of structures, communication capabilities, causal dependencies, and pointers or links, making these features easily understandable also to non-specialists. In several cases graphs provide a representation of models or systems at the "right" level of abstraction: often a single graph corresponds to an equivalence class of terms, up to an axiomatic specification equating systems considered as topologically indistinguishable. For example, the order in which the states or the transitions of an automaton are listed is inessential. Furthermore, as drawings are usually understood "up to isomorphism", if the concrete identities of certain syntactical entities are irrelevant (for example, the name of the states of a finite automaton; or the name of bound variables), it is sufficient to avoid depicting them in the drawing (a state is uniquely identified by the graphical components it is represented by; a bound variable is represented by an unlabeled node and the references to it with edges, making α-conversion unnecessary).

Unfortunately, though, graphical representations are much more difficult to handle and to analyse than linear ones. In general, a graphical model needs to be encoded into a linear syntax, in order to exploit tools like theorem provers or model checkers to verify certain properties on it. Clearly, the set-theoretical presentation of a graph, made of a tuple including the sets of nodes and of edges and various functions involving them (connection, labeling, ...), does provide a linear syntax, but it is very hard to parse and manipulate for human beings. A much handier syntax can be defined by introducing an equational signature,

whose operator symbols are interpreted as operations on graphs and where the axioms formalise suitable properties of such operators: then the terms of the initial algebra can be interpreted as graphs and manipulated algebraically.

In recent works [3, 5, 4] we explored graphical notations that treat as first-class citizens the sharing of possibly bound names, as well as the nesting of structures, a feature that emerges as a recurrent pattern in the conceptual modeling of systems. We introduced a class of hierarchical graphs and a corresponding equational algebra, that we proved to be sound and complete. The algebra for hierarchical graphs has been used to encode several process calculi featuring sophisticated notions of nesting and restriction (including π-calculus [13], Sagas [6], and CaSPiS [1], among others). However, a proper definition of *concurrent rewriting* over this algebra is still missing. The recent theory of rewriting in adhesive categories [11, 12], which lifts the realm of graph transformation to more general structures, cannot be applied here, as adhesivity fails for graphs with nesting. Other models of hierarchical graph transformation (as those proposed in [7, 10, 14]) are also not directly applicable, because of evident differences in the graphical models. Instead of defining a notion of rewriting over graphs with nesting from scratch, we propose to exploit a faithful encoding of such graphs to *term graphs*, as illustrated in [2] by exploiting an already known axiomatization of them [9]. In fact, term graph rewriting [15] is an established research field with a well-understood theory, and through this encoding we expect to borrow not only an adequate definion of rewriting for our hierarchical graphs, but possibly also results concerning termination, confluence, concurrency, as well as powerful analysis techiques. All the concepts will be illustrated by showing how the Calculus of Mobile Ambient [8] can be conveniently encoded into our framework.

References

1. Boreale, M., Bruni, R., De Nicola, R., Loreti, M.: Sessions and pipelines for structured service programming. In: Barthe, G., de Boer, F.S. (eds.) FMOODS 2008. LNCS, vol. 5051, pp. 19–38. Springer (2008)
2. Bruni, R., Corradini, A., Gadducci, F., Lluch Lafuente, A., Montanari, U.: On gs-monoidal theories for graphs with nesting (2010), submitted
3. Bruni, R., Corradini, A., Montanari, U.: Modeling a service and session calculus with hierarchical graph transformation (2010), submitted
4. Bruni, R., Gadducci, F., Lluch Lafuente, A.: An algebra of hierarchical graphs and its application to structural encoding. Scientific Annals in Computer Science (2010), to appear
5. Bruni, R., Gadducci, F., Lluch Lafuente, A.: A graph syntax for processes and services. In: Jianwen, S., Laneve, C. (eds.) WS-FM 2009. LNCS, vol. 6194. Springer (2010)
6. Bruni, R., Melgratti, H.C., Montanari, U.: Theoretical foundations for compensations in flow composition languages. In: Palsberg, J., Abadi, M. (eds.) POPL 2005. pp. 209–220. ACM (2005)
7. Busatto, G., Kreowski, H.J., Kuske, S.: Abstract hierarchical graph transformation. Mathematical Structures in Computer Science 15(4), 773–819 (2005)

8. Cardelli, L., Gordon, A.D.: Mobile ambients. Theoretical Computer Science 240(1), 177–213 (2000)
9. Corradini, A., Gadducci, F.: An algebraic presentation of term graphs, via gs-monoidal categories. Applied Categorical Structures 7, 299–331 (1999)
10. Drewes, F., Hoffmann, B., Plump, D.: Hierarchical graph transformation. Journal on Computer and System Sciences 64(2), 249–283 (2002)
11. Ehrig, H., Habel, A., Padberg, J., Prange, U.: Adhesive high-level replacement categories and systems. In: Ehrig, H., Engels, G., Parisi-Presicce, F., Rozenberg, G. (eds.) ICGT 2004. LNCS, vol. 3256, pp. 144–160. Springer (2004)
12. Lack, S., Sobociński, P.: Adhesive and quasiadhesive categories. Theoretical Informatics and Applications 39(2), 511–546 (2005)
13. Milner, R.: Communicating and Mobile Systems. Cambridge University Press (1992)
14. Palacz, W.: Algebraic hierarchical graph transformation. Journal of Computer and System Sciences 68, 497–520 (2004)
15. Sleep, M., Plasmeijer, M., van Eekelen, M. (eds.): Term Graph Rewriting: Theory and Practice. Wiley (1993)