Term Graph Representations for Cyclic Lambda-Terms*

Clemens Grabmayer
Department of Philosophy
Utrecht University
The Netherlands
clemens@phil.uu.nl

Jan Rochel
Department of Computing Sciences
Utrecht University
The Netherlands
jan@rochel.info

We study various representations for cyclic λ-terms as higher-order or as first-order term graphs. We focus on the relation between 'λ-higher-order term graphs' (λ-ho-term-graphs), which are first-order term graphs endowed with a well-behaved scope function, and their representations as 'λ-term-graphs', which are plain first-order term graphs with scope-delimiter vertices that meet certain scoping requirements. Specifically we tackle the question: Which class of first-order term graphs admits a faithful embedding of λ-ho-term-graphs in the sense that (i) the homomorphism-based sharing-order on λ-ho-term-graphs is preserved and reflected, and (ii) the image of the embedding corresponds closely to a natural class (of λ-term-graphs) that is closed under homomorphism?

We systematically examine whether a number of classes of λ-term-graphs have this property, and we find a particular class of λ-term-graphs that satisfies this criterion. Term graphs of this class are built from application, abstraction, variable, and scope-delimiter vertices, and have the characteristic feature that the latter two kinds of vertices have back-links to the corresponding abstraction.

This result puts a handle on the concept of subterm sharing for higher-order term graphs, both theoretically and algorithmically: We obtain an easily implementable method for obtaining the maximally shared form of λ-ho-term-graphs. Furthermore, we open up the possibility to pull back properties from first-order term graphs to λ-ho-term-graphs, properties such as the complete lattice structure of bisimulation equivalence classes with respect to the sharing order.

1 Introduction

Cyclic lambda-terms typically represent infinite λ-terms. In this paper we study term graph representations of cyclic λ-terms and their respective notions of homomorphism, or functional bisimulation.

The context in which the results presented in this paper play a central role is our research on subterm sharing as present in terms of languages such as the λ-calculus with letrec [8, 1], with recursive definitions [2], or languages with μ-recursion [3], and our interest in describing maximal sharing in such settings. Specifically we want to obtain concepts and methods as follows:

- an efficient test for term equivalence with respect to α-renaming and unfolding;
- a notion of 'maximal subterm sharing' for terms in the respective language;
- the efficient computation of the maximally shared form of a term;
- a sharing (pre-)order on unfolding-equivalent terms.

Now our approach is to split the work into a part that concerns properties specific to concrete languages, and into a part that deals with aspects that are common to most of the languages with constructs for expressing subterm sharing. To this end we set out to find classes of term graphs that facilitate faithful interpretations of terms in such languages as (higher-order, and eventually first-order) term graphs, and that are 'well-behaved' in the sense that maximally shared term graphs do always exist. In this way the

*This work was started, and in part carried out, within the framework of the project NWO project *Realising Optimal Sharing (ROS)*, project number 612.000.935, under the direction of Vincent van Oostrom and Doaitse Swierstra.

Submitted to:
TERMGRAPH 2012

task can be divided into two parts: an investigation of sharing for term graphs with higher-order features (the aim of this paper), and a study of language-specific aspects of sharing (the aim of a further paper).

Here we study a variety of classes of term graphs for denoting cyclic λ-terms, term graphs with higher-order features and their first-order 'implementations'. All higher-order term graphs we consider are built from three kinds of vertices, which symbolize applications, abstractions, and variable occurrences, respectively. They also carry features that describe notions of *scope*, which are subject to certain conditions that guarantee the meaningfulness of the term graph (that a λ-term is denoted), and in some cases are crucial to define *binding*. The first-order implementations do not have these additional features, but they may contain scope-delimiter vertices.

In particular we study the following three kinds (of classes) of term graphs:

λ-Higher-order-term-graphs (Section 3) are extensions of first-order term graphs by adding a scope function that assigns a set of vertices, its scope, to every abstraction vertex. There are two variants, one with and one without an edge (a *back-link*) from each variable occurrence to its corresponding abstraction vertex. The class with back-links is related to *higher-order term graphs* as defined by Blom in [4], and in fact is an adaptation of that concept for the purpose of representing λ-terms.

Abstraction-prefix based λ-higher-order-term-graphs (Section 4) do not have a scope function but assign, to each vertex w, an abstraction prefix consisting of a word of abstraction vertices that includes those abstractions for which w is in their scope (it actually lists all abstractions for which w is in their 'extended scope' [6]). Abstraction prefixes are aggregations of scope information that is relevant for and locally available at individual vertices.

λ-Term-graphs with scope delimiters (Section 5) are plain first-order term graphs intended to represent higher-order term graphs of the two sorts above, and in this way stand for λ-terms. Instead of relying upon additional features for describing scopes, they use scope-delimiter vertices to signify the end of scopes. Variable occurrences as well as scoping delimiters may or may not have back-links to their corresponding abstraction vertices.

Each of these classes induces a notion of homomorphism (functional bisimulation) and bisimulation. Homomorphisms increase sharing in term graphs, and in this way induce a sharing order. They preserve the unfolding semantics of term graphs[1], and therefore are able to preserve λ-terms that are denoted by term graphs in the unfolding semantics. Term graphs from the classes we consider always represent finite or infinite λ-terms, and in this sense are not 'meaningless'. But this is not shown here. Instead, we lean on motivating examples, intuitions, and the concept of higher-order term graph from [4].

We establish a bijective correspondence between the former two classes, and a correspondence between the latter two classes that is 'almost bijective' (bijective up to sharing or unsharing of scope delimiter vertices). All of these correspondences preserve and reflect the sharing order. Furthermore, we systematically investigate which specific class of λ-term-graphs is closed under homomorphism and renders the mentioned correspondences possible. We prove (in Section 6) that this can only hold for a class in which both variable-occurrence and scope-delimiter vertices have back-links to corresponding abstractions, and establish (in Section 7) that the subclass containing only λ-term-graphs with eager application of scope-closure satisfies these properties. For this class the correspondences allow us:

- to transfer properties known for first-order term graphs, such as the existence of a maximally shared form, from λ-term-graphs to the corresponding classes of higher-order λ-term-graphs;
- to implement maximal sharing for higher-order λ-term-graphs (with eager scope closure) via bisimulation collapse of the corresponding first-order λ-term-graphs (see algorithm in Section 8).

We stress that this paper in its present form is only a report about work in progress, and, while a number of proofs are included, predominantly has the character of an extended abstract.

[1] While this is well-known for first-order term graphs, it can also be proved for the higher-order term graphs considered here.

2 Preliminaries

By $\mathbb{N}$ we denote the natural numbers including zero. For words w over an alphabet A we denote the length of w by $|w|$. For a function $f : A \to B$ we denote by $\mathrm{dom}(f)$ the domain, and by $\mathrm{im}(f)$ the image of f; and for $A_0 \subseteq A$ we denote by $f|_{A_0}$ the restriction of f to A_0.

Let Σ be a signature with arity function $ar : \Sigma \to \mathbb{N}$. A *term graph over* Σ is a tuple $\langle V, lab, args, r\rangle$ where V is a set of *vertices*, $lab : V \to \Sigma$ the *(vertex) label function*, $args : V \to V^*$ the *argument function* that maps every vertex v to the word $args(v)$ consisting of the $ar(lab(v))$ successor vertices of v (hence it holds $|args(v)| = ar(lab(v))$), r, the *root* is a vertex in V, and where every vertex is reachable from the root (by a path that arises by repeatedly going from a vertex to one of its successors). (Note this reachability condition, and mind the fact that term graphs may have infinitely many vertices.) By a Σ-*term-graph* we mean a term graph over Σ. And by $\mathrm{TG}(\Sigma)$ we mean the class of all term graphs over Σ.

Let G be a term graph over signature Σ. As useful notation for picking out any vertex or the i-th vertex from among the ordered successors of a vertex v in G we define the (not indexed) edge relation $\rightarrowtail \subseteq V \times V$, and for each $i \in \mathbb{N}$ the indexed edge relation $\rightarrowtail_i \subseteq V \times V$, between vertices by stipulating that:

$$\begin{aligned} w \rightarrowtail_i w' &:\iff \exists w_0, \ldots, w_n \in V.\ args(w) = w_0 \ldots w_n \ \wedge\ w' = w_i \\ w \rightarrowtail w' &:\iff \exists i \in \mathbb{N}.\ w \rightarrowtail_i w' \end{aligned}$$

holds for all $w, w' \in V$. We write $w \overset{f}{\rightarrowtail}_i w'$ if $w \rightarrowtail_i w' \wedge lab(w) = f$ holds for $w, w' \in V$, $i \in \mathbb{N}$, $f \in \Sigma$, to indicate the label at the source of an edge. A *path* in G is a tuple of the form $\langle w_0, k_0, w_1, k_1, w_2, \ldots, w_{n-1}, k_{n-1}, w_n\rangle$ where $w_0, \ldots, w_n \in V$ and $n, k_0, k_1, \ldots, k_{n-1} \in \mathbb{N}$ such that $w_0 \rightarrowtail_{k_0} w_1 \rightarrowtail_{k_1} w_2 \cdots w_{n-1} \rightarrowtail_{k_{n-1}} w_n$ holds; paths will usually be denoted in the latter form, using indexed edge relations. An *access path* of a vertex w of G is a path that starts at the root of G, ends in w, and does not visit any vertex twice. Note that every vertex w has at least one access path: since every vertex in a term graph is reachable from the root, there is a path π from r to w; then an access path of w can be obtained from π by repeatedly cutting out *cycles*, that is, parts of the path between different visits to one and the same vertex.

In the sequel, let $G_1 = \langle V_1, lab_1, args_1, r_1\rangle$, $G_2 = \langle V_2, lab_2, args_2, r_2\rangle$ be term graphs over signature Σ.

A *homomorphism*, also called a *functional bisimulation*, from G_1 to G_2 is a morphism from the structure $\langle V_1, lab_1, args_1, r_1\rangle$ to the structure $\langle V_2, lab_2, args_2, r_2\rangle$, that is, a function $h : V_1 \to V_2$ such that, for all $v \in V_1$ it holds:

$$\left.\begin{aligned} lab_1(v) &= lab_2(h(v)) && \text{(labels)} \\ \bar{h}(args_1(v)) &= args_2(h(v)) && \text{(arguments)} \\ h(r_1) &= r_2 && \text{(roots)} \end{aligned}\right\} \tag{1}$$

where $\bar{h}$ is the homomorphic extension of h to words over V_1, that is, to the function $\bar{h} : V_1^* \to V_2^*$, $v_1 \ldots v_n \mapsto h(v_1) \ldots h(v_n)$. In this case we write $G_1 \trianglerighteq_h G_2$, or $G_2 \trianglelefteq_h G_1$. And we write $G_1 \trianglerighteq G_2$, or for that matter $G_2 \trianglelefteq G_1$, if there is a homomorphism (a functional bisimulation) from G_1 to G_2.

Let $\mathsf{f} \in \Sigma$. We write $G_1 \trianglerighteq^{\mathsf{f}} G_2$ or $G_2 \trianglelefteq^{\mathsf{f}} G_1$ if there is a homomorphism h between G_1 and G_2 with the property that for all $w_1, w_2 \in V_1$ with $w_1 \neq w_2$ it holds that $h(w_1) = h(w_2) \Rightarrow lab_1(w_1) = lab_1(w_2) = \mathsf{f}$, that is, if h only 'shares', or 'identifies', vertices when they have label f. If h is such a homomorphism, we also write $G_1 \trianglerighteq^{\mathsf{f}}_h G_2$ or $G_2 \trianglelefteq^{\mathsf{f}}_h G_1$.

A *bisimulation* between G_1 and G_2 is a term graph $G = \langle R, lab, args, r\rangle$ over Σ with $R \subseteq V_1 \times V_2$ and $r = \langle r_1, r_2\rangle$ such that $G_1 \trianglelefteq_{\pi_1} G \trianglerighteq_{\pi_2} G_2$ where π_1 and π_2 are projection functions, defined, for $i \in \{1, 2\}$, by $\pi_i : V_1 \times V_2 \to V_i$, $\langle v_1, v_2\rangle \mapsto v_i$. In this case we write $G_1 \leftrightarrows_R G_2$. And we write $G_1 \leftrightarrows G_2$ if there is a bisimulation between G_1 and G_2.

Alternatively, bisimulations for term graphs can be defined directly as relations on the vertex sets, obtaining the same notion of bisimilarity. In this formulation, a bisimulation between G_1 and G_2 is a relation $R \subseteq V_1 \times V_2$ such that the following conditions hold, for all $\langle v, v' \rangle \in R$:

$$\langle r_1, r_2 \rangle \in R \qquad \text{(roots)}$$
$$lab_1(v) = lab_2(v') \qquad \text{(labels)}$$
$$\langle args_1(v), args_2(v') \rangle \in R^* \qquad \text{(arguments)}$$

where $R^* := \left\{ \langle v_1 \cdots v_k, v'_1 \cdots v'_k \rangle \mid v_1, \ldots, v_k \in V_1, v'_1, \ldots, v'_k \in V_2 \text{ for } k \in \mathbb{N} \text{ such that} \langle v_i, v'_i \rangle \in R \text{ for all } 1 \leq i \leq k \right\}$.

Bisimulation is an equivalence relation on the class $\mathrm{TG}(\Sigma)$ of term graphs over a signature Σ. The homomorphism (functional bisimulation) relation $\twoheadrightarrow$ is a pre-order on term graphs over a given signature Σ, and it induces a partial order on isomorphism equivalence classes of term graphs over Σ. We will refer to $\twoheadrightarrow$ as the *sharing pre-order*, and will speak of it as *sharing order*, dropping the 'pre'. The bisimulation equivalence class $[[G]_\sim]_{\leftrightarrows} := \{[G']_\sim \mid G' \leftrightarrows G\}$ of the isomorphism equivalence class $[G]_\sim$ of a term graph G is ordered by homomorphism $\twoheadrightarrow$ such that $\langle [[G]_\sim]_{\leftrightarrows}, \twoheadrightarrow \rangle$ is a complete lattice [3, 9].Note that, different from e.g. [9], we use the order relation $\twoheadrightarrow$ in the same direction as $\leq$: if $G_1 \twoheadrightarrow G_2$, then G_2 is greater or equal to G_1 in the ordering $\twoheadrightarrow$ (indicating that sharing is typically increased from G_1 to G_2).

Let $\mathcal{K} \subseteq \mathrm{TG}(\Sigma)$ be a subclass of the term graphs over Σ, for a signature Σ. We say that $\mathcal{K}$ is *closed under homorphism* (*closed under bisimulation*) if $G \twoheadrightarrow G'$ (resp. $G \leftrightarrows G'$) for $G, G' \in \mathrm{TG}(\Sigma)$ with $G \in \mathcal{K}$ implies $G' \in \mathcal{K}$. Note these concepts are invariant under considering other signatures Σ' with $\mathcal{K} \subseteq \mathrm{TG}(\Sigma')$.

3 λ-higher-order-Term-Graphs

By Σ^λ we designate the signature $\{@, \lambda\}$ with $ar(@) = 2$, and $ar(\lambda) = 1$. By Σ^λ_i, for $i \in \{0, 1\}$, we denote the extension $\Sigma^\lambda \cup \{0\}$ of Σ^λ where $ar(0) = i$. The classes of term graphs over Σ^λ_0 and Σ^λ_1 are denoted by $\mathcal{T}_0$ and $\mathcal{T}_1$, respectively.

Let $G = \langle V, lab, args, r \rangle$ be a term graph over a signature extending Σ^λ or Σ^λ_i, for $i \in \{0, 1\}$. By $V(\lambda)$ we designate the set of *abstraction vertices* of G, that is, the subset of V consisting of all vertices with label λ; more formally, $V(\lambda) := \{v \in V \mid lab(v) = \lambda\}$. Analogously, the sets $V(@)$ and $V(0)$ of *application vertices* and *variable vertices* of G are defined as the sets consisting of all vertices in V with label @ or label 0, respectively. Whether the variable vertices have an outgoing edge depends on the value of i. The intention is to consider two variants of term graphs, one with and one without variable back-links to their corresponding abstraction.

A 'λ-higher-order-term-graph' consists of a Σ^λ_i-term-graph together with a scope function that maps abstraction vertices to their scopes ('extended scopes' in [6]), which are subsets of the set of vertices.

Definition 1 (λ-ho-term-graph) Let $i \in \{0, 1\}$. A *λ-ho-term-graph* (short for *λ-higher-order-term-graph*) *over* Σ^λ_i, is a five-tuple $\mathcal{G} = \langle V, lab, args, r, Sc \rangle$ where $G_\mathcal{G} = \langle V, lab, args, r \rangle$ is a Σ^λ_i-term-graph, called the term graph *underlying* $\mathcal{G}$, and $Sc : V(\lambda) \to \wp(V)$ is the *scope function* of $\mathcal{G}$ (which maps an abstraction vertex v to a set of vertices called its *scope*) that together with $G_\mathcal{G}$ fulfills the following conditions: For all $k \in \{0, 1\}$, all vertices $w, w_0, w_1 \in V$, and all abstraction vertices $v, v_0, v_1 \in V(\lambda)$ it holds:

$$\Rightarrow \; r \notin Sc^-(v) \qquad \text{(root)}$$
$$\Rightarrow \; v \in Sc(v) \qquad \text{(self)}$$
$$v_1 \in Sc^-(v_0) \;\Rightarrow\; Sc(v_1) \subseteq Sc^-(v_0) \qquad \text{(nest)}$$
$$w \twoheadrightarrow_k w_k \;\wedge\; w_k \in Sc^-(v) \;\Rightarrow\; w \in Sc(v) \qquad \text{(closed)}$$

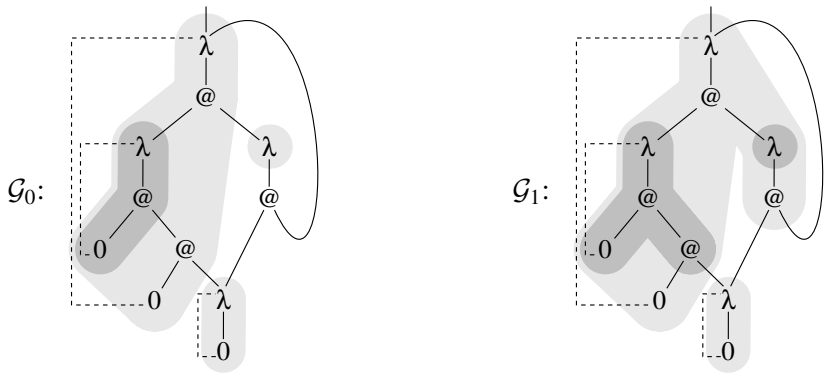


Figure 1: $\mathcal{G}_0$ and $\mathcal{G}_1$ are λ-ho-term-graphs in $\mathcal{H}^{\lambda}_i$ whereby the dotted back-link edges are present for $i = 1$, but absent for $i = 0$. The underlying term graphs of $\mathcal{G}_0$ and $\mathcal{G}_1$ are identical but their scope functions (signified by the shaded areas) differ. While in $\mathcal{G}_0$ scopes are chosen as small as possible, which we refer to as 'eager scope closure', in $\mathcal{G}_1$ some scopes are closed only later in the graph.

$$w \in V(0) \;\Rightarrow\; \exists v_0 \in V(\lambda).\, w \in Sc^-(v_0) \qquad \text{(scope)}_0$$

$$w \in V(0) \;\wedge\; w \twoheadrightarrow_0 w_0 \;\Rightarrow\; \begin{cases} w_0 \in V(\lambda) \;\wedge \\ \;\wedge\; \forall v \in V(\lambda). \\ \qquad (w \in Sc(v) \Leftrightarrow w_0 \in Sc(v)) \end{cases} \qquad \text{(scope)}_1$$

where $Sc^-(v) := Sc(v) \setminus \{v\}$. Note that if $i = 0$, then $(\text{scope})_1$ is trivially true and hence superfluous, and if $i = 1$, then $(\text{scope})_0$ is redundant, because it follows from $(\text{scope})_1$ in this case. For $w \in V$ and $v \in V(\lambda)$ we say that v is a *binder for* w if $w \in Sc(v)$, and we designate by $\mathrm{bds}(w)$ the set of binders of w.

The classes of λ-ho-term-graphs over Σ^{λ}_0 and Σ^{λ}_1 will be denoted by $\mathcal{H}^{\lambda}_0$ and $\mathcal{H}^{\lambda}_1$.

See Fig. 1 for two different λ-ho-term-graphs over Σ^{λ}_i both of which represent the same term in the λ-calculus with letrec, namely **letrec** $f = \lambda x.\,(\lambda y.\, y\,(x\,g))\,(\lambda z.\, g\,f)$, $g = \lambda u.\, u$ **in** f.

The following lemma states some basic properties of the scope function in λ-ho-term-graphs. Most importantly, scopes in λ-ho-term-graphs are properly nested, in analogy with scopes in finite λ-terms.

Lemma 2 *Let $i \in \{0,1\}$, and let $\mathcal{G} = \langle V, lab, args, r, Sc\rangle$ be a λ-ho-term-graph over Σ^{λ}_i. Then the following statements hold for all $w \in V$ and $v, v_1, v_2 \in V(\lambda)$:*

(i) If $w \in Sc(v)$, then v is visited on every access path of w, and all vertices on access paths of w after v are in $Sc^-(v)$. Hence (since $G_{\mathcal{G}}$ is a term graph, every vertex has an access path) $\mathrm{bds}(w)$ is finite.

(ii) If $Sc(v_1) \cap Sc(v_2) \neq \varnothing$ for $v_1 \neq v_2$, then $Sc(v_1) \subseteq Sc^-(v_2)$ or $Sc(v_2) \subseteq Sc^-(v_1)$. As a consequence, if $Sc(v_1) \cap Sc(v_2) \neq \varnothing$, then $Sc(v_1) \subsetneqq Sc(v_2)$ or $Sc(v_1) = Sc(v_2)$ or $Sc(v_2) \subsetneqq Sc(v_1)$.

(iii) If $\mathrm{bds}(w) \neq \varnothing$, then $\mathrm{bds}(w) = \{v_0, \ldots, v_n\}$ for $v_0, \ldots, v_n \in V(\lambda)$ and $Sc(v_n) \subsetneqq Sc(v_{n-1}) \ldots \subsetneqq Sc(v_0)$.

Proof. Let $i \in \{0,1\}$, and let $\mathcal{G} = \langle V, lab, args, r, Sc\rangle$ be a λ-ho-term-graph over Σ^{λ}_i.

For showing (i), let $w \in V$ and $v \in V(\lambda)$ be such that $w \in Sc(v)$. Suppose that $\pi : r = w_0 \twoheadrightarrow_{k_0} w_1 \twoheadrightarrow_{k_1} w_2 \cdots \twoheadrightarrow_{k_{n-1}} w_n = w$ is an access path of w. If $w = v$, then nothing remains to be shown. Otherwise $w_n = w \in Sc^-(v)$, and, if $n > 0$, then by (closed) it follows that $w_{n-1} \in Sc(v)$. This argument can be repeated to find subsequently smaller i with $w_i \in Sc(v)$ and $w_{i+1}, \ldots, w_n \in Sc^-(v)$. We can proceed as long as

$w_i \in Sc^-(v)$. But since, due to (root), $w_0 = r \notin Sc^-(v)$, eventually we must encounter an i_0 such that such that $w_{i_0+1}, \ldots, w_n \in Sc^-(v)$ and $w_{i_0} \in Sc(v) \setminus Sc^-(v)$. This implies $w_{i_0} = v$, showing that v is visited on π.

For showing (ii), let $w \in V$ and $v_1, v_2 \in V(\lambda)$, $v_1 \neq v_2$ be such that $w \in Sc(v_1) \cap Sc(v_2)$. Let π be an access path of w. Then it follows by (i) that both v_1 and v_2 are visited on π, and that, depending on whether v_1 or v_2 is visited first on π, either $v_2 \in Sc^-(v_1)$ or $v_1 \in Sc^-(v_2)$. Then due to (nest) it follows that either $Sc(v_2) \subseteq Sc^-(v_1)$ holds or $Sc(v_1) \subseteq Sc^-(v_2)$.

Finally, statement (iii) is an easy consequence of statement (ii). □

Remark 3 The notion of λ-ho-term-graph is an adaptation of the notion of 'higher-order term graph' by Blom [4, Def. 3.2.2] for the purpose of representing finite or infinite λ-terms or cyclic λ-terms, that is, terms in the λ-calculus with letrec. In particular, λ-ho-term-graphs over Σ_1^λ correspond closely to higher-order term graphs over signature Σ^λ. But they differ in the following respects:

Abstractions: Higher-order term graphs in [4] are graph representations of finite or infinite terms in Combinatory Reduction Systems (CRSs). They typically contain abstraction vertices with label □ that represent CRS-abstractions. In contrast, λ-ho-term-graphs have abstraction vertices with label λ that denote λ-abstractions.

Signature: Whereas higher-order term graphs in [4] are based on an arbitrary CRS-signature, λ-ho-term-graphs over Σ_1^λ only contain the application symbol @ and the variable-occurrence symbol 0 in addition to the abstraction symbol λ.

Variable back-links and variable occurrence vertices: In the formalization of higher-order term graphs in [4] there are no explicit vertices that represent variable occurrences. Instead, variable occurrences are represented by back-link edges to abstraction vertices. Actually, in the formalization chosen in [4, Def. 3.2.1], a back-link edge does not directly target the abstraction vertex v it refers to, but ends at a special variant vertex $\bar{v}$ of v. (Every such variant abstraction vertex $\bar{v}$ could be looked upon as a variable vertex that is shared by all edges that represent occurrences of the variable bound by the abstraction vertex v.)

In λ-ho-term-graphs over Σ_1^λ a variable occurrence is represented by a variable-occurrence vertex that as outgoing edge has a back-link to the abstraction vertex that binds the occurrence.

conditions on the scope function: While the conditions (root), (self), (nest), and (closed) on the scope function in higher-order term graphs in [4, Def. 3.2.2] correspond directly to the respective conditions in Def. 1, the difference between the condition (scope) there and $(\text{scope})_1$ in Def. 1 reflects the difference described in the previous item.

free variables: Whereas the higher-order term graphs in [4] cater for the presence of free variables, free variables have been excluded from the basic format of λ-ho-term-graphs.

Definition 4 (homomorphism, bisimulation) Let $i \in \{0,1\}$. Let $\mathcal{G}_1$ and $\mathcal{G}_2$ be λ-ho-term-graphs over Σ_i^λ with $\mathcal{G}_k = \langle V_k, lab_k, args_k, r_k, Sc_k \rangle$ for $k \in \{1,2\}$.

A *homomorphism*, also called a *functional bisimulation*, from $\mathcal{G}_1$ to $\mathcal{G}_2$ is a morphism from the structure $\langle V_1, lab_1, args_1, r_1, Sc_1 \rangle$ to the structure $\langle V_2, lab_2, args_2, r_2, Sc_2 \rangle$, that is, a function $h : V_1 \to V_2$ such that, for all $v \in V_1$ the conditions (labels), (arguments), and (roots) in in (1) are satisfied, and additionally, for all $v \in V_1(\lambda)$:

$$\bar{\bar{h}}(Sc_1(v)) = Sc_2(h(v)) \qquad \text{(scope functions)} \tag{2}$$

where $\bar{\bar{h}}$ is the homomorphic extension of h to sets over V_1, that is, to the function $\bar{h} : \wp(V_1) \to \wp(V_2)$, $A \mapsto \{h(a) \mid a \in A\}$. If there exists a homomorphism (a functional bisimulation) h from $\mathcal{G}_1$ to $\mathcal{G}_2$, then we write $\mathcal{G}_1 \twoheadrightarrow_h \mathcal{G}_2$ or $\mathcal{G}_2 \twoheadleftarrow_h \mathcal{G}_1$, or, dropping h as subscript, $\mathcal{G}_1 \twoheadrightarrow \mathcal{G}_2$ or $\mathcal{G}_2 \twoheadleftarrow \mathcal{G}_1$.

A *bisimulation* between $\mathcal{G}_1$ and $\mathcal{G}_2$ is a term graph $\mathcal{G} = \langle R, lab, args, r, Sc \rangle$ over Σ with $R \subseteq V_1 \times V_2$ and $r = \langle r_1, r_2 \rangle$ such that $\mathcal{G}_1 \twoheadleftarrow_{\pi_1} \mathcal{G} \twoheadrightarrow_{\pi_2} \mathcal{G}_2$ where π_1 and π_2 are projection functions, defined, for $i \in \{1,2\}$, by $\pi_i : V_1 \times V_2 \to V_i$, $\langle v_1, v_2 \rangle \mapsto v_i$. If there exists a bisimulation R between $\mathcal{G}_1$ and $\mathcal{G}_2$, then we write $\mathcal{G}_1 \leftrightarrows_R \mathcal{G}_2$, or just $\mathcal{G}_1 \leftrightarrows \mathcal{G}_2$.

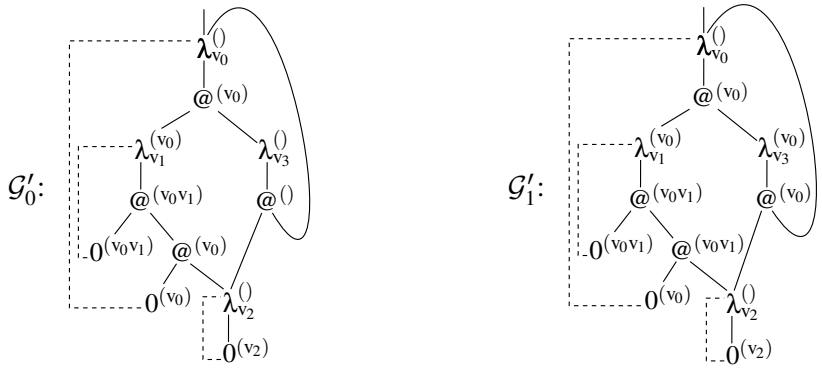


Figure 2: The λ-ap-ho-term-graphs corresponding to the λ-ho-term-graphs in Fig 1. The subscripts of abstraction vertices indicate their names. The super-scripts of vertices indicate their abstraction-prefixes. A precise formulation of this correspondence is given in Example 11.

4 Abstraction-prefix based λ-h.o.-term-graphs

By an 'abstraction-prefix based λ-higher-order-term-graph' we will mean a term-graph over Σ_i^λ for $i \in \{0,1\}$ that is endowed with a correct abstraction prefix function that maps abstraction vertices v to words of vertices that represent the sequence of abstractions that have v in their scope. The conceptual difference between the abstraction-prefix function and the scope function is that the former makes the most essential scoping information locally available. It explicitly states all 'extended scopes' (induced by the transitive closure of the in-scope relation, see [6]) in which a node resides in the order of their nesting. This approach leads to simpler correctness conditions.

Definition 5 (correct abstraction-prefix function for Σ_i^λ-term-graphs) Let $G = \langle V, lab, args, r\rangle$ be, for an $i \in \{0,1\}$, a Σ_i^λ-term-graph.

A function $P : V \to V^*$ from vertices of G to words of vertices is called an *abstraction-prefix function* for G. Such a function is called *correct* if for all $w, w_0, w_1 \in V$ and $k \in \{0,1\}$:

$$\begin{aligned}
&\Rightarrow\ P(r) = \varepsilon && \text{(root)}\\
w \in V(\lambda)\ \wedge\ w \rightarrowtail_0 w_0\ &\Rightarrow\ P(w_0) \leq P(w)w && (\lambda)\\
w \in V(@)\ \wedge\ w \rightarrowtail_k w_k\ &\Rightarrow\ P(w_k) \leq P(w) && (@)\\
w \in V(0)\ &\Rightarrow\ P(w) \neq \varepsilon && (0)_0\\
w \in V(0)\ \wedge\ w \rightarrowtail_0 w_0\ &\Rightarrow\ w_0 \in V(\lambda)\ \wedge\ P(w_0)w_0 = P(w) && (0)_1
\end{aligned}$$

Note that analogously as in Def. 1, if $i = 0$, then $(0)_1$ is trivially true and hence superfluous, and if $i = 1$, then $(0)_0$ is redundant, because it follows from $(0)_1$ in this case.

We say that G *admits* a correct abstraction-prefix function if such a function exists for G.

Definition 6 (λ-ap-ho-term-graph) Let $i \in \{0,1\}$. A *λ-ap-ho-term-graph* (short for *abstraction-prefix based λ-higher-order-term-graph*) over signature Σ_i^λ is a five-tuple $\mathcal{G} = \langle V, lab, args, r, P\rangle$ where $G_\mathcal{G} = \langle V, lab, args, r\rangle$ is a Σ_i^λ-term-graph, called the term graph *underlying* $\mathcal{G}$, and P is a correct abstraction-prefix function for $G_\mathcal{G}$. The classes of λ-ap-ho-term-graphs over Σ_i^λ will be denoted by $\mathcal{H}_i^{(\lambda)}$.

See Fig. 2 for two examples, which correspond, as we will see, to the λ-ho-term-graphs in Fig. 1.

The following lemma states some basic properties of the scope function in λ-ap-ho-term-graphs.

Lemma 7 *Let $i \in \{0,1\}$ and let $\mathcal{G} = \langle V, lab, args, r, P\rangle$ be a λ-ap-ho-term-graph over Σ_i^λ. Then the following statements hold:*

(i) Suppose that, for some $v, w \in V$, v occurs in $P(w)$. Then $v \in V(\lambda)$, occurs in $P(w)$ only once, and every access path of w passes through v, but does not end there, and thus $w \neq v$. Furthermore it holds: $P(v)\,v \leq P(w)$. In particular, if $P(w) = p\,v$, then $P(v) = p$.

(ii) Vertices in abstraction prefixes are abstraction vertices, and hence P is of the form $P : V \to (V(\lambda))^*$.

(iii) For all $v \in V(\lambda)$ it holds: $v \notin P(v)$.

(iv) While access paths might end in vertices in $V(0)$, they only pass through vertices in $V(\lambda) \cup V(@)$.

Proof. Let $i \in \{0,1\}$ and let $\mathcal{G} = \langle V, lab, args, r, P\rangle$ be a λ-ap-ho-term-graph over Σ_i^λ.

For showing (i), let $v, w \in V$ be such that v occurs in $P(w)$. Suppose further that π is an access path of w. Note that when walking through π the abstraction prefix starts out empty (due to (root)), and is expanded only in steps from vertices $v' \in V(\lambda)$ (due to (λ), $(@)$, and $(0)_1$) in which just v' is added to the prefix on the right (due to (λ)). Since v occurs in $P(w)$, it follows that $v \in V(\lambda)$, that v must be visited on π, and that π continues after the visit to v. That π is an access path also entails that v is not visited again on π, hence that $w \neq v$ and that v occurs only once in $P(w)$, and that $P(v)\,v$, the abstraction prefix of the successor vertex of v on π, is a prefix of the abstraction prefix of every vertex that is visited on π after v.

Statements (ii) and (iii) follow directly from statement (i).

For showing (iv), consider an access path $\pi : r = w_0 \rightarrowtail \cdots \rightarrowtail w_n$ that leads to a vertex $w_n \in V(0)$. If $i = 0$, then there is no path that extends π properly beyond w_n. So suppose $i = 1$, and let $w_{n+1} \in V$ be such that $w_n \rightarrowtail_0 w_{n+1}$. Then $(0)_1$ implies that $P(w_n) = P(w_{n+1})\,w_{n+1}$, from which it follows by (i) that w_{n+1} is visited already on π. Hence π does not extend to a longer path that is again an access path. □

Definition 8 (homomorphism, bisimulation) Let $i \in \{0,1\}$. Let $\mathcal{G}_1$ and $\mathcal{G}_2$ be λ-ap-ho-term-graphs over Σ_i^λ with $\mathcal{G}_k = \langle V_k, lab_k, args_k, r_k, P_k\rangle$ for $k \in \{1,2\}$.

A *homomorphism*, also called a *functional bisimulation*, from $\mathcal{G}_1$ to $\mathcal{G}_2$ is a morphism from the structure $\langle V_1, lab_1, args_1, r_1, P_1\rangle$ to the structure $\langle V_2, lab_2, args_2, r_2, P_2\rangle$, that is, a function $h : V_1 \to V_2$ such that, for all $v \in V_1$ the conditions (labels), (arguments), and (roots) in in (1) are satisfied, and additionally, for all $v \in V_1$:

$$\bar{h}(P_1(v)) = P_2(h(v)) \qquad \text{(abstraction-prefix functions)} \tag{3}$$

where $\bar{h}$ is the homomorphic extension of h to words over V_1. In this case we write $\mathcal{G}_1 \twoheadrightarrow_h \mathcal{G}_2$, or $\mathcal{G}_2 \twoheadleftarrow_h \mathcal{G}_1$. And we write $\mathcal{G}_1 \twoheadrightarrow \mathcal{G}_2$, or for that matter $\mathcal{G}_2 \twoheadleftarrow \mathcal{G}_1$, if there is a homomorphism (a functional bisimulation) from $\mathcal{G}_1$ to $\mathcal{G}_2$.

A *bisimulation* between $\mathcal{G}_1$ and $\mathcal{G}_2$ is a term graph $\mathcal{G} = \langle R, lab, args, r, Sc\rangle$ over Σ with $R \subseteq V_1 \times V_2$ and $r = \langle r_1, r_2\rangle$ such that $\mathcal{G}_1 \twoheadleftarrow_{\pi_1} \mathcal{G} \twoheadrightarrow_{\pi_2} \mathcal{G}_2$ where π_1 and π_2 are projection functions, defined, for $i \in \{1,2\}$, by $\pi_i : V_1 \times V_2 \to V_i$, $\langle v_1, v_2\rangle \mapsto v_i$. If there exists a homomorphism (a functional bisimulation) h from $\mathcal{G}_1$ to $\mathcal{G}_2$, then we write $\mathcal{G}_1 \twoheadrightarrow_h \mathcal{G}_2$ or $\mathcal{G}_2 \twoheadleftarrow_h \mathcal{G}_1$, or, dropping h as subscript, $\mathcal{G}_1 \twoheadrightarrow \mathcal{G}_2$ or $\mathcal{G}_2 \twoheadleftarrow \mathcal{G}_1$.

The following proposition defines mappings between λ-ho-term-graphs and λ-ap-ho-term-graphs by which we establish a bijective correspondence between the two classes. For both directions the underlying λ-term-graph remains unchanged. A_i derives an abstraction-prefix function P from a scope function by assigning to each vertex a word of its binders in the correct nesting order. B_i defines its scope function Sc by assigning to each λ-vertex v the set of vertices that have v in their prefix (along with v since a vertex never has itself in its abstraction prefix).

Proposition 9 *For each $i \in \{0,1\}$, the mappings A_i and B_i are well-defined between the class of λ-ho-term-graphs over Σ^λ_i and the class of λ-ap-ho-term-graphs over Σ^λ_i:*

$$\left.\begin{array}{l} A_i : \mathcal{H}^\lambda_i \to \mathcal{H}_i{}^{(\lambda)},\ \mathcal{G} = \langle V, lab, args, r, Sc\rangle \mapsto A_i(\mathcal{G}) := \langle V, lab, args, r, P\rangle \\ \qquad\qquad \text{where } P : V \to V^*,\ w \mapsto v_0 \cdots v_n \text{ if } \mathrm{bds}(w) \setminus \{w\} = \{v_0, \ldots, v_n\} \text{ and} \\ \qquad\qquad\qquad\qquad Sc(v_n) \subsetneqq Sc(v_{n-1}) \ldots \subsetneqq Sc(v_0) \end{array}\right\} \tag{4}$$

$$\left.\begin{array}{l} B_i : \mathcal{H}_i{}^{(\lambda)} \to \mathcal{H}^\lambda_i,\ \mathcal{G} = \langle V, lab, args, r, P\rangle \mapsto A_i(\mathcal{G}) := \langle V, lab, args, r, Sc\rangle \\ \qquad\qquad \text{where } Sc : V(\lambda) \to \wp(V),\ v \mapsto \{w \in V \mid v \text{ occurs in } P(w)\} \cup \{v\} \end{array}\right\} \tag{5}$$

Theorem 10 (correspondence of λ-ho-term-graphs with λ-ap-ho-term-graphs) *For each $i \in \{0,1\}$ it holds that the mappings A_i in* (4) *and B_i in* (5) *are each other's inverse; thus they define a bijective correspondence between the class of λ-ho-term-graphs over Σ^λ_i and the class of λ-ap-ho-term-graphs over Σ^λ_i. Furthermore, they preserve and reflect the sharing orders on $\mathcal{H}^\lambda_i$ and on $\mathcal{H}_i{}^{(\lambda)}$:*

$$\begin{array}{lll} (\forall \mathcal{G}_1, \mathcal{G}_2 \in \mathcal{H}^\lambda_i) & \mathcal{G}_1 \trianglerighteq \mathcal{G}_2 & \iff A_i(\mathcal{G}_1) \trianglerighteq A_i(\mathcal{G}_1) \\ (\forall \mathcal{G}_1, \mathcal{G}_2 \in \mathcal{H}_i{}^{(\lambda)}) & B_i(\mathcal{G}_1) \trianglerighteq B_i(\mathcal{G}_1) & \iff \mathcal{G}_1 \trianglerighteq \mathcal{G}_2 \end{array}$$

Example 11 The λ-ho-term-graphs in Fig. 1 correspond to the λ-ap-ho-term-graphs in Fig. 2 via the mappings A_i and B_i as follows: $A_i(\mathcal{G}_0) = \mathcal{G}'_0$, $A_i(\mathcal{G}_1) = \mathcal{G}'_1$, $B_i(\mathcal{G}_0) = \mathcal{G}'_0$, $A_i(\mathcal{G}_1) = \mathcal{G}'_1$.

For λ-ho-term-graphs over the signature Σ^λ_0 (that is, without variable back-links) essential binding information is lost when looking only at the underlying term graph, to the extent that λ-terms cannot be unambiguously represented anymore. For instance the λ-ho-term-graphs that represent the λ-terms $\lambda xy.xy$ and $\lambda xy.xx$ have the same underlying term graph. The same holds for λ-ap-ho-term-graphs.

This is not the case for λ-ho-term-graphs (λ-ap-ho-term-graphs) over Σ^λ_1, because the abstraction vertex to which a variable-occurrence vertex belongs is uniquely identified by the back-link. This is the reason why the following notion is only defined for the signature Σ^λ_1.

Definition 12 (λ-term-graph over Σ^λ_1) A term graph G over Σ^λ_1 is called a *λ-term-graph* over Σ^λ_1 if G admits a correct abstraction-prefix function. By $\mathcal{T}_1{}^{(\lambda)}$ we denote the class of λ-term-graphs over Σ^λ_1.

In the rest of this section we examine, and then dismiss, a naive approach to implementing functional bisimulation on λ-ho-term-graphs or λ-ap-ho-term-graphs, which is to apply the homomorphism on the underlying term graph, hoping that this application would simply extend to the λ-ho-term-graph (λ-ap-ho-term-graph) without further ado. We demonstrate that this approach fails, concluding that a faithful first-order implementation of functional bisimulation must not be negligent of the scoping information.

Definition 13 (scope- and abstraction-prefix-forgetful mappings) Let $i \in \{0,1\}$. The *scope-forgetful mapping* ScF^λ_i and the *abstraction-prefix-forgetful mapping* $PF_1{}^{(\lambda)}$ map λ-ho-term-graphs in $\mathcal{T}_i{}^{(\lambda)}$, and respectively, λ-ho-term-graphs in $\mathcal{T}_i{}^{(\lambda)}$ to their underlying term graphs:

$$\begin{array}{l} ScF^\lambda_i : \mathcal{H}^\lambda_i \to \mathcal{T}_i,\ \langle V, lab, args, r, Sc\rangle \mapsto \langle V, lab, args, r\rangle \\ PF_i{}^{(\lambda)} : \mathcal{H}_i{}^{(\lambda)} \to \mathcal{T}_i,\ \langle V, lab, args, r, P\rangle \mapsto \langle V, lab, args, r\rangle \end{array}$$

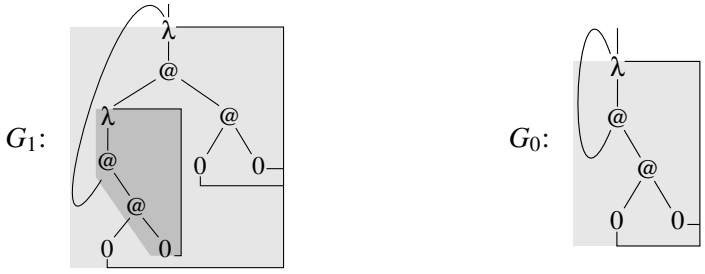
Definition 14 Let $\mathcal{G}$ be a λ-ho-term-graph over Σ^λ_i for $i \in \{0,1\}$ with underlying term graph $ScF^\lambda_i(\mathcal{G})$. And suppose that $ScF^\lambda_i(\mathcal{G}) \trianglerighteq_h G'$ holds for a term graph G' over Σ^λ_i and a functional bisimulation h. We say that *h extends to a functional bisimulation on $\mathcal{G}$* if G' can be endowed with a scope function to obtain a λ-ho-term-graph $\mathcal{G}'$ with $ScF^\lambda_i(\mathcal{G}') = G'$ and such that it holds $\mathcal{G} \trianglerighteq_h \mathcal{G}'$.

We say that a class $\mathcal{K}$ of λ-ho-term-graphs is *closed under functional bisimulations on the underlying term graphs* if for every $\mathcal{G} \in \mathcal{K}$ and for every homomorphism h on the term graph underlying $\mathcal{G}$ that witnesses $ScF_i^\lambda(\mathcal{G}) \rightrightarrows_h G'$ for a term graph G' there exists $\mathcal{G}' \in \mathcal{K}$ with $ScF_i^\lambda(\mathcal{G}') = G'$ such that $\mathcal{G} \rightrightarrows_h \mathcal{G}'$ holds, that is, h is also a homorphism between $\mathcal{G}$ and $\mathcal{G}'$.

These notions are also extended, by analogous stipulations, to λ-ap-ho-term-graphs over Σ_i^λ for $i \in \{0,1\}$ and their underlying term graphs.

Proposition 15 *Neither the class $\mathcal{H}_1^\lambda$ of λ-ho-term-graphs nor the class ${\mathcal{H}_1}^{(\lambda)}$ of λ-ap-ho-term-graphs is closed under functional bisimulations on the underlying term graphs.*

Proof. In view of Thm. 10 it suffices to show the statement for $\mathcal{H}_1^\lambda$. We show that not every functional bisimulation on the term graph underlying a λ-ho-term-graph over Σ_1^λ extends to a functional bisimulation on the higher-order term graphs. Consider the following term graphs G_0 and G_1 over Σ_1^λ (at first, please ignore the scope shading):



There is an obvious homomorphism h that witnesses $G_1 \rightrightarrows_h G_0$. Both of these term graphs extend to λ-ho-term-graphs by suitable scope functions (one possibility per term graph is indicated by the scope shadings above; G_1 actually admits two possibilities). However, h does not extend to any of the λ-ho-term-graphs $\mathcal{G}_1$ and $\mathcal{G}_0$ that extend G_1 and G_0, respectively. □

The next proposition is merely a reformulation of Prop. 15.

Proposition 16 *The scope-forgetful mapping ScF_1^λ on $\mathcal{H}_1^\lambda$ and the abstraction-prefix-forgetful mapping ${PF_1}^{(\lambda)}$ on ${\mathcal{H}_1}^{(\lambda)}$ preserve, but do not reflect, the sharing orders on these classes. In particular:*

$$(\forall \mathcal{G}_1, \mathcal{G}_2 \in {\mathcal{H}_1}^{(\lambda)}) \qquad \mathcal{G}_1 \rightrightarrows \mathcal{G}_2 \implies {PF_1}^{(\lambda)}(\mathcal{G}_1) \rightrightarrows {PF_1}^{(\lambda)}(\mathcal{G}_2)$$
$$(\exists \mathcal{G}_1, \mathcal{G}_2 \in {\mathcal{H}_1}^{(\lambda)}) \qquad \mathcal{G}_1 \not\rightrightarrows \mathcal{G}_2 \ \wedge \ {PF_1}^{(\lambda)}(\mathcal{G}_1) \rightrightarrows {PF_1}^{(\lambda)}(\mathcal{G}_2)$$

As a consequence of this proposition it is not possible to faithfully implement functional bisimulation on λ-ho-term-graphs and λ-ap-ho-term-graphs by only considering the underlying term graphs, and in doing so neglecting[2] the scoping information from the scope function, or respectively, from the abstraction prefix function. In order to yet be able to implement functional bisimulation of λ-ho-term-graphs and λ-ap-ho-term-graphs in a first-order setting, in the next section we introduce a class of first-order term graphs that accounts for scoping by means of scope delimiter vertices.

5 λ-Term-Graphs with Scope Delimiters

For all $i \in \{0,1\}$ and $j \in \{1,2\}$ we define the extensions $\Sigma_{i,j}^\lambda := \Sigma^\lambda \cup \{0, \mathsf{S}\}$ of the signature Σ^λ where $ar(0) = i$ and $ar(\mathsf{S}) = j$, and we denote the class of term graphs over signature $\Sigma_{i,j}^\lambda$ by $\mathcal{T}_{i,j}$.

[2] In the case of Σ_1^λ implicit information about possible scopes is being kept, due to the presence of back-links from variable occurrence vertices to abstraction vertices. But this is not enough for reflecting the sharing order under the forgetful mappings.

Let G be a term graph with vertex set V over a signature extending $\Sigma^{\lambda}_{i,j}$ for $i \in \{0,1\}$ and $j \in \{1,2\}$. We denote by $V(\mathsf{S})$ the subset of V consisting of all vertices with label S, which are called the *delimiter vertices* of G. Delimiter vertices signify the end of an 'extended scope' [6]. They are analogous to occurrences of function symbols S in representations of λ-terms in a nameless de-Bruijn index [5] form in which Dedekind numerals based on 0 and the successor function symbol S are used (this form is due to Hendriks and van Oostrom, see also [10], and is related to their end-of-scope symbol Λ [7]).

Analogously as for the classes $\mathcal{H}^{\lambda}_{i}$ and $\mathcal{H}_{i}{}^{(\lambda)}$, the index i will determine whether in correctly formed λ-term-graphs (defined below) variable vertices have back-links to the corresponding abstraction. Here additionally scope-delimiter vertices have such back-links (if $j = 2$) or not (if $j = 1$).

Definition 17 (correct abstraction-prefix function for $\Sigma^{\lambda}_{i,j}$-term-graphs) Let $G = \langle V, lab, args, r\rangle$ be a $\Sigma^{\lambda}_{i,j}$-term-graph for an $i \in \{0,1\}$ and an $j \in \{1,2\}$.

A function $P : V \to V^*$ from vertices of G to words of vertices is called an *abstraction-prefix function* for G. Such a function is called *correct* if for all $w, w_0, w_1 \in V$ and $k \in \{0,1\}$ it holds:

$$
\begin{array}{rcll}
 & \Rightarrow & P(r) = \varepsilon & (\text{root}) \\
w \in V(\lambda) \ \wedge\ w \rightarrowtail_0 w_0 & \Rightarrow & P(w_0) = P(w)w & (\lambda) \\
w \in V(@) \ \wedge\ w \rightarrowtail_k w_k & \Rightarrow & P(w_k) = P(w) & (@) \\
w \in V(0) & \Rightarrow & P(w) \neq \varepsilon & (0)_0 \\
w \in V(0) \ \wedge\ w \rightarrowtail_0 w_0 & \Rightarrow & w_0 \in V(\lambda) \ \wedge\ P(w_0)w_0 = P(w) & (0)_1 \\
w \in V(\mathsf{S}) \ \wedge\ w \rightarrowtail_0 w_0 & \Rightarrow & P(w_0)v = P(w) \text{ for some } v \in V & (\mathsf{S})_1 \\
w \in V(\mathsf{S}) \ \wedge\ w \rightarrowtail_1 w_1 & \Rightarrow & w_1 \in V(\lambda) \ \wedge\ P(w_1)w_1 = P(w) & (\mathsf{S})_2
\end{array}
$$

Note that analogously as in Def. 1 and in Def. 6, if $i = 0$, then $(0)_1$ is trivially true and hence superfluous, and if $i = 1$, then $(0)_0$ is redundant, because it follows from $(0)_1$ in this case. Additionally, if $j = 1$, then $(\mathsf{S})_2$ is trivially true and therefore superfluous.

Definition 18 (λ-term-graph over $\Sigma^{\lambda}_{i,j}$) Let $i \in \{0,1\}$ and $j \in \{1,2\}$. A *λ-term-graph (with scope-delimiters)* over $\Sigma^{\lambda}_{i,j}$ is a $\Sigma^{\lambda}_{i,j}$-term-graph that admits a correct abstraction-prefix function. The class of λ-term-graphs over $\Sigma^{\lambda}_{i,j}$ is denoted by $\mathcal{T}^{(\lambda)}_{i,j}$.

See Fig. 3 for examples, that, as we will see, correspond to the ho-term-graphs in Fig. 1 and in Fig. 2.

Lemma 19 *Let $i \in \{0,1\}$ and $j \in \{1,2\}$, and let $G = \langle V, lab, args, r\rangle$ be a λ-term-graph over $\Sigma^{\lambda}_{i,j}$. Then the statements* (i)–(iii) *in Lemma 7 hold, and additionally:*

(iv) Access paths may end in vertices in $V(0)$, but only pass through vertices in $V(\lambda) \cup V(@) \cup V(\mathsf{S})$, and depart from vertices in $V(\mathsf{S})$ only via indexed edges ${}^{\mathsf{S}}\!\!\rightarrowtail_0$.

(v) There exists precisely one correct abstraction-prefix function on G.

Proof. That also here statements (i)–(iii) in Lemma 7 hold, and that statement (iv) holds, can be shown analogously as in the proof of the respective items of Lemma 7. For (v) it suffices to observe that if P is a correct abstraction-prefix function for G, then, for all $w \in V$, the value $P(w)$ of P at w can be computed by choosing an arbitrary access path π from r to w and using the conditions (λ), $(@)$, and $(\mathsf{S})_0$ to determine in a stepwise manner the values of P at the vertices that are visited on π. Hereby note that in every transition along an edge on π the length of the abstraction prefix only changes by at most 1. □

Now we define a precise relationship between λ-term-graphs and λ-ap-ho-term-graphs via translation mappings between these classes:

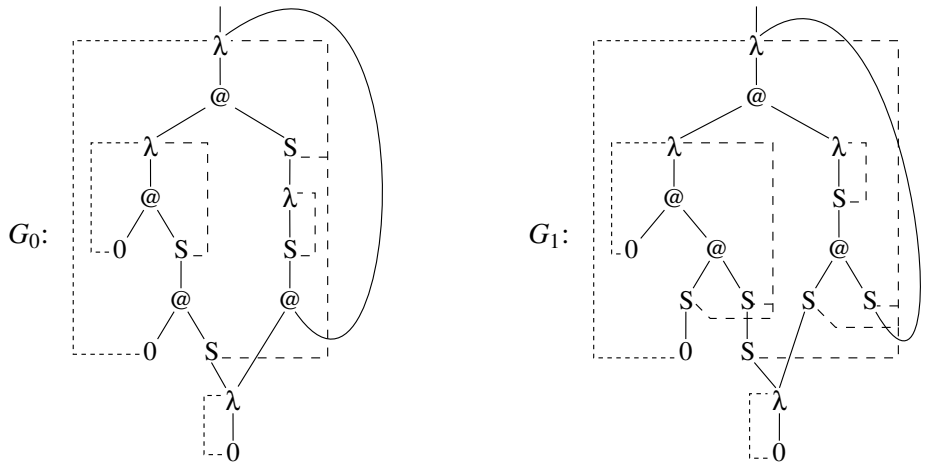


Figure 3: The λ-term-graphs corresponding to the λ-ap-ho-term-graphs from Fig 2 and the λ-ho-term-graphs from Fig 1. A precise formulation of this correspondence is given in Example 23.

The mapping $G_{i,j}$ (see Prop. 20) produces a λ-term-graph for any given λ-ap-ho-term-graph by adding to the original set of vertices a number of delimiter vertices at the appropriate places. That is, at every position where the abstraction prefix decreases by n elements, n S-vertices are inserted. In the image, the original abstraction prefix is retained as part of the vertices. This can be considered intermittent information used for the purpose of defining the edges of the image.

The mapping $\mathcal{G}_{i,j}$ (see Prop. 21) back to λ-ap-ho-term-graphs is simpler because it only has to erase the S-vertices, and add the correct abstraction prefix that exists for the λ-term-graph to be translated.

Proposition 20 *Let $i \in \{0,1\}$ and $j \in \{1,2\}$. The mapping $G_{i,j}$ defined below is well-defined between the class of λ-term-graphs over $\Sigma^{\lambda}_{i,j}$ and the class of λ-ap-ho-term-graphs over Σ^{λ}_{i}:*

$$G_{i,j} : \mathcal{H}_i^{(\lambda)} \to \mathcal{T}_{i,j}^{(\lambda)},\ \ \mathcal{G} = \langle V, lab, args, r, P\rangle \mapsto G_{i,j}(\mathcal{G}) := \langle V', lab', args', r'\rangle$$

where:

$$V' := \{\langle w, P(w)\rangle \mid w \in V\} \cup \left\{\langle w,k,w',p\rangle \;\middle|\; w,w' \in V,\ w \rightarrowtail_k w',\ \begin{array}{l} V(w) = \lambda \ \wedge\ P(w') < p \le P(w)\, w \\ \vee\ V(w) = @ \ \wedge\ P(w') < p \le P(w) \end{array}\right\}$$

$$r' := \langle r, \varepsilon\rangle \qquad\qquad lab' : V' \to \Sigma^{\lambda}_{i,j},\ \ \begin{array}{l} \langle w, P(w)\rangle \mapsto lab'(\langle w, P(w)\rangle) := lab(w) \\ \langle w,k,w',p\rangle \mapsto lab'(w,k,w',p) := \mathsf{S} \end{array}$$

and $args' : V' \to (V')^*$ is defined such that for the induced indexed successor relation $\rightarrowtail'_{(\cdot)}$ it holds:

$$w \rightarrowtail_k w_k \ \wedge\ \#\mathrm{del}(w,k) = 0 \implies \langle w, P(w)\rangle \rightarrowtail'_k \langle w_k, P(w_k)\rangle$$

$$w \rightarrowtail_0 w_0 \ \wedge\ \#\mathrm{del}(w,0) > 0 \ \wedge\ lab(w) = \lambda \ \wedge\ P(w) = P(w_0)\, v\, p$$
$$\implies \langle w, P(w)\rangle \rightarrowtail'_0 \langle w, 0, w_0, P(w)\, w\rangle \ \wedge\ \langle w, 0, w_0, P(w_0) v\rangle \rightarrowtail'_0 \langle w_0, P(w_0)\rangle$$

$$w \rightarrowtail_k w_k \ \wedge\ \#\mathrm{del}(w,k) > 0 \ \wedge\ lab(w) = @ \ \wedge\ P(w) = P(w_k)\, v\, p$$
$$\implies \langle w, P(w)\rangle \rightarrowtail'_k \langle w, k, w_k, P(w)\rangle \ \wedge\ \langle w, k, w_k, P(w_k) v\rangle \rightarrowtail'_0 \langle w_k, P(w_k)\rangle$$

$$w \rightarrowtail_k w_k \ \wedge\ \#\mathrm{del}(w,k) > 0 \ \wedge\ \langle w,k,w_k,p\,v\rangle, \langle w,k,w_k,p\rangle \in V' \implies \langle w,k,w_k,p\,v\rangle \rightarrowtail'_0 \langle w,k,w_k,p\rangle$$

$$w \rightarrowtail_k w_k \ \wedge\ \#\mathrm{del}(w,k) > 0 \ \wedge\ \langle w,k,w_k,p\,v\rangle \in V' \ \wedge\ j = 2 \implies \langle w,k,w_k,p\,v\rangle \rightarrowtail'_1 \langle w_k, P(w_k)\rangle$$

for all $w, w_0, w_1, v \in V$, $k \in \{0,1\}$, $p \in V^*$, *and where the function* #del *is defined as:*

$$\#\mathrm{del}(w,k) := \begin{cases} |P(w)| - |P(w')| & \text{if } w \in V(@) \wedge w \rightarrowtail_k w' \\ |P(w)| + 1 - |P(w')| & \text{if } w \in V(\lambda) \wedge w \rightarrowtail_k w' \\ 0 & \text{otherwise} \end{cases}$$

Proposition 21 *Let* $i \in \{0,1\}$ *and* $j \in \{1,2\}$. *The mapping* $\mathcal{G}_{i,j}$ *defined below is well-defined between the class of* λ*-term-graphs over* $\Sigma^{\lambda}_{i,j}$ *and the class of* λ*-ap-ho-term-graphs over* Σ^{λ}_{i}*:*

$$\begin{aligned} \mathcal{G}_{i,j} : \mathcal{T}^{(\lambda)}_{i,j} \to \mathcal{H}_i^{(\lambda)},\ G = \langle V, lab, args, r\rangle \mapsto \mathcal{G}_{i,j}(G) := \langle V', lab', args', r', P'\rangle \\ \text{where } V' := V(\lambda) \cup V(@) \cup V(0),\ lab' := lab|_{V'},\ r' := r, \\ args' : V' \to (V')^* \text{ so that for the induced indexed succ. relation } \rightarrowtail'_{(\cdot)}: \\ v_0 \rightarrowtail'_k v_1 :\Leftrightarrow v_0 \rightarrowtail_k \cdot \left(\overset{\mathsf{S}}{\rightarrowtail}_0\right)^* v_1 \quad \text{(for all } v_0, v_1 \in V',\ k \in \{0,1\}) \\ P' := P|_{V'} \text{ for the correct abstraction-prefix function } P \text{ for } G. \end{aligned}$$

Theorem 22 (correspondence between λ-ap-ho-term-graphs with λ-term-graphs) *Let* $i \in \{0,1\}$ *and* $j \in \{1,2\}$. *The mappings* $\mathcal{G}_{i,j}$ *from Prop. 21 and* $G_{i,j}$ *from Prop. 20 define a correspondence between the classes of* λ*-term-graphs over* $\Sigma^{\lambda}_{i,j}$ *and of* λ*-ap-ho-term-graphs over* Σ^{λ}_{i} *with the following properties:*

(i) $\mathcal{G}_{i,j} \circ G_{i,j} = \mathrm{id}_{\mathcal{H}_i^{(\lambda)}}$.

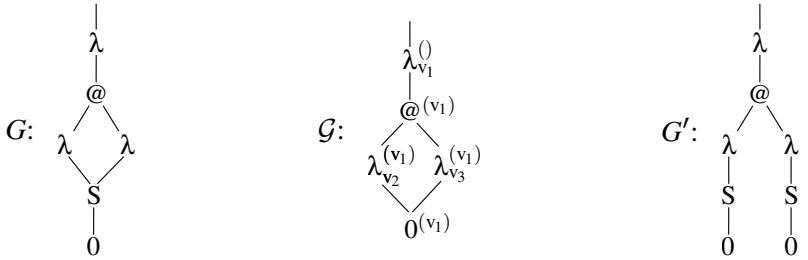
(ii) For all $G \in \mathcal{T}^{(\lambda)}_{i,j}$: $(G_{i,j} \circ \mathcal{G}_{i,j})(G) \rightrightarrows^{\mathsf{S}} G$.

(iii) $\mathcal{G}_{i,j}$ *and* $G_{i,j}$ *preserve and reflect the sharing orders on* $\mathcal{H}_i^{(\lambda)}$ *and on* $\mathcal{T}^{(\lambda)}_{i,j}$:

$$\begin{aligned} &(\forall \mathcal{G}_1, \mathcal{G}_2 \in \mathcal{H}_i^{(\lambda)}) && \mathcal{G}_1 \rightrightarrows \mathcal{G}_2 \iff G_{i,j}(\mathcal{G}_1) \rightrightarrows G_{i,j}(\mathcal{G}_2) \\ &(\forall G_1, G_2 \in \mathcal{T}^{(\lambda)}_{i,j}) && \mathcal{G}_{i,j}(G_1) \rightrightarrows \mathcal{G}_{i,j}(G_2) \iff G_1 \rightrightarrows G_2 \end{aligned}$$

Example 23 The λ-ap-ho-term-graphs in Fig. 2 correspond to the λ-ap-ho-term-graphs in Fig. 3 via the mappings $G_{i,j}$ and $\mathcal{G}_{i,j}$ as follows: $G_{i,j}(\mathcal{G}_0) = G'_0$, $G_{i,j}(\mathcal{G}_1) = G'_1$, $\mathcal{G}_{i,j}(G_0) = \mathcal{G}'_0$, $\mathcal{G}_{i,j}(G_1) = \mathcal{G}'_1$.

Remark 24 The correspondence in Theorem 22 is not a bijection since $\mathcal{G}_{i,j}$ is not injective. This can be seen for the following graphs (here with $i = 0$ and $j = 1$) where we have $\mathcal{G}_{0,1}(G) = \mathcal{G} = \mathcal{G}_{0,1}(G')$:



Obviously λ-ap-ho-term-graphs are not capable of reproducing the different degrees of S-sharing.

6 Not closed under bisimulation and functional bisimulation

In this section we collect all negative results concerning closedness under bisimulation and functional bisimulation for the classes of λ-term-graphs as introduced in the previous section.

Proposition 25 *None of the classes $\mathcal{T}_1^{(\lambda)}$ and $\mathcal{T}_{i,j}^{(\lambda)}$, for $i \in \{0,1\}$ and $j \in \{1,2\}$, of λ-term-graphs are closed under bisimulation.*

This proposition is an immediate consequence of the next one, which can be viewed as a refinement, because it formulates non-closedness of classes of λ-term-graphs under specializations of bisimulation, namely for functional bisimulation (under which some classes are not closed), and for converse functional bisimulation (under which none of the classes considered here is closed).

Proposition 26 *The following statements hold:*

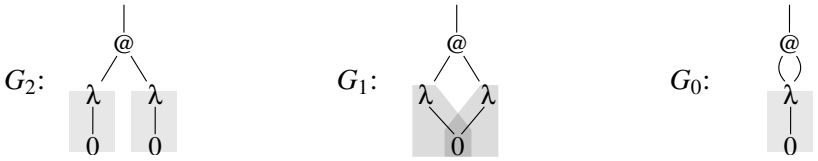
(i) None of the classes $\mathcal{T}_{\mathbf{0},j}^{(\lambda)}$ for $j \in \{1,2\}$ of λ-term-graphs are closed under functional bisimulation $\twoheadrightarrow$, or under converse functional bisimulation $\twoheadleftarrow$.

(ii) None of the classes $\mathcal{T}_{\mathbf{1}}^{(\lambda)}$ and $\mathcal{T}_{\mathbf{1},j}^{(\lambda)}$ for $j \in \{1,2\}$ of λ-term-graphs are closed under converse functional bisimulation.

(iii) The class $\mathcal{T}_{\mathbf{1},\mathbf{1}}^{(\lambda)}$ of λ-term-graphs is not closed under functional bisimulation.

(iv) The class $\mathcal{T}_{\mathbf{1},\mathbf{2}}^{(\lambda)}$ of λ-term-graphs is not closed under functional bisimulation.

Proof. For showing (i), let Δ be one of the signatures $\Sigma_{0,j}^{\lambda}$. Consider the following term graphs over Δ:



Note that G_2 represents the syntax tree of the nameless de-Bruijn-index notation $(\lambda 0)\,(\lambda 0)$ for the λ-term $(\lambda x.x)\,(\lambda x.x)$. Then it holds: $G_2 \twoheadrightarrow G_1 \twoheadrightarrow G_0$. But while G_2 and G_0 admit correct abstraction-prefix functions over Δ (nestedness of the implicitly defined scopes, here shaded), and consequently are λ-term-graphs over Δ, this is not the case for G_1 (overlapping scopes). Hence the class of λ-term-graphs over Δ is closed neither under functional bisimulation nor under converse functional bisimulation.

For showing (ii), let Δ be one of the signatures Σ_1^{λ} and $\Sigma_{1,j}^{\lambda}$. Consider the term graphs over Δ:

Then it holds: $G_1' \twoheadrightarrow G_0'$. But while G_0' admits a correct abstraction-prefix function, and therefore is a λ-term-graph, over Δ, this is not the case for G_1' (due to overlapping scopes). Hence the class of λ-term-graphs over Δ is not closed under converse functional bisimulation.

For showing (iii), consider the following term graphs over $\Sigma_{1,1}^{\lambda}$:

Then it holds that $G_1'' \rightrightarrows G_0''$. However, while G_1'' admits a correct abstraction-prefix function, and hence is a λ-term-graph over $\Sigma^{\lambda}_{1,1}$, this is not the case for G_0'' (due to overlapping scopes). Therefore the class of λ-term-graphs over $\Sigma^{\lambda}_{1,1}$ is not closed under functional bisimulation.

For showing (iv), consider the following term graphs over $\Sigma^{\lambda}_{1,2}$:

Then it holds that $G_1''' \rightrightarrows G_0'''$. However, while G_1''' admits a correct abstraction-prefix function, and hence is a λ-term-graph over $\Sigma^{\lambda}_{1,2}$, this is not the case for G_0''' (overlapping scopes). Therefore the class of λ-term-graphs over $\Sigma^{\lambda}_{1,2}$ is not closed under functional bisimulation. The scopes defined implicitly by these graphs are larger than necessary: they do not exhibit 'eager scope closure', see Section 7. □

As an easy consequence of Prop. 25, and of Prop. 26, (i) and (ii), together with the examples used in the proof, we obtain the following two propositions.

Proposition 27 *Let $i \in \{0,1\}$. None of the classes $\mathcal{H}^{\lambda}_i$ of λ-ho-term-graphs, or $\mathcal{H}_i{}^{(\lambda)}$ of λ-ap-ho-term-graphs are closed under bisimulations on the underlying term graphs.*

Proposition 28 *The following statements hold:*

(i) Neither the class $\mathcal{H}^{\lambda}_0$ nor the class $\mathcal{H}_0{}^{(\lambda)}$ is closed under functional bisimulations, or under converse functional bisimulations, on the underlying term graphs.

(ii) Neither the class $\mathcal{H}^{\lambda}_1$ of λ-ho-term-graphs nor the class $\mathcal{H}_1{}^{(\lambda)}$ of λ-ap-ho-term-graphs is closed under converse functional bisimulations on underlying term graphs.

Note that Prop. 28, (i) is a strengthening of the statement of Prop. 15 earlier.

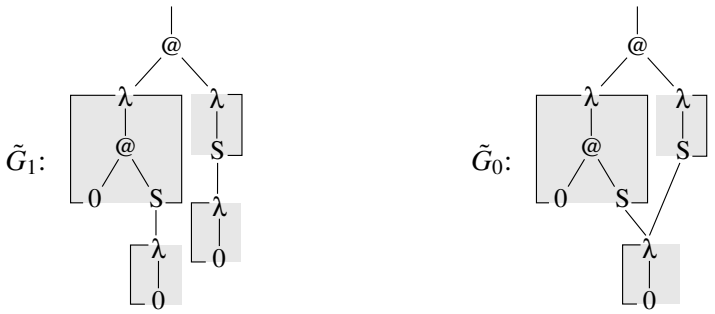
7 Closed under functional bisimulation

The negative results gathered in the last section might seem to show our enterprise in a quite poor state: For the classes of λ-term-graphs we introduced, Prop. 26 only leaves open the possibility that the class $\mathcal{T}_1{}^{(\lambda)}$ is closed under functional bisimulation. Actually, $\mathcal{T}_1{}^{(\lambda)}$ is closed (we do not prove this here), but that does not help us any further, because the correspondences in Thm. 22 do not apply to this class, and worse still, Prop. 16 rules out simple correspondences for $\mathcal{T}_1{}^{(\lambda)}$. So in this case we are left without the satisfying correspondences to λ-ho-term-graphs and λ-ap-ho-term-graphs that yet exist for the other classes of λ-term-graphs, but which in their turn are not closed under functional bisimulation.

But in this section we establish that the class $\mathcal{T}_{1,2}{}^{(\lambda)}$ is very useful after all: its restriction to term graphs with eager application of scope closure is in fact closed under functional bisimulation.

The reason for the non-closedness of $\mathcal{T}_{1,2}{}^{(\lambda)}$ under functional bisimulation consists in the fact that λ-term-graphs over $\Sigma^{\lambda}_{1,2}$ do not necessarily exhibit 'eager scope closure': for example in the term graph G_1''' from the proof of Prop. 26, (iv), the scopes of the two topmost abstractions are not closed on the paths to variable occurrences belonging to the bottommost abstractions. For the following variation $\tilde{G}_1$

of G_1 with eager scope closure the problem disappears:



Its bisimulation collapse $\tilde{G}_0$ has again a correct abstraction-prefix function and hence is a λ-term-graph.

Definition 29 (eager-scope, and fully back-linked, λ-term-graphs) Let $G = \langle V, lab, args, r\rangle$ be a λ-term-graph over $\Sigma^{\lambda}_{1,j}$ for $j \in \{1,2\}$ with abstraction-prefix function $P : V \to V^*$. We call G an *eager-scope λ-term-graph (over $\Sigma^{\lambda}_{1,j}$)* if it holds:

$$\begin{aligned}\forall w, v \in V.\ \forall p \in V^*.\ P(w) = pv \ \Rightarrow\ \exists n \in \mathbb{N}\, \exists w_1, \ldots, w_{n-1} \in V.\ & \\ w \rightarrowtail w_1 \rightarrowtail \ldots \rightarrowtail w_{n-1} \rightarrowtail_0 v & \\ \wedge\ w_{n-1} \in V(0) \ \wedge\ \forall 1 \le i \le n-1.\ P(w) \le P(w_i)\ , &\end{aligned}$$

that is, if for every vertex w in G with a non-empty abstraction-prefix $P(w)$ that ends with v there exists a path from w to v in G via vertices with abstraction-prefixes that extend $P(w)$ and finally a variable-occurrence vertex before reaching v. By ${}^{\mathrm{eag}}\mathcal{T}^{(\lambda)}_{i,j}$ we denote the subclass of $\mathcal{T}^{(\lambda)}_{i,j}$ consisting of all eager-scope-λ-term-graphs. And we say that G is *fully back-linked* if it holds:

$$\forall w, v \in V_1.\ \forall p \in V_1^*.\ P(w) = pv \ \Rightarrow\ w \rightarrowtail^* v\ , \tag{6}$$

that is, if for all vertices w of G_1, the last vertex v in the abstraction-prefix of w is reachable from v. Note that eager-scope implies fully back-linkedness for λ-term-graphs.

Lemma 30 *Let G be a fully back-linked λ-term-graph in $\mathcal{T}^{(\lambda)}_{1,2}$ with vertex set V, and let P be its abstraction-prefix function. Let G' be a term graph over $\Sigma^{\lambda}_{1,2}$ (thus in $\mathcal{T}_{1,2}$) such that $G \rightrightarrows_h G'$. Then it holds:*

$$\forall v_1, v_2 \in V.\ h(v_1) = h(v_2) \ \Rightarrow\ \bar{h}(P(v_1)) = \bar{h}(P(v_2)) \tag{7}$$

where $\bar{h}$ is the homomorphic extension of h to words over V.

Proof (Idea). Let G_1, G_2 be as assumed in the lemma, and let h be a homomorphism that witnesses $G_1 \rightrightarrows_h G_2$. We will use the following distance parameter for vertices of G_1: Let, for all $w \in V_1$, $d_{\lambda,P}(w)$ be either 0 if $P(w)$ is empty, or otherwise the minimum length of a path in G_1 from w to the last vertex in the abstraction-prefix $P(w)$. Thus due to (6), $d_{\lambda,P}(w) \in \mathbb{N}$ for all vertices w of G_1. Now (7) can be proved by induction on $\max\{d_{\lambda,P}(v_1), d_{\lambda,P}(v_2)\}$ with a subinduction on $\max\{|P(v_1)|, |P(v_2)|\}$. □

This lemma is the crucial stepping stone for the proof of the following theorem.

Theorem 31 (preservation of λ-term-graphs over $\Sigma^{\lambda}_{1,2}$ under homomorphism) *Let G and G' be term graphs over $\Sigma^{\lambda}_{1,2}$ such that G is a λ-term-graph in $\mathcal{T}^{(\lambda)}_{1,2}$, and $G \rightrightarrows_h G'$ holds for a homomorphism h.*

If G is fully back-linked, then also G' is a λ-term-graph in $\mathcal{T}^{(\lambda)}_{1,2}$, which is fully back-linked. If, in addition, G is an eager-scope λ-term-graph, then so is G'.

Corollary 32 *The subclass ${}^{\mathrm{eag}}\mathcal{T}_{1,2}^{(\lambda)}$ of the class $\mathcal{T}_{1,2}$ that consists of all eager-scope λ-term-graphs in $\mathcal{T}_{1,2}^{(\lambda)}$ is closed under functional bisimulation.*

Since the counterexample in the proof of Prop. 26, (iii) used eager-scope λ-term-graphs, it rules out a statement analogous to Cor. 32 for the class $\mathcal{T}_{1,1}^{(\lambda)}$. Such a statement for $\mathcal{T}_{0,1}^{(\lambda)}$ and $\mathcal{T}_{0,2}^{(\lambda)}$ is ruled out similarly, with respect to an appropriate definition of 'eager-scope' for λ-term-graphs over $\Sigma_{0,1}^{\lambda}$ and $\Sigma_{0,2}^{\lambda}$.

Corollary 33 *Let h be a functional bisimulation from an eager-scope λ-term-graph G over $\Sigma_{1,2}^{\lambda}$ to a term graph G' over $\Sigma_{1,2}^{\lambda}$ (h witnesses $G \rightrightarrows_h G'$). Then G' is an eager-scope λ-term-graph as well, and h extends to a functional bisimulation from $\mathcal{G}_{1,2}(G)$ to $\mathcal{G}_{1,2}(G')$ (thus h also witnesses $\mathcal{G}_{1,2}(G) \rightrightarrows_h \mathcal{G}_{1,2}(G')$).*

8 Conclusion

We first defined higher-order term graph representations for cyclic λ-terms:

- λ-ho-term-graphs in $\mathcal{H}_i^{\lambda}$, an adaptation of Blom's 'higher-order term graphs' [4], which possess a scope function that maps every abstraction vertex v to the set of vertices that are in the scope of v.
- λ-ap-ho-term-graphs in $\mathcal{H}_i{}^{(\lambda)}$, which instead of a scope function carry an abstraction-prefix function that assigns to every vertex w information about the scoping structure relevant for w. Abstraction prefixes are closely related to the notion of 'generated subterms' for λ-terms [6]. The correctness conditions here are simpler and more intuitive than for λ-ho-term-graphs.

These classes are defined for $i \in \{0,1\}$, according to whether variable occurrences have back-links to abstractions (for $i = 1$) or not (for $i = 0$). Our main statements about these classes are:

- a bijective correspondence between $\mathcal{H}_i^{\lambda}$ and $\mathcal{H}_i{}^{(\lambda)}$ via mappings A_i and B_i that preserve and reflect the sharing order (Thm. 10);
- the naive approach to implementing homomorphisms on theses classes (ignoring all scoping information and using only the underlying first-order term graphs) fails (Prop. 16).

The latter was the motivation to consider first-order term graph implementations with scope delimiters:

- λ-term-graphs in $\mathcal{T}_{i,j}^{(\lambda)}$ (with $i \in \{0,1\}$ and $j = 2$ or $j = 1$ for scope delimiter vertices with or without back-links, respectively), which are first-order term graphs without a higher-order concept, but for which correctness conditions are formulated via the existence of an abstraction-prefix function.

The most important results linking these classes with λ-ap-ho-term-graphs are:

- an 'almost bijective' correspondence between the classes $\mathcal{H}_i{}^{(\lambda)}$ and $\mathcal{T}_{i,j}^{(\lambda)}$ via mappings $G_{i,j}$ and $\mathcal{G}_{i,j}$ that preserve and reflect the sharing order (Thm. 22);
- the subclass ${}^{\mathrm{eag}}\mathcal{T}_{1,2}^{(\lambda)}$ of eager-scope λ-term-graphs in $\mathcal{T}_{1,2}^{(\lambda)}$ is closed under homomorphism (Cor. 32).

The correspondences together with the closedness result allow us to derive methods to handle homomorphisms between eager higher-order term graphs in $\mathcal{H}_1^{\lambda}$ and $\mathcal{H}_1{}^{(\lambda)}$ in a straightforward manner by implementing them via homomorphisms between first-order term graphs in $\mathcal{T}_{1,2}^{(\lambda)}$.

$$\begin{array}{ccc} {}^{\mathrm{eag}}\mathcal{H}_1^{\lambda} & & \\ {\scriptstyle A_1}\downarrow\ \uparrow{\scriptstyle B_1} & & \\ {}^{\mathrm{eag}}\mathcal{H}_1{}^{(\lambda)} & & \\ {\scriptstyle G_{1,2}}\downarrow\ \uparrow{\scriptstyle \mathcal{G}_{1,2}} & & \\ {}^{\mathrm{eag}}\mathcal{T}_{1,2}^{(\lambda)} & & \end{array} \qquad \begin{array}{ccc} \mathcal{G}_0 & \xmapsto[h]{} & \mathcal{G}_0' \\ {\scriptstyle A_1}\downarrow & & \uparrow{\scriptstyle B_1} \\ \mathcal{G}_1 & \xmapsto[h]{} & \mathcal{G}_1' \\ {\scriptstyle G_{1,2}}\downarrow & & \uparrow{\scriptstyle \mathcal{G}_{1,2}} \\ G & \xmapsto[h']{} & G' \end{array}$$

For example, the property that a unique maximally shared form exists for λ-term-graphs in $\mathcal{T}_{1,2}^{(\lambda)}$ (which

can be computed as the bisimulation collapse that is guaranteed to exist for first-order term graphs) can now be transferred to eager-scope λ-ap-ho-term-graphs and λ-ho-term-graphs via the correspondence mappings (see the diagram above). For this to hold it is crucial that ${}^{\text{eag}}\mathcal{T}_{1,2}^{(\lambda)}$ is closed under homomorphism, and that the correspondence mappings preserve and reflect the sharing order. The maximally shared form $\max_{\text{eag}}\mathcal{H}_1^{\lambda}(\mathcal{G})$ of an eager λ-ho-term-graph $\mathcal{G}$ can furthermore be computed as:

$$\max_{\text{eag}}\mathcal{H}_1^{\lambda}(\mathcal{G}) = (B_1 \circ \mathcal{G}_{1,2} \circ \max_{\text{eag}}\mathcal{T}_{1,2}^{(\lambda)} \circ G_{1,2} \circ A_1)(\mathcal{G}).$$

where $\max_{\text{eag}}\mathcal{T}_{1,2}^{(\lambda)}$ maps every λ-term-graph in $\mathcal{T}_{1,2}^{(\lambda)}$ to its bisimulation collapse. For obtaining $\max_{\text{eag}}\mathcal{T}_{1,2}^{(\lambda)}$ fast algorithms for computing the bisimulation collapse of first-order term graphs can be utilized.

While we have explained this result here only for term graphs with eager scope-closure, the approach can be generalized to non-eager-scope term graphs. To this end scope delimiters have to be placed also underneath variable vertices. Then variable occurrences do not implicitly close all open extended scopes, but every extended scope that is open at some position must be closed explicitly by scope delimiters on all (maximal) paths from that position. The resulting graphs are fully back-linked, and then Thm. 31 guarantees that the arising class of λ-term-graphs is again closed under homomorphism.

For our original intent of getting a grip on maximal subterm sharing in the λ-calculus with letrec or μ, however, only eager scope-closure is practically relevant, since it facilitates a higher degree of sharing.

Ultimately we expect that these results allow us to develop solid formalizations and methods for subterm sharing in higher order languages with sharing constructs.

Acknowledgement. We thank the reviewers for their comments, and for pointing out inaccuracies.

References

[1] Zena M. Ariola and Stefan Blom. Cyclic Lambda Calculi. In Martin Abadi and Takayasu Ito, editors, *Proceedings of the TACS'97 Lecture Notes in Computer Science*, pages 77–106. Springer Berlin / Heidelberg, 1997.

[2] Zena M. Ariola and Jan Willem Klop. Cyclic lambda graph rewriting. In *Proceedings of the Symposium on Logic in Computer Science (LICS) 1994*, pages 416 –425, July 1994.

[3] Zena M. Ariola and Jan Willem Klop. Equational term graph rewriting. *Fundamenta Informaticae*, 26(3):207–240, 01 1996.

[4] Stefan Blom. *Term Graph Rewriting – Syntax and Semantics*. PhD thesis, Vrije Universiteit Amsterdam, 2001.

[5] N. G. de Bruijn. Lambda calculus notation with nameless dummies, a tool for automatic formula manipulation, with application to the Church-Rosser theorem. *Indagationes Mathematicae*, 34:381–392, 1972.

[6] Clemens Grabmayer and Jan Rochel. Expressibility in the Lambda-Calculus with Letrec. Technical Report arXiv:1208.2383, `http://arxiv.org`, August 2012. `http://arxiv.org/abs/1208.2383`.

[7] Dimitri Hendriks and Vincent van Oostrom. ʎ. In F. Baader, editor, *Proceedings CADE-19*, volume 2741 of *Lecture Notes in Artificial Intelligence*, pages 136–150. Springer–Verlag, 2003.

[8] Simon L. Peyton Jones. *The Implementation of Functional Programming Languages*. Prentice-Hall, Inc., 1987.

[9] Terese. *Term Rewriting Systems*, volume 55 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 2003.

[10] Vincent van Oostrom, Kees-Jan van de Looij, and Marijn Zwitserlood. Lambdascope. Extended Abstract for the Workshop on Algebra and Logic on Programming Systems (ALPS), Kyoto, April 10th 2004, 2004. `http://www.phil.uu.nl/~oostrom/publication/pdf/lambdascope.pdf`.