Submitted for publication 2005 Preliminary Version

Bigraphical Semantics of Higher-Order Mobile Embedded Resources with Local Names

Mikkel Bundgaard [1] Thomas Hildebrandt [2]

Department of Theoretical Computer Science
IT University of Copenhagen
Denmark

Abstract

Bigraphs have been introduced with the aim to provide a topographical meta-model for mobile, distributed agents that can manipulate their own linkages and nested locations, generalising both characteristics of the π-calculus and the Mobile Ambients calculus. We give the first bigraphical presentation of a non-linear, higher-order process calculus with nested locations, non-linear active process mobility, and local names, the calculus of *Higher-Order Mobile Embedded Resources (Homer)*. The presentation is based on Milner's recent presentation of the λ-calculus in local bigraphs. The combination of non-linear active process mobility and local names requires a new definition of parametric reaction rules and a representation of the location of names. We suggest *localised bigraphs* as a generalisation of local bigraphs in which links can be further localised.

Key words: bigraphs, local names, non-linear process mobility

Introduction

The theory of *Bigraphical Reactive Systems* (BRS) [12] has been proposed as a topographical meta-model for mobile, distributed agents that can manipulate their own linkages and nested locations.

A bigraph consists of two structures: the *place graph* and the *link graph*. The *place graph* is a tuple of unordered trees that represents the topology of the system. The roots of the trees are referred to as *regions* and the nodes are often referred to as *places* and may represent locations or other process constructors such as e.g. action prefixing. Some of the leaves may be *sites* (also referred to as holes) making the bigraph a (multi-hole) context. Each non-site place is typed with a *control* and has a number of *ports* linked together

[1] Email: mikkelbu@itu.dk
[2] Email: hilde@itu.dk

This is a preliminary version submitted for publication.
Funded in part by the Danish Research Agency grant no.: 2059-03-0031 (LaCoMoCo)
URL: www.itu.dk/theory/research/CoMo/Papers/bigraphsHomer.pdf

by the link graph. The *link graph* represents the connectivity in the system, corresponding to shared names in the π-calculus. Free names are represented by links connected to a set of names in the (outer) *interface* of the bigraph.

In so-called *pure* bigraphs, the place and link graph can be considered to be orthogonal structures, since the nesting of the places and the connections of the links have no interrelationship. Pure bigraphs are sufficient to represent calculi such as the pure Mobile Ambient calculus. The orthogonality breaks when we move to so-called *binding* and *local* bigraphs. Binding bigraphs were introduced in [11] to capture the notions of binding and scope of names as found in the π-calculus. In binding bigraphs we allow for a node to have *binding ports*, and require that any other port linked to the same link as a binding port to be within the node of the binding port. In [14], Milner refines the definition of binding bigraphs into *local bigraphs*. In local bigraphs, the free names (i.e. names in the interface) are all explicitly located at the regions of the bigraph, the same name possibly located at several regions. Correspondingly, holes (i.e. sites) are explicitly annotated by a set of names connected to links. Local bigraphs are used to facilitate the presentation of the λ-calculus in [15], which demonstrates how higher-order processes (process passing) can be presented in the bigraphical framework using explicit substitutions.

In the present paper we give the first bigraphical presentation of the combination of active processes in nested locations as present in the Mobile Ambients, non-linear higher-order process passing (by explicit substitution) as present in the λ-calculus and local names as present in the π-calculus. It turns out that the combination of non-linear, active process mobility and local names needs special care, i.e. we can not simply combine the previous presentations of the Mobile Ambients, the λ-calculus, and the π-calculus.

We take as our starting point the calculus of (asynchronous) Higher-Order Mobile Embedded Resources (Homer) [8]. Homer is a pure higher-order calculus inspired by prior higher-order calculi such as Plain CHOCS [18] and HOπ [17], and can be regarded as an extension of the λ-calculus to contain nested, active locations and concurrent synchronisation over (nested) named channels. It is also a natural subclass of bigraphs for studying active, mobile processes in nested locations. Basically, asynchronous Homer has two constructors for located resources $\overline{\delta}\langle r\rangle$ (passive) and $\delta[r]$ (active) where δ is a sequence of names representing the address of the resource. These two constructors correspond respectively to a passive and an active bigraph control with ports connected to the links δ. The interactions are controlled by two corresponding constructors for moving located resources $\delta(x)\,.\,p$ (receive) and $\overline{\delta}(x)\,.\,p$ (take), denoting respectively the usual input-prefixed process waiting to receive a (passive) process on the channel δ, and an input action for taking an *active* process from location δ, in both cases substituting the moved resource in for x in p. We allow interactions with arbitrarily deeply nested, active processes by simply composing addresses. In the example below we send the resource r down to the nested address ab (composed of a and b), and it is received at the address

b residing in the location a

$$\overline{ab}\langle r\rangle \mid a[b(x)\,.\,q \mid q'] \searrow a[\,q[r/x] \mid q'\,] \ . \tag{1}$$

Dually, we can also take up resources from nested locations as in

$$a[b[r] \mid p] \mid \overline{ab}(x)\,.\,q \searrow a[p] \mid q[r/x] \ . \tag{2}$$

As usual, we let $(n)p$ denote a process p in which the name n is local. With local names we also need to handle scope extension. For most of the process constructors scope extension is as expected, but when a resource is moved it may be necessary to extend the scope of a name through the boundary of a location. For instance, if the resource r contains the name n free, we will expect the reaction

$$a[(n)(b[r] \mid p)] \mid \overline{ab}(x)\,.\,q \searrow (n)(a[p] \mid q[r/x]) \ , \tag{3}$$

where we have *vertically*, through the location boundary, extended the scope of n to cover all possible occurrences of the name n. In the Mobile Ambients calculus vertical scope extension is performed in the structural congruence (along with the usual scope extension)

$$m[(n)p] \equiv (n)m[p] \ , \text{ if } n \neq m \ . \tag{4}$$

However, as also discovered in [5] this rule is not sound when mobile processes may be copied. There exists several solutions to this problem, all of them exclude the vertical scope extension in the structural congruence (4), and instead extend the scope in the reaction relation. This extension is either done *eagerly*, meaning that we always extend the scope, or *if and only if* the name n is free in r. In Homer we have chosen the latter solution, which corresponds to the usual semantics of e.g. HOπ. Combined with nested locations it has the consequence that a context can test if a name is free in a process, and so for any non-trivial congruence related processes must have the same set of free names (see, e.g., [8] for a detailed discussion). It is sometimes useful, however, to be able to abstract from free, but non-accessible names, as e.g. in the *perfect firewall equation* $(n)(n[p]) \approx \mathbf{0}$, stating that the behaviour of a computing resource at a local location is unobservable. To facilitate this we type processes explicitly with a set of names $\tilde{n}$ containing the free names. The *typed* perfect firewall equation then becomes $(n)(n[p]) : \tilde{n} \approx \mathbf{0} : \tilde{n}$ for $fn(p)\backslash\{n\} \subseteq \tilde{n}$. Interestingly, it turns out that for this equation to hold we also need to explicitly annotate all located sub-resources with a type, which is done by extending the syntax to $\overline{\delta}\langle r\rangle_{\tilde{n}}$ and $\delta[r]_{\tilde{n}}$.

Related Work

The Homer calculus were introduced in [8] together with labelled transition bisimulation congruences, and an encoding in Homer of the synchronous π-

calculus without summation were presented in [2,3]. Composite names in send and receive prefixes are also found in the π-calculus with polyadic synchronisation [4], however, the dual prefixes for active processes are not considered.

In [12,11] Jensen and Milner set up the basic theory of BRSs and exhibit a bigraphical presentation of the asynchronous π-calculus Aπ and prove that the derived labelled transition system (LTS) and its bisimilarity match closely the traditional LTS and bisimilarity of Aπ. Milner gives in [13] a bigraphical presentation of condition-event Petri nets and Jensen gives in his forthcoming thesis a presentation of the Mobile Ambient calculus [10]. Milner has refined the theory of binding bigraphs [14], utilised in [15] to give a bigraphical presentation of a variant of the λ-calculus with explicit substitutions. Several aspects of the current paper are inspired by this presentation.

Explicit substitutions have been widely applied in the setting of functional programming languages, primarily to bridge the gap between the abstract definition of a programming language and the concrete implementation. In the seminal work of Abadi et al. [1] on $\lambda\sigma$, a λ-calculus with explicit substitutions, the substitutions are propagated throughout the term and applied locally. The approach chosen in this paper differs from this solution, in the same way as Milner's λ-calculus did, since we also perform the substitution 'at a distance'. We also employ an explicit garbage collection of substitutions as also found in [16]. Explicit substitutions have also appeared in process calculi for concurrency and mobility. In particular the π-calculus has been augmented with explicit substitutions in several variants, e.g. using a global environment for the substitutions [7] or using De Bruijn indices and handling the name instantiation using a term rewrite system [9].

The paper is structured as follows: In Sec. 1.1 we briefly review the main concepts of local bigraphs, and in Sec. 1.2 we present calculus of Homerσ. Sec. 2 contains the presentation of Homerσ as a BRS, ending with the suggestion of localised bigraphs as a generalisation of local bigraphs in which links can be further localised.

1 Preliminaries

In this section we first briefly recall the main concepts of the theory of local bigraphs [14], and give a new definition of parametric reaction rules. We then present the asynchronous variant of the calculus Homer introduced in [8], but extended with explicit substitutions to present the higher-order process passing of Homer in the bigraphical framework.

1.1 Local Bigraphs

We refer the reader to [12] for the basic static and dynamic theory of (pure and binding) bigraphs and [14] and [15] for the remaining details about local bigraphs. In this paper we will primarily use a simple term language, introduced in the above mentioned papers, instead of the graphical representation of bigraphs. The term language consists of the following constructors: $h \mid\mid g$ and $h \mid g$ are the parallel product and prime parallel product of two bigraphs h and g, respectively. The prime parallel product merges the regions of two single-region (prime) bigraphs. The parallel product places the regions next to each other. The closure constructor $/n \circ g$ is the bigraph g, where we have removed the outer name n by replacing the name with an edge in g.

The *outer face* of a local bigraph is a pair $\langle m, \overrightarrow{X} \rangle$, where m is the number of regions and $\overrightarrow{X}$ is a vector of length m, such that X_i is the set of local names attached to the $i'th$ region. Similarly, the *inner face* is a pair $\langle n, \overrightarrow{Y} \rangle$ where n is the number of sites, $|\overrightarrow{Y}| = n$ and Y_i is the local names attached to the $i'th$ site. We can compose two bigraphs H and G, if the outer face of G and inner face of H matches, resulting in the bigraph $H \circ G$, where the content of the regions of G have been inserted into the respective sites of H, and the links connected to the local names in the i'th region of G are connected to the links connected to the corresponding local names of the i'th site of H.

A bigraph *signature* $\mathcal{K}$ is a set of controls and provides for each control K a pair of finite ordinals, the number of binding and free ports, the *binding arity* h and the *free arity* k, written $K : h \to k$. It also determines which controls are atomic, and which of the non-atomic controls are active.

A *ground reaction rule* is a pair (r, r') of ground bigraphs with the same outer face. Given a set of ground rules, the *reaction relation*, $\longrightarrow\!\!\!\!\!\triangleright$, is the least relation such that $D \circ r \longrightarrow\!\!\!\!\!\triangleright D \circ r'$ for each active context D and each ground rule (r, r'). Parametric reaction rules allow for the rules to contain parameters, that can be replicated, discarded, or just moved. A *parametric reaction rule* has a *redex* R and *reactum* R', and takes the form $(R : I \to K, R' : I' \to K, \eta)$, with inner faces $I = \langle m, \overrightarrow{X} \rangle$ and $I' = \langle m', \overrightarrow{X}' \rangle$, and $\eta : m' \to m$ is a map of ordinals, inducing the instantiation $\overline{\eta}$. For every parameter $d : I$ the parametric reaction rule generates a ground reaction rule $(R \circ d, R' \circ \overline{\eta}(d))$. Differently from the original definition in [14], we require that all outer names of a parameter are specified explicitly by the parametric reaction rule, to ensure that we handle scope extension properly. The *instantiation* maps a parameter for the redex to a parameter for the reactum and thereby allows for the rules to replicate some of their parameters and discard others. More precisely, a ground bigraph $a : \langle m, \overrightarrow{X} \rangle$ with no closed links crossing regions can be factorised uniquely into prime bigraphs as $a = c_0 \mid\mid \cdots \mid\mid c_{m-1}$, with $c_i : X_i$. For a map $\eta : m' \to m$ we then define the instantiation $\overline{\eta}$ as

$$\overline{\eta}(a) : \langle m', \overrightarrow{X'} \rangle \stackrel{def}{=} c_{\eta(0)} \mid\mid \cdots \mid\mid c_{\eta(m'-1)} \text{ , where } X'_j \stackrel{def}{=} X_{\eta(j)} \text{ for all } j \in m'.$$

1.2 Higher-Order Mobile Embedded Resources

We assume an infinite set of *names* $\mathcal{N}$ ranged over by m and n, and let $\tilde{n}$ range over finite sets of names. We let γ range over (possibly empty) sequences

$$\frac{}{\tilde{x} \vdash \mathbf{0} : \tilde{n}} \qquad \frac{\tilde{x} \vdash p : \tilde{n}_1 \qquad \tilde{x} \vdash q : \tilde{n}_2}{\tilde{x} \vdash p \mid q : \tilde{n}_1 \cup \tilde{n}_2} \qquad \frac{}{\tilde{x} \vdash (-)_{\tilde{n}} : \tilde{n}}$$

$$\frac{}{\tilde{x}x \vdash x : \tilde{n}} \qquad \frac{\tilde{x}x \vdash p : \tilde{n} \qquad \vdash q : \tilde{m}}{\tilde{x} \vdash p[x := q : \tilde{m}] : \tilde{n} \cup \tilde{m}} \qquad \frac{\tilde{x} \vdash p : \tilde{n}n}{\tilde{x} \vdash (n)p : \tilde{n}}$$

$$\frac{\tilde{x}x \vdash p : \tilde{n}}{\tilde{x} \vdash \varphi(x) \,.\, p : \tilde{n} \cup fn(\varphi)} \qquad \frac{\tilde{x} \vdash r : \tilde{m}}{\tilde{x} \vdash \varphi[\![r]\!]_{\tilde{m}} : \tilde{m} \cup fn(\varphi)}$$

Table 1
Typing rules for Homerσ

of names, and let δ range over non-empty sequences of names, referred to as *addresses* and let $|\delta|$ denote the length of the address δ, also we let $\varphi ::= \delta \mid \overline{\delta}$. We assume an infinite set of *process variables* $\mathcal{V}$ ranged over by x and y, and let $\tilde{x}$ range over finite sets of variables. The set $\mathcal{P}$ of *process expressions* for the calculus Homerσ of (asynchronous) Higher-Order Mobile Embedded Resources with explicit substitution is then defined as follows

$$\begin{array}{llll}
\textit{Processes:} & p, q, r & ::= & \mathbf{0} \;\mid\; \pi \,.\, p \;\mid\; p \mid q \;\mid\; (n)p \;\mid \\
 & & & p[x := q : \tilde{n}] \;\mid\; x \;\mid\; \overline{\delta}\langle r \rangle_{\tilde{n}} \;\mid\; \delta[r]_{\tilde{n}} \\
\textit{Prefixes:} & \pi & ::= & \delta(x) \;\mid\; \overline{\delta}(x)
\end{array}$$

The complementary actions $\overline{\delta}\langle r \rangle_{\tilde{n}}$ and $\delta(x)$ are the usual prefixes of Plain CHOCS [18] or HOπ, except that we allow sequences of names as addresses instead of only a name, and we explicit type the resource r. As described in the introduction, the actions $\delta[r]_{\tilde{n}}$ and $\overline{\delta}(x)$ are responsible for adding active process mobility to the calculus. We write $\varphi[\![r]\!]_{\tilde{n}}$ for $\delta[r]_{\tilde{n}}$ or $\overline{\delta}\langle r \rangle_{\tilde{n}}$. The process $p[x := q : \tilde{n}]$ is an explicit syntactic substitution, representing the processes p in a context that can substitute q (of type $\tilde{n}$) in for x. The typing rules to be defined below ensures that q is closed and that the free names of q are contained in $\tilde{n}$. As usual, we let the restriction operator (n) bind the name n, and let the prefixes $\varphi(x)$ and $p[x := q : \tilde{n}]$ bind the variable x.

Contexts $\mathcal{C}$ are defined by taking the grammar for processes and augmenting it with a symbol called a *hole*, written $(-)_{\tilde{n}}$ Note that holes are typed, only a process with type $\tilde{n}$ can be placed in a hole $(-)_{\tilde{n}}$.

We define the valid typing judgements of the form $\tilde{x} \vdash p : \tilde{n}$ inductive by the rules in Tab. 1. From now on we will only consider well-typed processes. Note that a process p is well-typed with respect to a finite set of variables $\tilde{x}$ and names $\tilde{n}$, written $\tilde{x} \vdash p : \tilde{n}$, if and only if the free names (variables) of p are included in the set $\tilde{n}$ ($\tilde{x}$), and for every sub-term $\varphi[\![r]\!]_{\tilde{m}}$ and $q[x := r : \tilde{m}]$ in p we have that r can be typed with the type $\tilde{m}$. We define the free names and free variables as usual with the addition that the free names of $\varphi[\![r]\!]_{\tilde{n}}$ and $p[x := r : \tilde{n}]$ are defined as $fn(\varphi) \cup \tilde{n}$ and $fn(p) \cup \tilde{n}$, respectively.

We say that a process with no free variables is *closed* and let $\mathcal{P}\sigma_c$ denote the set of closed processes. We let $\mathcal{P}\sigma_{/\alpha}$ (and $\mathcal{P}\sigma_{c/\alpha}$) denote the set of α-equivalence classes of (closed) process expressions, and we consider processes up to α-equivalence. We omit trailing $\mathbf{0}$s, write $\vdash p : \tilde{n}$ for $\emptyset \vdash p : \tilde{n}$, and let prefixing and restriction be right associative and bind stronger than explicit substitution and let explicit substitution bind stronger than parallel composition. For a set of names $\tilde{n} = \{n_1, \ldots, n_k\}$ we let $(\tilde{n})p$ denote $(n_1)\cdots(n_k)p$. We write $\tilde{m}\tilde{n}$ for $\tilde{m} \cup \tilde{n}$, always assuming $\tilde{m} \cap \tilde{n} = \emptyset$.

1.3 Reaction Semantics

We provide Homerσ with a reaction semantics defined using structural congruence, evaluation contexts, and reaction rules. A binary relation $\mathcal{R}$ on well-typed processes is called *well-typed* if and only if it relates processes p and q with the same type $\tilde{n}$ $(\tilde{x})$, written $\tilde{x} \vdash p \,\mathcal{R}\, q : \tilde{n}$. We will only consider well-typed relations in this paper. A relation $\mathcal{R}$ is called a *congruence* if and only if it is a well-typed equivalence relation and it satisfies that $\tilde{x} \vdash p \,\mathcal{R}\, q : \tilde{n}$ implies $\tilde{x}' \vdash \mathcal{C}(p) \,\mathcal{R}\, \mathcal{C}(q) : \tilde{n}'$ for all contexts $\mathcal{C}$.

Structural congruence $\equiv_\sigma$ is defined as the least congruence on well-typed processes relating $\tilde{x} \vdash p \equiv_\sigma q : \tilde{n}$, if $\tilde{x} \vdash p : \tilde{n}$, $\tilde{x} \vdash q : \tilde{n}$, and $p \equiv_\sigma q$ can be derived using the following rules

$$p \mid \mathbf{0} \equiv_\sigma p \qquad (p \mid p') \mid p'' \equiv_\sigma p \mid (p' \mid p'') \qquad p \mid q \equiv_\sigma q \mid p$$

$$(n)p \mid q \equiv_\sigma (n)(p \mid q),\ \text{if } n \notin \mathit{fn}(q) \qquad \pi \,.\, (n)p \equiv_\sigma (n)\pi \,.\, p,\ \text{if } n \notin \mathit{fn}(\pi)$$

$$(n)(m)p \equiv_\sigma (m)(n)p \qquad (n)p \equiv_\sigma p,\ \text{if } n \notin \mathit{fn}(p)$$

$$(n)(p[x := r : \tilde{n}]) \equiv_\sigma (n)p[x := r : \tilde{n}],\ \text{if } n \notin \tilde{n}$$

As Homerσ permits reactions arbitrarily deep in the location hierarchy and also permits reactions between a process and an arbitrarily deeply nested sub-resource, we define the concepts of evaluation and path contexts. An *evaluation context* $\mathcal{E}$ is a context with no free variables and whose hole is not guarded by a prefix, nor does it occur as the object of a send constructor

$$\mathcal{E} ::= (-)_{\tilde{n}} \ \mid\ \mathcal{E} \mid p \ \mid\ (n)\mathcal{E} \ \mid\ \delta[\mathcal{E}]_{\tilde{n}},\ \text{for } p \in \mathcal{P}\sigma_c \ .$$

We define a family of multi-hole *path contexts* $\mathcal{C}^{\tilde{n}}_{\gamma}$, indexed by a path address $\gamma \in \mathcal{N}^*$ and a set of names $\tilde{n}$, inductively in $\tilde{n}$ and γ

$$\mathcal{C}^{\emptyset}_{\epsilon} ::= (-)_{\tilde{n}} \quad \text{and} \quad \mathcal{C}^{\tilde{n}\tilde{m}}_{\delta\gamma} ::= \delta[(\tilde{n})(\mathcal{C}^{\tilde{m}}_{\gamma} \mid (-)_{\tilde{n}'})]_{\tilde{m}'} \ ,$$

whenever $\tilde{n} \cap \gamma = \emptyset$. Note that the evaluation context $\delta[\mathcal{E}]_{\tilde{n}}$ enables internal reactions of active resources, and that for a path context $\mathcal{C}^{\tilde{n}}_{\gamma}$, the path address γ indicates the path under which the context's hole is found, and the set of names $\tilde{n}$ indicates the bound names of the hole. The side condition in

$$(send\sigma) \quad \vdash \overline{\gamma\delta}\langle r\rangle_{\tilde{n}} \mid \mathcal{C}^{\tilde{m}}_{\gamma}(\delta(x) \,.\, p, \overrightarrow{p}\,) \searrow_{\sigma} \tilde{n} \odot \mathcal{C}^{\tilde{m}}_{\gamma}(p[x := r : \tilde{n}], \overrightarrow{p}\,) : \tilde{n}' \;\;,$$
$$\text{if } \tilde{m} \cap (\delta \cup \tilde{n}) = \emptyset$$

$$(take\sigma) \quad \vdash \mathcal{C}^{\tilde{m}}_{\gamma}(\delta[r]_{\tilde{n}}, \overrightarrow{p}\,) \mid \overline{\gamma\delta}(x) \,.\, p \searrow_{\sigma} (\tilde{n} \cap \tilde{m})\big(\tilde{n} \odot \mathcal{C}^{\tilde{m}}_{\gamma}(\mathbf{0}, \overrightarrow{p}\,) \mid p[x := r : \tilde{n}]\big) : \tilde{n}' \;\;,$$
$$\text{if } \tilde{m} \cap (\delta \cup fn(p)) = \emptyset$$

$$(apply\sigma) \quad \vdash \mathcal{C}(x)[x := r : \tilde{n}] \searrow_{\sigma} \tilde{n} \odot \mathcal{C}(r)[x := r : \tilde{n}] : \tilde{n}' \;\;,$$
$$\text{if } \mathcal{C} \text{ does not bind } x \text{ or the names in } \tilde{n}$$

$$(garbage\sigma) \vdash p[x := q : \tilde{n}] \searrow_{\sigma} p : \tilde{n}' \;\;, \text{ if } x \notin fv(p)$$

Table 2
Reaction rules for Homerσ

the definition of path contexts ensures that none of the names in the path address of the hole are bound. The bound names ($\tilde{n}$) in the definition of path contexts are needed since the structural congruence does not permit vertical scope extension, as described in the introduction.

We handle the vertical scope extension and the update of the type annotation of a location using an *open* operator, defined on path contexts. We define an *open* operator on path contexts $\tilde{m} \odot \mathcal{C}^{\tilde{n}}_{\gamma}$ inductively by:

$$\begin{aligned} \tilde{m} \odot \mathcal{C}^{\emptyset}_{\epsilon} &= \mathcal{C}^{\emptyset}_{\epsilon} \\ \tilde{m} \odot \mathcal{C}^{\tilde{n}_1\tilde{n}_2}_{\delta\gamma} &= \delta[(\tilde{n}_1 \setminus \tilde{m})(\tilde{m} \odot \mathcal{C}^{\tilde{n}_2}_{\gamma} \mid (-)_{\tilde{n}'})]_{\tilde{m}' \cup \tilde{m}} \;\;, \end{aligned}$$

if $\mathcal{C}^{\tilde{n}_1\tilde{n}_2}_{\delta\gamma} = \delta[(\tilde{n}_1)(\mathcal{C}^{\tilde{n}_2}_{\gamma} \mid (-)_{\tilde{n}'})]_{\tilde{m}'}$ and $\tilde{m} \cap \tilde{n}_1\tilde{n}_2 \cap fn(\mathcal{C}^{\tilde{n}_1\tilde{n}_2}_{\delta\gamma}) = \emptyset$. Intuitively, the open operator in $\tilde{m} \odot \mathcal{C}^{\tilde{n}}_{\gamma}$ removes the names $\tilde{m}$ from the bound names of the hole and adds them to the type annotation of the locations that are part of the address path. When applied in the reaction rule, the latter condition of the open operator can always be met by α-conversion, the condition ensures us that we can extend the scope by using the open operator and place the restriction at top level, without any name captures.

As for the structural congruence, we define the reaction relation for Homerσ, written $\searrow_{\sigma}$, as the least well-typed relation on well-typed closed processes satisfying the rules in Tab. 2 and closed under all evaluation contexts $\mathcal{E}$ and structural congruence.

The ($send\sigma$) rule expresses how a passive resource r is sent (down) to the (sub) location γ, where it is received at the address δ. The side conditions ensure the location path is not bound in the context and that no free names of r get bound during movement. The open operator only extends the type annotation of the locations constituting the location path and does not lift any restrictions. The ($take\sigma$) rule captures that a computing resource r is taken from the (sub) location γ, where it is running at the address δ. Again, the side conditions ensure that the location path is not bound in the context, and that no free names are bound, when we lift the restriction. It is possible that the open operator both lifts restrictions and extends the type annotation of the locations. The rule ($apply\sigma$) replaces one occurrence of the variable (arbitrarily deep in the context) with the content of the explicit substitution. Note that we overload the use of $\odot$ in ($apply\sigma$), applying the operator to a general context and not only a path context. However, the result of the operator is the same, it extends the type annotation of all the locations (and send constructors) containing this occurrence of the variable. The latter condition of the rule can always be satisfied using α-conversion of the context. The ($garbage\sigma$) rule is responsible for garbage collecting superfluous substitutions.

The types ensure that no names can disappear from the free names of a location or from top-level during reaction. Locations or send constructors in the process that receives a resource r can get their type annotation extended by the type of r that do not already appear in their annotation.

2 Bigraphical Semantics of Homerσ

In this section we give the bigraphical presentation of Homerσ as the BRS $\mathcal{H}$omerσ. First, we present the signature for $\mathcal{H}$omerσ, and give a fully compositional translation of Homerσ-terms into bigraphs. Second, we translate the path contexts and the reaction relation. An important criteria for the presentation is to show that there is a static and operational correspondence between Homerσ and its presentation as a BRS, meaning that structural congruence of Homerσ corresponds to graph isomorphism in the bigraphical presentation, and that reactions match.

The signature has controls **rece** and **take** representing the two input prefixes, and **send**, and **loca** representing the two kinds (passive and active) of located resources. Controls **var**, **sub**, and **def** represent a variable and the constructs for explicit substitutions, respectively. Finally, the signature also has atomic controls **tname** (abbreviation for **typename**) and **ann** (abbreviation for **annotation**) to represent the explicit type annotation of resource and send constructors. We will discuss this in more detail after having presented the reaction rules in the bigraphical framework. Note that since path addresses are represented with one port for each element in the sequence, we have an infinite family of controls indexed by the length of the address. In total, the signature for $\mathcal{H}$omerσ is defined as follows.

- The controls **var**: $0 \to 1$ and **tname**: $0 \to 1$ are atomic
- The families of controls: $\mathbf{rece}_{|\delta|}$: $1 \to |\delta|$, $\mathbf{take}_{|\delta|}$: $1 \to |\delta|$, and $\mathbf{send}_{|\delta|}$: $0 \to |\delta|$ are all inactive
- The family of controls $\mathbf{loca}_{|\delta|}$: $0 \to |\delta|$ is active
- The controls **def**: $0 \to 1$, **sub**: $1 \to 0$, and **ann**: $0 \to 0$ are inactive

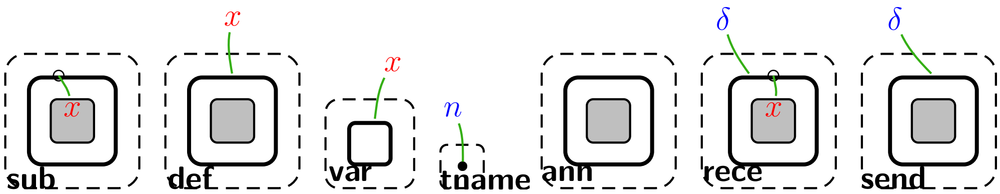


Figure 1. Ions and atoms for Homerσ

Note that we have no controls for restriction and the inactive process. This is to ensure the static correspondence, as stated in Thm. 2.2.

In Fig. 1 we depict the ions and the atoms used in the translation, we have left out the controls **take** and **loca** as they are similar to **rece** and **send**, respectively. We have chosen to depict the control **tname** as just a dot, •, in order to be able to distinguish graphically between **tname** and **var** controls. Following the convention of Milner [15], we write $\mathbf{var}_x$ and $\mathbf{tname}_n$ for the atoms, and we denote the ions as follows

$$\mathbf{sub}_{(x)} \mathbin{\overline{\oplus}} \mathbf{id}_Z \quad \mathbf{def}_x \mathbin{\overline{\oplus}} \mathbf{id}_Z \quad \mathbf{ann} \mathbin{\overline{\oplus}} \mathbf{id}_Z \quad \mathbf{rece}_{\delta(x)} \mathbin{\overline{\oplus}} \mathbf{id}_Z \quad \mathbf{send}_\delta \mathbin{\overline{\oplus}} \mathbf{id}_Z \ .$$

We write the binding port names in parenthesis and last. We use the $\overline{\oplus}$ operator to extend a bigraph with an identity wiring, hereby extending the inner and outer face of the bigraph. So the ion $\mathbf{send}_\delta \mathbin{\overline{\oplus}} \mathbf{id}_Z$ has Z as inner names and $Z \cup \delta$ as outer names.

2.1 The Translation

We have a fully compositional translation from Homerσ to bigraphs.

Definition 2.1 (Translation of Homerσ-terms into bigraphs) *We define the translation of a Homerσ-term p inductively in the inference of $\tilde{x} \vdash p : \tilde{n}$*

$$
\begin{aligned}
[\![\tilde{x} \vdash \mathbf{0} : \tilde{n}]\!] &= \tilde{n} \mathbin{\overline{\oplus}} \tilde{x} \\
[\![\tilde{x} \vdash p \mid q : \tilde{n}_1 \cup \tilde{n}_2]\!] &= [\![\tilde{x} \vdash p : \tilde{n}_1]\!] \mid [\![\tilde{x} \vdash q : \tilde{n}_2]\!] \\
[\![\tilde{x} \vdash (n)p : \tilde{n}]\!] &= /n \circ ([\![\tilde{x} \vdash p : \tilde{n}n]\!]) \\
[\![\tilde{x}x \vdash x : \tilde{n}]\!] &= \mathbf{var}_x \mathbin{\overline{\oplus}} \tilde{n} \mathbin{\overline{\oplus}} \tilde{x} \\
[\![\tilde{x} \vdash p[x := r : \tilde{n}'] : \tilde{n} \cup \tilde{n}']\!] &= (\mathbf{sub}_{(x)} \mathbin{\overline{\oplus}} \mathbf{id}_{\tilde{n},\tilde{x}})([\![\tilde{x}x \vdash p : \tilde{n}]\!] \mid \\
&\qquad (\mathbf{def}_x \mathbin{\overline{\oplus}} \mathbf{id}_{\tilde{n}'})([\![\vdash r : \tilde{n}']\!] \mid (\mathbf{ann} \mathbin{\overline{\oplus}} \mathbf{id}_{\tilde{n}'})[\![\tilde{n}']\!])) \\
[\![\tilde{x} \vdash \delta[r]_{\tilde{n}'} : \tilde{n}' \cup fn(\delta)]\!] &= (\mathbf{loca}_\delta \mathbin{\overline{\oplus}} \mathbf{id}_{\tilde{n},\tilde{x}})([\![\tilde{x} \vdash r : \tilde{n}']\!] \mid (\mathbf{ann} \mathbin{\overline{\oplus}} \mathbf{id}_{\tilde{n}'})[\![\tilde{n}']\!]) \\
[\![\tilde{x} \vdash \overline{\delta}\langle r\rangle_{\tilde{n}'} : \tilde{n}' \cup fn(\delta)]\!] &= (\mathbf{send}_\delta \mathbin{\overline{\oplus}} \mathbf{id}_{\tilde{n},\tilde{x}})([\![\tilde{x} \vdash r : \tilde{n}']\!] \mid (\mathbf{ann} \mathbin{\overline{\oplus}} \mathbf{id}_{\tilde{n}'})[\![\tilde{n}']\!]) \\
[\![\tilde{x} \vdash \delta(x) \, . \, p : \tilde{n} \cup fn(\delta)]\!] &= (\mathbf{rece}_{\delta(x)} \mathbin{\overline{\oplus}} \mathbf{id}_{\tilde{n},\tilde{x}})[\![\tilde{x}x \vdash p : \tilde{n}]\!] \\
[\![\tilde{x} \vdash \overline{\delta}(x) \, . \, p : \tilde{n} \cup fn(\delta)]\!] &= (\mathbf{take}_{\delta(x)} \mathbin{\overline{\oplus}} \mathbf{id}_{\tilde{n},\tilde{x}})[\![\tilde{x}x \vdash p : \tilde{n}]\!]
\end{aligned}
$$

and we translate the type annotations as follows: $[\![\tilde{n}]\!] = \prod_{n \in \tilde{n}} \mathbf{tname}_n$.

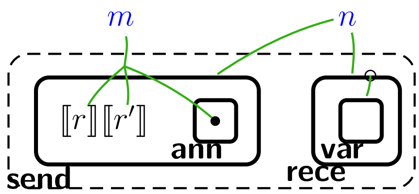


Figure 2. Example on translation of the term $\overline{n}\langle r \mid r'\rangle_{\{m\}} \mid n(x) \,.\, x$ into a bigraph

We represent $\mathbf{0}$ as an empty bigraph with the correct outer face, parallel composition is represented by the prime product, and we use a closure $/n$ to represent the restriction of the name n. A variable is represented as a node of control **var** which is connect to the name x. We represent the explicit substitutions in Homerσ in the same way as [15], except that we have augmented the explicit substitution with a type annotation.

The two constructors $\delta[r]_{\tilde{n}}$ and $\overline{\delta}\langle r\rangle_{\tilde{n}}$ are represented by a place with the corresponding control containing the representation of the resource r and the representation of the type annotation as a set of **tname** nodes enclosed by a place with control **ann**. The two prefixes $\delta(x) \,.\, p$ and $\overline{\delta}(x) \,.\, p$ are encoded straightforwardly by a node of the respective control, where the variable x is bound in the enclosed encoding of p, and we require that x and $\tilde{x}$ are disjoint. As an example on the translation from Homerσ-terms to bigraphs, we depict in Fig. 2 the result of the translation of $\overline{n}\langle r \mid r'\rangle_{\{m\}} \mid n(x) \,.\, x$. The static correspondence, stated by the theorem below, is proven in App. B.

Theorem 2.2 (Static correspondence) *$\tilde{x} \vdash p \equiv_\sigma q : \tilde{n}$ if and only if $[\![\tilde{x} \vdash p : \tilde{n}]\!] = [\![\tilde{x} \vdash q : \tilde{n}]\!]$.*

In order to present the reaction rules of Homerσ we first present the path contexts and the open operation. We define the translation of a path context $\mathcal{C}^{\tilde{n}}_{\gamma}$ into a bigraph of a certain form, called a *path bigraph*, inductively in the structure of $\mathcal{C}^{\tilde{n}}_{\gamma}$

$$
\begin{aligned}
[\![\vdash \mathcal{C}^{\emptyset}_{\epsilon} : \tilde{n}'']\!] &= \mathsf{id}_{\tilde{n}''} \\
[\![\vdash \mathcal{C}^{\tilde{n}\tilde{m}}_{\delta\gamma} : \tilde{n}'']\!] &= (\mathsf{loca}_\delta \,\overline{\oplus}\, \mathsf{id}_{\tilde{n}''})(/\tilde{n} \circ ([\![\vdash \mathcal{C}^{\tilde{m}}_{\gamma} : \tilde{n}']\!] \mid \mathsf{id}_{\tilde{n}'}) \mid (\mathsf{ann} \,\overline{\oplus}\, \mathsf{id}_{\tilde{m}'})[\![\tilde{m}']\!])
\end{aligned}
$$

if $\mathcal{C}^{\tilde{n}\tilde{m}}_{\delta\gamma} = \delta[(\tilde{n})(\mathcal{C}^{\tilde{m}}_{\gamma} \mid (-)_{\tilde{n}'})]_{\tilde{m}'}$. We let F, F' range over path bigraphs. And as for Homerσ we will sometimes use subscript to denote the address of the hole and superscript to denote the bound names of the hole. We define an *open operator* on path bigraphs, $\tilde{m} \odot_b F$, extending the type annotations with $\tilde{m}$

$$
\begin{aligned}
\tilde{m} \odot_b \mathsf{id}_{\tilde{n}} &= \mathsf{id}_{\tilde{n}\cup\tilde{m}} \\
\tilde{m} \odot_b F &= (\mathsf{loca}_\delta \,\overline{\oplus}\, \mathsf{id}_{\tilde{n}'',\tilde{m}})(/(\tilde{n} \setminus \tilde{m}) \circ ((\tilde{m} \odot_b [\![\vdash \mathcal{C}^{\tilde{m}}_{\gamma} : \tilde{n}']\!]) \mid \mathsf{id}_{\tilde{n}'}) \mid \\
&\qquad (\mathsf{ann} \,\overline{\oplus}\, \mathsf{id}_{\tilde{m}',\tilde{m}})[\![\tilde{m}' \cup \tilde{m}]\!])
\end{aligned}
$$

if $F = (\mathbf{loca}_\delta \mathbin{\overline{\oplus}} \mathbf{id}_{\tilde{n}''})(/\tilde{n} \circ ([\![\vdash \mathcal{C}_\gamma^{\tilde{m}} : \tilde{n}']\!] \mid \mathbf{id}_{\tilde{n}'}) \mid (\mathbf{ann} \mathbin{\overline{\oplus}} \mathbf{id}_{\tilde{m}'})[\![\tilde{m}']\!])$. Note that we cannot just juxtaposition the type annotations as $[\![\tilde{m}']\!] \mid [\![\tilde{m}]\!]$, since we represent the individual elements of the type annotations explicitly with one node per element in the annotation, as this would result in our annotations being multisets rather than sets. In App. A we present a sorting, which describes the bigraphs corresponding to Homerσ processes.

2.2 Reaction Rules of Homerσ

In this subsection we present the reaction rules of Homerσ.

Definition 2.3 (reaction rules of Homerσ) *We define the four reaction rules of Homerσ below*

Send:

$R = (\mathbf{send}_{\gamma\delta} \mathbin{\overline{\oplus}} \mathbf{id}_{\tilde{n}})\big(\mathbf{id}_{\tilde{n}} \mid (\mathbf{ann} \mathbin{\overline{\oplus}} \mathbf{id}_{\tilde{n}})\big) \mid F_\gamma \circ (\mathbf{rece}_{\delta(x)} \mathbin{\overline{\oplus}} \mathbf{id}_{\tilde{n}'})$
$R' = (\tilde{n} \odot_b F_\gamma) \circ (\mathbf{sub}_{(x)} \mathbin{\overline{\oplus}} \mathbf{id}_{\tilde{n}'})(\mathbf{id}_{x\tilde{n}'} \mid (\mathbf{def}_x \mathbin{\overline{\oplus}} \mathbf{id}_{\tilde{n}})(\mathbf{id}_{\tilde{n}} \mid (\mathbf{ann} \mathbin{\overline{\oplus}} \mathbf{id}_{\tilde{n}})))$
$\eta = \{0 \mapsto 2, 1 \mapsto 0, 2 \mapsto 1\}$

Take:

$R = F_\gamma^{\tilde{m}} \circ (\mathbf{loca}_\delta \mathbin{\overline{\oplus}} \mathbf{id}_{\tilde{n}})(\mathbf{id}_{\tilde{n}} \mid (\mathbf{ann} \mathbin{\overline{\oplus}} \mathbf{id}_{\tilde{n}})) \mid (\mathbf{take}_{\gamma\delta(x)} \mathbin{\overline{\oplus}} \mathbf{id}_{\tilde{n}'})$
$R' = /(\tilde{m} \cap \tilde{n}) \circ ((\tilde{n} \odot_b F_\gamma^{\tilde{m}}) \circ \mathbf{0}) \mid (\mathbf{sub}_{(x)} \mathbin{\overline{\oplus}} \mathbf{id}_{\tilde{n}'})(\mathbf{id}_{x\tilde{n}'} \mid (\mathbf{def}_x \mathbin{\overline{\oplus}} \mathbf{id}_{\tilde{n}})(\mathbf{id}_{\tilde{n}} \mid (\mathbf{ann} \mathbin{\overline{\oplus}} \mathbf{id}_{\tilde{n}})))$
$\eta = \{0 \mapsto 2, 1 \mapsto 0, 2 \mapsto 1\}$

Apply:

$R = (\mathbf{sub}_{(x)} \mathbin{\overline{\oplus}} \mathbf{id}_{\tilde{n}'})(\mathcal{C} \circ \mathbf{var}_x \mid (\mathbf{def}_x \mathbin{\overline{\oplus}} \mathbf{id}_{\tilde{n}})(\mathbf{id}_{\tilde{n}} \mid (\mathbf{ann} \mathbin{\overline{\oplus}} \mathbf{id}_{\tilde{n}})))$
$R' = (\mathbf{sub}_{(x)} \mathbin{\overline{\oplus}} \mathbf{id}_{\tilde{n}'})(\tilde{n} \odot_b \mathcal{C} \circ \mathbf{id}_{\tilde{n}} \mid (\mathbf{def}_x \mathbin{\overline{\oplus}} \mathbf{id}_{\tilde{n}})(\mathbf{id}_{\tilde{n}} \mid (\mathbf{ann} \mathbin{\overline{\oplus}} \mathbf{id}_{\tilde{n}})))$
$\eta = \{0, 1 \mapsto 0, 2 \mapsto 1\}$

Garbage:

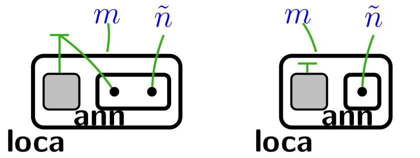
$R = (\mathbf{sub}_{(x)} \mathbin{\overline{\oplus}} \mathbf{id}_{\tilde{n}'})\big(\mathbf{id}_{\tilde{n}'} \mid (\mathbf{def}_x \mathbin{\overline{\oplus}} \mathbf{id}_{\tilde{n}})\big), \quad R' = \mathbf{id}_{\tilde{n}'}, \quad \eta = \{0 \mapsto 0\}$

In all the rules we have chosen to enumerate the holes from left to right in the terms representing the bigraphs, but omitting the last k holes in the $k+1$-hole path contexts F_γ and $F_\gamma^{\tilde{m}}$ on which the instantiation acts as the identity. In both the rules Send and Take the path bigraph F_γ does not bind the names in δ. In both rules the content of the **ann** node is used in the open operator, that is the set $\tilde{n}$. Both rules mimic their counterparts in Homerσ closely. Note that it is crucial that we have explicitly typed the parameters of the parametric reaction rule, and that we do not allow parameters to contain outer names not mentioned explicitly in the rules. In the rule Apply we utilise a general Homerσ context $\mathcal{C}$ satisfying the sorting requirement and that it does not close the variable-link x. The reaction rule Garbage, which discards the explicit substitution, is defined as in [15].

The proof of the operational correspondence, stated in the theorem below, is given in App. D.

Theorem 2.4 (Operational correspondence) *For every well-typed process* $\vdash p : \tilde{n}$, *we have*

$$\vdash p \searrow_\sigma p' : \tilde{n} \text{ if and only if } [\![\vdash p : \tilde{n}]\!] \rightarrow [\![\vdash p' : \tilde{n}]\!] \ .$$



$(n)m[(-)_n]_{\tilde{n}n}$ $\quad$ $m[(n)(-)_n]_{\tilde{n}}$

Figure 3. Location of a restriction

Now, let us take a closer look at the use of the type annotations. As mentioned in the introduction we have to be careful when combining local names and non-linear process passing. Since the two processes

$$(n)m[P] \quad \text{and} \quad m[(n)P] \qquad (\text{assuming } n \neq m) \tag{5}$$

are not structural congruent in general, they should not give rise to isomorphic bigraphs under the translation. If we consider our encoding without type annotations, then the two processes in (5) will give rise to isomorphic bigraphs, since we have no means to detect whether the closure occur outside or inside the location. In BRSs which copy parameters this would lead to the same kind of problems as mentioned in the introduction. In Fig. 3 we have illustrated how the type annotations helps us in distinguishing the two bigraphs. If the name appears in the type annotation, then the closure must be outside the location and every copy of the parameter will share this link. On the other hand, if the restricted name does not appear in the type annotation then every copy of the parameter will have a distinct link.

An immediate suggestion for an alternative to the type annotations is to represent name closures explicitly as a control with a binding port. However, then the usual scope condition would require the place with the binding port in the representation of $(n)p$ to be *around* the process p, which would break the usual structural congruence equalities such as $(n)(m)p \equiv_\sigma (m)(n)p$ and $(n)p \mid q \equiv_\sigma (n)(p \mid q)$, for $n \notin \mathit{fn}(q)$.

Recently Jensen and Milner have proposed a solution to the same problem of copying parameters with closed links unambiguously. In their solution they make use of an atomic **res** place for the restriction with a new kind of *outward-binding* port. The sole purpose of the **res** place is to facilitate this binding port, but contrary to the binding ports in normal binding bigraphs, this port does not bind inside the node, but instead it binds inside the parent node. Besides this change the port behaves as a traditional binding port. This explicit representation of restriction using one **res** place per restriction behaves well wrt the structural equalities above, but instead it breaks the equalities: $\pi \,.\, (n)p \equiv_\sigma (n)\pi \,.\, p$, if $n \notin \mathit{fn}(\pi)$ $\quad$ and $\quad$ $(n)p \equiv_\sigma p$, if $n \notin \mathit{fn}(p)$. More importantly, this solution does neither provide the desired *bisimulation* congruence. The typed perfect firewall equation $(n)(n[p]) : \tilde{n} \approx \mathbf{0} : \tilde{n}$ given in the introduction will only hold if $\mathit{fn}(p) \subseteq \{n\}$. The reason is that without the explicit localisation of links within active sub locations we loose local information

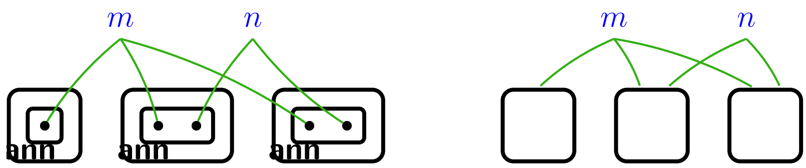


Figure 4. Original representation and using localised links

about the outer names of a process when we place it in a context.

2.3 Bigraphs with Localised Links

Since the type annotations in Homerσ are sets, we needed a way to associate an arbitrary number of names to a place in an *unordered* way. In the left-hand side of Fig. 4 we have sketched a situation where we have 3 places representing the Homerσ process $\delta[\mathbf{0}]_m \mid \delta[\mathbf{0}]_{m,n} \mid \delta[\mathbf{0}]_{m,n}$, where we have omitted the links δ. The solution used in this paper (and also used in the encoding of "The Game of Life" in [6]) is to introduce an **ann** place as a child of the place, and let it contain one **tname** place per name that we want to associate with the grand-parent place.

The annotation of names to places suggests an extension to local bigraphs in which one can associate names directly to a place in an *unordered* way, as illustrated on the right-hand side of Fig. 4, which we will call *localised links*. A direct consequence of this extension will be that we can remove the controls **tname** and **ann** from the encoding and instead represent the type annotations directly using localised links.

We do not propose localised links as a replacement for traditional links, but rather as an extension to these, as we still also want to be able to connect links to ordered ports, e.g. when representing $m[p]_{\{m\}}$ the name m will both be connected to the port corresponding to the address of the location, and localised in the place because of the type annotation.

Formally, we suggest to introduce a new function to the definition of a local bigraph. For a local bigraph $G : \langle m, \overrightarrow{X}\rangle \to \langle n, \overrightarrow{Y}\rangle$ with the set of edges E and the set of places V, we let the function *localise* map edges and outer names to a set of places, $\mathit{localise} : E \uplus Y \to \texttt{Pow}(V)$. We require that this map satisfies a scoping condition as for traditional links. We define the composition of two bigraphs

$$F : \langle m, \overrightarrow{X}\rangle \to \langle n, \overrightarrow{Y}\rangle \qquad \text{with places } V\text{, edges } E\text{, and function } \mathit{localise}$$
$$G : \langle l, \overrightarrow{Z}\rangle \to \langle m, \overrightarrow{X}\rangle \qquad \text{with places } V'\text{, edges } E'\text{, and function } \mathit{localise}'$$

as usual for local bigraphs. The localisation function $\mathit{localise}'' : E \uplus E' \uplus Y \to \mathcal{P}(V) \uplus \mathcal{P}(V')$ for $F \circ G$ is defined as follows (using the link map, *link*, of F)

$$\mathit{localise}''(x) = \begin{cases} \mathit{localise}'(x) & \text{if } x \in E' \ , \\ \mathit{localise}(x) \biguplus_{x' \in X \text{ and } \mathit{link}(x') = x} \mathit{localise}'(x') & \text{if } x \in E \uplus Y \ . \end{cases}$$

The locations of an edge in E' remain unchanged by the composition, whereas for a name in Y or an edge in E we might need to combine the locations of *localise* and *localise*$'$, if a name in X links to the name or edge, respectively.

3 Conclusions and Further Work

We have presented a higher-order calculus with non-linear active process mobility and local names, Homerσ as a bigraphical reactive system Ɦomerσ. We prove that structural congruence of Homerσ corresponds to graph isomorphism in Ɦomerσ and that there is a tight operational correspondence between the reaction relation of Homerσ and the reaction relation of Ɦomerσ. The presentation highlights the importance of keeping explicit track of the free names of parameters in reaction rules of bigraphs. It also address the issue of localisation of names (links) which suggests an extension to local bigraphs called *bigraphs with localised links*.

Several interesting questions arise from the work done in this paper. First and foremost, we plan to examine the labelled transition bisimulation congruence derivable using the general theory of bigraphs and compare it to the labelled transition bisimulation congruences for Homer in [8]. In this process we plan to examine proof techniques known from calculi for concurrency and mobility in the setting of bigraphs. Especially we plan to investigate the notion of *up-to* proof techniques related to bisimulation equivalences in bigraphs. We would also plan to further examine the extension of localised links, both with respect to facilitate encodings as bigraphical reactive systems and with respect to the behavioural theory of bigraphical reactive systems, in particular if the extension retains relative pushouts.

Currently several proposals exists for expressing constraints on the possible nesting of nodes, the linkage between ports etc. It would be interesting to see whether the sorting presented in App. A can be expressed in these settings, and in particular if we can enforce a more strict control with the movement and locations of closed free links. Hence to capture some of the same information as the outward-binding node, but without introducing an explicit node representing the restriction.

Acknowledgement

The authors would like to thank the members of the *Bigraphical Programming Languages*-group at ITU for helpful discussions about how to encode certain properties and constructs in bigraphs. We would also like to thank Robin Milner and Ole Høgh Jensen for providing access to early drafts of their papers.

References

[1] Abadi, M., L. Cardelli, P.-L. Curien and J.-J. Levy, *Explicit substitutions*, Journal of Functional Programming **1** (1991), pp. 375–416.

[2] Bundgaard, M., T. Hildebrandt and J. C. Godskesen, *A CPS encoding of name-passing in higher-order mobile embedded resources*, in: J. Baeten and F. Corradini, editors, *Proceedings of the 11th International Workshop on Expressiveness in Concurrency (EXPRESS'04)*, Electronic Notes in Theoretical Computer Science **128** (2004), pp. 131–150.

[3] Bundgaard, M., T. Hildebrandt and J. C. Godskesen, *A CPS encoding of name-passing in higher-order mobile embedded resources*, Theoretical Computer Science (2005), accepted for publication in a special issue of TCS.

[4] Carbone, M. and S. Maffeis, *On the expressive power of polyadic synchronisation in π-calculus*, Nordic Journal of Computing **10** (2003), pp. 70–98.

[5] Castagna, G., J. Vitek and F. Z. Nardelli, *The Seal calculus* (2004), accepted for publication in *Information and Computation*.

[6] Debois, S. and T. C. Damgaard, *Bigraphs by example*, Technical Report TR-2005-61, IT University of Copenhagen (2005).

[7] Ferrari, G., U. Montanari and P. Quaglia, *A π-calculus with explicit substitutions*, Theoretical Computer Science **168** (1996), pp. 53–103.

[8] Hildebrandt, T., J. C. Godskesen and M. Bundgaard, *Bisimulation congruences for Homer — a calculus of higher order mobile embedded resources*, Technical Report TR-2004-52, IT University of Copenhagen (2004).

[9] Hirschkoff, D., *Handling substitutions explicitely in the π-calculus*, in: *Proceedings of Second International Workshop on Explicit Substitutions: Theory and Applications to Programs and Proofs (WESTAPP'99)*, 1999, pp. 28–43.
URL http://cermics.enpc.fr/~dh/sigma/full.ps.gz

[10] Jensen, O. H., "Mobile Processes in Bigraphs," Ph.D. thesis, Department of Computer Science, Aalborg University (2005), forthcoming.

[11] Jensen, O. H. and R. Milner, *Bigraphs and transitions*, in: *Proceedings of the 30rd ACM SIGPLAN-SIGACT symposium on Principles of programming languages (POPL'03)* (2003), pp. 38–49.

[12] Jensen, O. H. and R. Milner, *Bigraphs and mobile processes (revised)*, Technical Report UCAM-CL-TR-580, University of Cambridge, Computer Laboratory (2004).
URL http://www.cl.cam.ac.uk/TechReports/UCAM-CL-TR-580.pdf

[13] Milner, R., *Bigraphs for petri nets*, in: J. Desel, W. Reisig and G. Rozenberg, editors, *Lectures on Concurrency and Petri Nets: Advances in Petri Nets*, Lecture Notes in Computer Science **3098** (2004), pp. 686–701.

[14] Milner, R., *Bigraphs whose names have multiple locality*, Technical Report UCAM-CL-TR-603, University of Cambridge, Computer Laboratory (2004). URL http://www.cl.cam.ac.uk/TechReports/UCAM-CL-TR-603.pdf

[15] Milner, R., *Local bigraphs, confluence and λ-calculus* (2004), draft of October 31, 2004.

[16] Rose, K. H., *Explicit substitution — tutorial & survey*, Lecture Series LS-96-3, BRICS, Department of Computer Science, University of Aarhus (1996), v+150 pp.

[17] Sangiorgi, D., "Expressing Mobility in Process Algebras: First-Order and Higher-Order Paradigms," Ph.D. thesis, Department of Computer Science, University of Edinburgh (1992).

[18] Thomsen, B., *Plain CHOCS: A second generation calculus for higher order processes*, Acta Informatica **30** (1993), pp. 1–59.

A A Simple Sorting on 'Homerσ

In this appendix we present a simple sorting to ensure that we only work with a subset of ground bigraphs, that is the bigraphs that are 'correct' with respect to our encoding. The sorting introduces a requirement on the possible nesting of nodes and on how the linkage is performed, particularly that the sets of free names and variables are kept disjoint. We need some nomenclature to differentiate the different kinds of links and ports before stating the definition of the class of bigraphs that we are interested in. We have two kinds of ports: name- and variable-ports.

- The *name-ports* are the port of a **tname** node and all the free ports of a **rece**, **take**, **send**, or a **loca** node.
- The *variable-ports* are the free port of a **def** node or a **var** node or the binding port of a **sub**, **rece**, or a **take** node.

In the same way we define two kinds of links:

- A *name-link* is a link with only name-ports, and if free a name.
- A *variable-link* is a link with only variable-ports connected to it, and if free a variable name.

Definition A.1 (bigraphs good for Homerσ) *We define a sub-class $\mathcal{I}$ of ground bigraphs in 'Homerσ as the bigraphs that satisfy the following requirements*

- *We only allow name- and variable-links as links in the bigraph.*
- *A variable-link can be connected to any number of* **var***-ports.*
  - · *If a variable-link is bound by either a* **rece***-or a* **take***-port, then it contains no* **def***-ports.*

· *If a variable-link is bound by a port on a* **sub***-node v, then it also has one unique* **def***-port, which resides on a child of v, and this is the only location where a* **def** *node can occur.*

- *A name-link can be connected to any number of name-ports.*
- *For every pair of distinct* **tname** *nodes enclosed in the same* **ann** *node their name-ports must be connected to distinct links.*
- *Every* **loca***,* **send***, and* **def** *node must contain an unique* **ann** *child node, and these are the only locations where* **ann** *nodes can occur.*
- *All* **tname** *nodes must be in a* **ann** *node and no other kind of nodes can reside here.*

We have introduced all the abovementioned restrictions to enforce that we only work with bigraphs, that have a structure corresponding to how we interpret Homerσ in bigraphs. In Homerσ the sets of names and variables are by definition disjoint, but since we use the links of bigraphs to encode both sets, we need some additional requirements to enforce the distinction in kinds of links.

The requirements enforce that a **loca** node and a **send** node contains unique **ann** node. We also require that **def** can only appear as a child of a **sub** node. Finally, we require that the **tname** nodes representing a type annotation only occur in a **ann** node and that they are unique, in the sense that they all are linked to different name-links.

Proposition A.2 (invariant) *The class of bigraphs $\mathcal{I}$ is preserved by the reaction relation $\twoheadrightarrow$ defined in Sec. 2.2 and contains all images of the translation given in Def. 2.1.*

B Static Correspondence

In this appendix we prove that two Homerσ-processes are structural congruent if and only if their image under the encoding are isomorphic. We prove each direction separately.

Proposition B.1 $\tilde{x} \vdash p \equiv_\sigma q : \tilde{n}$ *implies* $[\![\tilde{x} \vdash p : \tilde{n}]\!] = [\![\tilde{x} \vdash q : \tilde{n}]\!]$.

Proof Since the translation is compositional we can consider each of the axioms defining $\equiv_\sigma$ separately. We only present some of the cases

- Each of the axioms

  $$\tilde{x} \vdash p \mid \mathbf{0} \equiv_\sigma p : \tilde{n} \quad \tilde{x} \vdash (p \mid p') \mid p'' \equiv_\sigma p \mid (p' \mid p'') : \tilde{n} \quad \tilde{x} \vdash p \mid q \equiv_\sigma q \mid p : \tilde{n}$$

  follows directly from the translation, since we translate parallel composition in Homerσ as the prime product in bigraphs '$|$', which can be shown to be associative and commutative, and as we translate $\mathbf{0}$ into the unit for $|$.

- To prove the case for the axiom for reordering of restrictions

$$\tilde{x} \vdash (n)(m)p \equiv_\sigma (m)(n)p : \tilde{n}$$

  we show that the two bigraphs $[\![\tilde{x} \vdash (n)(m)p : \tilde{n}]\!]$ and $[\![\tilde{x} \vdash (m)(n)p : \tilde{n}]\!]$ can be constructed in the same manner (we assume that m and n are distinct and names of p). We construct $[\![\tilde{x} \vdash p : \tilde{n}nm]\!]$ and add two edges to its link graph e_m and e_n and make all points of m (n) point to e_m (e_n). Finally we remove the names m and n.

- The axiom for scope extension

$$\tilde{x} \vdash (n)p \mid q \equiv_\sigma (n)(p \mid q) : \tilde{n}, \text{ if } n \notin fn(q)$$

  can be proven in the same way. We construct the bigraphs $[\![\tilde{x} \vdash (n)p \mid q : \tilde{n}]\!]$ and $[\![\tilde{x} \vdash (n)(p \mid q) : \tilde{n}]\!]$ in the following way. Without loss of generality we assume that $\tilde{n} = \tilde{n}_1 \cup \tilde{n}_2$, where $\tilde{n}_1 n \tilde{x}$ and $\tilde{n}_2 \tilde{x}$ are the names in the outer face of $[\![\tilde{x} \vdash p : \tilde{n}_1 n]\!]$ and $[\![\tilde{x} \vdash q : \tilde{n}_2]\!]$, respectively. First we build $[\![\tilde{x} \vdash p : \tilde{n}_1 n]\!]$ and $[\![\tilde{x} \vdash q : \tilde{n}_2]\!]$ and combine them using the prime product, then we add one edge e_n to the link graph of this bigraph and make all points of the name n point to e_n. Since $n \notin fn(q)$ we only touch points in $[\![\tilde{x} \vdash p : \tilde{n}_1 n]\!]$. Finally we remove the name n.

- For the remaining cases we proceed in the same manner by exhibiting a constructing that forms both bigraphs.

□

Proposition B.2 *If* $[\![\tilde{x} \vdash p : \tilde{n}]\!] = [\![\tilde{x} \vdash q : \tilde{n}]\!]$ *then* $\tilde{x} \vdash p \equiv_\sigma q : \tilde{n}$.

From Prop. B.1 and Prop. B.2 it follows that two Homerσ-processes are structural congruent if and only if their image under the encoding are isomorphic.

Theorem B.3 (Static correspondence) $\tilde{x} \vdash p \equiv_\sigma q : \tilde{n}$ *if and only if* $[\![\tilde{x} \vdash p : \tilde{n}]\!] = [\![\tilde{x} \vdash q : \tilde{n}]\!]$.

C Mimicking Reactions

In this appendix we present how reactions in Homerσ are mimicked by the encoding as a BRS. We consider the following reactions, where we have omitted the top-level types.

$$\begin{gathered}
\overline{on}\langle r \mid r' \rangle_{\{m\}} \mid o[n(x) \,.\, x]_{\{n\}} \searrow_\sigma \\
o[x[x := (r \mid r') : \{m\}]]_{\{n,m\}} \searrow_\sigma \\
o[(r \mid r')[x := (r \mid r') : \{m\}]]_{\{n,m\}} \searrow_\sigma \\
o[r \mid r']_{\{n,m\}}
\end{gathered}$$

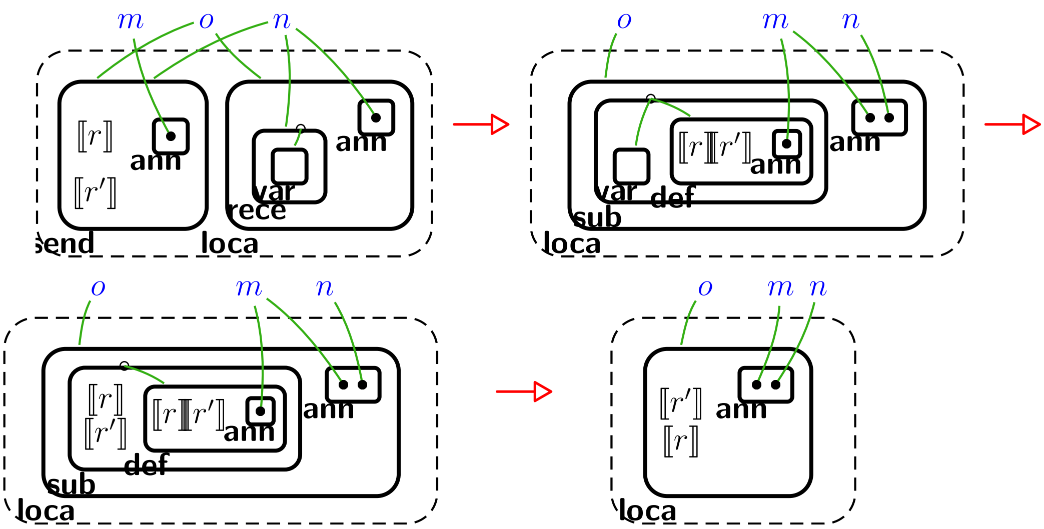


Figure C.1. Mimicking $\overline{on}\langle r \mid r'\rangle_{\{m\}} \mid o[n(x) \,.\, x]_{\{n\}} \searrow_\sigma^* o[r \mid r']_{\{n,m\}}$

using the rules, *send*σ, *apply*σ, and *garbage*σ. In the second line we have the location o containing the process variable x enclosed in an explicit substitution, which can substitute $r \mid r'$ of type $\{m\}$ in for x. In bigraphs we have the matching sequence of reactions depicted in Fig. C.1. Note that we have chosen not to draw the possible free name m of r and r'.

D Operational Correspondence

In this appendix we prove the main theorem of the paper, the operational correspondence between reactions in Homerσ and reactions in its encoding as a BRS Ħomerσ. By inspecting the translation we can easy see that evaluation contexts in Homerσ are translated to active contexts, and conversely if the image under the translation is an active context then the preimage must have been an evaluation context.

We follow the same method as Jensen and Milner by first characterising the reactions in both Homerσ and Ħomerσ by the forms of the expressions involved. Then we use the definition of the translation to connect the characterisations. We only present two of the cases (*garbage*σ) and (*send*σ) the remaining two are similar. Prop. D.1 and Prop. D.2 characterise the reaction relations $\searrow_\sigma$ and $\longrightarrow$ (for the rules (*garbage*σ) and Garbage, respectively) in terms of the form of the processes and bigraphs.

Proposition D.1 *$\vdash p \searrow_\sigma p' : \tilde{n}$ by the rule (garbageσ) if and only if p and p' are of the forms*

$$\vdash p \;\equiv_\sigma\; \mathcal{E}(q[x := r : \tilde{n}']) : \tilde{n}$$

$$\vdash p' \equiv_\sigma\; \mathcal{E}(q) : \tilde{n} \;\;,$$

if $x \notin fn(q)$ and for an evaluation context $\mathcal{E}$.

Proposition D.2 *$g \rightharpoondown g'$ by the rule Garbage if and only if g and g' are of the forms*

$$\begin{aligned} g &= E \circ \big((\mathsf{sub}_{(x)} \mathbin{\overline{\oplus}} \mathsf{id}_{\tilde{n}})\big(h \mid (\mathsf{def}_x \mathbin{\overline{\oplus}} \mathsf{id}_{\tilde{n}'})h'\big)\big) \\ g' &= E \circ h \ , \end{aligned}$$

if the outer face of h is $\tilde{n}$ and E is an active context.

Since the outer face of h is $\tilde{n}$, it means that h cannot be connected to the binder x in the surrounding **sub** control.

Lemma D.3 (operational correspondence on (*garbage*σ) and Garbage) *$\vdash p \searrow_\sigma p' : \tilde{n}$ by the rule (garbageσ) if and only if $[\![\vdash p : \tilde{n}]\!] \rightharpoondown [\![\vdash p' : \tilde{n}]\!]$ by the rule Garbage.*

Proof From Prop. D.1 we know that $\vdash p \searrow_\sigma p' : \tilde{n}$ if and only if p and p' have the forms

$$\begin{aligned} \vdash p &\equiv_\sigma \ \mathcal{E}(q[x := r : \tilde{n}']) : \tilde{n} \\ \vdash p' &\equiv_\sigma \ \mathcal{E}(q) : \tilde{n} \ , \end{aligned}$$

and $x \notin fn(q)$ and from α-conversion we can assume that all bound names are distinct and disjoint from the free names, and without loss of generality that the hole of $\mathcal{E}$ is annotated with the type $\tilde{n}''$. From the correspondence between structural congruence and graph isomorphism we have

$$\begin{aligned} [\![\vdash p : \tilde{n}]\!] &= [\![\vdash \mathcal{E} : \tilde{n}]\!] \circ ([\![\vdash q[x := r : \tilde{n}'] : \tilde{n}'']\!]) \\ &= [\![\vdash \mathcal{E} : \tilde{n}]\!] \circ ((\mathsf{sub}_{(x)} \mathbin{\overline{\oplus}} \mathsf{id}_{\tilde{n}''})([\![\vdash q : \tilde{n}'']\!] \mid (\mathsf{def}_x \mathbin{\overline{\oplus}} \mathsf{id}_{\tilde{n}'})h')) \\ [\![\vdash p' : \tilde{n}]\!] &= [\![\vdash \mathcal{E} : \tilde{n}]\!] \circ ([\![\vdash q : \tilde{n}'']\!]) \ , \end{aligned}$$

since $x \notin fn(q)$ and letting $h' = [\![\vdash r : \tilde{n}']\!] \mid (\mathsf{ann} \mathbin{\overline{\oplus}} \mathsf{id}_{\tilde{n}'})[\![\tilde{n}']\!]$. By Prop. D.2 this holds if and only if $[\![\vdash p : \tilde{n}]\!] \rightharpoondown [\![\vdash p' : \tilde{n}]\!]$. $\square$

We proceed in the same manner with the case for (*send*σ). Prop. D.4 and Prop. D.5 characterise the reaction relations $\searrow_\sigma$ and $\rightharpoondown$ (for the rules (*send*σ) and Send, respectively) in terms of the form of the processes and bigraphs.

Proposition D.4 *$\vdash p \searrow_\sigma \ \vdash p' : \tilde{n}$ by the rule (sendσ) if and only if p and p' are of the forms*

$$\begin{aligned} \vdash p &\equiv_\sigma \ \mathcal{E}(\overline{\gamma\delta}\langle r\rangle_{\tilde{n}'} \mid \mathcal{C}_\gamma^{\tilde{m}}(\delta(x) \,.\, q, \overrightarrow{q})) : \tilde{n} \\ \vdash p' &\equiv_\sigma \ \mathcal{E}(\tilde{n}' \odot \mathcal{C}_\gamma^{\tilde{m}}(q[x := r : \tilde{n}'], \overrightarrow{q})) : \tilde{n} \ , \end{aligned}$$

if $\tilde{m} \cap (\delta \cup \tilde{n}') = \emptyset$ and for an evaluation context $\mathcal{E}$.

Proposition D.5 *$g \rightharpoondown g'$ by the rule Send if and only if g and g' are of the* *forms*

$$g = E \circ ((\textsf{send}_{\gamma\delta} \mathbin{\overline{\oplus}} \textsf{id}_{\tilde{n}'})\big(h \mid (\textsf{ann} \mathbin{\overline{\oplus}} \textsf{id}_{\tilde{n}'})h'\big) \mid F_\gamma \circ ((\textsf{rece}_{\delta(x)} \mathbin{\overline{\oplus}} \textsf{id}_{\tilde{n}'''})h'''))$$
$$g' = E \circ ((\tilde{n}' \odot_b F_\gamma) \circ (\textsf{sub}_{(x)} \mathbin{\overline{\oplus}} \textsf{id}_{\tilde{n}''' \cup \tilde{n}'})(h''' \mid (\textsf{def}_x \mathbin{\overline{\oplus}} \textsf{id}_{\tilde{n}'})(h \mid (\textsf{ann} \mathbin{\overline{\oplus}} \textsf{id}_{\tilde{n}'})h'))) \enspace ,$$

if the outer face of h and h' are $\tilde{n}'$, of h''' is $\tilde{n}'''x$, E is an active context with inner face $\tilde{n}''$, and F_γ is a path bigraph with inner face $\tilde{n}'''$.

Note that we leave the last k holes in the $k+1$-hole path contexts F_γ unspecified.

Lemma D.6 (operational correspondence on (*send*σ) and Send) *$\vdash p \searrow_\sigma p' : \tilde{n}$ by the rule (sendσ) if and only if $[\![\vdash p : \tilde{n}]\!] \twoheadrightarrow [\![\vdash p' : \tilde{n}]\!]$ by the rule Send.*

Proof From Prop. D.4 we know that $\vdash p \searrow_\sigma p' : \tilde{n}$ if and only if p and p' have the forms

$$\vdash p \equiv_\sigma \mathcal{E}(\overline{\gamma\delta}\langle r\rangle_{\tilde{n}'} \mid \mathcal{C}^{\tilde{m}}_\gamma(\delta(x)\,.\,q, \overrightarrow{q})) : \tilde{n}$$
$$\vdash p' \equiv_\sigma \mathcal{E}(\tilde{n}' \odot \mathcal{C}^{\tilde{m}}_\gamma(q[x := r : \tilde{n}'], \overrightarrow{q})) : \tilde{n} \enspace ,$$

if $\tilde{m} \cap (\delta \cup \tilde{n}') = \emptyset$ and from α-conversion we can assume that all bound names are distinct and disjoint from the free names, and without loss of generality that the hole of $\mathcal{E}$ is annotated with $\tilde{n}''$. From the correspondence between structural congruence and graph isomorphism we have

$$\begin{aligned}
[\![\vdash p : \tilde{n}]\!] &= [\![\vdash \mathcal{E} : \tilde{n}]\!] \circ ([\![\vdash \overline{\gamma\delta}\langle r\rangle_{\tilde{n}'} \mid \mathcal{C}^{\tilde{m}}_\gamma(\delta(x)\,.\,q, \overrightarrow{q}) : \tilde{n}'']\!]) \\
&= [\![\vdash \mathcal{E} : \tilde{n}]\!] \circ ((\textsf{send}_{\gamma\delta} \mathbin{\overline{\oplus}} \textsf{id}_{\tilde{n}'})([\![\vdash r : \tilde{n}']\!] \mid ((\textsf{ann} \mathbin{\overline{\oplus}} \textsf{id}_{\tilde{n}'})[\![\tilde{n}']\!])) \mid \\
&\qquad [\![\vdash \mathcal{C}^{\tilde{m}}_\gamma : \tilde{n}'']\!] \circ (\textsf{rece}_{\delta(x)} \mathbin{\overline{\oplus}} \textsf{id}_{\tilde{n}'''})[\![x \vdash q : \tilde{n}''']\!]) \\
[\![\vdash p' : \tilde{n}]\!] &= [\![\vdash \mathcal{E} : \tilde{n}]\!] \circ ([\![\vdash \tilde{n}' \odot \mathcal{C}^{\tilde{m}}_\gamma(q[x := r : \tilde{n}'], \overrightarrow{q}) : \tilde{n}'']\!]) \\
&= [\![\vdash \mathcal{E} : \tilde{n}]\!] \circ ((\tilde{n}' \odot_b [\![\vdash \mathcal{C}^{\tilde{m}}_\gamma]\!]) \circ (\textsf{sub}_{(x)} \mathbin{\overline{\oplus}} \textsf{id}_{\tilde{n}''' \cup \tilde{n}'}) \\
&\qquad ([\![x \vdash q : \tilde{n}''']\!] \mid (\textsf{def}_x \mathbin{\overline{\oplus}} \textsf{id}_{\tilde{n}'})([\![\vdash r : \tilde{n}']\!] \mid (\textsf{ann} \mathbin{\overline{\oplus}} \textsf{id}_{\tilde{n}'})[\![\tilde{n}']\!])))
\end{aligned}$$

By Prop. D.5 this holds if and only if $[\![\vdash p : \tilde{n}]\!] \twoheadrightarrow [\![\vdash p' : \tilde{n}]\!]$. □

Theorem D.7 (Operational correspondence) *For every well-typed process $\vdash p : \tilde{n}$, we have*

$$\vdash p \searrow_\sigma p' : \tilde{n} \text{ if and only if } [\![\vdash p : \tilde{n}]\!] \twoheadrightarrow [\![\vdash p' : \tilde{n}]\!] \enspace .$$