Research Summary

Analysing Graph Transformation Systems using Extended Methods from Constraint Handling Rules

Frank Raiser

Faculty of Engineering and Computer Sciences, Ulm University, Germany
frank.raiser@uni-ulm.de

1 Motivation

Constraint Handling Rules (CHR) [1] has become a general-purpose rule-based programming language throughout the last decade. The relations to many other formalisms have been investigated [2] and often results could be transferred from CHR to other formalisms, or vice versa.

Graph Transformation Systems (GTS) [3], which have been developed in the 60ies and have become increasingly popular, have not been compared to CHR before. GTS and CHR appear to be very similar on a cursory glance, as they are both non-deterministic rule-based state transition systems. However, the fact that confluence is decidable in CHR [1] and undecidable in GTS [4] warrants a closer investigation of the two formalisms. Hence, I want to apply analysis methods of CHR to GTS, concentrating on confluence analysis.

2 Existing Work

A solid mathematical basis for algebraic graph transformation systems is given in [3] that is based on category theory. CHR has been compared with several other formalisms [2], but to the best of my knowledge, there either was no comparison of confluence, or confluence of terminating systems was decidable in both. A direct comparison of GTS with CHR has not appeared in the literature before.

3 Goals

Through this thesis I want to gain further insights into the relation between these two important rule-based formalisms. I want to embed GTS in CHR and compare the approaches to deciding confluence. I also want to find out which elements of CHR are responsible for the decidability of confluence. Furthermore, I want to investigate similar analysis methods, like operational equivalence, that have been developed for CHR and transfer them to GTS.

4 Preliminary Results

I have succeeded in embedding GTS in CHR [5]. I also managed to give a characterization of the sufficient criterion for confluence of GTS in CHR using the notion of observable confluence [6]. The embedding also proved viable to transfer the notion of operational equivalence to GTS [7].

Confluence analysis requires to test CHR states for equivalence. The comparison of GTS confluence with that in CHR, together with results on the linear logic semantics of CHR gave further insights into state equivalence in CHR. Recently, I have succeeded in providing an axiomatic definition for it [8] with two significant results for CHR research: firstly, the correspondence between state equivalence and rule application, that has been taken for granted for over a decade, could be proved for the first time. And secondly, this work provides the foundation for a new view on CHR's operational semantics as a state transition system over equivalence states. Thus, the investigation of state equivalence in the context of confluence gave rise to results relevant to the operational semantics.

5 Open Issues and Expected Achievements

An open issue is to extend the notion of operational equivalence to get a stronger charaterization of equivalence of terminating GTS, similar to my confluence characterization. Furthermore, using my observations on confluence I intend to determine if a subclass of GTS exists that more closely resembles CHR such that confluence in this subclass becomes decidable.

References

1. Frühwirth, T.: Constraint Handling Rules. Cambridge University Press (2009) to appear.
2. Sneyers, J., Van Weert, P., Schrijvers, T., De Koninck, L.: As time goes by: Constraint Handling Rules – A survey of CHR research between 1998 and 2007. Accepted by *Journal of Theory and Practice of Logic Programming* (2008)
3. Ehrig, H., Ehrig, K., Prange, U., Taentzer, G.: Fundamentals of Algebraic Graph Transformation. Springer-Verlag (2006)
4. Plump, D.: Confluence of graph transformation revisited. In Middeldorp, A., van Oostrom, V., van Raamsdonk, F., de Vrijer, R.C., eds.: Processes, Terms and Cycles. Volume 3838 of Lecture Notes in Computer Science., Springer-Verlag (2005) 280–308
5. Raiser, F.: Graph Transformation Systems in CHR. In Dahl, V., Niemelä, I., eds.: Logic Programming, 23rd International Conference, ICLP 2007. Volume 4670 of Lecture Notes in Computer Science., Porto, Portugal, Springer-Verlag (September 2007) 240–254
6. Raiser, F., Frühwirth, T.: Strong joinability analysis for graph transformation systems in CHR. In: 5th International Workshop on Computing with Terms and Graphs, TERMGRAPH'09. (2009)
7. Raiser, F., Frühwirth, T.: Operational equivalence of graph transformation systems. In: Sixth International Workshop on Constraint Handling Rules. (2009) submitted.
8. Raiser, F., Betz, H., Frühwirth, T.: Equivalence of CHR states revisited. In: Sixth International Workshop on Constraint Handling Rules. (2009) submitted.