Constraints and Application Conditions: From Graphs to High-Level Structures

Hartmut Ehrig¹, Karsten Ehrig¹, Annegret Habel², and Karl-Heinz Pennemann²

¹ Technische Universität Berlin, Germany
  {ehrig,karstene}@cs.tu-berlin.de
² Carl von Ossietzky Universität Oldenburg, Germany
  {habel,k.h.pennemann}@informatik.uni-oldenburg.de

Abstract. Graph constraints and application conditions are most important for graph grammars and transformation systems in a large variety of application areas. Although different approaches have been presented in the literature already there is no adequate theory up to now which can be applied to different kinds of graphs and high-level structures. In this paper, we introduce an improved notion of graph constraints and application conditions and show under what conditions the basic results can be extended from graph transformation to high-level replacement systems. In fact, we use the new framework of adhesive HLR categories recently introduced as combination of HLR systems and adhesive categories. Our main results are the transformation of graph constraints into right application conditions and the transformation from right to left application conditions in this new framework.

1 Introduction

According to the requirements of several application areas the rules of a graph grammar have been equipped in [4] by a very general notion of application conditions. In a subsequent paper [8], the notion of application conditions is restricted to contextual conditions like the existence or non-existence of certain nodes and edges or certain subgraphs in the given graph. In [9], the authors introduce graphical consistency constraints, also called graph constraints, that express very basic conditions on graphs as e.g. the existence or uniqueness of certain nodes and edges in a graphical way.

Basic results for graph constraints and application conditions have been shown in [9, 10] first for the single and later in the double pushout approach for different kinds of graphs. Unfortunately there is no adequate theory up to now which can be applied not only to graphs but also to high-level structures in the sense of [5].

A new version of high-level replacement systems, called adhesive HLR systems, has been introduced in [6] combining HLR systems in the sense of [5] and adhesive categories (see [11]). This new framework has been used not only to re-formulate the basic results like local Church Rosser, Parallelism and Concurrency...
Theorem from [5], but also to present an improved version of the Embedding Theorem [3] and the local Confluence Theorem, known as Critical Pair Lemma [12]. Moreover it can be applied to all kinds of graphs and Petri nets satisfying the HLR1 and HLR2 conditions in [5] and also to typed attributed graphs in [7].

In this paper we use adhesive HLR categories and systems to improve and generalize the basic notions and results for constraints and application conditions from graphs to high-level structures. For this purpose we present an improved notion of graph constraints, based on positive and negative atomic constraints, and of application conditions, based on atomic conditional conditions. In our main theorems we show how to transform constraints into right application conditions, and right into left application conditions in the framework of adhesive HLR systems. As additional condition we only need finite coproducts and a suitable $E-M$-factorization which is valid in all our example categories.

The paper is organized as follows. In section 2 we present our improved notions of graph constraints and application conditions. In section 3 we give a short introduction of adhesive HLR categories together with some basic properties. Then we generalize graph constraints and application conditions to the framework of adhesive HLR categories. In section 4, we present the main results for graphs and high-level structures and give several illustrating examples for graphs and place transition nets. A conclusion including further work is given in section 5.

## 2 Constraints and Application Conditions for Graphs

In the following, we assume that the reader is familiar with the notions of graphs and graph morphisms, see e.g. [3, 2]. Graph constraints, first investigated by [9], allow to express basic conditions on graphs as e.g. the existence or uniqueness of certain nodes and edges in a graphical way.

**Definition 1 (graph constraint).** An atomic graph constraint is of the form $\text{PC}(a)$ or $\text{NC}(a)$ where $a: P \to C$ is an arbitrary graph morphism. It is said to be a positive or negative atomic graph constraint, respectively. A graph constraint is a Boolean formula over atomic graph constraints, i.e. every atomic graph constraint is a graph constraint and, for every index set $I$ and every family $(c_i)_{i \in I}$ of graph constraints, $\land_{i \in I} c_i$ and $\lor_{i \in I} c_i$ are graph constraints. A graph $G$ satisfies $\text{PC}(a)$ ($\text{NC}(a)$), written $G \models \text{PC}(a)$ ($\text{NC}(a)$), if for every injective morphism $p: P \to G$ there exists (does not exist) an injective morphism $q: C \to G$ such that $q \circ a = p$.

![Fig. 1. Satisfiability of atomic constraints.](image-url)