Nested Constraints and Application Conditions for High-Level Structures

Annegret Habel and Karl-Heinz Pennemann

Carl v. Ossietzky University of Oldenburg, Germany
{habel,k.h.pennemann}@informatik.uni-oldenburg.de

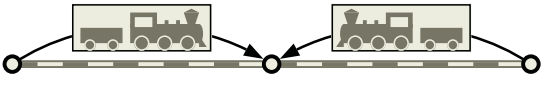
Abstract. Constraints and application conditions are most important for transformation systems in a large variety of application areas. In this paper, we extend the notion of constraints and application conditions to nested ones and show that nested constraints can be successively transformed into nested right and left application conditions.

1 Introduction

Constraints and application conditions are most important for transformation systems in a large variety of application areas, especially in the area of safety-critical systems e.g. the specification of railroad control systems and access control policies [10]. Application conditions for rules were investigated e.g. in [2, 6, 8, 10, 1]. They define classes of morphisms and thus restrict the applicability of their rules. Constraints, also called consistency constraints, were studied e.g. in [8, 10, 1]. They are properties on objects which have to be satisfied. In the graph case, simple constraints like the existence or uniqueness of certain nodes and edges can be expressed.

In this paper, we extend the existing theory on constraints and application conditions [1] to nested constraints and application conditions as proposed in [15]. We show that nested constraints can be transformed into nested right application conditions, and nested right application conditions can be transformed into nested left application conditions.

The transformation results are proved for high-level structures and are applied exemplarily for the case of graphs. The concepts are illustrated by a simple railroad system. The specification of a railroad system is given in terms of rail net graphs, constraints, rules for moving the trains, and application conditions. We study the integration of general rail net constraints into rail net application conditions for the movement of trains.



The paper is organized as follows. In section 2, we give a short introduction of adhesive HLR categories together with their basic properties. In section 3 we generalize constraints and application conditions to nested constraints and

H.-J. Kreowski et al. (Eds.): Formal Methods (Ehrig Festschrift), LNCS 3393, pp. 293–308, 2005.

application conditions in the framework of adhesive HLR categories. In sections 4, 5 and 6, we prove the main transformation results and give some applications of the results. A conclusion including further work is given in section 7.

2 Preliminaries

The main idea of high-level replacement systems is to generalize the concepts of graph replacement from graphs to all kinds of structures which are of interest in Computer Science and Mathematics. In the following sections, we will consider constraints and application conditions in adhesive HLR-categories (see [3]) and prove our transformation results on this general level. This has the advantage, that our results apply not only for graphs, but also for other high-level structures, e.g. typed attributed graphs, Petri-nets or algebraic specifications. We consider adhesive HLR-categories to benefit from the combined advantages of HLR and of adhesive categories [11]: the sound theory of the HLR framework and the much smoother requirements for adhesive categories in comparison with the variety of HLR preconditions.

Definition 1 (adhesive HLR-category). A category $\mathcal{C}$ with a morphism class M is called *adhesive HLR category*, if 1) M is a class of monomorphisms closed under compositions and decompositions ($g \circ f \in M$, $g \in M$ implies $f \in M$), 2) $\mathcal{C}$ has pushouts and pullbacks along M-morphisms, i.e. pushouts and pullbacks, where at least one of the given morphisms is in M, and M-morphisms are closed under pushouts and pullbacks, i.e. whenever a given morphism is in M, then the opposite morphism is in M, as well, and 3) pushouts in $\mathcal{C}$ along M-morphisms are VK-squares (see e.g. [3]).

Examples for adhesive HLR categories for the class M of all monomorphism include *Sets*, *Graphs*, *PT-Nets* and several other variants of graphs and nets, like typed, labeled and attributed graphs, hypergraphs and high-level nets. Further examples can be found, e.g. in [3].

Example 1 (rail net graph). The railroad system (similar to [12]) models the movement of one or more trains on a net of railroad tracks. The basic items are simple tracks from which the net is synthesized. The net together with trains on it forms the static part of the system, while movement of trains constitute the dynamic part. The static part is given by a labeled directed rail net graph:

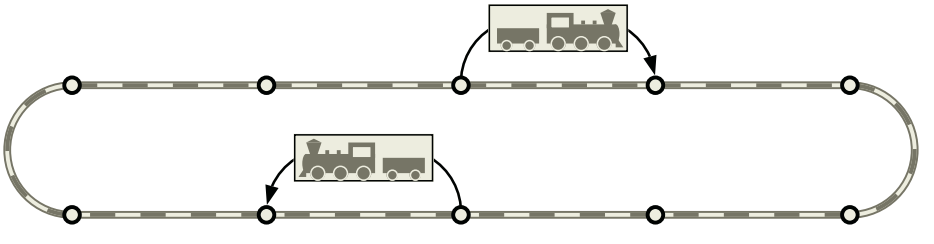


Fig. 1. A simple railway model.