A Bigraph Reactive Systems Realtion Model

Maxime Beauquier
Carsten Schürmann

IT University Technical Report Series **TR-2010-TR 2010-126**

ISSN 1600–6100 **June 2010**

ISSN 1600–6100

ISBN 9788779492158

Copies may be obtained by contacting:

IT University of Copenhagen
Rued Langgaards Vej 7
DK-2300 Copenhagen S
Denmark

Telephone:	**+45 72 18 50 00**
Telefax:	**+45 72 18 50 01**
Web	**`www.itu.dk`**

Abstract. In this paper, we present a model based on relations for *bigraphical reactive systems* [6]. Its defining characteristics are that validity and reaction relations are captured as traces in a multi-set rewriting system. The relational model is derived from Milner's graphical definition and directly amenable to implementation.

Introduction

Milner's bigraphical reactive systems [6], or BRSs in short, are formulated in terms of category theory. They encompass earlier models, such as CCS [4], the π-calculus [5], and Petri nets [7]. However, as with other categorical models, it is not immediately clear how to implement a logical framework that could check, for example, the well-formedness or correctness of a BRS, or just execute reaction rules. On the other hand there are mature implementations of logical frameworks, e.g. Celf [8] that already provide many of the algorithms that one would need for such an implementation. In particular, Celf provides support for linearity and concurrency using a kind of structural congruence that arises naturally from the definition of equivalence in the type theory CLF [2]. In this paper we show that the two are deeply connected. In particular, we formulate a *bigraph relational model* for BRSs and demonstrate how to piggy-bag on Celf's implementation by reusing algorithms, such as unification, type checking, type inference, logic programming, and multi-set rewriting.

A BRS consists of a bigraph and a set of reaction rules. The bigraph consists of a *place graph*, that usually models the hierarchical (physical) structure of the concurrent system to be modeled, and the *link graph* that establishes the communication structure between the different places. By the virtue of this definition alone, a bigraph does not have any dynamic properties. It is best understood as a snapshot of a concurrent system at a particular point of time.

What makes a bigraph reactive is the accompanying set of reaction rules. A reaction rule can be thought of as a rewrite rule, except that the left and the right hand side are graphs rather than terms. Consequently, matching the left hand side of a reaction rule with a subgraph of the current bigraph is conceptually and computationally not as straightforward as for example first-order unification.

As an alternative, we relate bigraphical reactive systems to something that we understand well: unification modulo structural congruence in the setting of CLF [9]. CLF is a type theory that conservatively extends the λ-calculus and serves to model truly concurrent systems. CLF follows the standard judgements-as-types encoding paradigm, which means that derivations of the validity of a BRS, traces of the operational semantics, etc. are encoded as CLF objects of the corresponding CLF type. The extensions include, for example, type families that are indexed by objects, a dependent type constructor $\Pi x : A. B(x)$ as a generalisation of the usual function type constructor, linear type constructors, for example $A \multimap B$, that capture the nature of resource consumption, and also a monadic type constructor $\{A\}$. This type is inhabited by objects of type A, such that two objects are considered (structurally) congruent if and only if they differ only in the order in which subterms are evaluated.

The Celf [8] system is an implementation of CLF type theory that provides a concept of logic variables that are logically well understood in terms of linear contextual modal type theory and a rich set of algorithms, including a sophisticated unification algorithm, type inference, type checking, and a logical programming language that supports both backward chaining proof search and forward chaining multi-set rewriting.

The main contribution of this paper is a model for BRS that we will call *bigraph relational model* that follows closely the graphical interpretation of the categorical model proposed by Milner [6]. The key idea is to encode the validity relation of a bigraph by a set of rewrite rules. A specific bigraph is valid if and only if a multi-set of assumptions (representing the structure the bigraph) can be rewritten to the empty multi-set. We also give a direct and elegant encoding of reaction rules in CLF. Furthermore, we prove that Celf's operational interpretation of these rules is *adequate* in the sense that it coincides with the intended meaning of those rules. As a consequence, Celf's multi-set rewriting engine implements the reactive behaviour of a BRS.

The running example of this paper is CCS without replication and naming. For illustrative purposes, we sketch the corresponding bigraph relational model and give the implementation in Celf. The source code is available from `www.itu.dk/∼beauquie/brs`.

The remainder of this paper is structured as follow: in Section 1 we reiterate the formal definition of the bigraph structure. In Section 2 we define reaction rules and therefore BRS. Then, we define the bigraph relational model in Section 3, and show that the structures properties of bigraphs are respected. Furthermore we define encoding and interpretation functions, and show that the bigraph relational model is adequate. We define the encoding of the reaction rules and prove adequacy in Section 4. Finally, we show the implementation in Celf in Section 5 before we conclude and assess results in Section 5.

1 Bigraphs

The definition used in this paper can be found in [6]. A bigraph consists of a set of nodes. Each node is characterised by a type, which we call *control*. Each *control* is defined by a number of *ports*. We write K for the set of *controls*, and $arity : K \to \mathbb{N}$ a map from controls to the number of ports. Together they form what we call the *signature* $\Sigma = (K, arity)$ of a bigraph. The roots of the place graph and the link graph, are called *roots* and *outer names*, respectively. They form the *outer interface* of the bigraph. The leafs of the place graph and the link graph, are called *sites* and *inner names*, respectively. They form the *inner interface*. A *site* should be thought of as a hole, which can be filled and properly connected with another bigraph. More formally, we define:

Definition 1 (Bigraph). *A bigraph B under a signature Σ is defined as*

$$B = (V_B, E_B, P_B, ctrl_B, prnt_B, link_B) : \langle m, X \rangle \to \langle n, Y \rangle,$$

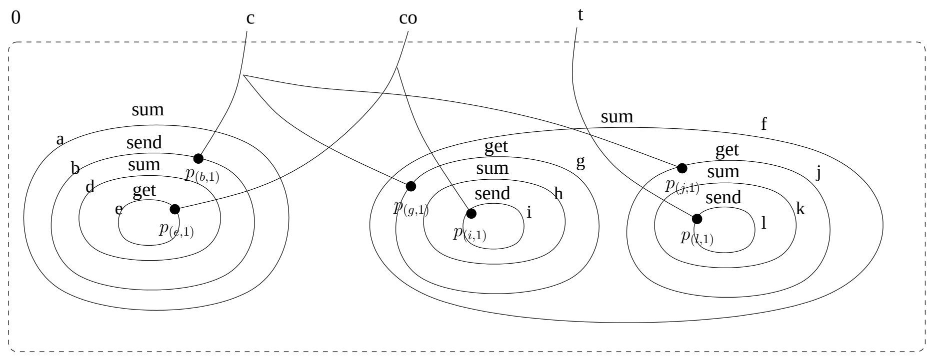

Fig. 1. Vending machine example.

where m and n are finite ordinals that respectively refer to the sites *and the* roots *of the bigraph structure. X (resp. Y, V_B, E_B) refers to a finite set of* inner names *(resp.* outer names, nodes, edges*). P_B represents a set of* ports*. It is defined as*

$$P_B = \{(v, i) | i \in arity(ctrl_B\ v)\}$$

$ctrl_B : V_B \rightarrow K$ assigns controls to nodes. The place graph *establishes a tree shaped parent ordering among all nodes and is defined by $prnt_B : V_B \cup m \rightarrow V_B \cup n$. The relation $link_B : X \cup P_B \rightarrow E_B \cup Y$ maps the union of inner names and ports to the union of edges and outer names, which represents the hypergraph called the* link graph*.*

In this definition, m, n, X, Y, V_B, and E_B are all assumed to be disjoint. The functions arity, $ctrl_B$, $prnt_B$, and $link_B$ are assumed to be total.

$\langle m, X\rangle$ and $\langle n, Y\rangle$ are called the interfaces *of the bigraph, whereby the former is also referred to as the* inner interface *and the latter the* outer interface*.*

Definition 2 (Ground Bigraph). *A bigraph is* ground*, if its inner interface is empty: $\langle 0, \emptyset\rangle$, also written ϵ.*

As a running example we use CCS without replication (!) and new name (ν) used to model a vending machine: $\overline{c}.co \mid c.\overline{co} + c.\overline{t}$. Here c represents a coin, co a cup of coffee, and t a cup of tea. The corresponding bigraph is presented in Figure 1, where

$\Sigma = (\{\texttt{get}, \texttt{send}, \texttt{sum}\}, \{(\texttt{get}, 1), (\texttt{send}, 1), (\texttt{sum}, 0)\}),$
$V_B = \{a, b, d, e, f, g, h, i, j, l\}, E_B = \emptyset,$
$P_B = \{p_{(b,1)}, p_{(e,1)}, p_{(g,1)}, p_{(i,1)}, p_{(j,1)}, p_{(l,1)}\},$
$ctrl_B = \{(a, \texttt{sum}), (b, \texttt{send}), (d, \texttt{sum}), (e, \texttt{get}), (f, \texttt{sum}), (g, \texttt{get}),$
$(h, \texttt{sum}), (i, \texttt{send}), (j, \texttt{get}), (k, \texttt{sum}), (l, \texttt{send})\},$
$prnt_B = \{(a, 0), (f, 0), (b, a), (d, b), (e, d), (g, f), (h, g), (i, h), (j, f), (k, j), (l, k)\},$
$link_B = \{(p_{(b,1)}, c), (p_{(e,1)}, co), (p_{(g,1)}, c), (p_{(i,1)}, co), (p_{(j,1)}, c), (p_{(l,1)}, t),$
$m = 0, X = \emptyset, n = 1,$ and, $Y = \{c, co, t\}$

Bigraphs are closed by composition and juxtaposition, which are defined as follows:

Definition 3 (Composition). *Let $F : \langle k, X\rangle \to \langle m, Y\rangle$ and $G : \langle m, Y\rangle \to \langle n, Z\rangle$ be two bigraphs under the same signature Σ with disjoint sets of nodes and edges. The composition $G \circ F$ is defined as:*

$$G \circ F = (V, E, P, ctrl, prnt, link) : \langle k, X\rangle \to \langle n, Z\rangle$$

where: $V = V_G \uplus V_F$, $E = E_G \uplus E_F$, $ctrl = ctrl_G \uplus ctrl_F$

$$prnt\ x = \begin{cases} prnt_F\ x & \text{if } x \in k \uplus V_F \text{ and } prnt_F\ x \in V_F \\ prnt_G\ j & \text{if } x \in k \uplus V_F \text{ and } prnt_F\ x = j \in m \\ prnt_G\ x & \text{if } x \in V_G \end{cases}$$

and

$$link\ x = \begin{cases} link_F\ x & \text{if } x \in X \uplus P_F \text{ and } link_F\ x \in E_F \\ link_G\ y & \text{if } x \in X \uplus P_F \text{ and } link_F\ x = y \in Y \\ link_G\ x & \text{if } x \in P_G \end{cases}$$

Intuitively, $prnt_{(G\circ F)}$ is defined as the union of $prnt_G$ and $prnt_F$ where each root r of F and each site s of G such that $s = r$ satisfies the following: If $prnt_G\ s = y$ and $prnt_F\ x = r$ then $prnt_{(G\circ F)}\ x = y$ for all x, y. The definition of $link_{(G\circ F)}$ is defined analogously.

Definition 4 (Juxtaposition). *Let $F = (V_F, E_F, P_F, ctrl_F, prnt_F, link_F) : \langle k, X\rangle \to \langle m, Y\rangle$ and $G = (V_G, E_G, P_G, ctrl_G, prnt_G, link_G) : \langle l, W\rangle \to \langle n, Z\rangle$ be two bigraphs under the same signature Σ that as above have disjoint nodes and edges. The* juxtaposed *bigraph $G \otimes F$ is defined as follows.*

$$\begin{aligned} G \otimes F =& (V_F \uplus V_G, E_F \uplus E_G, P_F \uplus P_G, ctrl_F \uplus ctrl_G, prnt_F \uplus prnt'_G, \\ & link_F \uplus link_G) : \langle k + l, X \uplus W\rangle \to \langle m + n, X \uplus Z\rangle, \end{aligned}$$

where $prnt'_G\ (k + i) = m + j$ whenever $prnt_G\ i = j$.

The composition of two bigraphs can be seen as plugging one bigraph structure into the other. The juxtaposition on the other hand can be seen as putting two disjoint bigraphs next to each other.

2 Bigraphical Reactive System

A BRS consists of a ground bigraph, which is also called the *agent* and a set of reaction rules. In this paper, we discuss two kinds of reaction rules, those that disallow the matching of sites (which are said to be *ground*) and those that do not (which are said to be *parametric*).

Definition 5 (Ground Reaction Rule). *A ground rewriting rule consists of two ground bigraphs, the redex and the reactum, with the same interfaces* ($L : \epsilon \to J,\ R : \epsilon \to J$).

When we *apply* a ground reaction rule to an agent B, we require that it can be decomposed into $B \equiv C \circ L$. The result of the application is an agent $B' \equiv C \circ R$, which is justified because L and R have the same interface.

Parametric reaction rules differ from ground reaction rules by allowing L and R to contain sites, which may move from one to another node, be copied, or simply deleted. To this effect parametric reaction rules define a relation η between sites of the reactum and sites of the redex. Note the particular direction of η and the link graph inner interfaces of the redex and the reactum are empty.

Definition 6 (Parametric Reaction Rule). *A parametric reaction rule is a triple of two bigraphs and a total function from R's sites to L's sites η*

$$(L : \langle m, X\rangle \to J,\ R : \langle m', X\rangle \to J,\ \eta : m' \to m)$$

The semantics of parametric BRS is also based on decomposition of the agent, where an instantiation function is deduced from η to compute the new tails of the bigraph structure. This function is defined up to an equivalence. The $\simeq$-equivalence is defined as follows:

Definition 7 (Equivalence). *Let B and G be two bigraphs with the same interface $\langle k, X\rangle \to \langle m, Y\rangle$. B and G are called $\simeq$-equivalent (*lean* equivalent) if there exist two bijections $\rho_V : V_B \to V_G$ and $\rho_E : E_B \to E_G$ that respect the structure, in the following sense:*

- *ρ preserve controls, i.e. $ctrl_G \circ \rho_V = ctrl_B$, and therefore induces a bijection $\rho_P : P_F \to P_G$ defined as $\rho_P\ (v, i) = ((\rho_V\ v), i)$.*
- *ρ commutes with the structural maps as follow:*

$$prnt_G \circ (Id_m \uplus \rho_V) = (Id_n \uplus \rho_V) \circ prnt_B$$
$$link_G \circ (Id_X \uplus \rho_P) = (Id_Y \uplus \rho_E) \circ link_B.$$

Definition 8 (Instantiation). *Let $F \equiv G \circ (d_0 \otimes \cdots \otimes d_{m-1}) : \langle n, Y\rangle \to \langle m, X\rangle$ be a bigraph where G is a link graph, the d's have no inner names and a unary outer face and $\eta : m' \to m$ the relation on sites from Definition 6. We refer to an instantiation of η on F as $\overline{\eta}$ that is defined as follows:*

$$\overline{\eta}\ F = G \circ (d'_0 \| \ldots \| d'_{m'-1})$$

where $\forall j \in m', d'_j \simeq d_{(\eta\ j)}$. Following [6] we write $d \| d'$ for $d \otimes d'$ where d and d' can share edges and outer names.

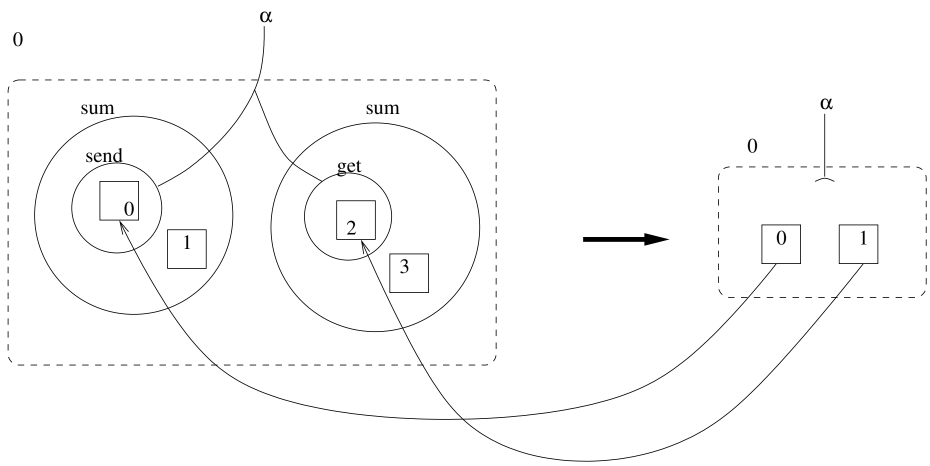

Fig. 2. The bigraph encoding for the CCS τ-transition rule.

The τ-reaction rule of CCS, $(\alpha.P + P') \mid (\overline{\alpha}.Q + Q') \to^{\tau} Q \mid P$ is encoded as parametric reaction rule presented in figure 2, in this case $\eta = \{(0,0),(1,2)\}$.

Let $(B, \mathcal{R})$ be a BRS, and $(L, R, \eta) \in \mathcal{R}$ the parameterised reaction rule. If $B \equiv C \circ (D \otimes L) \circ C'$ then we can apply the reaction rule $\mathcal{R}$ and rewrite the bigraph B into $B' \equiv C \circ (D \otimes R) \circ (\overline{\eta}\, C')$.

Finally, it is possible to derive ground reaction rules as ground instances of parametric reaction rules.

It has been shown in [6] that a rewriting step in a BRS $(B, \mathcal{R})$ with a parametric rewrite rule corresponds a rewriting step in the ground bigraph reactive system $(B, \mathcal{R}')$ where $\mathcal{R}'$ is the set of reaction rules obtained as ground instances of $\mathcal{R}$ and B.

3 Bigraph Relational Model

Next, we tackle the definition of the bigraph relational model that arises form the graphical presentation of the categorical model. In our model, bigraphs are identified by name. The disjoint sets of names n, V_B, m, P_B, Y, X, E_B, K are expressed as *base* relation symbols `bigraph`, `root`, `node`, `site`, `port`, `o_name`, `i_name`, `e_name`, `control`. The *arity* function is encoded as `arity`, a ternary relation symbol indexed by `control`, a natural number, and a `bigraph`. And similarly, relations $prnt_B$, $link_B$, $ctrl_B$, and P_B are encoded by the following *operational* relation symbols:

- `prnt S D B` where `S` $\in$ `node` $\cup$ `site` and `D` $\in$ `node` $\cup$ `root`,
- `link S D B` where `S` $\in$ `i_name` $\cup$ `port` and `D` $\in$ `o_name` $\cup$ `e_name`,
- `lc A K B` where `A` $\in$ `node` and `K` $\in$ `control`,
- `lp P A B` where `P` $\in$ `port` and `A` $\in$ `node`,

base cases

$\{$`is_root R, has_child_p (dst_r R) z`$\} \uplus \Delta \mapsto \Delta$ (dr)

$\{$`is_o_name O, has_child_l (dst_o O) z`$\} \uplus \Delta \mapsto \Delta$ (do)

$\{$`is_e_name E, has_child_l (dst_e E) z`$\} \uplus \Delta, \mapsto \Delta$ (de)

recursive cases

$\{$`is_port P, lp P A, vp A (s z), link (src_p P) D,` (lgpsz)
`has_child_l D (s N)`$\} \uplus \Delta \mapsto \{$`has_child_l D N`$\} \uplus \Delta$

$\{$`is_i_name I, link (src_i I) D, has_child_l D (s N)`$\}$ (lgi)
$\uplus \Delta \mapsto \{$`has_child_l D N`$\} \uplus \Delta$

$\{$`is_site S, prnt (src_s S) D, has_child_p D (s N)`$\} \uplus \Delta$ (lgs)
$\mapsto \{$`has_child_p D N`$\} \uplus \Delta$

$\{$`is_node A, has_child_p (dst_n A) z, prnt (src_n A) D,` (pgnz)
`has_child_p D (s N), lc A K`$\} \uplus \Delta \mapsto \{$`has_child_p D N`$\} \uplus \Delta$ if $arity$ `K` $= 0$

$\{$`is_node A, has_child_p (dst_n A) z, prnt (src_n A) D,` (pgns)
`has_child_p D (s N), lc A K`$\} \uplus \Delta$
$\mapsto \{$`has_child_p D N, vp A N'`$\} \uplus \Delta$ if $arity$ `K` $= N' > 0$

$\{$`is_port P, lp P A, vp A (s (s N')),` (lgps)
`link (src_p P) D, has_child_l D (s N)`$\} \uplus \Delta$
$\mapsto \{$`vp A (s N'), has_child_l D N`$\} \uplus \Delta$

Fig. 3. Bigraph Validity

We declare the following *structural* relation symbols as well:

- `is_root R B` where `R` $\in$ `root` and `R` belongs to `B`,
- `is_node N B` where `N` $\in$ `node` and `N` belongs to `B`,
- `is_site S B` where `S` $\in$ `site` and `S` belongs to `B`,
- `is_port P B` where `P` $\in$ `port` and `P` belongs to `B`,
- `is_o_name O B` where `O` $\in$ `o_name` and `O` belongs to `B`,
- `is_i_name I B` where `I` $\in$ `i_name` and `I` belongs to `B`,
- `is_e_name E B` where `E` $\in$ `e_name` and `E` belongs to `B`.

Next, we sketch an algorithm for deciding the validity of a bigraph in the relational model. The algorithm is deceptively simple: using the rules depicted in Figure 3, we rewrite the encoding of a bigraph to the empty set by checking the validity of the place graph and the link graph and the valid use of control's arity. As we will show below, the algorithm is confluent and strongly normalising. The rules are partially sequentialised in such a way that children are rewritten before the parents and nodes before ports. We make this information explicit and define three more operational relational symbols one for the place graph, an other one for the link graph, and one for the control's arity.

$$
\begin{aligned}
[\![B]\!] = & \biguplus_{i=0}^{|V_B|-1} \big(\texttt{is\_node}\ a_i \uplus \texttt{lc}\ a_i\ (ctrl_B\ a_i) \uplus \texttt{prnt}\ a_i\ (prnt_B\ a_i) \\
& \qquad \uplus \texttt{has\_child\_p}\ a_i\ |\{y \mid prnt_B\ y = a_i\}|\big) \\
& \biguplus_{i=0}^{|E_B|-1} \big(\texttt{is\_e\_name}\ b_i\big) \uplus \biguplus_{i=0}^{|Y|-1} \big(\texttt{is\_o\_name}\ g_i\big) \\
& \biguplus_{i=0}^{|P_B|-1} \big(\texttt{is\_port}\ c_i \uplus \texttt{lp}\ c_i\ (\pi_1\ c_i) \\
& \qquad \uplus \texttt{link}\ c_i\ (link_B\ c_i)\big) \\
& \biguplus_{i=0}^{m-1} \big(\texttt{is\_site}\ d_i \uplus \texttt{prnt}\ d_i\ (prnt_B\ d_i)\big) \\
& \biguplus_{i=0}^{|X|-1} \big(\texttt{is\_i\_name}\ e_i \uplus \texttt{prnt}\ e_i\ (prnt_B\ e_i)\big) \\
& \biguplus_{i=0}^{n-1} \big(\texttt{has\_child\_p}\ f_i\ |\{y \mid prnt_B\ y = f_i\}| \\
& \qquad \uplus \texttt{is\_root}\ f_i\big)
\end{aligned}
$$

Fig. 4. Encoding function from a bigraph structure.

- `has_child_p D N B` where `D` $\in$ `node` $\cup$ `root`, `N` a natural number, and `B` a bigraph name.

- `has_child_l D N B` where `D` $\in$ `e_name` $\cup$ `o_name`, `N` a natural number, and `B` a bigraph name.

- `vp A N B` where `A` $\in$ `node`, `N` a natural number, and `B` a bigraph name.

The encoding of bigraph B is now straightforward. It is defined as the multiset $S = [\![B]\!]$ in Figure 4.

Example 1. The bigraph B from Figure 1, is represented as follows:

$[\![B]\!] = \{$is_root 0 B, has_child_p 0 (s (s z)) B
is_node a B, lc a get B, has_child_p a (s z) B,
is_node b B, lc b send B, has_child_p b (sz) B,
is_node d B, lc d sum B, has_child_p d (s z) B,
is_node e B, lc e get B, has_chidl_p e z B,
is_node f B, lc f sum B, has_child_p f (s (s z)) B,
is_node g B, lc g get B, has_child_p g (s z),
is_node h B, lc h sum B, has_child_p h (s z) B,
is_node i B, lc i send B, has_child_i g z,
is_node j B, lc j get B, has_child_p d (s z) B,
is_node k B, lc k sum B, has_chidl_p k z B,
is_node l B, lc l send B, has_child_p l z B,
is_port $p_{(b,1)}$ B, lp $p_{(b,1)}$ b B, is_port $p_{(e,'1)}$ B, lp $p_{(e,1)}$ e B,
is_port $p_{(g,1)}$ B, lp $p_{(g,1)}$ g B, is_port $p_{(i,1)}$ B, lp $p_{(i,1)}$ i B,
is_port $p_{(j,1)}$ B, lp $p_{(j,1)}$ j B, is_port $p_{(l,1)}$ B, lp $p_{(l,1)}$ l B,
is_o_name c B, has_child_l c (s (s (s z))) B,
is_o_name co B, has_child_l co (s (s z)) B,
is_o_name t B, has_child_l t (s z) B,
prnt a 0 B, prnt f 0 B, prnt b a B, prnt d b B, prnt e d B, prnt g f B,
prnt h g B, prnt i h B, prnt j f B,prnt k j B, prnt l k B,
link $p_{(b,1)}$ c B, link $p_{(e,1)}$ c B, link $p_{(g,1)}$ c B,
link $p_{(i,1)}$ co B, link $p_{(j,1)}$ co B, link $p_{(l,1)}$ t B$\}$

For reasons of convenience, we omit bigraph names from relations, and we use uppercase characters for logic variables in rewrite rules. Furthermore we use lower case z and s for zero and successor.

We show that the rewriting system is strongly normalising and implements a decision procedure for checking the validity of bigraphs.

Lemma 1 (SN). *This multi-set rewriting system is strongly normalising for any finite multi-set S.*

Proof. By a trivial induction of the size of the set S.

In the following we write $S \longrightarrow S'$ for transitive closure of $\mapsto$ from Figure 3. We say S is *valid* if and only if $S \longrightarrow \emptyset$, S contains only unique elements and operational symbols has_child_p, has_child_l and vp are unique on their first argument, respectively node $\cup$ root, o_name $\cup$ e_name and port.

Lemma 2. *Let S be valid. Then the relations in S defined by the operational symbols are total, acyclic, and single valued.*

Proof. Every operational symbols are consumed by the rewriting rules with their structural symbols associated, which also defined their domain. Therefore operational relations are total and since every structural element is unique they are also single valued relations.

By typing, `link`, `lc` and `lp` are acyclic but `prnt`.

Proof by contradiction, suppose that `prnt` represent an acyclic relation, then there exists $x_0, \ldots, x_{(n-1)} \in \texttt{node}$ such that $\forall j < n, \texttt{prnt}\ x_j\ x_{(j+1)} \in S$ and $\texttt{prnt}\ x_{(n-1)}\ x_0$. Only rules and are able to consume a `prnt` symbol, therefore $\texttt{prnt}\ x_i\ x_{(i+1)}$ and $\texttt{prnt}\ x_{(n-1)}\ x_0$ are consumed with the `has_child_p` symbol for x_i and $x_{(n-1)}$, and the `has_child_p` symbol for $x_{(i+1)}$ and x_0 are decremented. In particular, $\texttt{prnt}\ x_0\ x_1$ is consumed with $\texttt{has\_child\_p}\ x_0\ \texttt{z}$ and $\texttt{has\_child\_p}\ x_{(n-1)}\ \texttt{s}n$ is decremented and $\texttt{prnt}\ x_{(n-1)}\ x_0$ is consumed with $\texttt{has\_child\_p}\ x_{(n-1)}\ x_0$ and $\texttt{has\_child\_p}\ x_{(n-1)}\ \texttt{s}n'$ is decremented. Therefore, the `has_child_p` symbols of x_0 and $x_{(n-1)}$ have both to be decremented and consume, which leads to a contradiction with the hypothesis S is valid: either $S \not\longrightarrow \emptyset$ or S do not have at most one `has_child_p` symbol for each different node.

Lemma 3. *Let S be valid.*

- $\forall y \in \texttt{node} \cup \texttt{root}$*,* $\forall N$ *a natural number,* $\texttt{has\_child\_p y N} \in S$ *implies* $|\{x \mid \texttt{prnt}\ x\ y \in S\}| = N$

- $\forall y \in \texttt{edge} \cup \texttt{o\_name}$*,* $\forall N$ *a natural number,* $\texttt{has\_child\_l y N} \in S$ *implies* $|\{x \mid \texttt{link}\ x\ y \in S\}| = N$

- $\forall v \in \texttt{node}$*,* $\forall N$ *a natural number,* $\texttt{vp v N} \in S$ *implies* $|\{x \mid \texttt{lp x v} \in S\}| = N$*.*

Proof.
- `has_child` symbol: let n, the natural number in argument of one of these symbols. Let x be something involve in a `has_child` symbol, by induction on n.
  - $n = 0$, if there is an other `prnt` where x is in the parent position, then it will broke the hypothesis.
  - $n = n' + 1$, then there exists a set S' such that $S \longrightarrow S' \longrightarrow \emptyset$ where a `prnt` symbol with x as parent is consumed and the `has_child` symbol of x is decremented from n to n'. Since the rewriting rules does not break the unicity and $S' \longrightarrow \emptyset$, by the induction hypothesis, n' is the number of `prnt` symbols that involve x as a parent in S'.
- `vp` symbol: let v a `node`, p a `port` and n a natural number such that $\texttt{vp v n} \in S$. Note that, by $S \longrightarrow \emptyset$, $n < 0$. By the validity of S, $S \longrightarrow S' \longrightarrow \emptyset$, where $\texttt{vp p (s n')} \in S$ and $\texttt{s}\ n' = n$, if the Rule (pgnz) is used, $n = 1$ and there is no more `vp` symbols related to p. Otherwise, the Rule (pgns) is used, $n < 1$, $\texttt{vp p n'} \in S'$ and the induction hypothesis can be applied on S'.

Lemma 4 (Normal Form). *Let S be valid. Then $\emptyset$ is the unique normal form of S with respect to $\longrightarrow$.*

$$
[\![S]\!]^\star = \begin{cases} (V_B, E_B, P_B, prnt_B, ctrl_B, link_B) : \langle m, X\rangle \to \langle n, Y\rangle \\ \quad \text{under the signature } (K, arity) & \text{if } S \longrightarrow \emptyset. \\ \text{undefined} & \text{otherwise.} \end{cases}
$$

where for some $\mathtt{B}$ bigraph name

$$
\begin{aligned}
&K = \{k \mid \forall \mathtt{A} \text{ a node}, \mathtt{lc}\ \ k\ \ \mathtt{A}\ \ \mathtt{B} \in S\},\\
&arity = \{(k, n) \mid \forall \mathtt{A} \text{ a node}, \mathtt{vp}\ \ \mathtt{A}\ \ n\ \ \mathtt{B} \in S \wedge \mathtt{lc}\ \ \mathtt{A}\ \ k\ \ \mathtt{B} \in S\},\\
&\quad \vee\, \mathtt{lc}\ \ \mathtt{A}\ \ k\ \ \mathtt{B} \in S \wedge \neg\exists n', \mathtt{vp}\ \ \mathtt{A}\ \ n'\ \ \mathtt{B} \in S \wedge n = 0\}\\
&V_B = \{v \mid \mathtt{is\_node}\ \ v\ \ \mathtt{B} \in S\},\ E_B = \{e \mid \mathtt{is\_e\_name}\ \ e\ \ \mathtt{B} \in S\}\\
&E_B = \{(v, i) \mid \mathtt{is\_port}\ \ (v, i)\ \ \mathtt{B} \in S\},\ prnt_B = \{(x, y) \mid \mathtt{prnt}\ \ x\ y\ \ \mathtt{B} \in S\}\\
&ctrl_B = \{(v, k) \mid \mathtt{lc}\ \ v\ k\ \ \mathtt{B} \in S\},\ link_B = \{(x, y) \mid \mathtt{link}\ \ x\ y\ \ \mathtt{B} \in S\}\\
&m = \{r \mid \mathtt{is\_root}\ \ r\ \ \mathtt{B} \in S\},\ X = \{x \mid \mathtt{is\_i\_name}\ \ x\ \ \mathtt{B} \in S\}\\
&n = \{s \mid \mathtt{is\_site}\ \ s\ \ \mathtt{B} \in S\},\ Y = \{y \mid \mathtt{is\_o\_name}\ \ y\ \ \mathtt{B} \in S\}
\end{aligned}
$$

Fig. 5. Interpretation function.

Proof. Since strong normalisation has already been proven in Lemma 1, we prove local confluence in order to apply the Newman's lemma. There are 81 critical pairs that respect the hypothesis, some are trivial, and show that rules are truly commutative, the Rules (dr) and (do) for instance. Some others are almost commutative, basically, for parent relation symbols, such as `prnt` or `link`, or when a `port` structural symbol is consumed, they share respectively a `has_child` symbol or a `vp` symbol. It is not truly commutative because the rule does not use the same instance of these additional symbols, but the critical pair can be join by applying the other rule with the new additional symbol.

By induction over the rewrite trace, we can easily convince ourselves that $valid S$ holds if and only if S encodes a bigraph.

Theorem 1 (Inversion). *If S is valid then there exists a bigraph B, s.t. $[\![B]\!] = S$.*

Proof. Lemma 2 and 3 ensure that the set holds the properties of the graphical definition of a bigraph.

This theorem guarantees that the encoding as defined in Figure 4 has an inverse (which is only defined on valid sets). It is defined in Figure 5 and for which we write $[\![S]\!]^\star$. Furthermore, we have shown that there exists a bijection between bigraphs and their representations as valid multi-sets.

Theorem 2 (Adequacy). *Let B be a bigraph under a signature $\Sigma = (K, arity)$, then $[\![[\![B]\!]]\!]^\star \simeq B$.*

Proof. With theorem 1, we provide by cases on element of B:

– $\forall r \in m_B$, `root r` $\in [\![B]\!]$ there are only $r \in m_{[\![[\![B]\!]]\!]}$.
– $\forall n \in V_B, k \in K, ctrl_B\ n = k, \forall d \in V_B \uplus m, prnt\ n = d$, `node n`, `lc n k` and `prnt n d` $\in [\![B]\!]$, there are only $n \in V_{[\![[\![B]\!]]\!]}, ctrl_{[\![[\![B]\!]]\!]}\ n = k$ and $prnt_{[\![[\![B]\!]]\!]}\ n = d$.
– $\forall o \in Y_B$, `is_o_name o` $\in [\![B]\!]$ there are only $o \in Y_{[\![[\![B]\!]]\!]}$
– $\forall e \in E_B$, `is_e_name e` $\in [\![B]\!]$ there are only $e \in E_{[\![[\![B]\!]]\!]}$
– $\forall s \in n_B$, `is_site s` $\in [\![B]\!]$ there are only $s \in n_{[\![[\![B]\!]]\!]}$
– $\forall i \in X_B$, `is_i_name i` $\in [\![B]\!]$ there are only $i \in X_{[\![[\![B]\!]]\!]}$
– $\forall k \in K_B, \forall n \in \mathbb{N}, arity_B\ k = n$, `arity k n` $\in [\![B]\!]$, then $arity_{[\![[\![B]\!]]\!]}\ k = n$.

Next, we show that composition and juxtaposition of bigraphs are provided "for free" in the bigraph relational model. They basically correspond to multi-set union.

Let C be a bigraph. Next, we partition $[\![C]\!]$ into three parts, $\widetilde{C} \uplus out_C \uplus in_C$ where

1. out_C contains only references to roots `is_root`, outer names `is_o_name`, place graph parent relations `prnt` on roots and link graph parent relations `link` on outer names,
2. in_C contains only references to sites `i_site`, inner names `is_i_name`, place graph parent relations `prnt` on sites and link graph parent relations `link` on inner names,
3. and $\widetilde{C} = [\![C]\!] \setminus out_C \setminus in_C$.

Also, let B be a bigraph such that $B = C \circ C'$ for two bigraphs, C and C'. We define the set $eq_{CC'}$ as follows :

$$\begin{aligned} eq_{CC'} = & \{\texttt{prnt x y} \mid \texttt{is\_root r} \in [\![C']\!] \wedge \texttt{is\_site s} \in [\![C]\!] \\ & \wedge \mid \texttt{s} \mid = \mid \texttt{r} \mid \ \wedge \texttt{prnt x r} \in [\![C']\!] \wedge \texttt{prnt s y} \in [\![C]\!]\} \\ & \cup \{\texttt{link x y} \mid \texttt{is\_i\_name i} \in [\![C]\!] \wedge \texttt{is\_o\_name o} \in [\![C']\!] \\ & \wedge \mid \texttt{i} \mid = \mid \texttt{o} \mid \ \wedge \texttt{link i y} \in [\![C]\!] \wedge \texttt{link x o} \in [\![C']\!]\} \end{aligned}$$

Note, that $eq_{CC'}$ is built from $out_{C'}$ and in_C.

Lemma 5. *Let B be a bigraph. If $B \equiv C \circ C'$ then $[\![B]\!] = out_c \cup \widetilde{C} \cup eq_{CC'} \cup \widetilde{C'} \cup in_{C'}$.*

Proof. Following the definition 3 of composition, the outer interface of B is the one of C, here out_c.the inner interface is the one of C', here $in_{C'}$. $V_B = V_C \cup V_{C'}$ with $V_C \cap V_{C'} = \emptyset$, here it is the union of nodes of $[\![C]\!]$ and $[\![C']\!]$. Same things for E_B and $ctrl_B$. The $prnt$ and $link$ maps, that does not involve sites in C and roots in C' are in $\widetilde{C'}$ and $\widetilde{C}$. And the actual composition is defined in $eq_{CC'}$ by $f_B\, x = f_C\, j$ where f is a short cut for $prnt$ or $link$, $f'_C\, x = i$ and $i = j$.

Lemma 6. *Let B be a bigraph. If $B \equiv C \otimes C'$ then $[\![B]\!] = \widetilde{C} \cup \widetilde{C'} \cup out_C \cup in_C \cup out_{C'} \cup in_{C'} = [\![C]\!] \cup [\![C']\!]$.*

Proof. The definition of the juxtaposition directly implies this statement.

4 Modeling Reaction Rules

In this section, we illustrate how we model reaction rules.

4.1 Ground Reaction Rule

Recall that applying a ground reaction rule (L, R) to an agent B proceeds by decomposing B into $C \circ L$ for some bigraph C and then replacing L by R. Therefore, in the model, we only need to partition the agent $[\![B]\!]$ into two sets with respect to C, one that is affected by the reaction rule (here $\widetilde{L} \cup eq_{CL}$) and the other that is not.

$$[\![C \circ L]\!] = out_C \cup \widetilde{C} \cup eq_{CL} \cup \widetilde{L}$$

$$\downarrow$$

$$[\![C \circ R]\!] = out_C \cup \widetilde{C} \cup eq_{CR} \cup \widetilde{R}$$

For a given bigraph C, we can think of a ground reaction rule as a multi-set rewrite rule that replaces among other things the set eq_{CL} by eq_{CR}:

$$\widetilde{L} \cup eq_{CL} \mapsto \widetilde{R} \cup eq_{CR}$$

We can rid this rule of the dependency on C. We know, first, that the inner interface of C must be the same as the outer interface of L (and therefore also R). Second, the components in the interface that depend on C are only roots and outer names. Therefore, instead of quantifying over C, we can reformulate the rule by simply quantifying over the aforementioned components.

4.2 Parametric Reaction Rule

The parametric reaction rule from Definition 6 is a triple that consists of two bigraphs, and a function η that maps sites from the reactum to sites in the redex.

A parametric reaction rule is applied to an agent B if B can be decomposed into $C \circ (D \otimes L) \circ C'$ where C, D and C' are bigraphs and L is the redex. The result of the application is the agent $C \circ (D \otimes R) \circ (\overline{\eta}\ C')$ where $\overline{\eta}$ is defined in Definition 8. The basic idea is essentially the same as in the ground case, therefore we proceed analogously and model decomposition by partition

$$\begin{aligned}[\![C \circ (D \otimes L) \circ C']\!] &= out_C \cup \widetilde{C} \cup eq_{C((D \otimes L) \circ C')} \cup ((D \widetilde{\otimes L)} \circ C') \cup eq_{LC'} \\ &\quad \cup \widetilde{C'} \cup in_{((D \otimes L) \circ C')} \\ &= out_C \cup \widetilde{C} \cup eq_{C((D \otimes L) \circ C')} \cup \widetilde{L} \cup \widetilde{D} \cup eq_{DC'} \cup eq_{LC'} \cup \widetilde{C'}\end{aligned}$$

and model parametric reaction rule as a multi-set rewriting rule.

$$
\begin{aligned}
&eq_{C((D\otimes L)\circ C')} \cup \widetilde{L} \cup \widetilde{D} \cup eq_{DC'} \cup eq_{LC'} \cup \widetilde{C'} \\
&\quad \mapsto eq_{C((D\otimes R)\circ(\overline{\eta}\, C'))} \cup \widetilde{R} \cup \widetilde{D} \cup eq_{D(\overline{\eta}\, C')} \cup eq_{L(\overline{\eta}\, C')} \cup \widetilde{(\overline{\eta}\, C')}
\end{aligned}
$$

Differently from above, the formulation of the rule is not only dependent on C, but also on C'. This time, however things are not as direct, in part because the inner interfaces of the redex and the reactum do not match. In Definition 6, we use η to coerce one to the other, which means that the interfaces between $(\overline{\eta}\, C')$ and R actually do match.

Applying $\overline{\eta}$ to C' is algorithmically simple: on the place graph, the operation recursively *deletes* all sites that are not in the range of η, *moves* all sites that have a unique image under η, or *copies* all sites that do not have a unique image under η; on the link graph, it only administers links from and to ports (as proposed in [6]). The computational essence of these operations is captured in terms of a few multi-set rewriting rules, that iterates over the place graph, which we discuss in more detail in Section 5.

4.3 Meta Theory

We show that modeling ground and the parametric reduction rules as rewrite rules is sound and complete.

Theorem 3 (Soundness). *Let B, B' be agents, $\mathcal{R}$ a set of reaction rules and $(L, R, \eta) \in \mathcal{R}$. If B can be rewritten into B' by L, R, η and $W \mapsto Z$ is the corresponding rewriting system, then the following diagram commutes:*

$$
\begin{tikzcd}
B \arrow[r, "{(L,R,\eta)}"] \arrow[d, "{[\![\cdot]\!]}"'] & B' \arrow[d, "{[\![\cdot]\!]}"] \\
{[\![B]\!]} \arrow[r, "W \mapsto Z"] & {[\![B']\!]}
\end{tikzcd}
$$

Proof. Ground reaction rule. L and R are ground and the graph of η is empty. $B \equiv C \circ L$ and $B' \equiv C \circ R$, by Lemma 5 and because interfaces of L and R are the same, $[\![B]\!] = \widetilde{C} \cup \widetilde{L} \cup out_C \cup in_L \cup eq_{CL}$ and $[\![B']\!] = \widetilde{C} \cup \widetilde{R} \cup out_C \cup in_L \cup eq_{CR}$, therefore $[\![B']\!]$ is the result of one step of the multi-rewriting system $\widetilde{L} \cup eq_{CL} \to \widetilde{R} \cup eq_{CR}$ applied on $[\![B]\!]$. *Parametric reaction rule.* Analogous.

Conversely, all multi-set rewriting system that encode a ground reaction rule respects the semantics of ground BRS.

Theorem 4 (Completeness). *Let X, Y be valid sets. Furthermore let $W \mapsto Z$ be sets such that there exists an (L, R, η) that is a reaction rule that can be applied on $[\![X]\!]^\star$ and $W \mapsto Z$ is the modeled reaction rule of (L, R, η), then the following diagram commutes:*

$$\begin{tikzcd}
X \arrow[r, "W \mapsto Z"] \arrow[d, "\llbracket\cdot\rrbracket^\star"'] & Y \arrow[d, "\llbracket\cdot\rrbracket^\star"] \\
\llbracket X\rrbracket^\star \arrow[r, "(L, R, \eta)"] & \llbracket Y\rrbracket^\star
\end{tikzcd}$$

Proof. Ground reaction rule. Using Lemma 5 and Theorem 2, $\widetilde{L} \cup eq_{XL} \in X$ implies that $\llbracket X\rrbracket^\star \equiv C \circ L$. Therefore $\widetilde{R} \cup eq_{XR} \in X$ implies that $\llbracket Y\rrbracket^\star \equiv C \circ R$. *Parametric reaction rule.* Analogously.

5 Implementation in Celf

We turn now to the original motivation of this work and evaluate the bigraph relational model empirically. The very nature of the rules depicted in Figure 3 suggests a language based on multi-set rewriting, such as Maude, Elan, λProlog, CHR, or Celf. Because of Celf's features, in particular linearity and higher-order abstract syntax, we have decided to use Celf as our implementation platform.

And indeed, the implementation of the bigraph relational model is straightforward. Roots, nodes, sites, etc. are encoded using Celf's intuitionistic features, and the evidence that something is a root, a parent, or a port is captured by linear assumptions using dependent types. Consequently a bigraph is represented by the Celf context. The multi-set rewriting rules as depicted in Figure 3 and the reaction rules from Section 4 are encoded using linear types and the concurrency modality. For example, the rewrite rule

$$\{\texttt{is\_root R, has\_child\_p (dst\_r R) z}\} \uplus \Delta \mapsto \Delta$$

is implemented in Celf as a constant

```
dr : is_root R B * has_child_p (dst_r R) z B -o {1}.
```

where `is_root` carries a reference to the bigraph it is a root for, and all uppercase variables are implicitly Π quantified. Celf provides a sophisticated type inference algorithm that infers all omitted types (or terminates with an error if those cannot be found).

Celf also comes with a forward directed logic programming engine in the style of Lollimon [3], which resembles the CHR evaluation engine. During operation, the uppercase variable names are replaced by logic variables, which are subsequently instantiated by unification if the rule is applied. Note that the properties of the encoded bigraph reactive system are preserved: If the reaction rules are strongly normalising then so is their encoding. As an illustration of our experiments we depict an encoding of bigraph validity (see Figure 3) as a type family `valid` and a few Celf declarations in Figure 5. Note how similar the two figures are.

The higher-order nature of Celf allows us to express rewriting rules that dynamically introduce new rewriting rules on the fly. In Celf, rewriting rules are

```
%% base cases
dr : is_root R B * has_child_p (dst_r R) z B -o {1}.
do : is_o_name O B * has_child_l (dst_o O) z B -o {1}.
de : is_e_name E B * has_child_l (dst_e E) z B -o {1}.

%% recursive cases
lgpsz : is_port P Bi * lp P A Bi * vp A (s z) Bi * link (src_p P) D Bi
  * has_child_l D (s N) Bi -o {has_child_l D N Bi}.
lgi : is_i_name I Bi * link (src_i I) D Bi
  * has_child_l D (s N) Bi -o {has_child_l D N Bi}.
pgs : is_site S Bi * prnt (src_s S) D Bi * has_child_p D (s N) Bi
  -o {has_child_p D N Bi }.
pgnz : is_node A Bi * has_child_p (dst_n A) z Bi * prnt (src_n A) D Bi
  * has_child_p D (s N) Bi * lc A K Bi -o arity K z -> {has_child_p D N Bi}.
pgns : is_node A Bi * has_child_p (dst_n A) z Bi * prnt (src_n A) D Bi
  * has_child_p D (s N) Bi * lc A K Bi -o arity K (s N')
    -> {has_child_p D N Bi * vp A (s N') Bi}.
lgps : is_port P Bi * lp P A Bi * vp A (s (s  N)) Bi
  * link (src_p P) D Bi * has_child_l D (s N') Bi
    -o {vp A (s  N) Bi * has_child_l D N' Bi}.
```

Fig. 6. The implementation of the valid relation in Celf.

first-class citizens. The logical principle behind this technique is called *embedded implications*, as popularised by λProlog. By nesting them we achieve elegant encodings.

An example is the encoding of a parametric bigraph reaction rule $(L, R, \overline{\eta})$. The definition of the Celf signature is rather involved, where we use linearity and token system $(\llbracket\emptyset, \emptyset\rrbracket)$ in order to sequentialise the reaction rule. Below we give an algorithm that computes the Celf declaration

$$\texttt{rule}_{(L,R,\overline{\eta})} : \llbracket(m, \eta)\rrbracket \otimes \left(\llbracket(\emptyset, \emptyset)\rrbracket \multimap \{\widetilde{L} \otimes eq_{XL} \multimap \{\widetilde{R} \otimes eq_{XR})\}\}\right)$$

from the sites m in L and η.[1] Recall the three auxiliary operations *delete*, *move*, and *copy* that are triggered depending on the cardinality $c = |\{x \mid x \in m, (x, y) \in \eta\}|$. In the case that is $c = 0$, we first colour all the direct children of node $prnt_L\ x$ (using `tmp`) that contain the site x with colour `tmp_prnt`. Second we remove the colour information from all of siblings of x that are also present in L. Third, for each coloured node, we start a recursive descent phase (using `del`) to trigger the deletion of the node and its children.

[1] In the interest of clarity, we omit all references to the bigraph identifiers from Celf type constructors.

$$
\begin{aligned}
&[\![x \in m, (x,y) \notin \eta]\!] = \\
&\qquad \texttt{has\_child\_p}\ \ (prnt_L\ x)\ (\texttt{s}^k\ N) \otimes \texttt{tmp}\ \ (prnt_L\ x)\ (s^k\ N) \\
&\qquad\quad \otimes (\texttt{tmp}\ \ (prnt_L\ x)\ \texttt{z} \multimap \{\bigotimes_{i=0}^{k} \texttt{tmp\_prnt}\ \ S_i\ D_i \\
&\qquad\qquad \multimap \{\bigotimes_{i=0}^{k} \texttt{prnt}\ \ S_i\ D_i \otimes \texttt{del}\ \ (prnt_L\ x)\ N \otimes (\texttt{del}\ \ (prnt_L\ x)\ \texttt{z} \\
&\qquad\qquad\quad \multimap \{\texttt{has\_child\_p}\ \ (prnt_L\ x)\ (\texttt{s}^k \texttt{z}) \otimes [\![m \setminus \{x\}, \eta]\!]\})\}\})
\end{aligned}
$$

In the case that $c = 1$, we do something very similar as in the previous case, except that we move instead of delete. This case is conceptually easier because we can skip the recursive descent phase.

$$
\begin{aligned}
&[\![x \in m, (x,y) \in \eta]\!] = \\
&\qquad \texttt{has\_child\_p}\ \ (prnt_R\ y)\ (\texttt{s}^k\ N) \otimes \texttt{tmp\_move}\ \ (prnt_L\ x)\ (s^k\ N) \\
&\qquad\quad \otimes (\texttt{tmp\_move}\ \ (prnt_L\ x)\ \texttt{z} \multimap \{\texttt{move}\ \ (prnt_L\ x)\ (prnt_R\ x)\ N \\
&\qquad\qquad \otimes (\texttt{move}\ \ (prnt_L\ x)\ (prnt_R\ x)\ \texttt{z} \\
&\qquad\qquad \multimap \{\texttt{has\_child\_p}\ \ (prnt_R\ y)\ (\texttt{s}^k\ \texttt{z}) \otimes [\![m \setminus \{x\}, \eta \setminus \{(x,y)\}]\!]\})\})
\end{aligned}
$$

The case that $c > 1$ is again similar, except that this time we need to recursively copy the graph rooted in $prnt_L\ x$. While copying we are forced to create new nodes, ports, etc., which we get for free from the `Exists` connective that is part of Celf.

$$
\begin{aligned}
&[\![x \in m, (x,y) \in \eta]\!] = \\
&\qquad \texttt{has\_child\_p}\ \ (prnt_R\ x)\ (\texttt{s}^k\ N) \otimes \texttt{tmp\_copy}\ \ (prnt_L\ x)(\texttt{s}^k\ N) \\
&\qquad\quad \otimes (\texttt{tmp\_copy}\ \ (prnt_L\ x)\ \texttt{z} \multimap \{\bigotimes_{i=0}^{k} \texttt{tmp\_prnt}\ \ S_i\ D_i \\
&\qquad\qquad \multimap \{\bigotimes_{i=0}^{k} \texttt{prnt}\ \ S_i\ D_i \otimes \texttt{copy}\ \ (prnt_L\ x)\ (prnt_R\ y)\ N \\
&\qquad\qquad\quad \otimes (\texttt{copy}\ \ (prnt_L\ x)\ (prnt_R\ y)\ \texttt{z} \\
&\qquad\qquad\qquad \multimap \{\texttt{has\_child\_p}\ \ (prnt_R\ x)\ (\texttt{s}^k\ N) \otimes [\![m, \eta \setminus \{(x,y)\}]\!]\})\}\})
\end{aligned}
$$

In the base case, we define $[\![\emptyset, \emptyset]\!]$ as a Celf type constructor. Note that some cases, in particular move and copy, need additional rule in the case that the source node and the destination node are the same. This definition is well-formed because during the recursive calls either, m, η, or both get smaller.

Finally, we address the question of adequacy. Let $(B, \mathcal{R})$ be a BRS, and Γ the intuitionistic Celf context that contains the names of all ports, sites, roots, inner names, outer names, edges, nodes, controls, the graph of the arity function in B, and the translation of all reaction rules declared in $\mathcal{R}$.

Theorem 5 (Adequacy). *The agent B reduces to agent B' using the rules in $\mathcal{R}$ if and only if (in* Celf*) $\Pi\Gamma.\bigotimes[\![B']\!] \multimap \{C\}$ implies that $\Pi\Gamma.\bigotimes[\![B]\!] \multimap \{C\}$.*

Proof. By induction on the reduction sequence, using the definition of $[\![(m, \eta)]\!]$ and Theorems 3, 4.

Conclusion

In this paper we have describe a model for bigraph reactive systems, which we refer to as the bigraph relational model. We have shown that this model that is based on a multi-set rewriting system is amenable to implementation. The rewriting system ensures the validity of the encoding with respect to the bigraph structural properties. We have also shown that the semantics of BRS is precisely captured by the multi-set rewriting rules. Finally, we give an implementation of the bigraph relational model in Celf. Celf is a powerful tool: Linearity allows us to implement the reaction rules directly, its higher-order features take care of dynamic introduction of new rewrite rules and the creation of fresh names while copying the place graph where warranted.

References

1. Maxime Beauquier and Carstern Schürmann. Bigraphs in CLF. Technical report, tr 2010-126, IT University of Copenhagen, 2010.
2. Iliano Cervesato, Frank Pfenning, David Walker, and Kevin Watkins. A concurrent logical framework ii: Examples and applications. Technical report, cmu-cs-02-102, Caregie Mello University, Pittsburgh, PA 15213, 2002.
3. Pablo López, Frank Pfenning, Jeff Polakow, and Kevin Watkins. Monadic concurrent linear logic programming. In Pedro Barahona and Amy P. Felty, editors, *Proceedings of the 7th International ACM SIGPLAN Conference on Principles and Practice of Declarative Programming*, pages 35–46, Lisbon, Portugal, 2005.
4. Robin Milner. *A calculus of communicating systems.* Springer-Verlag, 1980.
5. Robin Milner. *Communicating and Mobile Systems: the Pi-Calculus.* Cambridge University Press, 1999.
6. Robin Milner. *The Space and Motion of Communication Agents.* Cambridge University Press, 2009.
7. Wolfgang Reisig. *Petri Nets.* Springer-Verlag, Berlin, 1985.
8. Anders Schack-Nielsen and Carsten Schürmann. Celf — a logical framework for deductive and concurrent systems (system description). In *IJCAR '08: Proceedings of the 4th international joint conference on Automated Reasoning*, pages 320–326, Berlin, Heidelberg, 2008. Springer-Verlag.
9. Anders Schack-Nielsen and Carsten Schürmann. Pattern Unification for the Lambda Calculus with Linear and Affine Types. *Electronic Proceedings in Theoretical Computer Science*, 34:101–116, 2010. In Proceedings LFMTP 2010.