Bisimulation Slices and Transfer Functions

Clemens Grabmayer

Gran Sasso Science Institute, L'Aquila, Italy
clemens.grabmayer@gssi.it

Abstract

We motivate basic concepts for the transfer of specifications from one system to another that we are developing in the context of the process semantics of regular expressions with respect to bisimilarity. Specifically we introduce: 'bisimulating slices', which relate parts of labeled transition systems (LTSs) without regard for the context; 'grounded bisimulation slices', which relate parts of a single LTS in such a way that these slices can easily be extended into bisimulations; 'transfer functions', which are functional bisimulations, and as such permit to transfer specifying expressions (process specifications, programs) between LTSs; 'local transfer functions', which are functional grounded bisimulation slices; and 'elevations of sets of states above' an LTS, which are partial unfoldings of an LTS.

We give definitions and state basic results that link them. While purpose-built for the transfer of regular-expression specifications of processes between LTSs, these ideas might be useful in other situations as well. We are interested in finding links to similar concepts.

We report on concepts that we currently develop for linking specifications of finite-state processes that are represented by regular expressions. The motivating goal is to prove equal, in an equational proof system, any two specifications of processes that have the same behavior in the specific sense of being bisimilar. Two expedient tools for this purpose are the minimization of labeled transition systems (LTSs) under bisimilarity, and the transfer of specifications via functional bisimulations. In the context of our work we have to deal with LTSs that are not minimized optimally, but in which self-bisimulations can be decomposed into 'slices' that relate states in common strongly connected components. In order to adapt to this situation, we aim to transfer specifications also via slices of bisimulations that behave as functions locally. For this reason we formally define, in Section 1, 'bisimulating slices' and 'grounded bisimulation slices'. Subsequently in Section 2 we link 'local transfer functions' (functional grounded bisimulation slices) to 'transfer functions' (functional bisimulations) via 'elevations' of their co-/domains. Finally we *informally* explain how we intend to use these concepts to show that specifications are provably invariant not only under transfer functions, but also under local transfer functions.

1 Bisimulating slices and grounded bisimulation slices

By a *labeled transition system (LTS)* we here mean a 4-tuple $\mathcal{L} = \langle T, A, \to, \downarrow\rangle$ where T is a set of *states*, A is a set of *actions*, $\to \subseteq T \times A \times T$ is a *transition relation*, and $\downarrow \subseteq T$ is a set of *terminating states* (or states *with immediate termination*).

We define a 'bisimulating slice' between two LTSs $\mathcal{L}_1$ and $\mathcal{L}_2$ as a binary relation B between their state sets for which the forth-, and the back-condition of a bisimulation is only required for transitions within the active domain of B, and within the active codomain of B, respectively. Our use of the term 'slice' is inspired by its use by Baeten, Bergstra, and Klop in [1] for patterns of context-free process graphs that facilitate the construction of regular infinite bisimulations.

Definition 1.1. We consider two LTSs $\mathcal{L}_i = \langle T_i, A, \to_i, \downarrow_i\rangle$ for $i \in \{1, 2\}$. A *bisimulating slice between* $\mathcal{L}_1$ *and* $\mathcal{L}_2$ is a binary relation $B \subseteq T_1 \times T_2$ with active domain $W_1 := \mathrm{dom}_{\mathrm{act}}(B) = \pi_1(B)$ and active codomain $W_2 := \mathrm{cod}_{\mathrm{act}}(B) = \pi_2(B)$, where $\pi_i : T_1 \times T_2 \to T_i, \pi_i(\langle t_1, t_2\rangle) = t_i$, for $i \in \{1, 2\}$, such that $B \neq \emptyset$, and for every $\langle t_1, t_2\rangle \in B$ the following conditions hold:

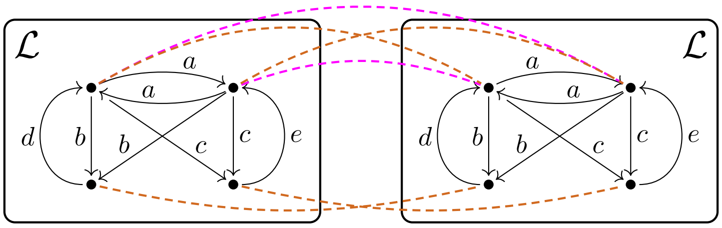

Figure 1: Example of a grounded bisimulation slice B on an LTS $\mathcal{L}$ given by the magenta links. By adding identity pairs, the brown links, a bisimulation $\overline{\overline{B}}$ on $\mathcal{L}$ is obtained, see Prop. 1.3.

$$\begin{aligned}
&(\text{forth})_{\text{s}} \quad \forall a \in A\, \forall t_1' \in T_1 \big(\, t_1 \xrightarrow{a}_1 t_1' \wedge t_1' \in W_1 \quad \Longrightarrow \quad \exists t_2' \in T_2 \big(\, t_2 \xrightarrow{a}_2 t_2' \wedge \langle t_1', t_2' \rangle \in B \,\big)\big)\,,\\
&(\text{back})_{\text{s}} \quad \forall a \in A\, \forall t_2' \in T_2 \big(\, \exists t_1' \in T_1 \big(\, t_1 \xrightarrow{a}_1 t_1' \wedge \langle t_1', t_2' \rangle \in B \,\big) \quad \Longleftarrow \quad t_2 \xrightarrow{a}_2 t_2' \wedge t_2' \in W_2 \,\big)\,,\\
&(\text{termination})_{\text{s}} \quad t_1{\downarrow_1} \quad \Longleftrightarrow \quad t_2{\downarrow_2}\,.
\end{aligned}$$

The condition $(\text{forth})_{\text{s}}$ entails $t_2' \in W_2$, and the condition $(\text{back})_{\text{s}}$ entails $t_1' \in W_1$. Furthermore, by a *bisimulation slice between* $\mathcal{L}_1$ *and* $\mathcal{L}_2$ we mean a bisimulating slice between $\mathcal{L}_1$ and $\mathcal{L}_2$ that is contained in a bisimulation between $\mathcal{L}_1$ and $\mathcal{L}_2$.

Bisimulation slices are related to bisimulations as follows. Every bisimulation slice B between LTSs $\mathcal{L}_1$ and $\mathcal{L}_2$ is a bisimulation between the full sub-LTS of $\mathcal{L}_1$ on $\text{dom}_{\text{act}}(B)$, and the full sub-LTS of $\mathcal{L}_2$ on $\text{cod}_{\text{act}}(B)$. A bisimulation slice B between LTSs $\mathcal{L}_1$ and $\mathcal{L}_2$ is a bisimulation if $\text{dom}_{\text{act}}(B)$ and $\text{cod}_{\text{act}}(B)$ are transition closed in $\mathcal{L}_1$ and in $\mathcal{L}_2$, respectively.

In order to guarantee that a bisimulating slice B on a single LTS $\mathcal{L}$ can be extended into a bisimulation on $\mathcal{L}$, different sufficient conditions are conceivable. The simplest one is to require that any transition from a state t_i with $i \in \{1, 2\}$ of a pair $\langle t_1, t_2 \rangle \in B$ that leaves the active domain of B, or accordingly, the active codomain of B, can be joined by a transition to the same target state with the same action label from the other state t_{3-i}. This idea leads us to the definition of 'grounded bisimulation slices', and the containment result in Prop. 1.3 below.

Definition 1.2. Let $\mathcal{L} = \langle T, A, \rightarrow, \downarrow \rangle$ be an LTS. A bisimulating slice (a bisimulation slice) *on* an LTS $\mathcal{L}$ is a bisimulating slice (and respectively, a bisimulation slice) between $\mathcal{L}$ and $\mathcal{L}$ itself.

We say that a bisimulating slice $B \subseteq T \times T$ on an LTS $\mathcal{L} = \langle T, A, \rightarrow, \downarrow \rangle$ is a *grounded bisimulation slice* if for all $\langle t_1, t_2 \rangle \in B$ the following additional forth/back conditions hold:

$$\begin{aligned}
&(\text{forth})_{\text{g}} \quad \forall a \in A\, \forall t_1' \in T_1 \big(\, t_1 \xrightarrow{a} t_1' \wedge t_1' \notin W_1 \quad \Longrightarrow \quad t_2 \xrightarrow{a} t_1' \wedge t_1' \notin W_2 \,\big)\,,\\
&(\text{back})_{\text{g}} \quad \forall a \in A\, \forall t_2' \in T_1 \big(\, t_1 \xrightarrow{a} t_2' \wedge t_2' \notin W_1 \quad \Longleftarrow \quad t_2 \xrightarrow{a} t_2' \wedge t_2' \notin W_2 \,\big)\,.
\end{aligned}$$

where $W_1 := \text{dom}_{\text{act}}(B)$ is the active domain, and $W_2 := \text{cod}_{\text{act}}(B)$ the active codomain of B.

Proposition 1.3. *For every grounded bisimulation slice* $B \subseteq T \times T$ *on an LTS* $\mathcal{L} = \langle T, A, \rightarrow, \downarrow \rangle$, *the relation* $\overline{\overline{B}} := B \cup {=}$ *is a bisimulation on* $\mathcal{L}$.

In Fig. 1 we provide an example for this proposition. We illustrate a grounded bisimulation slice B on an LTS $\mathcal{L}$, and its extension $\overline{\overline{B}}$ into a bisimulation on $\mathcal{L}$.

2 Transfer functions and local transfer functions

While the grounded bisimulation slice B in Fig. 1 is functional, this does not hold for the extending bisimulation. Hence specification transfer via B is not clear immediately. In this section we will link functional grounded bisimulation slices, which are graphs of a 'local transfer functions' according to the definition below, to 'transfer functions' that define a functional bisimulations.

Definition 2.1. A *transfer (partial) function* between LTSs $\mathcal{L}_1$ and $\mathcal{L}_2$, for $\mathcal{L}_i = \langle T_i, A, \to_i, \downarrow_i\rangle$ where $i \in \{1,2\}$, is a partial function $\phi : T_1 \rightharpoonup T_2$ whose graph $\{\langle t, \phi(t)\rangle \mid t \in T_1\}$ is a bisimulation between $\mathcal{L}_1$ and $\mathcal{L}_2$.

A *local-transfer (partial) function* on an LTS $\mathcal{L} = \langle T, A, \to, \downarrow\rangle$ is a partial function $\phi : T \rightharpoonup T$ whose graph $\{\langle t, \phi(t)\rangle \mid t \in T\}$ is a grounded bisimulation slice on $\mathcal{L}$.

Local-Transfer functions can be linked to transfer functions via the concept of 'elevation LTS' $\mathsf{E}_W(\mathcal{L})$ of a set W of states above a LTS $\mathcal{L} = \langle T, A, \to, \downarrow\rangle$, which is constructed as follows. Its set of states consists of two copies of the set T of states of $\mathcal{L}$, the 'ground floor' $T \times \{\mathbf{0}\}$, and the 'first floor' $T \times \{\mathbf{1}\}$. These two copies of T are linked by copies of the corresponding transitions of $\mathcal{L}$ with the exception that transitions $\langle t, a, t'\rangle$ of $\mathcal{L}$ with $t' \notin W$ do not give rise to a transition $\langle\langle t, \mathbf{1}\rangle, a, \langle t', \mathbf{1}\rangle\rangle$ on the first floor, but are redirected as transitions $\langle\langle t, \mathbf{1}\rangle, a, \langle t', \mathbf{0}\rangle\rangle$ to target the corresponding copy $\langle t', \mathbf{0}\rangle$ of t' on the ground floor. The sub-LTS of $\mathsf{E}_W(\mathcal{L})$ that consists of all transitions between vertices on the ground floor is an exact copy of the original 1-LTS $\mathcal{L}$. Yet within the elevation $\mathsf{E}_W(\mathcal{L})$ of W above $\mathcal{L}$, a number of vertices on the ground floor will have additional incoming transitions from vertices on the first floor.

Definition 2.2. Let $\mathcal{L} = \langle T, A, \to, \downarrow\rangle$ be an LTS, and let $W \subseteq T$ be a subset of the vertices of $\mathcal{L}$. The *elevation of W above $\mathcal{L}$* is the LTS $\mathsf{E}_W(\mathcal{L}) = \langle T_{\mathsf{E}_W}, A, \to_{\mathsf{E}_W}, \downarrow_{\mathsf{E}_W}\rangle$:

$$\begin{aligned} T_{\mathsf{E}_W} := T \times \{\mathbf{0},\mathbf{1}\}\ , \quad \to_{\mathsf{E}_W} := {} & \big\{\langle\langle t_1, \mathbf{1}\rangle, a, \langle t_2, \mathbf{1}\rangle\rangle \bigm| \langle t_1, a, t_2\rangle \in \to \wedge\, a \in A \wedge t_2 \in W\big\} \\ & \cup \big\{\langle\langle t_1, \mathbf{1}\rangle, a, \langle t_2, \mathbf{0}\rangle\rangle \bigm| \langle t_1, a, t_2\rangle \in \to \wedge\, a \in A \wedge t_2 \notin W\big\} \\ & \cup \big\{\langle\langle t_1, \mathbf{0}\rangle, a, \langle t_2, \mathbf{0}\rangle\rangle \bigm| \langle t_1, a, t_2\rangle \in \to\big\}\,. \\ \downarrow_{\mathsf{E}_W} := {} & \big\{\langle t, i\rangle \bigm| t \in T,\, i \in \{\mathbf{0},\mathbf{1}\}\,,\, t\downarrow\big\}\,, \end{aligned}$$

Lemma 2.3. *Let $\mathcal{L} = \langle T, A, \to, \downarrow\rangle$ be an LTS. Let $W \subseteq T$ be a subset of its set of states. Then the projection function $\pi_1 : T \times \{\mathbf{0},\mathbf{1}\} \to T$, $\langle t, \boldsymbol{i}\rangle \mapsto t$ is a transfer function from $\mathsf{E}_W(\mathcal{L})$ to $\mathcal{L}$.*

Proposition 2.4. *Let $\mathcal{L} = \langle T, A, \to, \downarrow\rangle$ be an LTS. Every local-transfer function $\phi : T \rightharpoonup T$ on $\mathcal{L}$ with field $W := \mathrm{field}(\phi) := \mathrm{dom}(\phi) \cup \mathrm{ran}(\phi)$ can be lifted to a transfer function:*

$$\begin{aligned} \widehat{\phi} : (T \times \{\mathbf{0},\mathbf{1}\}) & \rightharpoonup (T \times \{\mathbf{0},\mathbf{1}\}) \\ \langle t, \boldsymbol{i}\rangle & \longmapsto \widehat{\phi}(\langle t, \boldsymbol{i}\rangle) := \begin{cases} \langle \phi(t), \boldsymbol{i}\rangle & \text{if } \boldsymbol{i} = \mathbf{1} \wedge t \in \mathrm{dom}(\phi)\,, \\ \text{undefined} & \text{if } \boldsymbol{i} = \mathbf{1} \wedge t \notin \mathrm{dom}(\phi)\,, \\ \langle t, \boldsymbol{i}\rangle & \text{if } \boldsymbol{i} = \mathbf{0}\,, \end{cases} \end{aligned}$$

on the elevation $\mathsf{E}_W(\mathcal{L})$ of W above $\mathcal{L}$ such that the diagram below commutes on the first floor of $\mathsf{E}_W(\mathcal{L})$, i.e. $(\pi_1 \circ \widehat{\phi})(\langle t, \mathbf{1}\rangle) = (\phi \circ \pi_1)(\langle t, \mathbf{1}\rangle)$ for all $t \in \mathrm{dom}(\phi)$:

$$\begin{tikzcd} \mathcal{L} \arrow[r, harpoon, "\phi"] & \mathcal{L} \\ \mathsf{E}_W(\mathcal{L}) \arrow[u, "\pi_1"] \arrow[r, harpoon, "\widehat{\phi}"] & \mathsf{E}_W(\mathcal{L}) \arrow[u, "\pi_1"'] \end{tikzcd}$$

local transfer function ϕ (grounded functional bisimulation slice)
projections π_1 are transfer functions (functional bisimulations)
transfer function $\widehat{\phi}$ (functional bisimulation)

In Fig. 2 we illustrate the statement of this proposition for obtaining a transfer function from the local transfer function underlying the grounded bisimulation slice B in Fig. 1.

Finally we informally describe the purpose for which we are developing these concepts. We first explain how we used transfer of specifications of regular expressions via transfer functions in [2], and then sketch the extension of this technique to local transfer functions as defined here. By a specification of an LTS $\mathcal{L}$ we mean a function $S : T \to \mathit{Exp}$ from the states T of $\mathcal{L}$ to a set Exp of process expressions in a formalism like Basic Process Algebra BPA.

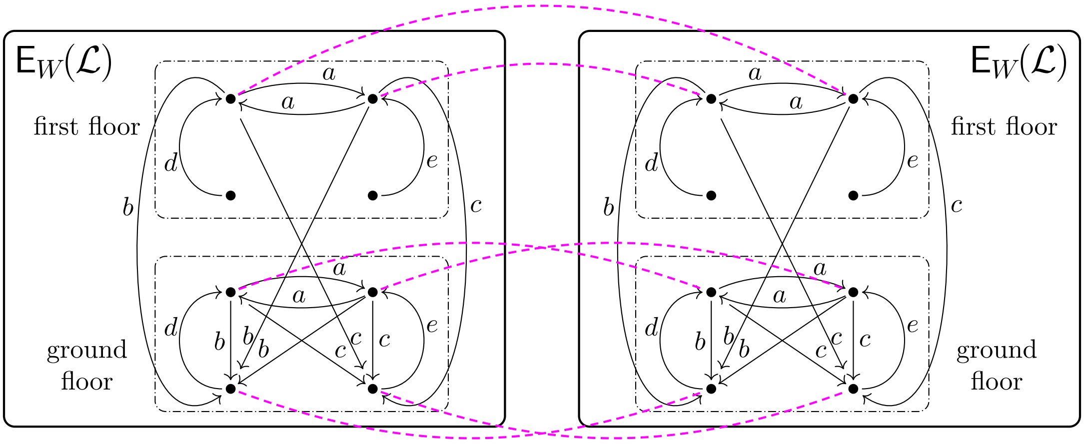

Figure 2: The dashed magenta links form a functional bisimulation that defines a transfer function $\widehat{\phi}$ on the elevation $\mathsf{E}_W(\mathcal{L})$ of $W := \mathrm{dom}_{\mathrm{act}}(B) \cup \mathrm{cod}_{\mathrm{act}}(B)$ over $\mathcal{L}$, for the grounded bisimulation slice B on the LTS $\mathcal{L}$ in Fig. 1. Hereby $\widehat{\phi}$ results via the statement of Prop. 2.4 from the local transfer function ϕ on $\mathcal{L}$ whose graph is the grounded bisimulation slice B in Fig. 1.

In a completeness proof [2] for a proof system P for the process semantics of regular expressions modulo bisimilarity we used the following technique for showing that process specifications are provably equal at bisimilar states. It is based on the following two lemmas: **(P)** LTS specifications can be <u>pulled</u> backwards over functional bisimulations. **(U)** Specifications of LTSs that satisfy a certain structural property $\mathfrak{P}$ are <u>unique</u> modulo provability in P. — On this basis, one can argue as follows. Suppose that $\phi : T_1 \to T_2$ is a functional bisimulation between LTSs $\mathcal{L}_1$ and $\mathcal{L}_2$ where $\mathcal{L}_1$ satisfies $\mathfrak{P}$. Let S_1 and S_2 be specifications of $\mathcal{L}_1$ and $\mathcal{L}_2$, respectively. Then by **(P)**, $S_2 \circ \phi$ is a specification of $\mathcal{L}_1$. By **(U)**, $S_1(t) =_{\mathsf{P}} (S_2 \circ \phi)(t) = S_2(\phi(t))$ holds for all states $t \in T_1$ of $\mathcal{L}_1$, where $=_{\mathsf{P}}$ indicates provable equality in P. This shows that the specifications S_1 of $\mathcal{L}_1$ and S_2 of $\mathcal{L}_2$ are provably equal in P at all bisimilar vertices that are linked by ϕ.

Based on Prop. 2.4, this technique for showing provable invariance of specifications, with respect to a proof system P, can be adapted from transfer functions to local-transfer functions, if additionally the structural property $\mathfrak{P}$ in **(U)** lifts to elevations of LTSs. Suppose that $\phi : T \rightharpoonup T$ is a local-transfer function on an LTS $\mathcal{L}$ that satisfies $\mathfrak{P}$. Let S be a specification of $\mathcal{L}$. Then by **(P)**, $S \circ \pi_1$ is a specification on $\mathsf{E}_W(\mathcal{L})$ for $W := \mathrm{field}(\phi)$, and the same holds for $S \circ \pi_1 \circ \widehat{\phi}$ on $\mathrm{dom}(\widehat{\phi})$. By the additional assumption, also $\mathsf{E}_W(\mathcal{L})$ satisfies $\mathfrak{P}$. Then by **(U)**, $S \circ \pi_1$ and $S \circ \pi_1 \circ \widehat{\phi}$ are provably equal in P, for all states in $\mathrm{dom}(\widehat{\phi}) \supseteq \mathrm{dom}(\phi) \times \{\mathbf{1}\}$. Hence $S(t) = (S \circ \pi_1)(\langle t, \mathbf{1}\rangle) =_{\mathsf{P}} S \circ \pi_1 \circ \widehat{\phi}(\langle t, \mathbf{1}\rangle) = (S \circ \phi \circ \pi_1)(\langle t, \mathbf{1}\rangle) = S(\phi(t))$, for all $t \in \mathrm{dom}(\phi)$, where we use diagram commutativity on the first floor as stated by Prop. 2.4.

References

[1] Jos C. M. Baeten, Jan A. Bergstra, and Jan Willem Klop. Decidability of Bisimulation Equivalence for Processes Generating Context-Free Languages. *J. ACM*, 40(3):653–682, 1993.

[2] Clemens Grabmayer and Wan Fokkink. A Complete Proof System for 1-Free Regular Expressions Modulo Bisimilarity. In *Proceedings of the 35th Annual ACM/IEEE Symposium on Logic in Computer Science*, LICS '20, page 465–478, New York, NY, USA, 2020. ACM.