Journal of Visual Languages and Computing 24 (2013) 192–206

Contents lists available at SciVerse ScienceDirect

Journal of Visual Languages and Computing

journal homepage: www.elsevier.com/locate/jvlc

Treewidth, pathwidth and cospan decompositions with applications to graph-accepting tree automata ☆,☆☆

Christoph Blume, H.J. Sander Bruggink, Martin Friedrich, Barbara König *

Universität Duisburg-Essen, Fakultät für Ingenieurwissenschaften, Abteilung für Informatik und Angewandte Kognitionswissenschaft, D-47048, Duisburg, Germany

ARTICLE INFO

Available online 2 November 2012

Keywords:
Cospans
Graph decompositions
Pathwidth
Treewidth
Tree automata

ABSTRACT

We will revisit the categorical notion of cospan decompositions of graphs and compare it to the well-known notions of path decomposition and tree decomposition from graph theory. More specifically, we will define several types of cospan decompositions with appropriate width measures and show that these width measures coincide with pathwidth and treewidth. Such graph decompositions of small width are used to efficiently decide graph properties, for instance via graph automata. Hence we will give an application by defining graph-accepting tree automata, thus integrating previous work by Courcelle into the setting of cospan decompositions. Furthermore we will show that regardless of whether we consider path or tree decompositions, we arrive at the same notion of recognizability.

1. Introduction

In graph rewriting the notion of cospan plays a major role: cospans can be seen as graphs equipped with an inner and an outer interface and they can be used as (atomic) building blocks for constructing or decomposing larger graphs. Furthermore cospans are a means to cast graph rewriting into the setting of reactive systems [1,2].

In graph theory there are different notions for decomposing graphs: path and tree decompositions [3], which at first glance seem to have a very different flavor than cospan decompositions. These notions lead to width measures such as pathwidth and treewidth and they are used to specify how similar a graph is to a path or a tree. Treewidth plays a major role in complexity theory: for instance Courcelle's theorem [4] states that every graph property that can be specified in monadic second-order graph logic can be checked in linear time on graphs of bounded treewidth. Furthermore there are intuitive game characterizations (robber and cops games) for treewidth.

In this paper we show that, when seen from the right perspective, graph decompositions based on cospans are in fact very similar to path and tree decompositions. In order to be able to state this formally we classify several types of cospan decompositions, which are sequences of cospans (with varying additional conditions). Obtaining the decomposed graph amounts to taking the colimit of the resulting diagram. We define width measures based on such decompositions and show that the width measures all coincide with pathwidth. In the second part of the paper the results are repeated for tree-like decompositions and treewidth, where the tree-like decompositions are trees where the edges are labeled with spans or cospans, and the decomposed graph is again obtained by taking the colimit.

Additionally, we define automata for such decompositions. For cospan decompositions we use automaton functors [5], which in [6] were used to check invariants

☆ Supported by the DFG-project GaReV.
☆☆ This paper has been recommended for acceptance by Shi Kho Chang

* Corresponding author. Tel.: +49 203 3793397; fax: +49 203 3793557.
E-mail addresses: christoph.blume@uni-due.de (C. Blume), sander.bruggink@uni-due.de (H.J.S. Bruggink), martin.friedrich@stud.uni-due.de (M. Friedrich), barbara_koenig@uni-due.de (B. König).

1045-926X/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.jvlc.2012.10.002

of graph transformation systems. Automaton functors work by decomposing a graph into (atomic) cospans, and then running a finite automaton on the sequence. This approach is an extension of the work by Courcelle and others on recognizable graph languages [4], which are in turn equivalent to the notion of inductive graph properties [7].

For tree-like decompositions, we define consistent tree automata. These automata input so-called term decompositions, which are tree-like decompositions in the form of first-order terms. They are usual tree automata [8,9] with the additional requirement that their behavior and acceptance are the same for different term decompositions of the same graph. A main result of the paper is that automaton functors and consistent tree automata accept the same language class, namely the class of recognizable graph languages.

As far as we know there have been only few investigations into the notions of pathwidth and treewidth in the context of graph rewriting. We are mainly aware of the relation between context-free (or hyperedge replacement) grammars and bounded treewidth that is discussed in [10–12]. It is shown that the language generated by a context-free grammar has always bounded treewidth, that is, there is an upper bound for the treewidth of every graph in the language. This also implies the well-known result that the language of all graphs is not context-free.

Interest in the relation between tree decompositions and graph rewriting seems to have declined, but in our opinion this area has a lot of potential for an increased interaction of graph transformation and graph theory, since graph decompositions and width measures are still of central interest to the graph theory community. As far as we are aware, the relation between cospan decompositions and tree and path decompositions has never been formally investigated and while the main ideas are fairly straightforward it turns out that there are some subtle issues to consider when translating one representation into the other. For instance, we found that there is more than one possible translation and more than one width measure.

The paper is organized as follows: In Section 2 we will introduce the preliminaries such as cospans, graph decompositions and tree automata. Then in Section 3 we will have a closer look at cospans, identifying also atomic cospans as building blocks. Then in Section 4 we will compare cospan decompositions with path decompositions and in Section 5 we will define graph automata as automaton functors for the category of cospans of graphs. In Section 6 we compare tree-like cospan decompositions with tree decompositions, and in Sections 7 and 8 we define term decompositions and tree automata operating on the them. Finally we will conclude with Section 9.

This paper is based on [13]. Sections 3, 4 and 6, in which the correspondence of the various types of decomposition is discussed, correspond to that paper; Sections 5, 7 and 8, which address (tree) automata, are new.

2. Preliminaries

By $\mathbb{N}_k$ we denote the set $\{1,\ldots,k\}$. The set of finite sequences over a set A, including the empty sequence ϵ, is denoted by A^*. Composition of two sequences $\vec{a}$ and $\vec{b}$ is denoted by juxtaposition, that is by $\vec{a}\,\vec{b}$.

If $f: A \to B$ is a function from A to B, we will implicitly extend it to subsets and sequences; for $A' \subseteq A$ and $\vec{a} = a_1 \ldots a_n \in A^*$: $f(A') = \{f(a) \mid a \in A'\}$ and $f(\vec{a}) = f(a_1) \ldots f(a_n)$.

2.1. Categories and cospans

We presuppose a basic knowledge of category theory. For an arrow f from A to B we write $f: A \to B$ and define $dom(f) = A$ and $cod(f) = B$. For arrows $f: A \to B$ and $g: B \to C$, the composition of f and g is denoted $(f;g): A \to C$. The category **Rel** has sets as objects and relations as arrows. Its subcategory **Set** has only the functional relations (functions) as arrows.

An initial object of a category **C** is an object 0, such that for each object $K \in \mathbf{C}$ there exists a unique morphism from 0 to K, which is denoted by $!_K: 0 \to K$.

Let **C** be a category in which all pushouts exist. A *concrete cospan* in **C** is a pair $\langle c_L, c_R \rangle$ of **C**-arrows with the same codomain: $J \xrightarrow{c_L} G \xleftarrow{c_R} K$. Two concrete cospans are isomorphic if their middle objects are isomorphic (such that the isomorphism commutes with the component morphisms of the concrete cospan). A *cospan* is an isomorphism class of concrete cospans. In the following we will confuse cospans and concrete cospans, in the sense that we represent cospans by giving a representative of the isomorphism class.

Composition of two cospans $\langle c_L, c_R \rangle, \langle d_L, d_R \rangle$ is computed by taking the pushout of the arrows c_R and d_L. Cospans are the arrows of so-called cospan categories. That is, for a category **C** with pushouts, the category *Cospan*(**C**) has the same objects as **C**. The isomorphism class of a cospan $c: J \xrightarrow{c_L} G \xleftarrow{c_R} K$ in **C** is an arrow from J to K in *Cospan*(**C**) and will be denoted by $c: J \leftrightarrow K$.

Spans are the dual notion of cospans, that is, they are (equivalence classes of) pairs of morphisms with the same domain.

Colimits can be seen as "generalized" pushouts. Given a collection (diagram) D of objects $\{A_1, \ldots, A_n\}$ and morphisms between them, the *colimit of D* is an object B together with morphisms $\mu_i: A_i \to B$ such that the diagram commutes, and for each object B' and morphism $\mu'_i: A_i \to B'$ where the diagram commutes, it holds that there exists a unique $h: B \to B'$ such that everything commutes. We will write $Colim(D) = B$ in this case.

2.2. Graphs and decompositions

A *hypergraph* over a set of labels Σ (in the following also simply called *graph*) is a structure $G = \langle V, E, att, lab \rangle$, where V is a finite set of nodes, E is a finite set of edges, $att: E \to V^*$ maps each edge to a finite sequence of nodes attached to it, and $lab: E \to \Sigma$ assigns a label to each edge. The size of the graph G, denoted $|G|$, is defined to be the cardinality of its node set, that is $|G| = |V|$. A *discrete graph* is a graph without edges; the discrete graph with node set $\mathbb{N}_k$ is denoted by D_k. We denote the *empty graph* by $\emptyset$ instead of D_0.

A graph morphism from a graph $G = \langle V_G, E_G, att_G, lab_G \rangle$ to a graph $H = \langle V_H, E_H, att_H, lab_H \rangle$ is a pair of maps