UNIVERSITÄT DUISBURG ESSEN

INFORMATIK

ABTEILUNG FÜR INFORMATIK UND ANGEWANDTE KOGNITIONSWISSENSCHAFT

FAKULTÄT FÜR INGENIEURWISSENSCHAFTEN

Technischer Bericht Nr. 2008-03

On the Recognizability of Arrow and Graph Languages

H. J. Sander Bruggink
Barbara König

10.07.2008

ISSN 1863-8554

IMPRESSUM:

Technische Berichte der Abteilung für Informatik und Angewandte Kognitionswissenschaft, Universität Duisburg-Essen

ISSN 1863-8554

Herausgeber:

Abteilung für Informatik und Angewandte Kognitionswissenschaft
Fakultät für Ingenieurwissenschaften
Universität Duisburg-Essen
Campus Duisburg
47048 Duisburg

http://duepublico.uni-duisburg-essen.de/informatik/berichte.xml

On the Recognizability of Arrow and Graph Languages*

H. J. Sander Bruggink and Barbara König

Universität Duisburg-Essen
{sander.bruggink,barbara_koenig}@uni-due.de
www.ti.inf.uni-due.de/people/{bruggink,koenig}

Abstract. In this paper we give a category-based characterization of recognizability. A recognizable subset of arrows is defined via a functor into the category of relations on sets, which can be seen as a straightforward generalization of a finite automaton. In the second part of the paper we apply the theory to graphs, and we show that our approach is a generalization of Courcelle's recognizable graph languages.

1 Introduction

Regular languages have been studied extensively in computer science and they have a large number of applications. For instance, more recent approaches take advantage of regular languages for model-checking [1] and termination analysis [11]. The notion of regularity can be straightforwardly generalized to trees and tree automata. Hence one can talk about regular tree languages and exploit the convenient closure properties that these languages enjoy.

Hence, as a next step, it is natural to ask for a natural notion of regular graph languages. There have been several attempts to answer this question [23, 18, 22, 5], all arriving at slightly different notions of regularity (also called recognizability), of which the notion of Courcelle [5, 7] emerged as the one which is widely accepted. To our knowledge these notions of recognizability have not been applied extensively to verification and it is not entirely clear how they relate to graph transformation specified by double-pushout rewriting [4], one of the standard graph transformation approaches.

Courcelle focuses on the notion of recognizability—which is equivalent to regularity in the case of word languages—and which characterizes languages as the inverse image of a monoid morphism from the monoid of words (with concatenation) into a finite monoid. Specifically, he extends the more general notion of Mezei and Wright [17] from recognizability in one-sorted algebras to recognizability in many-sorted algebras by considering a specific algebra of graphs.

Here we compare the algebraic notion of recognizability by Courcelle with a categorical notion of recognizability, strongly related to composition-representative subsets introduced by Griffing [12].

* Supported by the DFG project SANDS.

We show that if we use the notion of Griffing and work in the category of cospans of graphs, we recover exactly the notion of recognizability proposed by Courcelle. Although both approaches rely on the same basic ideas, the proof is non-trivial. Furthermore it gives us a close relation with the double-pushout approach which can be characterized as a reactive system over cospans [21].

In addition we extend the notion of Griffing with a so-called automaton functor that—when instantiated to the word case—specializes to the notion of deterministic or non-deterministic finite automaton. We show that standard constructions on finite automata such as determinization or minimization can also performed on automaton functors.

We will proceed as follows: in Sect. 2 we will briefly introduce the necessary concepts of category theory, graphs and graph morphisms. In Sect. 3, we will introduce our category-theoretic notion of recognizable graph language, and in Sect. 4 we show that this enjoys useful properties, such as closure properties. In Sect. 5 we will apply our notion of recognizable arrow languages to define a notion of recognizable language of graphs, by considering the category of cospans of graphs, and we show that we can restrict ourselves to cospans between discrete graphs, i.e. graphs consisting of nodes only, without affecting the notion of recognizability. Finally, we show that our approach is equivalent to Courcelle's notion of recognizability. All proofs which are not in the main body of the paper can be found in the appendix.

2 Preliminaries

We briefly review the concepts from category theory, graphs and graph morphisms that will play a role in this paper. For more detailed introductions we refer to [16] and [10].

2.1 Category theory

We presuppose basic knowledge of category theory. The identity arrow of an object G will be denoted by id_G (or just by id if G is clear from the context). Furthermore, arrow composition will be denoted by either ;, where the composed arrow which applies first f and then g is denoted by $f \ ; g$. The category $\mathcal{Rel}$ is the category which has sets as objects, relations as arrows and relation composition as composition operator. The category $\mathcal{Set}$ is the subcategory in which all arrows are in fact total functions. We mention here that a functor $\mathcal{A}$ from some category $\mathscr{C}$ into $\mathcal{Set}$ is easily extended to a functor from $\mathscr{C}$ to $\mathcal{Rel}$ by postcomposing it with the embedding functor.

In the rest of this section we define the more advanced concept of a cospan category, which will be used in Sect. 5. The idea of the cospan category is to have a category which has cospans as arrows.

Let $\mathscr{C}$ be a category in which all pushouts exist. A cospan c is a pair of $\mathscr{C}$-arrows $\langle c_\mathrm{L}, c_\mathrm{R}\rangle$ with the same codomain:

$$c\colon J \xrightarrow{c_\mathrm{L}} G \xleftarrow{c_\mathrm{R}} K \ .$$

Above, J and K are the domain (or *inner interface*) and codomain (or *outer interface*) of the cospan c, resp., i.e., the cospan can be considered as an arrow from J to K. The identity cospan for an object G is the cospan consisting of twice the identity arrow of G: $\langle \mathsf{id}_G, \mathsf{id}_G \rangle \colon G \to G \leftarrow G$. Let $c\colon J \to G \leftarrow M$ and $d\colon M \to H \leftarrow K$ be cospans (where the codomain of c equals the domain of d): The composition of c and d is defined by the commuting diagram

$$\begin{tikzcd} & & M \arrow[dl, "c_{\mathrm{R}}"'] \arrow[dr, "d_{\mathrm{L}}"] & & \\ J \arrow[r, "c_{\mathrm{L}}"] & G \arrow[dr, "f"'] \arrow[rr, phantom, "(\mathrm{PO})"] & & H \arrow[dl, "g"] & K \arrow[l, "d_{\mathrm{R}}"'] \\ & & M' & & \end{tikzcd}$$

where the middle diamond is a pushout, and the resulting composed cospan is defined as:

$$(c \mathbin{;} d)\colon J \xrightarrow{c_{\mathrm{L}} ; f} M' \xleftarrow{d_{\mathrm{R}} ; g} K \ .$$

We want to form a category which has cospans as arrows, but in order to have the new category satisfy all axioms of category theory, we have to do some work. Let two cospans

$$c\colon J \xrightarrow{c_{\mathrm{L}}} G \xleftarrow{c_{\mathrm{R}}} K \quad \text{and} \quad d\colon J \xrightarrow{d_{\mathrm{L}}} H \xleftarrow{d_{\mathrm{R}}} K$$

with the same interfaces be given. We define the equivalence relation $\sim$ as follows: $c \sim d$ if there exists an isomorphism k from G to H, such that $c_{\mathrm{L}} \mathbin{;} k = d_{\mathrm{L}}$ and $c_{\mathrm{R}} \mathbin{;} k = d_{\mathrm{R}}$. A *semi-abstract cospan* is a $\sim$-equivalence class of cospans. Note that all members of a semi-abstract cospan have the same domain and codomain, and we define the domain and codomain of the semi-abstract cospan to be the domain and codomain of its members.

Now, the category $\mathit{Cospan}(\mathscr{C})$ is defined as the category which has the objects of $\mathscr{C}$ as objects, and semi-abstract cospans as arrows.

2.2 Graphs and graph morphisms

Let a set Σ of labels be given. A *hypergraph* G (in the following also called a *graph*) is a four-tuple $\langle V_G, E_G, \mathsf{att}_G, \mathsf{lab}_G \rangle$, where V_G is the set of nodes of G, E_G is the set of edges, $\mathsf{att}_G\colon E_G \to V_G^*$ (where V_G^* is the set of sequences of elements of V_G) is the *attachment function* and $\mathsf{lab}_G\colon E_G \to \Sigma$ is the *labeling function*. By $\emptyset$ we denote the empty graph. A *graph morphism* f between two graphs G and H is a pair $\langle f_{\mathrm{V}}, f_{\mathrm{E}} \rangle$ of functions $f_{\mathrm{V}}\colon V_G \to V_H$ and $f_{\mathrm{E}}\colon E_G \to E_H$ such that the following hold:

$$f_{\mathrm{E}} \mathbin{;} \mathsf{lab}_H = \mathsf{lab}_G \quad \text{and} \quad f_{\mathrm{E}} \mathbin{;} \mathsf{att}_H = \mathsf{att}_G \mathbin{;} f_{\mathrm{V}}^*$$

where f_{V}^* is the natural extension of f_{V} to sequences.

We define the category $\mathcal{HGraph}$ of hypergraphs as the category which has as objects finite (hyper)graphs, and as arrows graph morphisms. Concretely, taking a pushout of morphisms $f: U \to G$, $g: U \to H$ in the category of graphs means to take the disjoint union of G and H and to factor through the smallest equivalence $\equiv$ (on nodes and edges) that satisfies $f(x) \equiv g(x)$ for all items (i.e., for all nodes and edges) x of U.

In particular, we will work with the category $\mathcal{C}ospan(\mathcal{HG}raph)$. Cospans of graphs are intimately connected with the double-pushout (DPO) approach to graph rewriting [21]. DPO rewriting rules are spans of graphs of the form

$$p\colon L \xleftarrow{\varphi_{\mathrm{L}}} I \xrightarrow{\varphi_{\mathrm{R}}} R \ .$$

We consider the cospans $\ell\colon \emptyset \to L \overset{\varphi_{\mathrm{L}}}{\leftarrow} I$ and $r\colon \emptyset \to R \overset{\varphi_{\mathrm{R}}}{\leftarrow} I$ as left and right-hand sides. Now, for a graph G let $[G]\colon \emptyset \to G \leftarrow \emptyset$ be the cospan consisting of G with empty source and target. Then DPO rewriting can also be defined as follows: the graph G rewrites to H by applying rule p if and only if $[G] = \ell \,;\, c$ and $[H] = r \,;\, c$ for some cospan c.

3 Recognizable Languages of Arrows

We consider a fixed category $\mathscr{C}$. In order to be able to talk about sets, we will require that $\mathscr{C}$ is *locally small*, i.e., for every two objects the class of arrows between them is a set, called *homset*. Furthermore in the following a subset of a homset will be called an *(arrow) language*.

Definition 3.1 (Recognizability). *Let $\mathscr{C}$ be a category. We consider a functor $\mathcal{A}\colon \mathscr{C} \to \mathcal{R}el$ where every object X of $\mathscr{C}$ is mapped to a finite set $\mathcal{A}(X)$ (called the set of* states *of X) and every arrow $f\colon X \to Y$ is mapped to a relation $\mathcal{A}(f) \subseteq \mathcal{A}(X) \times \mathcal{A}(Y)$. We assume that every set $\mathcal{A}(X)$ contains a distinguished set $\mathrm{I}^{\mathcal{A}}_X$ of* start states *and a distinguished set $\mathrm{F}^{\mathcal{A}}_X$ of* final states *as subsets.*

The functor $\mathcal{A}$ is also called automaton functor*. It is called* deterministic *whenever every relation $\mathcal{A}(f)$ is a function and every $\mathrm{I}^{\mathcal{A}}_X$ contains exactly one element; this element will be denoted by $\mathrm{i}^{\mathcal{A}}_X$.*

*Let J, K be two objects. The (J,K)-*language *$L_{J,K}(\mathcal{A})$ (of arrows from J to K) is defined as follows:*

> *$f\colon J \to K$ is contained in $L_{J,K}(\mathcal{A})$ if and only if there exist $s \in \mathrm{I}^{\mathcal{A}}_J$, $t \in \mathrm{F}^{\mathcal{A}}_K$ which are related by $\mathcal{A}(f)$.*

A language $L_{J,K}$ of arrows from J to K is recognizable in *$\mathscr{C}$ if it is the (J,K)-language of a an automaton functor $\mathcal{A}\colon \mathscr{C} \to \mathcal{R}el$.*

The intuition behind the definition is to have a mapping into a finite domain that respects compositionality and identities, that is, which is a functor. The functor property ensures that decomposing the arrow in different ways does not affect acceptance in any way. This is different from the case of words where there is essentially only one way to decompose a word into atomic components.

In a sense $\mathcal{A}(f)$ gives the transition relation of the finite automaton, only that here we do not have a single set of states, but a set of states for every object. This means that we have infinitely many sets (which corresponds to the infinitely many sorts in the case of [5]).

Example 3.2. Let $\mathcal{F}$ be a nondeterministic finite automaton over the alphabet Σ, with state set Q, start states $I \subseteq Q$, final states $F \subseteq Q$ and transition relation $\delta \subseteq Q \times \Sigma \to \mathcal{P}(Q)$. We consider the free monoid of Σ, i.e., the category with one object X and words over Σ as arrows from X to X. We construct the following automaton functor $\mathcal{A}$ which recognizes a language L if and only if L is accepted by $\mathcal{F}$.

- $\mathcal{A}(X) = Q$, with $\mathrm{I}^{\mathcal{A}}_X = I$ and $\mathrm{F}^{\mathcal{A}}_X = F$;
- for $w \in \Sigma^*$, $\mathcal{A}(w) = \{\langle s,t\rangle \mid t \in \hat{\delta}(s,w)\}$, where $\hat{\delta}$ is the extension of δ from letters to words given by
  - $\hat{\delta}(s,\epsilon) = \{s\}$;
  - $\hat{\delta}(s,wa) = \bigcup\{\delta(s',a) \mid s' \in \delta(s,w)\}$.

Example 3.3. Also tree automata can be seen as a special case of our notion of automaton functor: take as the category a Lawvere theory where objects are natural numbers and arrows from m to n are n-tuples of terms with m holes. Arrow composition is the usual term substitution.

As an example take two unary function symbols f, g, a binary function symbol h and a constant a. Now, the 3-tuple $\langle f(x_1), h(x_1,x_2), g(f(a))\rangle$ has two holes, i.e., two variables x_1, x_2. Hence it can be regarded as an arrow from 2 to 3. Then trees can be represented as arrows from 0 to 1, since they are single terms (or 1-tuples of terms) without variables.

Note that this specific instantiation has similarly been considered in [12].

Two automaton functors are equivalent if they recognize the same language. Sometimes we are mainly interested in languages of arrows which start at a fixed object, for instance the initial object; therefore we parametrize the notion of equivalence with a source object.

Definition 3.4. *Let $\mathcal{A}$ and $\mathcal{B}$ be automaton functors. $\mathcal{A}$ and $\mathcal{B}$ are said to be J-*equivalent *if they recognize the same languages $L_{J,K}$ of arrows from J to K, for arbitrary K. They are* equivalent *if they are J-equivalent for all $\mathscr{C}$-objects J.*

It is also possible to find a characterization of recognizable language in terms of congruence classes, similar to Myhill-Nerode equivalences in the case of regular string languages.

Definition 3.5 (Congruence). *Let $\mathscr{C}$ be a category. Let a family of relations*

$$\equiv_{\mathrm{R}} = \{R_{J,K} \mid J, K \text{ are objects of } \mathscr{C}\}$$

be given, where the components $R_{J,K}$ are equivalence relations on $\mathscr{C}$-arrows from J to K. We call $\equiv_{\mathrm{R}}$ a (right) congruence if the following holds for all arrows $a, a' \colon J \to K$, $b \colon K \to M$:

If $a\ R_{J,K}\ a'$, then $(a \,;\, b)\ R_{J,M}\ (a' \,;\, b)$.

A congruence $\equiv_{\mathrm{R}}$ is locally finite *if each $R_{J,K} \in \equiv_{\mathrm{R}}$ is an equivalence relation of finite index (i.e., it has finitely many equivalence classes).*

Let $a\colon J \to K$ be an arrow. In the following we will write $[\![a]\!]_{R_{J,K}}$, or simply $[\![a]\!]_{\mathrm{R}}$ if J and K are clear from the context, to denote the congruence class of which a is a member. We will also usually write $a \equiv_{\mathrm{R}} b$ instead of $a\ R_{J,K}\ b$.

Proposition 3.6. *Let $\mathscr{C}$ be a category, J, K $\mathscr{C}$-objects and $L_{J,K}$ a set of $\mathscr{C}$-arrows from J to K. The language $L_{J,K}$ is recognizable in $\mathscr{C}$, if and only if there exists a locally finite congruence $\equiv_{\mathrm{R}}$ such that $L_{J,K}$ is the union of some equivalence classes of $R_{J,K}$.*

Note that the proof of Prop. 3.6 only works if we fix a specific start object. Dually it would also have been possible to fix a target object and to concentrate on left congruences. But for our examples and for the comparison to the work of Courcelle fixing a start object seems to be more natural.

The paper by Griffing [12] does not introduce the notion of an automaton functor, but it shows that composition-representative subsets (which are our recognizable languages) can be characterized via locally finite congruences. In addition the paper gives another—equivalent—characterization via a functor from $\mathscr{C}$ into a category with finite homsets. The recognizable languages are then the preimages of subsets of a finite homset.

4 Determinism, Closure Properties and Minimization

One of the advantages of our characterization over characterizations utilizing finite homsets or congruences, is that we can talk explicitly about determinization and minimization. In this section we consider these two notions as well as closure properties. These results, different from the results in Sect. 5, are not particularly deep or surprising; usually they can be shown quite straightforwardly. We show them here for completeness and in order to illustrate that our notion of recognizability is reasonable.

Proposition 4.1. *For every automaton functor, there exists an equivalent deterministic automaton functor.*

Proof. (Sketch.) The construction is more or less equivalent to the case of finite automata: we replace every set of states by its powerset. □

Proposition 4.2 (Closure under boolean operators). *Suppose we have two recognizable languages of arrows, $L^1_{J,K}$ and $L^2_{J,K}$. Then also $L^1_{J,K} \cap L^2_{J,K}$, $L^1_{J,K} \cup L^2_{J,K}$ and $(L^1_{J,K})^{\mathrm{C}}$ (the complement of $L^1_{J,K}$) are recognizable.*

Proof. (Sketch.) Again the construction resembles the case of finite automata: we work with deterministic automaton functors, take the cross product of the state sets and define the final states appropriately. □

In the rest of this section we show that each deterministic automaton functor has a unique equivalent minimal one. The construction of a minimal automaton is analogous to the usual construction for finite automata: first we remove unreachable states, and then we fuse indistinguishable states.

The notion of minimality depends on the exact notion of equivalence we use. If we fix the source object in advance, the equivalence classes grow, and therefore smaller minimal automaton functors may be found. The constructions in the general case and the case with fixed source object are exactly the same, except for the fact that with a fixed source more states may be unreachable.

We give here two key definitions, of minimality and reachability, and mention the minimization result.

Definition 4.3. *Let $\mathcal{A}\colon \mathscr{C} \to \mathcal{R}el$ and $\mathcal{B}\colon \mathscr{C} \to \mathcal{R}el$ be automaton functors. We define:*

$$\mathcal{A} \leq \mathcal{B} \text{ if for all } G \in \mathcal{O}bj(\mathscr{C}),\ |\mathcal{A}(G)| \leq |\mathcal{B}(G)|$$

An automaton functor $\mathcal{A}$ is (J-)minimal *if, for all (J-)equivalent automaton functors $\mathcal{B}$ it holds that $\mathcal{A} \leq \mathcal{B}$.*

Definition 4.4. *Let $\mathcal{A}\colon \mathscr{C} \to \mathcal{R}el$ be an automaton functor and K an object of $\mathscr{C}$. A state $s \in \mathcal{A}(K)$ is* J-reachable*, if there exists an arrow $c\colon J \to K$ such that $s \in \mathcal{A}(c)(\mathrm{I}_J^{\mathcal{A}})$. The state s is* reachable*, if it is J-reachable for some J.*

$\mathcal{A}$ is said to be fully (J-)reachable *if for all $\mathscr{C}$-objects K and states $s \in \mathcal{A}(K)$, s is (J-)reachable.*

Proposition 4.5. *For each deterministic automaton functor $\mathcal{A}$, there exists a minimal, (J-)equivalent, deterministic automaton functor $\mathcal{A}^{\min}$ which recognizes the same language and which is unique up to isomorphism.*

Note that the results of this section depend on the specific nature of the categories $\mathcal{R}el$ and $\mathcal{S}et$. It would be interesting to find an abstract characterization which still allows the techniques and constructions of this section.

5 Recognizable Graph Languages

In this section we apply the theory of the previous sections to recognizing languages of graphs. First, we show how graph languages can be recognized by considering the category of (semi-abstract) cospans of graphs. Then we briefly introduce Courcelle's algebraic notion of recognizable graph languages, and finally we show that we can recognize the same graph languages as Courcelle.

5.1 Recognizing Languages of Graphs by Cospans

In the following, the category under consideration will be $\mathcal{C}ospan(\mathcal{HG}raph)$, i.e., the category of cospans of graphs, or put differently, the category of graphs with inner and outer interfaces. If we want to talk only about graphs without interfaces, we can restrict ourselves to languages of cospans with empty interfaces, i.e., cospans where the source and target is the empty graph.

Definition 5.1. *A set L of graphs is recognizable whenever*

$$L_{\emptyset,\emptyset} = \{[G]\colon \emptyset \to G \leftarrow \emptyset \mid G \in L\}$$

is a recognizable language in $\mathcal{C}ospan(\mathcal{HG}raph)$.

Below we give an example for a recognizable graph language. It is not surprising that it is recognizable since it is definable in monadic second-order graph logic and all such languages are recognizable in the sense of Courcelle [5, 7]. However, the example provides some intuition into the notion of recognizability.

Example 5.2 (k-Colorability). We set $\mathbb{N}_k = \{0, \dots, k-1\}$. Let G be a graph. A k-coloring of G is a function $f\colon V_G \to \mathbb{N}_k$ such that for all $e \in E_G$ and for all $v_1, v_2 \in \mathsf{att}_G(e)$ it holds that $f(v_1) \neq f(v_2)$ if $v_1 \neq v_2$. We show that the language of all k-colorable graphs is recognizable, by considering the following automaton functor $\mathcal{A}\colon \mathcal{C}ospan(\mathcal{HG}raph) \to \mathcal{R}el$:

– Every graph J is mapped to $\mathcal{A}(J)$, the set of all valid k-colorings of J:

$$\mathcal{A}(J) = \{f\colon V_J \to \mathbb{N}_k \mid f \text{ is a valid } k\text{-coloring of } J\} \enspace .$$

– For a cospan $c\colon J \to G \leftarrow K$ the relation $\mathcal{A}(c)$ relates two colorings f_J, f_K, whenever there exists a coloring f for G such that $f(c_{\mathrm{L}}(v)) = f_J(v)$ for every node $v \in V_J$ and $f(c_{\mathrm{R}}(v)) = f_K(v)$ for every node $v \in V_K$.

Specifically we have that $\mathcal{A}(\emptyset) = \{\emptyset\}$ where $\emptyset$ is the empty coloring. Then in order to accept all k-colorable graphs with empty interfaces we take $\mathrm{I}^{\mathcal{A}}_{\emptyset} = \mathrm{F}^{\mathcal{A}}_{\emptyset} = \{\emptyset\}$: a graph $\emptyset \to G \leftarrow \emptyset$ is accepted whenever the two empty mappings are related.

Note that it is well known that k-colorability of graphs is an *NP*-complete property. Intuitively this manifests itself in the fact that interfaces may grow unboundedly, leading to an exponential explosion of the size of the state sets. However, if we restrict ourselves to graphs of bounded treewidth, there are efficient algorithms for k-colorability (see also the related discussion in the conclusion).

Example 5.3. Let H be a fixed graph. We consider the language L_H of all graphs G for which a morphism $f\colon G \to H$ exists. The language L_H is recognizable whenever H is finite.

The functor $\mathcal{A}$ associates to every graph J the set of all morphisms $J \to H$. For a cospan $c\colon J \to G \leftarrow K$ it relates a morphism $f_J\colon J \to H$ to a morphism $f_K\colon K \to H$ whenever there exists a morphism $f\colon G \to H$ such that $c_{\mathrm{L}} \,;\, f = f_J$ and $c_{\mathrm{R}} \,;\, f = f_K$. All states are initial and final.

$$\begin{tikzcd}
J \arrow[r, "c_{\mathrm{L}}"] \arrow[dr, "f_J"'] & G \arrow[d, dotted, "f"] & K \arrow[l, "c_{\mathrm{R}}"'] \arrow[dl, "f_K"] \\
& H &
\end{tikzcd}$$

This is a weaker notion than recognizability and has been considered before (see for instance [14, 3]).

5.2 Robustness

We now show the robustness of recognizability by restricting ourselves to injective, edge-injective and discrete interfaces. These results will also be important for the comparison with Courcelle's notion of recognizability (see Sect. 5.4).

We use proof principles already explored in [9] where robustness proofs are based on the characterization of recognizable languages in terms of locally finite congruences (see Def. 3.5). The results and proofs all follow the same lines: let $\mathscr{D}$

be a subcategory of $\mathscr{C}$ and let J, K be two objects of $\mathscr{D}$. We want to show that every language of arrows from J to K is recognizable in $\mathscr{C}$ if and only if it is recognizable in $\mathscr{D}$. The direction from left to right is obvious, we simply restrict the congruence or the automaton functor accordingly. The real challenge is the direction from right to left. In this case take a congruence $\equiv_{\mathrm{R}}$ on $\mathscr{D}$ and construct a congruence $\equiv'_{\mathrm{R}}$ on $\mathscr{C}$ that is locally finite and refines $\equiv_{\mathrm{R}}$ when restricted to $\mathscr{D}$. Since $\equiv'_{\mathrm{R}}$ refines $\equiv_{\mathrm{R}}$ any union of equivalence classes can still be represented as a union of (possibly more) equivalence classes, and hence recognizability is preserved.

We now show the convenient fact that restricting our attention to cospans with injective interfaces does not limit the descriptive power of the formalism. Hence let $\mathscr{G} = \mathcal{C}ospan(\mathcal{HG}raph)$ be a cospan category and let $\mathscr{G}_{\mathrm{inj}}$ be its subcategory that consist only of the cospans with injective interface morphisms.

Proposition 5.4. *Let a class $L_{J,K}$ of graphs with injective interfaces J, K be called* injectively recognizable *whenever $L_{J,K}$ is recognizable in $\mathscr{G}_{\mathrm{inj}}$. Then $L_{J,K}$ is recognizable in $\mathscr{G}$ if and only if it is injectively recognizable.*

We now restrict our attention to the subcategory $\mathscr{G}_{\mathrm{einj}}$ of cospans with interface morphism which are injective on edges. The result is not that interesting in its own right, but it is a necessary auxiliary step for Prop. 5.6.

Corollary 5.5. *Let a class $L_{J,K}$ of graphs with edge-injective interfaces J, K be called* edge-injectively recognizable *whenever $L_{J,K}$ is recognizable in $\mathscr{G}_{\mathrm{einj}}$. Then $L_{J,K}$ is recognizable in $\mathscr{G}$ if and only if it is edge-injectively recognizable.*

Similar to the restriction to injective interfaces we now show the fact that restricting our attention to cospans with discrete interfaces does not limit the descriptive power of the formalism. This allows us to restrict our attention to discrete interfaces in the following. Let $\mathscr{G} = \mathcal{C}ospan(\mathcal{HG}raph)$ be a cospan category and let $\mathscr{G}_{\mathrm{dis}}$ be its subcategory that consist only of the cospans with discrete interfaces, i.e., with interface graphs that do not contain edges.

Proposition 5.6. *Let a class $L_{J,K}$ of graphs with discrete interfaces J, K be called* discretely recognizable *whenever $L_{J,K}$ is recognizable in $\mathscr{G}_{\mathrm{dis}}$. Then $L_{J,K}$ is recognizable in $\mathscr{G}$ if and only if it is discretely recognizable.*

In a sense the result above is mirrored in the fact that graph rewriting does not lose any expressive power if we restrict to discrete interfaces.

5.3 Courcelle's Algebra of Graphs

We will now give a short introduction to Courcelle's algebraic notion of recognizable graph languages [5, 7]. Courcelle's notion of recognizable graph language is widely accepted as a notion of regularity for graphs. Also, Courcelle showed that if a language is definable in monadic second-order logic then it is recognizable. In [13] Courcelle's notion has been found to be nearly identical to the notions finite graph property and compatible graph property, which were developed in different contexts. For instance compatible properties arise in connection with hyperedge replacement grammars.

In [7] the relevant graph algebra is called HR-algebra (and in addition another algebra, called VR-algebra is investigated). However, here in this paper we use the algebra introduced in [5], since it is closer to our notion of cospan composition and hence yields simpler proofs.

Note that there are two major differences between cospan composition and the algebra of graphs introduced below: first the graph algebra considers only discrete interfaces, and we have already shown how to bridge this gap via Prop. 5.6. Second, cospans have two interfaces whereas graphs in the algebra have only one.

First, we give some preliminary definitions. We set $\mathbb{N}_k = \{0, \ldots, k-1\}$. Let P, Q be arbitrary sets. For functions $f\colon \mathbb{N}_n \to P$ and $g\colon \mathbb{N}_m \to Q$, we define the function $f \odot g\colon \mathbb{N}_{n+m} \to P \cup Q$ as follows:

$$(f \odot g)(i) = \begin{cases} f(i) & \text{if } i < n \\ g(i-n) & \text{otherwise.} \end{cases}$$

An n-ary hypergraph is a pair $\mathbb{G} = \langle \mathsf{base}_{\mathbb{G}}, \zeta_{\mathbb{G}} \rangle$ consisting of a hypergraph $\mathsf{base}_{\mathbb{G}}$ and a mapping $\zeta_{\mathbb{G}}\colon \mathbb{N}_n \to V$, where V is the node set of $\mathsf{base}_{\mathbb{G}}$. The function $\zeta_{\mathbb{G}}$ is called the *interface* of the graph, and its range the *external nodes*. Basically, an n-ary hypergraph corresponds to a graph with an empty internal and discrete external interface.

In [2], the following atomic operations on n-ary graphs are defined:

Redefinition of external nodes. Let an n-ary hypergraph $\mathbb{G}$ be given, and let $\sigma\colon \mathbb{N}_m \to \mathbb{N}_n$ be a function. The redefinition of $\mathbb{G}$ under σ is:

$$\mathsf{redef}_{\sigma}(\mathbb{G}) = \langle \mathsf{base}_{\mathbb{G}}, (\sigma\,;\,\zeta_{\mathbb{G}}) \rangle\ .$$

Note that this means that $\mathsf{redef}_{\sigma}(\mathbb{G})$ is an m-ary graph.

Fusion of external nodes. Let $\mathbb{G}$ be an n-ary graph, and θ an equivalence relation on $\mathbb{N}_n$. The fusion of $\mathbb{G}$ over θ, denoted $\mathsf{fuse}_{\theta}(\mathbb{G})$, is obtained by fusing the nodes of $\mathbb{G}$ according to θ. The result is again an n-ary graph.

Disjoint union. Let $\mathbb{G}$ be an n-ary graph and $\mathbb{H}$ an m-ary graph. The disjoint union of $\mathbb{G}$ and $\mathbb{H}$ is defined as:

$$\mathbb{G} \oplus \mathbb{H} = \langle \mathsf{base}_{\mathbb{G}} \oplus \mathsf{base}_{\mathbb{H}}, \zeta_{\mathbb{G}} \odot \zeta_{\mathbb{H}} \rangle$$

(we assume here that the node sets of $\mathsf{base}_{\mathbb{G}}$ and $\mathsf{base}_{\mathbb{H}}$ are disjoint and that $\oplus$ denotes the disjoint union on base graphs.)

We will now define recognizability in the sense of Courcelle via congruences. There is an alternative, but equivalent, definition of recognizable subsets as preimages of algebra homomorphisms.

Definition 5.7. *Let* $\equiv_{\mathrm{C}}$ *be an equivalence on* n*-ary hypergraphs that relates only hypergraphs with the same arity. It is called* locally finite *if for each* n *there are only finitely many equivalence classes. It is called a* congruence *if the following conditions hold:*

- *if* $\mathbb{G} \equiv_{\mathrm{C}} \mathbb{H}$*, then* $\mathsf{redef}_\sigma(\mathbb{G}) \equiv_{\mathrm{C}} \mathsf{redef}_\sigma(\mathbb{H})$*;*
- *if* $\mathbb{G} \equiv_{\mathrm{C}} \mathbb{H}$*, then* $\mathsf{fuse}_\theta(\mathbb{G}) \equiv_{\mathrm{C}} \mathsf{fuse}_\theta(\mathbb{H})$*;*
- *if* $\mathbb{G}_1 \equiv_{\mathrm{C}} \mathbb{H}_1$ *and* $\mathbb{G}_2 \equiv_{\mathrm{C}} \mathbb{H}_2$*, then* $\mathbb{G}_1 \oplus \mathbb{G}_2 \equiv_{\mathrm{C}} \mathbb{H}_1 \equiv_{\mathrm{C}} \mathbb{H}_2$.

A set L *of* n*-ary graphs is called* Courcelle-recognizable *if it is the union of finitely many equivalence classes of a locally finite congruence.*

We will in the following show that the notion of recognizability of Courcelle coincides with our notion, hence the notion of "Courcelle-recognizability" is redundant. However, we will keep it for the moment in order to properly distinguish the two notions of recognizability.

5.4 Equivalence of the two Notions of Recognizability

In this subsection we show that our notion of recognizable graph language is equivalent to Courcelle's. In Courcelle's notion there is only one (discrete) interface. The role of cospan composition is played by the operators defined above.

Encouraged by the result of Prop. 5.6 we restrict our attention to cospans with discrete interfaces. The discrete graph with node set $\mathbb{N}_n$ will be called canonical n-graph, and will be represented by Dis_n; we make use of the fact that each discrete graph with n nodes is isomorphic to the canonical n-graph. To formalize the equivalence between both notions of recognizability, we must first associate (sets of) cospans of graphs with n-ary graphs.

Definition 5.8. *For each discrete graph* D *with* n *nodes we fix in advance an isomorphism* $\mathsf{di}_D \colon \mathsf{Dis}_n \to D$ *such that* $\mathsf{di}_{D_1 \oplus D_2} = \mathsf{di}_{D_1} \odot \mathsf{di}_{D_2}$.

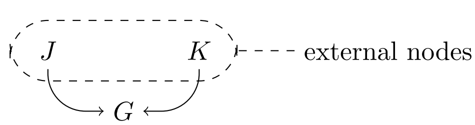
We define the function bend *which maps a cospan*

$$c \colon J \xrightarrow{c_{\mathrm{L}}} G \xleftarrow{c_{\mathrm{R}}} K$$

where J, K *are discrete interfaces with* n, m *nodes, resp., to an* $(n+m)$*-ary graph as follows:*

$$\mathsf{bend}(c) = \langle G, (\mathsf{di}_J \,;\, c_{\mathrm{L}}) \odot (\mathsf{di}_K \,;\, c_{\mathrm{R}})\rangle \ .$$

The name bend is inspired by the fact that the function basically 'bends' a cospan so that its inner and outer interface are together, and then interprets the resulting figure as a $(m+n)$-ary hypergraph, as illustrated below:



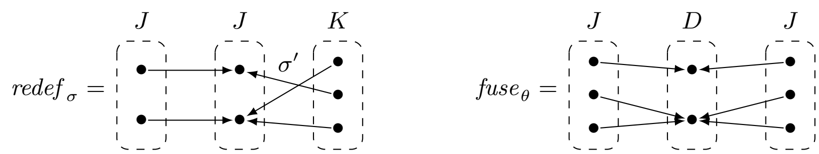


Fig. 1. *Simulating Courcelle's graph operation by cospans. On the left: example of cospan simulating redefinition of external nodes; on the right: example of cospan simulating fusion of external nodes.*

Theorem 5.9. *Let J be a discrete graph. A set of graphs L is the $(\emptyset, J)$-language of some automaton functor $\mathcal{A}$ if and only if $\mathsf{bend}(L)$ is Courcelle-recognizable.*

Proof. (Sketch.) We prove the theorem by simulating Courcelle's operations by cospan composition, and cospan composition by Courcelle's operations, so that the congruences can be transferred. Suppose J has n nodes. The simulations of Courcelle's operations work as follows (see Fig. 1):

– Let $\sigma \colon \mathbb{N}_m \to \mathbb{N}_n$ be a function, and K a discrete graph with m nodes. It can be considered as a graph morphism from Dis_m to Dis_n. Then $\sigma' = \mathsf{di}_K^{-1}$; σ ; di_J is the corresponding graph morphism from K to J. Postcomposing a cospan c with the cospan

$$redef_\sigma \colon J \xrightarrow{\mathsf{id}_J} J \xleftarrow{\sigma'} K$$

simulates performing the redef_σ operation on c.

– Let θ be an equivalence relation on the elements of $\mathbb{N}_n$ (i.e. an equivalence relation on the nodes of Dis_n). Suppose θ_{map} is the morphism which maps each element of $\mathbb{N}_n$ to its θ-equivalence class and let $\theta' = \mathsf{di}_J^{-1}$; θ_{map}. Then

$$fuse_\theta \colon J \xrightarrow{\theta'} D \xleftarrow{\theta'} J \ ,$$

where D is the discrete graph with node set $\{[\![v]\!]_\theta \mid v \in \mathbb{N}_n\}$, simulates the fuse_θ operation.

– Disjoint sum is simulated by disjoint sum in the category of graphs and we obtain the congruence property via the congruence property for cospan composition (see the extended proof in the appendix).

The simulation of cospan composition in Courcelle's algebra depends on the fact that $\mathsf{bend}(c\ ;\ d) = \mathsf{redef}_\sigma(\mathsf{fuse}_\theta(\mathsf{bend}(c) \oplus \mathsf{bend}(d)))$ for appropriate σ and θ. In Fig. 2 this is depicted for cospans c, d where c has inner interface $\emptyset$. $\square$

Note that one of the reasons why the proof works is the fact that the category of cospans is compact-closed, which means that certain "bending" laws, similar to the one above, hold.

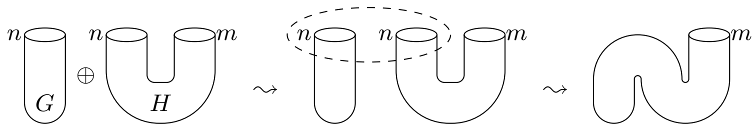


Fig. 2. *Simulating cospan composition with Courcelle's operations.* First, we construct the disjoint union of the graphs In the second step the indicated external nodes are fused and then removed from the external nodes by a redefinition.

6 Conclusion

We have shown that a very general categorical notion of recognizability via automaton functors (which is equivalent to a notion suggested by Griffing [12]) is equivalent to a notion of recognizability for graph languages by Courcelle whenever we consider the category of cospans of graphs. The proof of this equivalence is non-trivial.

Furthermore we investigated our notion of automaton functor and showed that it preserves several nice properties which are well-known for finite-state automata. Our main motivation behind this work is to provide automata-based techniques for verification and termination analysis of graph transformation systems. Some preliminary results on termination analysis are reported in [3].

Cospans of graphs were also investigated, in the context of gluing graph structures, by Rosebrugh, Sabadini and Walters [19, 20], but in [20] their graph structures represent automata which recognize word languages rather than graph languages. In the future we plan to explore the relations between their and our work in some more detail.

Naturally, efficiency questions arise. The automaton functor we are working with is only locally finite, i.e., the sets of states are finite for every interface, but interfaces might be arbitrarily large. This question has already been addressed by Courcelle, who characterized classes of graphs which can be recognized efficiently. He showed that for the HR-algebra of graphs a graph can be decomposed via interfaces whose size is bounded by $k + 1$ if and only if its treewidth is bounded by k [6, 7]. Hence a language L can be recognized efficiently (even in linear time!) if there is a bound on the treewidth of the graphs contained in L (see also [8]). This is also known as Courcelle's theorem and applies to properties such as k-colorability that would be NP-complete on graphs of unbounded treewidth.

Since with cospans we have a different notion of interface and different operations, this result by Courcelle does not carry over directly, although we have the same notion of recognizability. This is a point which has to be investigated further, but in order to arrive at a similar result we believe that it is necessary to equip our category with a monoidal operation $\oplus$ (which is the disjoint sum on cospans) and to require that an automaton functor preserves this monoidal operation. We conjecture that, at least in the case of graphs with empty inner interface, adding such a monoidal operation will not affect which sets of (cospans of) graphs are recognizable. Currently it seems that we can only guarantee linear-time algorithms for graphs of bounded pathwidth, since cospans allow only to construct path decompositions of graphs.

Of course, in order to obtain practical algorithms for recognizability, we have to find reasonable ways to represent and handle automaton functors, at least in the case of graphs of bounded treewidth. We have some preliminary ideas how this can be achieved, but it is an interesting problem that has to be studied further.

In this paper we mainly considered cospans of graphs, but there is a more general notion of (DPO) rewriting based on adhesive categories [15, 10]. Our notion of recognizability can be easily generalized to this setting, whereas it is not entirely clear how to extend Courcelle's algebra of graphs. One possible application of such a generalization provides us with a method to show that (recognizable) sets of objects in an adhesive category are invariant under DPO rewriting rules. Let $p\colon L \leftarrow I \rightarrow R$ be a DPO rule and let $\equiv_{\mathrm{R}}$ be a congruence[1] characterizing a language L of objects. We observe that whenever an object A is rewritten to B via p, $A \in L$ and $(0 \rightarrow L \leftarrow I) \equiv_{\mathrm{R}} (0 \rightarrow R \leftarrow I)$ (for an initial object 0), then we can conclude that $B \in L$.

Finally, an important result in Courcelle's work is that a language is recognizable whenever it is definable in monadic second-order logic [5, 7]. Currently we have no counterpart to this result, but it might be worthwhile to study it in a more categorical setting.

Acknowledgement

The authors would like to thank Tobias Heindel for his valuable comments on the contents of this paper.

References

1. A. Bouajjani, B. Jonsson, M. Nilsson, and T. Touili. Regular model checking. In *Proc. of CAV '00*, pages 403–418. Springer, 2000. LNCS 1855.
2. M. Bouderon and B. Courcelle. Graph expressions and graph rewritings. *Mathematical Systems Theory*, 20:81–127, 1987.
3. H.J.S. Bruggink. Towards a systematic method for proving termination of graph transformation systems (work-in-progress paper). In *Proc. of GT-VC '07 (Graph Transformation for Verification and Concurrency)*, 2007.
4. A. Corradini, U. Montanari, F. Rossi, H. Ehrig, R. Heckel, and M. Löwe. Algebraic approaches to graph transformation—part I: Basic concepts and double pushout approach. In G. Rozenberg, editor, *Handbook of Graph Grammars and Computing by Graph Transformation, Vol. 1: Foundations*, chapter 3. World Scientific, 1997.
5. B. Courcelle. The monadic second-order logic of graphs I. recognizable sets of finite graphs. *Information and Computation*, 85:12–75, 1990.

[1] One is here actually interested in the weaker notion of a Myhill well-quasi order instead of a congruence, but we omit the details.

6. B. Courcelle. Graph grammars, monadic second-order logic and the theory of graph minors. In *Graph Structure Theory*, pages 565–590. American Mathematical Society, 1991.
7. B. Courcelle. The expression of graph properties and graph transformations in monadic second-order logic. In G. Rozenberg, editor, *Handbook of Graph Grammars and Computing by Graph Transformation, Vol.1: Foundations*, chapter 5. World Scientific, 1997.
8. B. Courcelle and J. Lagergren. Equivalent definitions of recognizability for sets of graphs of bounded tree-width. *Mathematical Structures in Computer Science*, 6(2):141–165, 1996.
9. B. Courcelle and P. Weil. The recognizability of sets of graphs is a robust property. *Theoretical Computer Science*, 342(2–3):173–228, 2005.
10. H. Ehrig, K. Ehrig, U. Prange, and G. Taentzer. *Fundamentals of Algebraic Graph Transformation*. Springer, 2006.
11. A. Geser, D. Hofbauer, and J. Waldmann. Match-bounded string rewriting systems. *Applicable Algebra in Engineering, Communication and Computing*, 15(3–4):149–171, 2004.
12. G. Griffing. Composition-representative subsets. *Theory and Applications of Categories*, 11(19):420–437, 2003.
13. A. Habel, H.-J. Kreowski, and C. Lautemann. A comparison of compatible, finite and inductive graph properties. *Theoretical Computer Science*, 110(1):145–168, 1993.
14. V. Kuncak and M.C. Rinard. Existential heap abstraction entailment is undecidable. In *Proc. of SAS '03*, pages 418–438. Springer, 2003. LNCS 2694.
15. S. Lack and P. Sobociński. Adhesive and quasiadhesive categories. *RAIRO – Theoretical Informatics and Applications*, 39(3), 2005.
16. S. Mac Lane. *Categories for the Working Mathematician*. Springer-Verlag, 1971.
17. J. Mezei and J.B. Wright. Algebraic automata and context-free sets. *Information and Control*, 11:3–29, 1967.
18. A. Potthoff, S. Seibert, and W. Thomas. Nondeterminism versus determinism of finite automata over directed acyclic graphs. *Bulletin of the Belgian Mathematical Society*, 1:285–298, 1994.
19. R. Rosebrugh, N. Sabadini, and R.F.C. Walters. Generic commutative separable algebras and cospans of graphs. *Theory and applications of categories*, 15(6):164–177, 2005.
20. R. Rosebrugh, N. Sabadini, and R.F.C. Walters. Calculating colimits compositionally. 2007.
21. V. Sassone and P. Sobociński. Reactive systems over cospans. In *Proc. of LICS '05*, pages 311–320. IEEE, 2005.
22. T. Urvoy. Abstract families of graphs. In *Proc. of DLT '02*, pages 381–392. Springer, 2003. LNCS 2450.
23. K.-U. Witt. Finite graph-acceptors and regular graph-languages. *Information and Control*, 50(3):242–258, 1981.

A Proofs

3. Recognizable Languages of Arrows

Proposition 3.6. *Let $\mathscr{C}$ be a category, J, K $\mathscr{C}$-objects and $L_{J,K}$ a set of $\mathscr{C}$-arrows from J to K. The language $L_{J,K}$ is recognizable in $\mathscr{C}$, if and only if there exists a locally finite congruence $\equiv_{\mathrm{R}}$ such that $L_{J,K}$ is the union of some equivalence classes of $R_{J,K}$.*

Proof. ($\Rightarrow$) Let $\mathcal{A}$ be the automaton functor which recognizes the language $L_{J,K}$. We construct $\equiv_{\mathrm{R}} = \{R_{M,N} \mid M, N \in \mathscr{C}\}$ as follows. Let $a, a' \colon M \to N$ be $\mathscr{C}$-arrows. We define: $a \; R_{M,N} \; a'$ if and only if $\mathcal{A}(a) = \mathcal{A}(a')$. It is clear that each $R_{M,N}$ is an equivalence relation of finite index. The fact that $\equiv_{\mathrm{R}}$ is a congruence follows from the fact that $\mathcal{A}$ is a functor, and thus respects composition. Now we have that

$$L_{J,K} = \{[\![a]\!]_{\mathrm{R}} \mid a \colon J \to K \text{ and } \mathcal{A}(a)(\mathrm{I}_J^{\mathcal{A}}) \cap \mathrm{F}_K^{\mathcal{A}} \neq \emptyset\} \enspace .$$

($\Leftarrow$) This direction of the proof only works for sets of arrows starting from the fixed object J. Let a congruence $\equiv_{\mathrm{R}} = \{R_{M,N} \mid M, N \in \mathcal{O}bj(\mathscr{C})\}$ and the language $L_{J,K}$ be given. We define the deterministic automaton functor $\mathcal{A} \colon \mathscr{C} \to \mathcal{S}et$ as follows:

- for an object H, we define $\mathcal{A}(H) = \{[\![a]\!]_{\mathrm{R}} \mid a : J \to H\}$.
- for an arrow $b \colon H \to M$ and $[\![a]\!]_{\mathrm{R}} \in \mathcal{A}(H)$ we define $\mathcal{A}(b)([\![a]\!]_{\mathrm{R}}) = [\![a \,;\, b]\!]_{\mathrm{R}}$.

We show that $\mathcal{A}$ is a functor, i.e. that it respects composition and identities:

- *Composition.* Let $b \colon H \to M$ and $c \colon M \to N$ be $\mathscr{C}$-arrows.

$$\begin{aligned}
(\mathcal{A}(b) \,;\, \mathcal{A}(c)) \left([\![a]\!]_{R_{J,H}}\right) &= \mathcal{A}(c) \left(\mathcal{A}(b) \left([\![a]\!]_{R_{J,H}}\right)\right) \\
&= \mathcal{A}(c) \left([\![a \,;\, b]\!]_{R_{J,M}}\right) \\
&= [\![a \,;\, b \,;\, c]\!]_{R_{J,N}} \\
&= \mathcal{A}(b \,;\, c) \left([\![a]\!]_{R_{J,H}}\right)
\end{aligned}$$

- *Identities.* Let $a \colon J \to H$ be a $\mathscr{C}$-arrow. Then

$$\mathcal{A}(\mathsf{id}_H) \left([\![a]\!]_{\mathrm{R}}\right) = [\![a \,;\, \mathsf{id}_H]\!]_{\mathrm{R}} = [\![a]\!]_{\mathrm{R}} = \mathsf{id}_{\mathcal{A}(H)} \left([\![a]\!]_{\mathrm{R}}\right).$$

Finally, we de define $\mathrm{I}_J^{\mathcal{A}}$ and $\mathrm{F}_K^{\mathcal{A}}$ as follows:

$$\begin{aligned}
\mathrm{I}_J^{\mathcal{A}} &= \{\mathsf{id}_J\} \\
\mathrm{F}_K^{\mathcal{A}} &= \{[\![a]\!]_{\mathrm{R}} \mid a \in L_{J,K}\} \enspace .
\end{aligned}$$

□

Note that in our construction of an automaton functor from a given congruence, we construct an automaton functor which only recognizes the language $L_{J,K}$ correctly. For other choices of J and K, the recognized languages may not correspond. We leave it to further research to find a better construction.

4. Determinism, Closure Properties and Minimization

Proposition 4.1. *For every automaton functor, there exists an equivalent deterministic automaton functor.*

Proof. Let $\mathcal{A}\colon \mathscr{C} \to \mathscr{R}el$ be a non-deterministic automaton functor. We construct a deterministic automaton functor $\mathcal{A}_{\mathrm{D}}\colon \mathscr{C} \to \mathscr{S}et$ as follows:

– For each object $H \in \mathcal{O}bj(\mathscr{C})$ we take $\mathcal{A}_{\mathrm{D}}(H) = \wp(\mathcal{A}(H))$, with $\mathrm{I}_H^{\mathcal{A}_{\mathrm{D}}} = \{\mathrm{I}_H^{\mathcal{A}}\}$ and $\mathrm{F}_H^{\mathcal{A}_{\mathrm{D}}} = \{S \subseteq \mathcal{A}(H) \mid S \cap \mathrm{F}_H^{\mathcal{A}} \neq \emptyset\}$.
– For each arrow $c \in \mathcal{A}rr(\mathscr{C})$ we take

$$\mathcal{A}_{\mathrm{D}}(c)\,(S) = \bigcup_{s \in S} \mathcal{A}(c)\,(s) \quad .$$

It is clear that $\mathcal{A}$ and $\mathcal{A}_{\mathrm{D}}$ recognize the same language of arrows, and that $\mathcal{A}_{\mathrm{D}}$ is deterministic. □

Proposition 4.2 (Closure under boolean operators). *Suppose we have two recognizable languages of arrows, $L^1_{J,K}$ and $L^2_{J,K}$. Then also $L^1_{J,K} \cap L^2_{J,K}$, $L^1_{J,K} \cup L^2_{J,K}$ and $(L^1_{J,K})^{\mathrm{C}}$ (the complement of $L^1_{J,K}$) are recognizable.*

Proof. Let $\mathcal{A}_1$ be the automaton functor which recognizes $L^1_{J,K}$ and $\mathcal{A}_2$ the automaton functor which recognizes $L^2_{J,K}$. First we obtain equivalent deterministic automaton functors $\mathcal{A}'_1, \mathcal{A}'_2$, in the way of Prop. 4.1. Then we construct the required automaton functors as follows:

– For union and intersection, let $\mathcal{A}$ be defined as $\mathcal{A}(J) = \mathcal{A}'_1(J) \times \mathcal{A}'_2(J)$, and the cross-product of relations defined pointwise in the obvious ways. Furthermore, we take $\mathrm{I}_J^{\mathcal{A}} = \mathrm{I}_J^{\mathcal{A}'_1} \times \mathrm{I}_J^{\mathcal{A}'_2}$. Finally, we define $\mathrm{F}_J^{\mathcal{A}}$ as follows:
  • For union: $\mathrm{F}_J^{\mathcal{A}} = (\mathrm{F}_J^{\mathcal{A}_1} \times \mathcal{A}_2(J)) \cup (\mathcal{A}_1(J) \times \mathrm{F}_J^{\mathcal{A}_2})$
  • For intersection: $\mathrm{F}_J^{\mathcal{A}} = \mathrm{F}_J^{\mathcal{A}_1} \times \mathrm{F}_J^{\mathcal{A}_2}$
– For complement, we define for objects J and arrows c:

$$\mathcal{A}(J) = \mathcal{A}'_1(J) \qquad \mathrm{I}_J^{\mathcal{A}} = \mathrm{I}_J^{\mathcal{A}'_1}$$
$$\mathcal{A}(c) = \mathcal{A}'_1(c) \qquad \mathrm{F}_J^{\mathcal{A}} = (\mathrm{F}_J^{\mathcal{A}'_1})^{\mathrm{C}} \quad .$$

□

Definition A.1. *Let a deterministic automaton functor $\mathcal{A}\colon \mathscr{C} \to \mathscr{S}et$, a $\mathscr{C}$-object J and $s, t \in \mathcal{A}(J)$ be given. We define: $s \approx t$ if $\mathcal{A}(c)(s) \in \mathrm{F}_K^{\mathcal{A}} \Leftrightarrow \mathcal{A}(c)(t) \in \mathrm{F}_K^{\mathcal{A}}$ for all $\mathscr{C}$-arrows c from J to K.*

Lemma A.2. *A fully (J-)reachable, deterministic automaton functor $\mathcal{A}\colon \mathscr{C} \to \mathscr{S}et$ is (J-)minimal if and only if for all $K \in \mathcal{O}bj(\mathscr{C})$ and $s, t \in \mathcal{A}(K)$, where $s \neq t$, it holds that $s \not\approx t$.*

Proof. We only show the proof of the claim which is not restricted to any J. The proof in the restricted case is analogous.

($\Rightarrow$) If for some $K \in \mathcal{O}bj(\mathscr{C})$, $s, t \in \mathcal{A}(K)$, $s \neq t$, we have $s \approx t$, we may merge s and t (i.e. factor $\mathcal{A}(K)$ through the smallest equivalence relation such that $s \equiv t$), obtaining a smaller equivalent automaton functor.

($\Leftarrow$) Let $\mathcal{B}\colon \mathscr{C} \to \mathcal{S}et$ be a deterministic automaton functor such that for some $\mathscr{C}$-object K it holds that $|\mathcal{B}(K)| < |\mathcal{A}(K)|$. We need to prove that $\mathcal{A}$ and $\mathcal{B}$ are not equivalent.

For all $u \in \mathcal{A}(K)$, arbitrarily select a $\mathscr{C}$-arrow $f_u\colon G_u \to K$ such that $\mathcal{A}(f_u)(\mathrm{i}^{\mathcal{A}}_{G_u}) = u$. (Such an arrow exists because $\mathcal{A}$ is fully reachable by assumption.) Since $|\mathcal{B}(K)| < |\mathcal{A}(K)|$ there exist states $s, t \in \mathcal{A}(K)$, with $s \neq t$, such that

$$\mathcal{B}(f_s)(\mathrm{i}^{\mathcal{B}}_{G_s}) = \mathcal{B}(f_t)(\mathrm{i}^{\mathcal{B}}_{G_t}). \qquad (\star)$$

Since, by assumption, $s \not\approx t$, there is, for some H, an arrow $g\colon K \to H$ such that $\mathcal{A}(g)(s) \in \mathrm{F}^{\mathcal{A}}_H$ but $\mathcal{A}(g)(t) \notin \mathrm{F}^{\mathcal{A}}_H$ (or the other way around, a case which we will, without loss of generality, ignore). Therefore $(f_s \mathbin{;} g) \in L_{G_s,H}(\mathcal{A})$, but $(f_t \mathbin{;} g) \notin L_{G_t,H}(\mathcal{A})$.

However, by ($\star$), either both $f_s \mathbin{;} g$ and $f_t \mathbin{;} g$ are element of $L_{G_s,H}(\mathcal{B})$, or both are not. So we conclude that $\mathcal{A}$ and $\mathcal{B}$ are not equivalent. ◻

Proposition 4.5. *For each deterministic automaton functor $\mathcal{A}$, there exists a minimal, (J-)equivalent, deterministic automaton functor $\mathcal{A}^{\min}$ which recognizes the same language and which is unique up to isomorphism.*

Proof. Let $\mathcal{A}\colon \mathscr{C} \to \mathcal{S}et$ be a deterministic automaton functor. We define the minimal automaton functor $\mathcal{A}^{\min} : \mathscr{C} \to \mathcal{S}et$, using the equivalence relation $\approx$ defined above:

– $\mathcal{A}^{\min}(K) = \{[\![s]\!]_{\approx} \mid s \text{ is a } (J\text{-})\text{reachable state of } \mathcal{A}(K)\}$, and similarly for $\mathrm{I}^{\mathcal{A}^{\min}}_K$ and $\mathrm{F}^{\mathcal{A}^{\min}}_K$.
– $\mathcal{A}^{\min}(d)([\![s]\!]_{\approx}) = \{[\![s']\!]_{\approx} \mid s' \text{ is a } (J\text{-})\text{reachable state of } \mathcal{A}(M), s' \in \mathcal{A}(d)(s)\}$ for $d\colon K \to M$.

The facts that $\mathcal{A}^{\min}$ is minimal and unique up to isomorphism follow easily from Lemma A.2. ◻

5. Recognizable Graph Languages

5.2. Robustness

Proposition 5.4. *Let a class $L_{J,K}$ of graphs with injective interfaces J, K be called* injectively recognizable *whenever $L_{J,K}$ is recognizable in $\mathscr{G}_{\mathrm{inj}}$. Then $L_{J,K}$ is recognizable in $\mathscr{G}$ if and only if it is injectively recognizable.*

Proof. For both directions of the proof, we use the characterization of recognizability via congruences (Prop. 3.6).

($\Rightarrow$): Trivial, because $\mathscr{G}_{\mathrm{inj}}$ is a subcategory of $\mathscr{G}$, and thus we can use the congruence for $\mathscr{G}$ also in the case of $\mathscr{G}_{\mathrm{inj}}$.

($\Leftarrow$): Let $\equiv_{\mathrm{R}}$ be a congruence for $L_{J,K}$ in the sense of Def. 3.5 for the category $\mathscr{G}_{\mathrm{inj}}$. We now define an equivalence $\equiv'_{\mathrm{R}}$ on $\mathscr{G}$ and show that it is a congruence of finite index.

For this we first need the notion of *merger pairs*. For a cospan $c\colon J \xrightarrow{c_{\mathrm{L}}} G \xleftarrow{c_{\mathrm{R}}} K$ the set $M(c)$ of merger pairs consists of all pairs of cospans $\langle m_J, m_K \rangle$ of the form

$$m_J\colon J' \xrightarrow{\mathsf{id}} J' \xleftarrow{m_J^{\mathrm{R}}} J \quad , \quad m_K\colon K \xrightarrow{m_K^{\mathrm{L}}} K' \xleftarrow{\mathsf{id}} K'$$

such that m_J^{R} and m_K^{L} are surjective and $m_J \,;\, c \,;\, m_K$ is an injective cospan. In a sense the merger pairs induce an equivalence on the interfaces which relates at least the interface items which have the same image under c_{L} or c_{R}. Furthermore, since we only deal with finite graphs, there are only finitely many "merger cospans" up to isomorphism.

We now define $\equiv'_{\mathrm{R}}$ as follows: for two cospans $c_1\colon J \to G_1 \leftarrow K$ and $c_2\colon J \to G_2 \leftarrow K$ it holds that $c_1 \equiv'_{\mathrm{R}} c_2$ whenever $M(c_1) = M(c_2)$ and for all $\langle m_J, m_K \rangle \in M(c_1)$ we have $m_J \,;\, c_1 \,;\, m_K \equiv_{\mathrm{R}} m_J \,;\, c_2 \,;\, m_k$. Note that for an injective cospan c the pairs consisting of the identity cospans on J and K are among the merger pairs and hence $\equiv'_{\mathrm{R}}$ is a refinement of $\equiv_{\mathrm{R}}$. Furthermore it can be shown that $\equiv'_{\mathrm{R}}$ is locally finite whenever $\equiv_{\mathrm{R}}$ is.

We now have to show that $\equiv'_{\mathrm{R}}$ is a congruence. For this take two cospans $c_1 \equiv'_{\mathrm{R}} c_2$ with $c_i\colon J \to G_i \leftarrow K$, where $i \in \{1,2\}$, and another cospan $d\colon K \to H \leftarrow M$. We have to prove that $c_1 \,;\, d \equiv'_{\mathrm{R}} c_2 \,;\, d$.

Now take a merger pair of $c_1 \,;\, d$, i.e., choose $\langle m_J, m_M \rangle \in M(c_1 \,;\, d)$. We will show that there are cospans m_K, m'_K such that $\langle m_J, m_K \rangle \in M(c_1)$, $\langle m'_K, m_M \rangle \in M(d)$ and $m_J \,;\, c_1 \,;\, d \,;\, m_M = m_J \,;\, c_1 \,;\, m_K \,;\, m'_K \,;\, d \,;\, m_M$. For this consider the diagram below which represents the composition $(m_J \,;\, c_1) \,;\, (d \,;\, m_M)$ via pushouts where the order of composition is indicated by the brackets.

$$\begin{tikzcd}
J' \arrow[r] \arrow[ddrrrr, tail, bend right] & J' \arrow[dr] & J \arrow[l] \arrow[r] & G_1 \arrow[dl] & K \arrow[l] \arrow[r] & H \arrow[dr] & M \arrow[l] \arrow[r] & M' \arrow[dl] & M' \arrow[l] \arrow[ddllll, tail, bend left] \\
 & & G'_1 \arrow[drr] & & & & H' \arrow[dll] & & \\
 & & & & H_1 & & & &
\end{tikzcd}$$

Note that the two "triangles" on the left and right are pushouts and that the middle "pentagon" is also a pushout. In the following we will denote injective morphisms by $\rightarrowtail$ and surjective morphisms by $\twoheadrightarrow$.

Now consider the morphism $K \to H_1$ and take its factorization into a surjective morphism $K \twoheadrightarrow K'$, followed by an injective morphism $K' \rightarrowtail H_1$. In the category of graphs this factorization always exists and is unique. In addition take the pushouts of $K \to G'_1$, $K \twoheadrightarrow K'$ and of $K \to H'$, $K \twoheadrightarrow K'$, obtaining graphs G_1'' and H'' and corresponding mediating morphisms. Now the situation is as follows:

$$\begin{tikzcd}
J' \arrow[r] \arrow[ddrrrr, tail, bend right] & J' \arrow[dr] & J \arrow[l] \arrow[r] & G_1 \arrow[dl] & K \arrow[l] \arrow[r] \arrow[d, two heads] & H \arrow[dr] & M \arrow[l] \arrow[r] & M' \arrow[dl] & M' \arrow[l] \arrow[ddllll, tail, bend left] \\
& & G_1' \arrow[r, two heads] \arrow[drr] & G_1'' \arrow[dr, dotted, tail] & K' \arrow[l, tail] \arrow[r, tail] \arrow[d] & H'' \arrow[dl, dotted, tail] & H' \arrow[l, two heads] \arrow[dll, tail] & & \\
& & & & H_1 & & & &
\end{tikzcd}$$

Note also that the morphisms $J' \to J' \to G_1' \to G_1''$ and $M' \to M' \to H' \to H''$ must be injective, since they split the injective morphisms $J' \to H_1$, $M' \to H_1$.

We will now show that the square K', G_1'', H'', H_1 commutes and that it is a pushout. We are in the situation depicted on the left below where the outer square is a pushout. We have $K \twoheadrightarrow K' \rightarrowtail G_1'' \rightarrowtail H_1 = K \to G_1' \twoheadrightarrow G_1'' \rightarrowtail H_1 = K \to H' \twoheadrightarrow H'' \rightarrowtail H_1 = K \twoheadrightarrow K' \rightarrowtail H'' \rightarrowtail H_1$. And since $K \twoheadrightarrow K'$ is surjective we can conclude that $G_1' \twoheadrightarrow G_1'' \rightarrowtail H_1 = G_1' \rightarrowtail H'' \rightarrowtail H_1$. Now assume a commuting square K', G_1'', H'', X as shown below on the right. Since the outer square is a pushout we obtain a mediating morphism $H_1 \to X$. The triangles G_1'', H_1, X and H'', H_1, X commute since they commute if we precompose them with the morphisms $G_1' \twoheadrightarrow G_1''$ and $H' \twoheadrightarrow H''$ respectively. Furthermore the morphism $H_1 \to X$ must be unique since any other mediating morphism would also be a mediating morphism for the outer pushout.

$$\begin{tikzcd}
K \arrow[rr] \arrow[dd] \arrow[dr, two heads] & & H' \arrow[d, two heads] \\
& K' \arrow[r, tail] \arrow[d, tail] & H'' \arrow[d, tail] \\
G_1' \arrow[r, two heads] & G_1'' \arrow[r, tail] & H_1
\end{tikzcd}
\qquad
\begin{tikzcd}
K \arrow[rr] \arrow[dd] \arrow[dr, two heads] & & H' \arrow[d, two heads] & \\
& K' \arrow[r, tail] \arrow[d, tail] & H'' \arrow[d, tail] \arrow[dr, bend left] & \\
G_1' \arrow[r, two heads] & G_1'' \arrow[r, tail] \arrow[drr, bend right] & H_1 \arrow[dr, dotted] & \\
& & & X
\end{tikzcd}$$

Hence we get $m_K \colon K \twoheadrightarrow K' \xleftarrow{\mathsf{id}} K'$ and $m_K' \colon K' \xrightarrow{\mathsf{id}} K' \twoheadrightarrow K$ and from the diagram above it follows that $m_J \,;\, c_1 \,;\, d \,;\, m_M = m_J \,;\, c_1 \,;\, m_K \,;\, m_K' \,;\, d \,;\, m_M$. Since both $m_J \,;\, c_1 \,;\, m_K$ and $m_K' \,;\, d \,;\, m_M$ are injective cospans and $c_1 \equiv_{\mathrm{R}}' c_2$, we get $m_J \,;\, c_1 \,;\, m_K \,;\, m_K' \,;\, d \,;\, m_M \equiv_{\mathrm{R}} m_J \,;\, c_2 \,;\, m_K \,;\, m_K' \,;\, d \,;\, m_M$.

It is now left to show that $m_J \,;\, c_2 \,;\, m_K \,;\, m_K' \,;\, d \,;\, m_M = m_J \,;\, c_2 \,;\, d \,;\, m_M$ and that this is an injective cospan. We first consider the diagram for $(m_J \,;\, c_2) \,;$

$(d\,;m_M)$.

$$\begin{tikzcd}
J' \arrow[r] \arrow[ddrrrr, bend right, tail] & J' \arrow[dr] & J \arrow[l] \arrow[r] & G_2 \arrow[dl] & K \arrow[l] \arrow[r] & H \arrow[dr] & M \arrow[l] \arrow[r] & M' \arrow[dl] & M' \arrow[l] \arrow[ddllll, bend left, tail] \\
 & & G_2' \arrow[drr] & & & & H' \arrow[dll] & & \\
 & & & & H_2 & & & &
\end{tikzcd}$$

Next, we consider the factorization of $K \to G_2'$ into a surjective morphism followed by an injective morphism, i.e.,

$$K \to G_2' = K \overset{s_K}{\twoheadrightarrow} K_G \rightarrowtail G_2'$$

We construct the pushout of $s_K \colon K \twoheadrightarrow K_G$ and $K \to G_2'$ as shown below in two steps. First we take the pushout of s_K with itself, which gives us K_G and the identity morphisms since s_K is surjective. Then we take the pushout of $K_G \rightarrowtail G_2'$ and the identity on K_G, resulting in G_2'.

$$\begin{tikzcd}
K \arrow[r, two heads, "s_K"] \arrow[d, two heads, "s_K"] & K_G \arrow[r, tail] \arrow[d, "\mathsf{id}"] & G_2' \arrow[d, "\mathsf{id}"] \\
K_G \arrow[r, "\mathsf{id}"] & K_G \arrow[r, tail] & G_2'
\end{tikzcd}$$

Now consider the cospan $\hat{m} \colon K \overset{s_K}{\twoheadrightarrow} K_G \overset{\mathsf{id}}{\leftarrow} K_G$. Because of the considerations above we can conclude that $\langle m_J, \hat{m} \rangle \in M(c_2) = M(c_1)$. Furthermore we can show that $\hat{m}$ is uniquely characterized to be the most "general" partner for m_J. Consider for instance another cospan $m_K \colon K \to K' \leftarrow K'$ with $\langle m_J, m_K \rangle \in M(c_2)$. Then we obtain the diagram below on the left.

$$\begin{tikzcd}
K \arrow[r, two heads, "s_K"] \arrow[d, two heads] & K_G \arrow[r, tail] & G_2' \arrow[d, two heads] \\
K' \arrow[rr, tail] & & G_2''
\end{tikzcd}
\qquad
\begin{tikzcd}
K \arrow[r, two heads, "s_K"] \arrow[d, two heads] & K_G \arrow[r, tail] & G_2' \arrow[d, two heads] \\
K' \arrow[rr, tail] \arrow[ur, dotted, leftarrow] & & G_2''
\end{tikzcd}$$

Because of the diagonalization property of factorizations we obtain a unique morphism $K_G \to K'$ which makes the diagram commute (see diagram above on the right). This property uniquely characterizes s_K (up to isomorphism). The same characterization holds for the factorization of $K \to G_1'$ (since the merger pairs are identical) and we can conclude that $K \to G_1'$ factorizes into $K \overset{s_K}{\twoheadrightarrow} K_G \rightarrowtail G_1'$. Now consider also the factorization of $K \to H$ into $K \twoheadrightarrow K_H \rightarrowtail H$

and split the pushout of $K \to G_2'$ and $K \to H$ as shown below:

$$\begin{tikzcd}
K \arrow[r, two heads] \arrow[d, two heads, "s_K"] & K_H \arrow[r, tail] \arrow[d, two heads] & H \arrow[d, two heads] \\
K_G \arrow[r, two heads] \arrow[d, tail] & K'' \arrow[r, tail] \arrow[d] & \bullet \arrow[d, tail] \\
G_2' \arrow[r, two heads] & \bullet \arrow[r, tail] & H_2
\end{tikzcd}$$

Hence taking the pushout of $K \twoheadrightarrow K_G$ and $K \twoheadrightarrow K_H$ gives us the unique factorization of $K \to H_2$. The same is true in the case of H_1 where the factorization is $K \to H_2 = K \twoheadrightarrow K' \rightarrowtail H_1$. Hence K' and K'' must be isomorphic.

Furthermore in the diagram for $(m_J \mathbin{;} c_2) \mathbin{;} (d \mathbin{;} m_M)$ above we can split the middle pushout in a way identical to the splitting of $(m_J \mathbin{;} c_1) \mathbin{;} (d \mathbin{;} m_M)$ and obtain a pushout consisting of K', G_2'', H'', H_2 (see below).

$$\begin{tikzcd}
J' \arrow[r] \arrow[ddrrrr, tail, bend right] & J' \arrow[dr] & J \arrow[l] \arrow[r] & G_2 \arrow[dl] & K \arrow[l] \arrow[r] \arrow[d, two heads] & H \arrow[dr] & M \arrow[l] \arrow[r] & M' \arrow[dl] & M' \arrow[l] \arrow[ddllll, tail, bend left] \\
 & & G_2' \arrow[r, two heads] \arrow[drr] & G_2'' \arrow[dr, tail, dotted] & K' \arrow[l, tail] \arrow[r, tail] \arrow[d, tail] & H'' \arrow[dl, tail, dotted] & H' \arrow[l, two heads] \arrow[dll] & & \\
 & & & & H_2 & & & &
\end{tikzcd}$$

This implies that $m_J \mathbin{;} c_1 \mathbin{;} d \mathbin{;} m_M = m_J \mathbin{;} c_1 \mathbin{;} m_K \mathbin{;} m_K' \mathbin{;} d \mathbin{;} m_M$. Furthermore $J' \rightarrowtail H_2$ and $M' \rightarrowtail H_2$ must be injective: $J \rightarrowtail G_2''$ is injective (since it is the first leg of the injective cospan $m_J \mathbin{;} c_1 \mathbin{;} m_K$) and if we compose with the injective morphism $G_2'' \rightarrowtail H_2$ we obtain $J' \rightarrowtail H_2$. Second $M' \rightarrowtail H''$ is injective (since it is the second leg of the injective cospan $m_K' \mathbin{;} d \mathbin{;} m_M$) and if we compose with the injective morphism $H'' \rightarrowtail H_2$ we obtain $M' \rightarrowtail H_2$. Hence we have $M(c_1 \mathbin{;} d) = M(c_2 \mathbin{;} d)$ and this concludes the proof. $\square$

Corollary 5.5. *Let a class $L_{J,K}$ of graphs with edge-injective interfaces J, K be called* edge-injectively recognizable *whenever $L_{J,K}$ is recognizable in $\mathscr{G}_{\mathrm{einj}}$. Then $L_{J,K}$ is recognizable in $\mathscr{G}$ if and only if it is edge-injectively recognizable.*

Proof. We adapt the proof of Prop. 5.4 by extending the notion of merger pairs. For $c\colon J \to G \leftarrow K$ let $M_e(c)$ be the set of merger pairs $\langle m_J, m_K \rangle$ such that $m_J \mathbin{;} c \mathbin{;} m_K$ is edge-injective. In addition let $M_l(c)$ be the set of pairs of cospans $\langle m_J, m_K \rangle$ such that $m_J \mathbin{;} c \mathbin{;} m_K$ is edge-injective on the left leg and injective on the right[2]. Clearly $M_l(c) \subseteq M_e(c)$.

Given a congruence $\equiv_{\mathrm{R}}$ that characterizes $L_{J,K}$ we now define $\equiv_{\mathrm{R}}'$ as follows: for two cospans $c_i : J \to G_i \leftarrow K$, where $i \in \{1, 2\}$ it holds that $c_1 \equiv_{\mathrm{R}}' c_2$

[2] This asymmetry is caused by the fact that we need a congruence with respect to composition on the right.

whenever $M_l(c_1) = M_l(c_2)$, $M_e(c_1) = M_e(c_2)$ and for all $(m_J, m_K) \in M_e(c_1)$ we have $m_J \,;\, c_1 \,;\, m_K \equiv_{\mathrm{R}} m_J \,;\, c_2 \,;\, m_k$. Clearly $\equiv'_{\mathrm{R}}$ is a refinement of $\equiv_{\mathrm{R}}$ and it is of finite index.

The rest of the proof carries through as before with some slight adaptations: note that we obtain that $J' \to H_1$, $M' \to H_1$ are edge-injective (instead of injective as before). Again we factorize $K \to H_1$ into an injective and surjective morphism, i.e., $K \to H_1 = K \twoheadrightarrow K' \rightarrowtail H_1$. Since we have $M_l(c_1) = M_l(c_2)$ we obtain a factorization $K \twoheadrightarrow K' \rightarrowtail H_2$ for $K \to H_2$ with an identical morphism $K \twoheadrightarrow K'$.

Hence the rest of the proof works as before, we only have to show that $M_l(c_1 \,;\, d) = M_l(c_2 \,;\, d)$ (and the same for M_e). Take for instance $\langle m_K, m_M\rangle \in M_l(c_1 \,;\, d)$. In this case $m_K \,;\, c_1 \,;\, d \,;\, m_M$ is edge-injective on the left and injective on the right which means $J' \to H_1$ is edge-injective and $M' \rightarrowtail H_1$ is injective. This implies that $J' \to G''_1$ is edge-injective and $M' \rightarrowtail H''$ is injective (since they split (edge-)injective morphisms). Since $K' \rightarrowtail G''_1$ and $K' \rightarrowtail H''$ are injective this means that $\langle m_J, m_K\rangle \in M_l(c_1)$ and $\langle m'_K, m_M\rangle \in M(d)$. Hence also $\langle m_J, m_K\rangle \in M_l(c_2)$ and we have that $J' \to G''_2$ is edge-injective. So $J' \to H_2$ is edge-injective since we postcompose $J' \to G''_2$ with the injective morphism $G''_2 \rightarrowtail H_2$. And finally $M' \rightarrowtail H_2$ is injective since we post-compose $M' \rightarrowtail H''$ with the injective morphism $H'' \rightarrowtail H_1$. □

Proposition 5.6. *Let a class $L_{J,K}$ of graphs with discrete interfaces J, K be called* discretely recognizable *whenever $L_{J,K}$ is recognizable in $\mathscr{G}_{\mathrm{dis}}$. Then $L_{J,K}$ is recognizable in $\mathscr{G}$ if and only if it is discretely recognizable.*

Proof. For both directions of the proof, we use the characterization of regularity via congruences (Prop. 3.6).

($\Rightarrow$): Trivial, because $\mathscr{G}_{\mathrm{dis}}$ is a subcategory of $\mathscr{G}$, and thus we can use the congruence for $\mathscr{G}$ also in the case of $\mathscr{G}_{\mathrm{dis}}$.

($\Leftarrow$): We use the result that whenever a language is edge-injectively recognizable, then it is recognizable (Cor. 5.5). Hence it is sufficient to show that whenever a language is discretely recognizable, then it is edge-injectively recognizable. So let $\equiv_{\mathrm{R}}$ be a congruence on discrete cospans (as in Def. 3.5). We need the following operator discr: let

$$c\colon J \xrightarrow{c_{\mathrm{L}}} G \xleftarrow{c_{\mathrm{R}}} K$$

be a cospan which is injective on edges. Now take as J', K' be the discrete graphs underlying J, K (which are obtained by removing all edges). Furthermore remove from G all edges that are in the image of c_{L}. Now we get a new cospan $c' = \mathsf{discr}(c)\colon J' \to G' \leftarrow K'$.

Observe that we have $\mathsf{discr}(c_1 \,;\, c_2) = \mathsf{discr}(c_1) \,;\, \mathsf{discr}(c_2)$ whenever c_1, c_2 are both edge-injective and $c_1 \,;\, c_2$ is defined. This does not hold whenever c_1, c_2 are not injective on edges since on the right-hand side of the equation fewer edges might be merged.

Now define a new equivalence $\equiv'_{\mathrm{R}}$, on edge-injective cospans, with $c_1 \equiv'_{\mathrm{R}} c_2$ if and only if $\mathsf{discr}(c_1) \equiv_{\mathrm{R}} \mathsf{discr}(c_2)$. We now show that $\equiv'_{\mathrm{R}}$ is a congruence: let

$c_1 \equiv'_{\mathrm{R}} c_2$. Then we have that

$$\mathsf{discr}(c_1 \mathbin{;} d) = \mathsf{discr}(c_1) \mathbin{;} \mathsf{discr}(d) \equiv_{\mathrm{R}} \mathsf{discr}(c_2) \mathbin{;} \mathsf{discr}(d) = \mathsf{discr}(c_2 \mathbin{;} d) \ ,$$

which implies that $c_1 \mathbin{;} d \equiv'_{\mathrm{R}} c_2 \mathbin{;} d$. □

5.4. Equivalence of the two Notions of Recognizability

Definition A.3. *Let J be a discrete graph with n nodes. We define the following cospans with source J (See Fig. 1):*

(i) Let $s \colon K \to J$ be a morphism from a discrete graph K to J. Then we define:

$$\mathit{redef}_s \colon J \xrightarrow{\mathsf{id}_J} J \xleftarrow{s} K \ .$$

(ii) Let D be a discrete graph and $t \colon J \to D$ a morphism.

$$\mathit{fuse}_t \colon J \xrightarrow{t} D \xleftarrow{t} J \ .$$

In the following we use the monoidal operation $\oplus$ which is the disjoint sum on cospans. Formally it is obtained by taking the coproducts of the middle graph and of the inner and outer interfaces. The new interface morphisms are then obtained as mediating morphisms.

Lemma A.4. *Let the n-node discrete graph J and cospan $c \colon \emptyset \to G \leftarrow J$ be given.*

(i) Let $\sigma \colon \mathbb{N}_m \to \mathbb{N}_n$ be a function. For each m-node discrete graph K there exists an arrow $s \colon K \to J$ such that

$$\mathsf{bend}(c \mathbin{;} \mathit{redef}_s) = \mathsf{redef}_\sigma(\mathsf{bend}(c)) \ .$$

(ii) Let θ be an equivalence relation on $\mathbb{N}_n$. There exist a discrete graph D and arrow $t \colon J \to D$ such that

$$\mathsf{bend}(c \mathbin{;} \mathit{fuse}_t) = \mathsf{fuse}_\theta(\mathsf{bend}(c)) \ .$$

Proof. (i) We consider σ as a graph morphism from Dis_m to Dis_n and take $s = \mathsf{di}_K^{-1} \mathbin{;} \sigma \mathbin{;} \mathsf{di}_J$. The composed cospan $c \mathbin{;} \mathit{redef}_s$ now looks like

$$(c \mathbin{;} \mathit{redef}_s) \colon \emptyset \xrightarrow{c_{\mathrm{L}}} G \xleftarrow{s;c_{\mathrm{R}}} K \ .$$

Then we do (because J and K are discrete, we confuse graph morphisms and functions on nodes):

$$\begin{aligned}
\mathsf{bend}(c \mathbin{;} \mathit{redef}_s) &= \langle G, \mathsf{di}_K \mathbin{;} s \mathbin{;} c_{\mathrm{R}} \rangle \\
&= \langle G, \mathsf{di}_K \mathbin{;} \mathsf{di}_K^{-1} \mathbin{;} \sigma \mathbin{;} \mathsf{di}_J \mathbin{;} c_{\mathrm{R}} \rangle \\
&= \langle G, \sigma \mathbin{;} \mathsf{di}_J \mathbin{;} c_{\mathrm{R}} \rangle \\
&= \mathsf{redef}_\sigma(\langle G, \mathsf{di}_J \mathbin{;} c_{\mathrm{R}} \rangle) \\
&= \mathsf{redef}_\sigma(\mathsf{bend}(c)) \ .
\end{aligned}$$

(ii) Let θ_{map} be the function which maps each element of $\mathbb{N}_n$ to its respective θ-equivalence class, let D be the discrete graph with node set $\{[\![k]\!]_\theta \mid k \in \mathbb{N}_n\}$ and define the arrow t as $t = \mathsf{di}_J^{-1} \mathbin{;} \theta_{\mathrm{map}}$. Because cospan composition is define by a pushout, the cospan $(c \mathbin{;} \mathit{fuse}_t)$ is the same as c but with the nodes fused according to θ. $\square$

Lemma A.5. *For cospans $c \colon \emptyset \to G \leftarrow J$ and $d \colon \emptyset \to H \leftarrow M$ it holds that $\mathsf{bend}(c \oplus d) = \mathsf{bend}(c) \oplus \mathsf{bend}(d)$.*

Proof. From the following equation:

$$\begin{aligned}
\mathsf{bend}(c \oplus d) &= \langle G \oplus H, \mathsf{di}_{J \oplus K} \mathbin{;} c_{\mathrm{R}} \oplus d_{\mathrm{R}} \rangle \\
&= \langle G \oplus H, (\mathsf{di}_J ; c_{\mathrm{R}}) \odot (\mathsf{di}_K ; d_{\mathrm{R}}) \rangle \\
&= \langle G, \mathsf{di}_J ; c_{\mathrm{R}} \rangle \oplus \langle H, \mathsf{di}_K ; d_{\mathrm{R}} \rangle \\
&= \mathsf{bend}(c) \oplus \mathsf{bend}(d) \ .
\end{aligned}$$

$\square$

Lemma A.6. *Let cospans $c \colon I \to G \leftarrow J$ and $d \colon J \to H \leftarrow K$ be given, where I has m nodes, J has n nodes and K has k nodes. Then:*

$$\mathsf{bend}(c \mathbin{;} d) = \mathsf{redef}_\sigma(\mathsf{fuse}_\theta(\mathsf{bend}(c) \oplus \mathsf{bend}(d))) \ ,$$

where $\sigma \colon \mathbb{N}_{m+k} \to \mathbb{N}_{m+2n+k}$ is defined as

$$\sigma(x) = \begin{cases} x + 2n & \text{if } x \geq m \\ x & \text{otherwise,} \end{cases}$$

and θ is the smallest equivalence relation on $\mathbb{N}_{m+2n+k}$, which, for $m \leq x < m+n$, relates x to $x + n$.

Proof. We have that:

$$\mathbb{M}_0 = \mathsf{bend}(c) \oplus \mathsf{bend}(d) = \langle G \oplus H, (\mathsf{di}_I \mathbin{;} c_{\mathrm{L}}) \odot (\mathsf{di}_J \mathbin{;} c_{\mathrm{R}}) \odot (\mathsf{di}_J \mathbin{;} d_{\mathrm{L}}) \odot (\mathsf{di}_K \mathbin{;} d_{\mathrm{R}}) \rangle$$

The cospan composition of c and d looks as follows:

$$\begin{tikzcd}
 & & J \arrow[dl, "c_{\mathrm{R}}"'] \arrow[dr, "d_{\mathrm{L}}"] & & \\
I \arrow[r, "c_{\mathrm{L}}"] & G \arrow[dr, "f"'] & \text{(PO)} & H \arrow[dl, "g"] & K \arrow[l, "d_{\mathrm{R}}"'] \\
 & & M & &
\end{tikzcd}$$

Because the middle diamond is a pushout, we have that

$$\begin{aligned}
&\mathbb{M}_0' = \mathsf{fuse}_\theta(\mathbb{M}_0) = \\
&\qquad \langle M, (\mathsf{di}_I \mathbin{;} c_{\mathrm{L}} \mathbin{;} f) \odot (\mathsf{di}_J \mathbin{;} c_{\mathrm{R}} \mathbin{;} f) \odot (\mathsf{di}_J \mathbin{;} d_{\mathrm{L}} \mathbin{;} g) \odot (\mathsf{di}_K \mathbin{;} d_{\mathrm{R}} \mathbin{;} g) \rangle \ .
\end{aligned}$$

Finally,

$$\mathbb{M} = \mathsf{redef}_\sigma(\mathbb{M}'_0) = \langle M, (\mathsf{di}_I; c_\mathrm{L}; f) \odot (\mathsf{di}_K; d_\mathrm{R}; g)\rangle \ .$$

On the other hand, a simple application of the definition reveals that

$$\mathsf{bend}(c \ ; d) = \langle M, (\mathsf{di}_I; c_\mathrm{L}; f) \odot (\mathsf{di}_K; d_\mathrm{R}; g)\rangle \ ,$$

as required. ☐

Lemma A.7. *Let* $c\colon \emptyset \to G \leftarrow J$, $d\colon \emptyset \to H \leftarrow J$, $c'\colon \emptyset \to G' \leftarrow K$ *and* $d'\colon \emptyset \to H' \leftarrow K$ *be cospan, and* $\equiv_\mathrm{R}$ *a congruence on cospans. If* $\mathsf{bend}(c) = \mathsf{bend}(c')$ *and* $\mathsf{bend}(d) = \mathsf{bend}(d')$ *then it holds that* $c \equiv_\mathrm{R} d$ *if and only if* $c' \equiv_\mathrm{R} d'$*.*

Proof. Follows from the fact that if $\mathsf{bend}(c) = \mathsf{bend}(c')$ and $\mathsf{bend}(d) = \mathsf{bend}(d')$, an isomorphism $f\colon K \to J$ exists such that $c_\mathrm{R} = f \ ; {c_\mathrm{R}}'$. This allows us to construct the cospan

$$e\colon J \xrightarrow{\mathsf{id}_J} J \xleftarrow{f} K$$

such that $c \ ; e = c'$ and $d \ ; e = d'$, which, because $\equiv_\mathrm{R}$ is a congruence by assumption, proves the left-to-right direction of the proof. The other direction works symmetrically. ☐

Theorem 5.9. *Let J be a discrete graph. A set of graphs L is the $(\emptyset, J)$-language of some automaton functor $\mathcal{A}$ if and only if* $\mathsf{bend}(L)$ *is Courcelle-recognizable.*

Proof. $(\Rightarrow)$ By Prop. 3.6 there exists a locally finite congruence $\equiv_\mathrm{R} = \{R_{J,K} \mid K, M$ are discrete graphs$\}$ such that L is the union of some $R_{\emptyset,J}$-equivalence classes. We define:

$$\mathsf{bend}(c) \equiv_\mathrm{C} \mathsf{bend}(d) \text{ if } c \equiv_\mathrm{R} d \ .$$

This is well-defined because of Lemma A.7 and the fact that bend is surjective. We now show that $\equiv_\mathrm{C}$ is a congruence in Courcelle's sense.

– *Redefinition and fusion.* Let $\mathbb{G} = \langle G, \zeta_\mathbb{G}\rangle$, $\mathbb{H} = \langle H, \zeta_\mathbb{H}\rangle$ be n-ary graphs such that $\mathbb{G} \equiv_\mathrm{C} \mathbb{H}$, and consider the cospans

$$c\colon \emptyset \xrightarrow{c_\mathrm{L}} G \xleftarrow{c_\mathrm{R}} J \qquad d\colon \emptyset \xrightarrow{d_\mathrm{L}} H \xleftarrow{d_\mathrm{R}} J$$

such that $\mathsf{bend}(c) = \mathbb{G}$ and $\mathsf{bend}(d) = \mathbb{H}$. By definition it holds that $c \equiv_\mathrm{R} d$. By Lemma A.4 (i) $\mathsf{redef}_\sigma(\mathbb{G}) = \mathsf{bend}(c \ ; \mathit{redef}_s)$ and $\mathsf{redef}_\sigma(\mathbb{H}) = \mathsf{bend}(d \ ; \mathit{redef}_s)$ (for suitably chosen s). Because $\equiv_\mathrm{R}$ is a congruence, and $c \equiv_\mathrm{R} d$, it holds, by definition, that $\mathsf{redef}_\sigma(\mathbb{G}) \equiv_\mathrm{C} \mathsf{redef}_\sigma(\mathbb{H})$, as required. In the case of fusion, we arrive at the result analogously, by using Lemma A.4 (ii).

– *Disjoint union.* Let $\mathbb{G}_1, \mathbb{H}_1$ be n-ary graphs, and $\mathbb{G}_2, \mathbb{H}_2$ m-ary graphs, such that $\mathbb{G}_i \equiv_\mathrm{C} \mathbb{H}_i$, for $i \in \{1,2\}$. Consider cospans c_1, c_2, d_1, d_2 such that $\mathsf{bend}(c_i) = \mathbb{G}_i$ and $\mathsf{bend}(d_i) = \mathbb{H}_i$. By definition, it holds that $c_i \equiv_\mathrm{R} d_i$. By Lemma A.5 it holds that $\mathbb{G}_1 \oplus \mathbb{G}_2 = \mathsf{bend}(c_1 \oplus c_2)$ and $\mathbb{H}_1 \oplus \mathbb{H}_2 = \mathsf{bend}(d_1 \oplus d_2)$. Since it holds that

$$\begin{aligned} c_1 \oplus c_2 = c_1 \ ; (\mathsf{id}_{J_1} \oplus c_2) &\equiv_\mathrm{R} d_1 \ ; (\mathsf{id}_{J_1} \oplus c_2) \\ &= c_2 \ ; (d_1 \oplus \mathsf{id}_{J_2}) \equiv_\mathrm{R} d_2 \ ; (d_1 \oplus \mathsf{id}_{J_2}) = d_1 \oplus d_2 \ , \end{aligned}$$

by definition it must be the case that $\mathbb{G}_1 \oplus \mathbb{G}_2 \equiv_\mathrm{C} \mathbb{H}_1 \oplus \mathbb{H}_2$.

($\Leftarrow$) Let the congruence $\equiv_{\mathrm{C}}$ in the sense of Courcelle be given. We define the relation $\equiv_{\mathrm{R}}$ as follows:

$$c \equiv_{\mathrm{R}} d \text{ if } \mathsf{bend}(c) \equiv_{\mathrm{C}} \mathsf{bend}(d) \ .$$

We show that $\equiv_{\mathrm{R}}$ is a congruence. Let the following cospans be given:

$$\begin{aligned} c\colon I \to G \leftarrow J & \qquad\qquad \mathsf{bend}(c) = \mathbb{G} \\ d\colon I \to H \leftarrow J & \qquad\qquad \mathsf{bend}(d) = \mathbb{H} \\ p\colon J \to M \leftarrow K & \qquad\qquad \mathsf{bend}(p) = \mathbb{M} \ , \end{aligned}$$

where $c \equiv_{\mathrm{R}} d$. We need to show that $(c \mathbin{;} p) \equiv_{\mathrm{R}} (d \mathbin{;} p)$. By definition, it holds that $\mathbb{G} \equiv_{\mathrm{C}} \mathbb{H}$, and since $\equiv_{\mathrm{C}}$ is a congruence, it holds that

$$\mathsf{redef}_{\sigma}(\mathsf{fuse}_{\theta}(\mathbb{G} \oplus \mathbb{M})) \equiv_{\mathrm{C}} \mathsf{redef}_{\sigma}(\mathsf{fuse}_{\theta}(\mathbb{H} \oplus \mathbb{M})) \ ,$$

where σ, θ are as given in Lemma A.6. By the same lemma, it holds that

$$\begin{aligned} \mathsf{bend}(c \mathbin{;} p) &= \mathsf{redef}_{\sigma}(\mathsf{fuse}_{\theta}(\mathbb{G} \oplus \mathbb{M})) \\ \mathsf{bend}(d \mathbin{;} p) &= \mathsf{redef}_{\sigma}(\mathsf{fuse}_{\theta}(\mathbb{H} \oplus \mathbb{M})) \ , \end{aligned}$$

and therefore, by definition, $(c \mathbin{;} p) \equiv_{\mathrm{R}} (d \mathbin{;} p)$, as required. $\square$