

An adequate compositional encoding of bigraph structure in linear logic with subexponentials

Kaustuv Chaudhuri, Giselle Reis

▶ **To cite this version:**

Kaustuv Chaudhuri, Giselle Reis. An adequate compositional encoding of bigraph structure in linear logic with subexponentials. 20th International Conference on Logic for Programming, Artificial Intelligence and Reasoning (LPAR), Nov 2015, Suva, Fiji. pp.146–161, 10.1007/978-3-662-48899-7_11 . hal-01208362

HAL Id: hal-01208362
https://inria.hal.science/hal-01208362

Submitted on 2 Oct 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

An adequate compositional encoding of bigraph structure in linear logic with subexponentials

Kaustuv Chaudhuri[1] and Giselle Reis[1]

Inria & LIX/École Polytechnique, France
{kaustuv.chaudhuri,giselle.reis}@inria.fr

Abstract. In linear logic, formulas can be split into two sets: classical (those that can be used as many times as necessary) or linear (those that are consumed and no longer available after being used). Subexponentials generalize this notion by allowing the formulas to be split into many sets, each of which can then be specified to be classical or linear. This flexibility increases its expressiveness: we already have adequate encodings of a number of other proof systems, and for computational models such as concurrent constraint programming, in linear logic with subexponentials (SEL). Bigraphs were proposed by Milner in 2001 as a model for ubiquitous computing, subsuming models of computation such as CCS and the π-calculus and capable of modeling connectivity and locality at the same time. In this work we present an encoding of the bigraph structure in SEL, thus giving an indication of the expressive power of this logic, and at the same time providing a framework for reasoning and operating on bigraphs. Our encoding is adequate and therefore the operations of composition and juxtaposition can be performed on the logical level. Moreover, all the proof-theoretical tools of SEL become available for querying and proving properties of bigraph structures.

1 Introduction

Linear logic is excellent at *counting* elements of state since it interprets linear hypotheses as resources that are consumed upon use, and every linear hypothesis must be used in a proof. However, it is not particularly good at reasoning about *relationships* between elements. For example, an office building can have several rooms, with some rooms having a number of cubicles, and some cubicles containing several persons and computers. A precise description of this building must not only inventory all the rooms, cubicles, persons, and computers, but also state which component occurs inside which other components. In ordinary linear logic we can attempt to express the inclusion relation as a separate predicate, say a binary predicate $\mathtt{in}(x, y)$ that expresses that the entity with index x is contained in that with index y, but such predicates are nearly impossible to treat linearly as they may be consulted and composed repeatedly. Yet, treating them non-linearly is also problematic: if a person moves from one cubicle to another, for example, the old placement needs to be invalidated, but there is no way to "consume" a non-linear resource in linear logic.

Recently, a family of logics called *subexponential logics* (*SEL*) has emerged as a way to capture such intensional relationships between resources [12]. The idea is fairly old [6]: the exponential connectives ! and ? can be split into several different *flavors*, and the linear logic proof system would, *a priori*, make each version of the connectives independent of every other. If we impose a pre-order on them though, we can get a limited form of *promotion* where $?^u A$ entails $?^v A$ assuming the u version of the exponentials are smaller than the v version. In this view, $?^u$ corresponds, roughly, to a *placement* in the *zone* u, while $!^u$ corresponds to *checking* that every resource is placed in a zone larger than u. Subexponential logics are naturally much more expressive than just ordinary linear logic; for instance, even the propositional additive-free fragment is undecidable [5]. Moreover, they can easily express a wide spectrum of other deductive systems at a very high level of *representational adequacy* [14, 4, 13].

In this paper, we consider the use of *SEL* as a *logical framework* for specifying *bigraphs*, a model of computation proposed by Milner in 2001 [10]. Bigraphs are a very general model, subsuming standard process calculi such as CCS [9] and the π-calculus [8]. Our use of *SEL* for this purpose has two primary goals:

- It gives a good formal indication of the expressive power of *SEL* as a logical framework. If *SEL* can be used for bigraphs, it can probably also be used for any location-aware formalism. A number of such formalisms exist in the literature, such as ambients [3], brane calculi [2], and membrane computing [15].
- It provides an adequate *syntactic* treatment for bigraph structure.[1] By adequate, we mean that any structural operation, such as composition and juxtaposition, can be performed directly on the syntactical representation. Bigraphs certainly have a well developed categorical semantics, but formalizing that semantics can be rather heavyweight. We show straightforward encoding and decoding functions from bigraphs to *SEL* sequents. Such a representation of bigraphs benefits from all the proof-theoretic tools available for querying and proving properties.

In relation to the second goal, another syntactic treatment for bigraphs can be found in [1]. The authors encode a bigraph and its reaction rules in the Concurrent LF (*CLF*) type theory, where both the formation and the reactions of bigraphs are encoded as rewrite rules. The encoding of locations is done by using a (linear) predicate `has_child`, and thus presents the problems previously mentioned. In particular, their use of a multi-set of atoms for encoding a bigraph makes it impossible to query a static structure, or prove its correctness, without damaging the structure itself. The proof that a bigraph is valid in [1] indeed consists of rewriting it to the empty set. Analogously, a containment check cannot be made without consuming the atom(s) encoding the parent relations.

In principle, we could use the same approach with rewriting rules to check for the correctness of bigraphs in our encoding, but we have decided to take

[1] We posit that, given the way computer science is evolving, the lack of formal and mechanized reasoning capabilities for any formalisms can be fatal.

advantage of the expressiveness of subexponentials by having a more concise and conservative approach. In this work, we will encode bigraphs using fewer predicates than in [1] and in such a way that asking questions about the static structure will not cause any modifications to it. Our bigraph encoding changes only on actual operations, such as juxtaposition and compositions.

Our specific approach is to start with a *sequent calculus* for first-order classical *SEL* (Section 2.1). The classical dialect of *SEL* is used simply to keep the proof systems simple; our results extend to the intuitionistic dialect without any complications. The static definition of a bigraph is encoded in *SEL* as follows. The place graph of the bigraph is given in terms of atomic formulas placed in certain subexponential zones, and the *subexponential signature* encodes the parent relationship. The link graph of the bigraph is given in terms of eigenvariable parameters that are shared by every place that has a connection to a hyperedge. Finally, the inner and outer faces of the bigraph are expressed as purely linear formulas that are placed in the *default* zone of the *SEL*. This encoding will be an injection, *i.e.*, it will have a right-inverse that will be able to extract the original bigraph from its sequent encoding. Furthermore, the static structure of the bigraph can be *queried* by trying to derive certain formulas. For example, to check whether a given node occurs inside (perhaps with multiple levels of nesting) a given place is tantamount to deriving a corresponding !-formula.

On top of this static description of a single bigraph, we define bigraph composition by first moving to a *focused* proof system for *SEL*, called *SELF* (Section 2.2) [11]. Focusing is a general mechanism for treating certain combinations of inferences in a (well behaved) sequent calulus as *atomic* derived inference rules. Bigraph juxtaposition in our encoding corresponds to a multiplicative composition of the two sequents describing the two bigraphs, and the composition of the bigraphs is given as a small *theory* on these sequents. We then run focused proof search on this sequent, extended with the composition theory, until quiescence, *i.e.*, until no more focused steps are possible. The single unfinished leaf of the proof will then be the representation of the composed bigraph. This gives us the necessary *representational adequacy* theorem (Section 3.1).

2 Background

2.1 Subexponential Logic (*SEL*)

Subexponential logic (*SEL*) is a variant of linear logic with a family of exponential connectives—called *subexponentials* (a term coined in [12])—that are indexed and arranged in a pre-order [6, 11]. To keep things simple, we will use the classical dialect of *SEL* in this paper. The grammar of *formulas* $(A, B, \dots)$ is as follows:

$$\begin{array}{rcl} A, B, \dots & ::= & \underbrace{a}_{\text{LITERAL}} \mid \underbrace{A \otimes B \mid 1}_{\text{MULTIPLICATIVE}} \mid \underbrace{A \oplus B \mid 0}_{\text{ADDITIVE}} \mid \underbrace{\exists x.\, A}_{\text{QUANTIFIED}} \mid \underbrace{!^u A}_{\text{SUBEXPONENTIAL}} \\ & \mid & \underbrace{\neg a}_{\text{LITERAL}} \mid \underbrace{A \parr B \mid \bot}_{\text{MULTIPLICATIVE}} \mid \underbrace{A \mathbin{\&} B \mid \top}_{\text{ADDITIVE}} \mid \underbrace{\forall x.\, A}_{\text{QUANTIFIED}} \mid \underbrace{?^u A}_{\text{SUBEXPONENTIAL}} \end{array} \tag{1}$$

Each column in the grammar is a De Morgan dual pair, and we indicate the dual of A by $(A)^{\perp}$. The identifiers $a, b, \dots$ range over *atomic formulas*, which

$$\frac{}{\vdash \Gamma^{\mathcal{U}}, u{:}\neg\, a, a}\,\mathsf{init} \quad \frac{\vdash \Gamma^{\mathcal{U}}, \Delta_1, A \quad \vdash \Gamma^{\mathcal{U}}, \Delta_2, B}{\vdash \Gamma^{\mathcal{U}}, \Delta_1, \Delta_2, A \otimes B}\otimes \quad \frac{}{\vdash \Gamma^{\mathcal{U}}, 1}\,1 \quad \frac{\vdash \Gamma, A_i}{\vdash \Gamma, A_1 \oplus A_2}\oplus_i \quad \frac{\vdash \Gamma, [t/x]A}{\vdash \Gamma, \exists x.\, A}\,\exists$$

$$\frac{\vdash \Gamma, A, B}{\vdash \Gamma, A \parr B}\,\parr \quad \frac{\vdash \Gamma}{\vdash \Gamma, \bot}\,\bot \quad \frac{\vdash \Gamma, A \quad \vdash \Gamma, B}{\vdash \Gamma, A \,\&\, B}\,\& \quad \frac{}{\vdash \Gamma, \top}\,\top \quad \frac{\vdash \Gamma, A}{\vdash \Gamma, \forall x.\, A}\,\forall \quad \frac{\vdash \Gamma, u{:}A}{\vdash \Gamma, ?^u A}\,?$$

$$\frac{(u \leq \vec{v}) \quad \vdash \vec{v}{:}\vec{A}, C}{\vdash \Gamma^{\mathcal{U}}, \vec{v}{:}\vec{A}, !^u C}\,! \quad \frac{(u \notin \mathcal{U}) \quad \vdash \Gamma, A}{\vdash \Gamma, u{:}A}\,\mathsf{derelict} \quad \frac{(u \in \mathcal{U}) \quad \vdash \Gamma, u{:}A, A}{\vdash \Gamma, u{:}A}\,\mathsf{copy}$$

Fig. 1: A cut-free, one-sided, and zoned sequent calculus formulation of *SEL*. In the $\oplus_i$ rules, $i \in \{1, 2\}$. In the $\forall$ rule, x is not free in Γ.

are formulas of the form $\mathtt{p}(t_1, \ldots, t_n)$ where $\mathtt{p}$ stands for a predicate symbol and $t_1, \ldots, t_n$ are first-order terms. Each *term* is either a *variable* $(x, y, \ldots)$ or an *application* of the form $\mathtt{f}(s_1, \ldots, s_m)$ where $\mathtt{f}$ stands for a *function symbol* and the $s_1, \ldots, s_m$ are terms. Variables, predicates, and function symbols are assumed to be drawn from disjoint infinite sets and arities are fixed. We follow the usual syntactic convention of dropping the parentheses when the arity of a predicate or function symbol is 0. The subexponential connectives $!^u$ and $?^u$ are indexed by *subexponential labels* $(u, w, \ldots)$ that belong to a *subexponential signature*.

Definition 1. *A* subexponential signature Σ *is a structure* $\langle \Lambda, \mathcal{U}, \leq \rangle$ *where:*

- Λ *is a countable set of* labels*;*
- $\mathcal{U} \subseteq \Lambda \uplus \{\varepsilon\}$*, called the* unbounded labels*;*
- $\leq$ *is a pre-order on* $\Lambda \uplus \{\varepsilon\}$*—i.e., it is reflexive and transitive—and* $\leq$*-upwardly closed with respect to* $\mathcal{U}$*, i.e., for any* $u, w \in \Lambda \uplus \{\varepsilon\}$*, if* $u \in \mathcal{U}$ *and* $u \leq w$*, then* $w \in \mathcal{U}$*; and*
- ε *is* $\leq$*-minimal, i.e., for every* $u \in \Lambda \uplus \{\varepsilon\}$*,* $\varepsilon \leq u$*.*

We say that any label in $\Lambda \uplus \{\varepsilon\} \setminus \mathcal{U}$ *is* bounded*.*

Intuitively, unbounded labels indicate formulas which can be weakened or contracted, while bound labels indicate linear formulas. We will assume an ambient subexponential signature Σ unless we need to disambiguate particular instances of *SEL*, in which case we will use Σ in subscripts. For instance, the set $\mathcal{U}_\Sigma$ stands for the unbounded labels of Σ.

The true formulas of *SEL* are derived from a *sequent calculus* proof system. There are many variants of such calculi in the literature; here, to keep things simple, we will use a *zoned* formulation where the members of a sequent are *zoned formulas* of the form $u{:}A$ where A is a formula and $u \in \Lambda \uplus \{\varepsilon\}$. When it is unambiguous, we will drop the *default* zone label ε, *i.e.*, write $\varepsilon{:}A$ as just A. A *context* $(\Gamma, \Delta, \ldots)$ is a non-empty multiset of zoned formulas, and Γ, Δ and $\Gamma, u{:}A$ stand as usual for the multi-set union of Γ with Δ and $\{u{:}A\}$, respectively.

The inference rules for *SEL* sequents are displayed in Figure 1. Most of the rules are shared between *SEL* and ordinary linear logic and will not be elaborated upon here; we will therefore explain only the peculiarities of *SEL*.

Definition 2 (Notation). *For $\vec{v} = [v_1, \ldots, v_n]$ and $\vec{A} = [A_1, \ldots, A_n]$:*

- *$\vec{v}{:}\vec{A}$ stands for the context $v_1{:}A_1, \ldots, v_n{:}A_n$;*
- *$u \leq \vec{v}$ means that $u \leq v_1$, ..., and $u \leq v_n$; and*
- *$\vec{v} \in \mathcal{U}$ stands for $v_1 \in \mathcal{U}$, ..., and $v_n \in \mathcal{U}$.*

We write $\Gamma^{\mathcal{U}}$ for a context of the form $\vec{u}{:}\vec{A}$ where $\vec{u} \in \mathcal{U}$.

The rule for !, sometimes called *promotion*, has a side condition that checks that the label of the principal formula is less than the labels of all the other zoned formulas in the context that survive into the premise. This rule cannot be used if there are non-zoned formulas in the context, nor if the labels of some of the bound zoned-formulas are strictly smaller or incomparable with the subexponential label of the principal formula.

Theorem 1. *The following rules are admissible for SEL.*

$$\frac{}{\vdash u{:}(A)^{\perp}, A}\ \mathsf{init}^{*} \qquad \frac{\begin{array}{c}(\forall w{:}C \in \Gamma^{\mathcal{U}} \cup \Delta_1 \cup \{u{:}A\}, v \leq w) \\ \vdash \Gamma^{\mathcal{U}}, \Delta_1, u{:}A \quad \vdash \Gamma^{\mathcal{U}}, \Delta_2, v{:}(A)^{\perp}\end{array}}{\vdash \Gamma^{\mathcal{U}}, \Delta_1, \Delta_2}\ \mathsf{cut}$$

Proof (sketch). The init^{*} rule can be shown to be derivable by a straightforward structural induction on A, where reflexivity of $\leq$ is used in the cases for the subexponentials. For cut-admissibility, the standard permutative cut-reduction algorithm works. Transitivity of $\leq$ is used to justify the commutative cut case for !, while upward closure of $\leq$ with respect to $\mathcal{U}$ is used to justify permuting cuts past copy. □

Subexponential logic is considerably more expressive than ordinary linear logic. Even the propositional additive-free fragment can be undecidable with as few as three subexponentials [5]. By choosing the appropriate signature it is possible to encode many proof systems [14] and various models of concurrency in concurrent constraint programming [13] in a natural way. This expressivity is a result of the ability of the ! rule to "check" the emptiness of certain zones in the sequent. Note that this rule can only be applied if all (linear) formulas are in bigger zones, which means that smaller or unrelated zones must be empty. It is natural then to view the zones in a sequent as *locations* where a formula has been stored (by the ? rule).

2.2 Focusing (*SELF*)

In our intended use of *SEL* as a specification logic for bigraphs, we will sometimes need to reason about sequences of inference steps that happen atomically. These *derived* or *synthetic* inference steps will be informed by *focusing*, which is a general technique for determining a normal form for a sequent calculus that eliminates many unnecessary branching points. A focused proof consists of an alternation of two *phases*. The *asynchronous phase* uses[2] the inference rules in

[2] As usual in the view of the sequent calculus as a proof search formalism, we read inference rules from conclusion to premises.

$$\frac{}{\vdash \Omega^{\mathcal{U}}, u{:}\neg\, a, [a]}\ [\mathsf{init}] \qquad \frac{\vdash \Omega^{\mathcal{U}}, \Omega_1, [B] \quad \vdash \Omega^{\mathcal{U}}, \Omega_2, [C]}{\vdash \Omega^{\mathcal{U}}, \Omega_1, \Omega_2, [B \otimes C]}\ [\otimes] \qquad \frac{}{\vdash \Omega^{\mathcal{U}}, [\mathbf{1}]}\ [1] \qquad \frac{\vdash \Omega, [A_i]}{\vdash \Omega, [A_1 \oplus A_2]}\ [\oplus_i]$$

$$\frac{\vdash \Omega, [[t/x]A]}{\vdash \Omega, [\exists x.\, A]}\ [\exists] \qquad \frac{(u \leq \vec{w}) \quad \vdash \vec{w}{:}\vec{A}, C}{\vdash \Omega^{\mathcal{U}}, \vec{w}{:}\vec{A}, [!^u\, C]}\ [!] \qquad \frac{\vdash \Omega, N}{\vdash \Omega, [N]}\ [\mathsf{release}]$$

. . . . . .

Rules $\parr$, $\bot$, $\&$, $\top$, $\forall$, ? shared with *SEL*

$$\frac{(u \notin \mathcal{U}) \quad \vdash \Omega, [A]}{\vdash \Omega, u{:}A}\ \mathsf{ldecide} \qquad \frac{(u \in \mathcal{U}) \quad \vdash \Omega, u{:}A, [A]}{\vdash \Omega, u{:}A}\ \mathsf{udecide}$$

Fig. 2: The *SELF* inference system, a focused version of *SEL*. The context $\mathbf{\Omega}$ is such that if $A \in \mathbf{\Omega}$ then A is a positive formula or a negated atom.

the second line of Figure 1 that are invertible and can therefore always be safely applied. When no such rule is applicable, the proof enters a *synchronized phase* by selecting a formula for *focus*; this formula is decomposed under focus using the rules in the first line of Figure 1 until no such rule applies, in which case the phase switches back to asynchronous again. To be a bit more explicit, we add a new kind of *focused sequent* of the form $\vdash \Gamma; [A]$ where A is the focused formula.

The inference rules for the focused version of *SEL*, called *SELF*, are given in Figure 2. These rules are based on a division of the formulas of *SEL* into *positive formulas* $(P, Q, \dots)$ that come from the first line of (1), and *negative formulas* $(N, M, \dots)$ that come from the second line. We use the convention that $\mathbf{\Omega}$ stands for a context that consists of zoned formulas where if the zone label is ε then the corresponding formula is positive or a literal. As before $\mathbf{\Omega}^{\mathcal{U}}$ stands for a context of the form $\vec{u}{:}\vec{A}$ where $\vec{u} \in \mathcal{U}$. There are two *decision* rules, ldecide and udecide, for focusing on a zoned formula with a bound label and an unbound label respectively. The focus persists through to the immediate subformula in the synchronous phase rules $[\otimes]$, $[\oplus_i]$, and $[\exists]$. Ultimately, the focused formula becomes negative or a !-formula, in which case focus is released and the asynchronous phase begins, where the inference rules are identical to those of *SEL*. Finally, when no more asynchronous rules are available, a decision rule is used to enter the focused phase again.

Theorem 2 (Soundness and Completeness of *SELF*). *The sequent $\vdash \Gamma$ is derivable in SELF if and only if it is derivable in SEL.*

Proof (sketch). Soundness is immediate, since if we interpret $\vdash \mathbf{\Omega}, [A]$ as $\vdash \mathbf{\Omega}, A$ then we get a *SEL* proof from a *SELF* proof. For completeness, we can easily adapt any of the existing proofs for related systems, such as [11, chapter 5]. □

3 Bigraphs in *SEL*

Bigraphs were proposed by Milner [10] as a model for ubiquitous computing. It tries to fill in a gap between the current state of computational systems, which involves communications on a global scale and pervasive computing, and the available tools to model and reason about such systems. In bigraphs, *locality* and *connectivity* of agents are treated independently, and they might be equipped with reaction rules (*i.e.*, rewriting) for reconfiguration. It has been shown that bigraphs

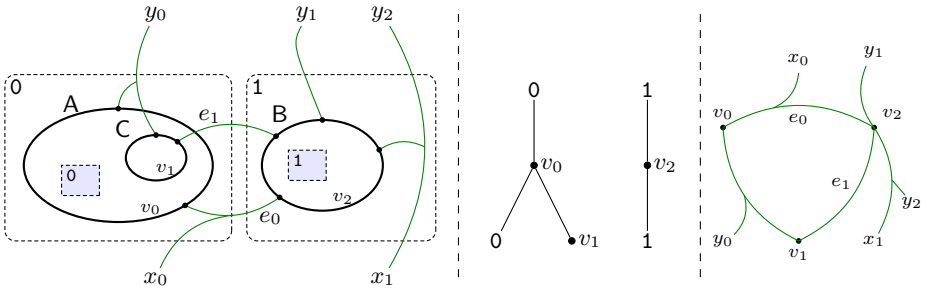


Fig. 3: A bigraph and its place and link graphs.

subsume the calculus of communicating systems (CCS) [9] and π-calculus [8]. Most definitions and examples in this section were taken from [9].

A bigraph is a combination of two graphs (hence the name): a *place graph* and a *link graph*. Both have in common the set of vertices, but the place graph is a set of trees while the link graph is a hypergraph (*i.e.*, one edge can connect any number of nodes). Figure 3 is an example of a bigraph and its components, each representing orthogonally the concepts of locality and connectivity.

Each node takes the role of a specific *control* that determines its ports. The controls available for a bigraph are defined in its *signature*, which does not have to be finite or even denumerable.

Definition 3 (Bigraph signature). *A signature $\mathcal{K}$ is a set of elements called* controls*, each with an* arity *($ar(K)$). The signature also determines which controls are* atomic *and which of the non-atomic controls are* active*.*

Depending on the system being represented, such signatures can be augmented with, for example, sorts that enforce what kind of controls a non-atomic control can contain, or signs that make sure only ports of opposite polarities are connected. For a discussion on adding sorts to our encoding, see Section 4. Given a signature, we can formally define a bigraph.

Definition 4 (Bigraphs). *A* bigraph $\mathcal{B}$ *over the signature $\mathcal{K}$ is defined as:*

$$\mathcal{B} = (V, E, ctrl, prnt, link) : \langle m, X\rangle \to \langle n, Y\rangle\,.$$

- *V is a set of node names.*
- *E is a set of edge names.*
- *m is the number of sites (i.e.,* holes *which can accommodate other bigraphs).*
- *X is a set of inner names.*
- *n is the number of roots (i.e., which can be accommodated in other bigraphs).*
- *Y is a set of outer names.*
- *$ctrl : V \to \mathcal{K}$ defines the role of each node.*
- *$prnt : m \uplus V \to n \uplus V$ defines the parent relation, or the place graph.*
- *$link : X \uplus P \to Y \uplus E$ defines the connectivity, or the link graph, where $P = \{(v, i) \mid 1 \le i \le ar(ctrl(v))\}$, i.e, (v, i) is the i^{th} port of node v.*

The sets of names V, E, X, and Y must be pairwise disjoint. The terms m and n are treated notationally as sets of all strictly smaller natural numbers, i.e., the sites are numbered $0, 1, \ldots, m-1$, and likewise for the roots.

For the bigraph of Figure 3, we have $\mathcal{K} = \{\mathsf{A} : 2, \mathsf{B} : 4, \mathsf{C} : 2\}$ and:

$$V = \{v_0, v_1, v_2\} \quad E = \{e_0, e_1\} \quad X = \{x_0, x_1\} \quad Y = \{y_0, y_1, y_2\} \quad m = n = 2$$

ctrl	*prnt*	*link*	
$v_0 \mapsto \mathsf{A}$	$0 \mapsto v_0$	$x_0 \mapsto e_0$	$(v_1, 2) \mapsto e_1$
$v_1 \mapsto \mathsf{C}$	$1 \mapsto v_2$	$x_1 \mapsto y_2$	$(v_2, 1) \mapsto y_1$
$v_2 \mapsto \mathsf{B}$	$v_0 \mapsto 0$	$(v_0, 1) \mapsto y_0$	$(v_2, 2) \mapsto y_2$
	$v_1 \mapsto v_0$	$(v_0, 2) \mapsto e_0$	$(v_2, 3) \mapsto e_0$
	$v_2 \mapsto 1$	$(v_1, 1) \mapsto y_0$	$(v_2, 4) \mapsto e_1$

3.1 Encoding bigraphs

Given a bigraph $\mathcal{B}$, we will show in this section how to encode it using a *SEL* sequent. Intuitively, each *place* is represented by a subexponential index; indices are related according to *prnt* in the place graph. Next, each place which has an assigned control (*i.e.*, the actual *nodes*) will be represented by a predicate, conveniently named `control` and placed in the zone corresponding to its parent. The *links* are encoded by using constants or variables that are shared among the controls if they are linked together. We will use a `typewriter` font to denote constants in our encoding.

Definition 5 (Encoding function). *Let $\mathcal{B} = (V, E, ctrl, prnt, link) : \langle m, X\rangle \to \langle n, Y\rangle$ be a bigraph over a signature $\mathcal{K}$, then:*

- *Term and atomic formulas:*
  - `id` *is a unique identifier.*
  - *Each element of V, E, and $\mathcal{K}$ is a constant.*
  - *Each element of X and Y is a variable.*
  - $\texttt{control}(\mathtt{K}, \mathtt{v}, L)$ *is an atomic formula, where L is a list of size $ar(\mathtt{K})$.*
  - $\texttt{in}(x, l, \texttt{id})$ *and* $\texttt{out}(y, \texttt{id})$ *are atomic formulas, where l is a constant or variable corresponding to an element $e \in E \cup Y$.*
  - $\texttt{site}(\texttt{id})$ *and* $\texttt{root}(\texttt{id})$ *are atomic formulas.*
- *Subexponential signature $\Sigma = \langle \Lambda, \{\infty\}, \leq\rangle$:*
  - *Each node name is duplicated as a label, i.e., $V \subseteq \Lambda$.*
  - *$r_0, \ldots, r_{n-1} \in \Lambda \setminus V$ are subexponential* variables *representing the roots.*
  - *$\forall (a, b) \in prnt.\ b \leq a \in \Sigma$.*[3]

The function $E(-)$ maps $\mathcal{B}$ to the SEL_Σ sequent:

$$\begin{aligned} \vdash_\Sigma & \{\texttt{in}(x, link(x), \texttt{id}) \mid x \in X\}, \{\texttt{out}(y, \texttt{id}) \mid y \in Y\}, \\ & \{r_i{:}\texttt{root}(\texttt{id}) \mid 0 \leq i \leq n-1\}, \{prnt(i){:}\texttt{site}(\texttt{id}) \mid 0 \leq i \leq m-1\}, \\ & \{prnt(\mathtt{v}){:}\texttt{control}(ctrl(\mathtt{v}), \mathtt{v}, [link((\mathtt{v}, 1)), \ldots, link((\mathtt{v}, ar(ctrl(\mathtt{v}))))]) \mid \mathtt{v} \in V\}. \end{aligned}$$

Note that all x, y and r_i are free in the resulting sequent.

[3] If a or b is a natural number $1 \leq i \leq m$ representing a root, than we map it to r_i in the subexponential signature.

As an example, if $\mathcal{B}$ is the bigraph of Figure 3, then $E(\mathcal{B})$ is the sequent:

$$\begin{aligned}\vdash_\Sigma\ & \texttt{in}(x_0, \texttt{e}_0, \texttt{id}), \texttt{in}(x_1, y_2, \texttt{id}),\\ & \texttt{out}(y_0, \texttt{id}), \texttt{out}(y_1, \texttt{id}), \texttt{out}(y_2, \texttt{id}),\\ & r_0{:}\texttt{root}(\texttt{id}), r_1{:}\texttt{root}(\texttt{id}), \texttt{v}_0{:}\texttt{site}(\texttt{id}), \texttt{v}_2{:}\texttt{site}(\texttt{id}),\\ & r_0{:}\texttt{control}(\texttt{A}, \texttt{v}_0, [y_0, \texttt{e}_0]), \texttt{v}_0{:}\texttt{control}(\texttt{C}, \texttt{v}_1, [y_0, \texttt{e}_1]),\\ & r_1{:}\texttt{control}(\texttt{B}, \texttt{v}_2, [y_1, y_2, \texttt{e}_0, \texttt{e}_1])\end{aligned}$$

Where $\Sigma = \langle \{r_0, r_1, \texttt{v}_0, \texttt{v}_1.\texttt{v}_2, 0, 1\}, \{\infty\}, \{r_0 \leq \texttt{v}_0 \leq \texttt{v}_1; \texttt{v}_0 \leq 0; r_1 \leq \texttt{v}_2 \leq 1\}\rangle$.

We will now show that it is possible to define a function that takes a sequent encoding a bigraph and returns the bigraph itself. In order to be a bigraph encoding, a *SEL* sequent must be of a specific shape and satisfy some conditions which are specified in the definition that follows.

Definition 6 (Decoding function). *Let S be the following SEL sequent:*

$$\begin{aligned}\vdash_\Sigma & \{\texttt{in}(x_i, l_{x_i}, \texttt{id}) \mid 0 \leq i \leq n_x - 1\}, \{\texttt{out}(y_i, \texttt{id}) \mid 0 \leq i \leq n_y - 1\},\\ & \{r_i{:}\texttt{root}(\texttt{id}) \mid 0 \leq i \leq n - 1\}, \{p_i{:}\texttt{site}(\texttt{id}) \mid 0 \leq i \leq n_v - 1\},\\ & \{p_i{:}\texttt{control}(\texttt{K}_j, \texttt{v}_i, L_j) \mid 0 \leq i \leq n_v - 1,\ 0 \leq j \leq n_k - 1\}\end{aligned}$$

for integers n, n_x, n_y, n_v and n_k, satisfying the following conditions:

- *x_i, y_i and r_i are free variables.*
- *L_i is a list of variables and constants.*
- *l_{x_i} is a variable that occurs as the first argument of an* `out` *predicate or a constant that occurs in a L_j for some j.*
- *p_i is either a variable that occurs as a label for some* `root`*, or a constant that occurs as the second argument for some* `control` *predicate different than $\texttt{v}_i$.*
- *$p_i \leq v_i \in \Sigma$ for each p_i labeling a $\texttt{control}(\texttt{K}_j, \texttt{v}_i, L_j)$.*
- *$p_i \leq i \in \Sigma$ for each p_i labeling a $\texttt{site}(\texttt{id})$.*
- *the relation $\leq$ in Σ defines a tree.*

Then, $D(S)$ is a bigraph $(V, E, ctrl, prnt, link) : \langle m, X\rangle \to \langle n, Y\rangle$ over a signature $\mathcal{K}$, where:

- *$\mathcal{K}$ contains all the $\texttt{K}_j$ with $ar(\texttt{K}_j) = |L_j|$;*
- *$V = \{v_i \mid 0 \leq i \leq n_v - 1\}$;*
- *E is the set of constants occurring in the lists L_i;*
- *$ctrl(v_i) = \texttt{K}_j$ if $l{:}\texttt{control}(\texttt{K}_j, \texttt{v}_i, L) \in S$ (for any L);*
- *$prnt$ is the same as the $\leq$ relation in $\Sigma \setminus \{\varepsilon, \infty\}$;*
- *$link(x_i) = l_{x_i}$ if $\texttt{in}(x_i, l_{x_i}, \texttt{id})$;*
- *$link((v, i)) = l$ if $\texttt{control}(_, \texttt{v}, L) \in S$ and $L[i] = l$;*
- *$X = \{x_i \mid 0 \leq i \leq n_x - 1\}$;*
- *$Y = \{y_i \mid 0 \leq i \leq n_y - 1\}$; and*
- *m is the number of $\texttt{site}(\texttt{id})$ predicates.*

We will now show that D is a left-inverse of E.

Theorem 3 (Representational adequacy). *Let $\mathcal{B}$ be a bigraph, $S = E(\mathcal{B})$ and $\mathcal{B}' = D(S)$. Then $\mathcal{B} = \mathcal{B}'$*[4]*.*

Proof (sketch). For each element $V, E, prnt, ctrl, link, m, n, X$ and Y of a bigraph, E maps it to a certain element of S and D maps it back. □

Theorem 4. *The function D is surjective, i.e., for every bigraph $\mathcal{B}$, there exists a SEL sequent S such that $D(S) = \mathcal{B}$.*

Proof. By the totality of E, we know that for every bigraph $\mathcal{B}$, there exists a sequent $S = E(\mathcal{B})$. From Theorem 3, we know that $D(S) = \mathcal{B}$. □

An encoding of a bigraph in *SEL* will allow us to use the tools available in the logic to reason about it. In particular, we can use the informations about subexponentials to easily deduce when a node v_i is contained in v_j, even if v_j is not v_i's immediate parent.

Theorem 5. *Let $\mathcal{B}$ be a bigraph over a signature $\mathcal{K}$. Assume that Σ is the subexponential signature generated by $E(\mathcal{B})$. If the sequent $\vdash_\Sigma !^{\mathtt{v}_j}(A)^\perp \parr ?^{\mathtt{v}_i} A$ is provable for an arbitrary A, then the node v_i is contained in node v_j (with an arbitrary number of nestings) in $\mathcal{B}$.*

Proof. While deriving this sequent in *SEL*, the promotion rule generates the side condition $\mathtt{v}_j \leq \mathtt{v}_i$. Since $\leq$ is transitive, and the parent relation is the kernel of $\leq$, we have $prnt^k(v_i) = v_j$ for some k, so v_i is contained in v_j in $\mathcal{B}$. □

3.2 Juxtaposition

One of the two basic combination mechanisms for bigraphs is juxtaposition, which places two disjoint bigraphs "side-by-side". The formal definition can be found in [10, Definition 2.7]. At a basic level, this is easily achieved in terms of *SEL* sequents as we merely have to take the multiset join of the two encodings of the constituent bigraphs. However, it is not entirely trivial, since the roots and sites need to be renumbered.

Definition 7 (Juxtaposition). *Given $\Sigma_1 = \langle \Lambda_1, \mathcal{U}_1, \leq_1 \rangle$ and $\Sigma_2 = \langle \Lambda_2, \mathcal{U}_2, \leq_2 \rangle$ with $\Lambda_1 \cap \Lambda_2 = \{\infty\}$ that are the subexponential signatures for the encoding of two bigraphs with interfaces $\langle n_1, X_1 \rangle \to \langle m_1, Y_1 \rangle$ and $\langle n_2, X_2 \rangle \to \langle m_2, Y_2 \rangle$ respectively, the signature $\Sigma_1 \mid \Sigma_2$ with corresponding substitution on locations σ is defined to be the structure $\langle \Lambda, \mathcal{U}, \leq \rangle$ where:*

- $\sigma = \{r_{n_1+i}/r_i \mid 0 \leq i < n_2\}$.[5]
- $\Lambda = \Lambda_1 \cup \Lambda_2\sigma$ *and* $\mathcal{U} = \mathcal{U}_1 \cup \mathcal{U}_2\sigma$.
- $x \leq y$ *iff* $x \leq_1 y$ *or* $x\sigma \leq_2 y\sigma$.

(We use $\mid$ instead of $\otimes$ to indicate juxtaposition to avoid confusion with the SEL connective.)

[4] We always consider bigraphs to be equal up to the renaming of elements.
[5] t/s denotes the substitution of s by t.

Theorem 6 (Adequacy of juxtaposition). *Given disjoint bigraphs $\mathcal{B}_1$ and $\mathcal{B}_2$, let $E(\mathcal{B}_1)$ be $\vdash_{\Sigma_1} \Gamma_1$ and $E(\mathcal{B}_2)$ be $\vdash_{\Sigma_2} \Gamma_2$, such that both encodings use the same identifier* $\mathtt{id}$*. Let σ be the substitution for $\Sigma_1 \mid \Sigma_2$. Then: $E(\mathcal{B}_1 \mid \mathcal{B}_2) = \vdash_{\Sigma_1|\Sigma_2} \Gamma_1, (\Gamma_2\sigma)$.*

Proof. Immediate by inspection.. □

Note that we only need to renumber the roots, since the sites are implicitly numbered in our encoding. This is a consequence of linearity (to distinguish sites) and subexponentials (used to name the immediate parent of every site).

3.3 Composition

The interface of a bigraph, *i.e.*, $\langle m, X\rangle \to \langle n, Y\rangle$, determines the available links and places for composition with another bigraph. A bigraph $\mathcal{B}_1 : \langle m_1, X_1\rangle \to \langle n_1, Y_1\rangle$ can be composed with a bigraph $\mathcal{B}_2 : \langle m_2, X_2\rangle \to \langle n_2, Y_2\rangle$ if $n_1 = m_2$ and $Y_1 = X_2$. This means that $\mathcal{B}_1$ has n_1 roots that will be placed inside $\mathcal{B}_2$'s m_2 sites, and all $\mathcal{B}_1$'s outer names Y_1 will be connected with $\mathcal{B}_2$'s inner names X_2. This composition is denoted by $\mathcal{B}_2 \circ \mathcal{B}_1$. For a formal definition of bigraph composition, see [10, Definition 2.5]. Such compositions can be adequately captured in our framework by a focused derivation of formulas which will encode the operations of (1) connecting links and (2) placing roots inside sites.

Let $S_1 = E(\mathcal{B}_1)$ be the sequent $\vdash_{\Sigma_1} \Gamma_1$ and $S_2 = E(\mathcal{B}_2)$ be $\vdash_{\Sigma_2} \Gamma_2$ such that the names in S_1 do not clash with those of S_2.[6]. The composition of these into one bigraph will require a common subexponential signature, defined below.

Definition 8. *Let $\Sigma_1 = \langle \Lambda_1, \mathcal{U}_1, \leq_1\rangle$ and $\Sigma_2 = \langle \Lambda_2, \mathcal{U}_2, \leq_2\rangle$ be such that $\Lambda_1 \cap \Lambda_2 = \{\infty\}$. Then, $\Sigma_2 \circ \Sigma_1$ is the signature $\langle \Lambda, \mathcal{U}, \leq\rangle\,\sigma$ such that*

$$\Lambda = \Lambda_1 \cup \Lambda_2, \quad \mathcal{U} = \mathcal{U}_1 \cup \mathcal{U}_2, \quad \leq = \leq_1 \cup \leq_2, \quad \sigma = \{\mathtt{v}_i/r_j \mid v_i \in \Lambda_2 \text{ and } r_j \in \Lambda_1\}.$$

The substitution σ defines a map between sites and roots.[7] *In the subexponential signature, this means that the root r_j will be instantiated with the index $\mathtt{v}_i$ at the same time that one of its successors (i.e., a site) is removed, which is equivalent to replacing a numeric successor of v_i by the subtree of r_j.*

The composed subexponential signature yields the definition of the formulas encoding the operations of linking and placing. Consider the sequent:

$$\vdash_{\Sigma} \Gamma_1, \Gamma_2, \infty{:}\exists e.\ (\mathtt{in}(e, e, \mathtt{id2}))^{\perp} \otimes (\mathtt{out}(e, \mathtt{id1}))^{\perp} \otimes \bot$$

where all free variables are interpreted as existential variables, *i.e.*, they are unified with terms in applications of the [init] rule. We will henceforth denote this existentially quantified formula in zone ∞ by $\mathtt{CF}(\mathcal{B}_1, \mathcal{B}_2)$. Focusing on $\mathtt{CF}(\mathcal{B}_1, \mathcal{B}_2)$ will result on three premises:

$$P_1 : \vdash_{\Sigma} \Gamma, (\mathtt{in}(\mathtt{e}, \mathtt{e}, \mathtt{id2}))^{\perp} \qquad P_2 : \vdash_{\Sigma} \Gamma', (\mathtt{out}(\mathtt{e}, \mathtt{id1}))^{\perp} \qquad P_3 : \vdash_{\Sigma} \Gamma'', \bot$$

[6] This is always possible due to renaming and α-equivalence.

[7] Note that, by the definition of substitution, the r_j must be pairwise distinct. In contrast, $\mathtt{v}_i$ can be repeated in case a node contains more than one site.

In order to prove P_1 and P_2 it must be the case that $\Gamma = \{\mathtt{in}(x, \mathtt{e}, \mathtt{id2})\}$ and $\Gamma' = \{\mathtt{out}(y, \mathtt{id1})\}$. Since these are linear resources, they will not be in Γ''. The unification of the atoms in the initial rule will generate the substitutions $[\mathtt{e}/x]$ and $[\mathtt{e}/y]$, which will be propagated to the formulas in Γ''. While the substitution on x will have no effect (each inner channel occurs only once in the encoded sequent, see Definition 5), the substitution on y will rename an outer channel to a concrete edge which is connected to another component. It will instantiate variables in some `control`'s list of links L, *i.e.*, a `control` in $\mathcal{B}_1$ which was connected to an outer channel y will become connected to a real edge `e` in $\mathcal{B}_2$.

The constants `id1` and `id2` guarantee that such channels belong to different bigraphs. Each time a focused derivation is performed on this formula, a connection will be made. All connections will have been made once the formula can no longer be derived.

It is also possible to perform the linking without using free variables and unification by first guessing a linking and then performing proof search to ensure that such a linking exists. Since the bigraphs are composable by assumption, a valid instantiation for the variables always exists. Our search procedure will have to check that all the in links of the outer bigraph have been instantiated in either view of the linking procedure.

The next step is to place $\mathcal{B}_1$'s roots inside $\mathcal{B}_2$'s sites. For this operation, we will need to reason on (and unify) subexponentials. The formula used for placement is constructed according to the sites available in $\mathcal{B}_2$. Let $L = \{l \mid l{:}\mathtt{site}(\mathtt{id2}) \in \Gamma_2\}$; then: $\vdash_\Sigma \Gamma_1, \Gamma_2, \left\{(?^l\, \mathtt{site}(\mathtt{id2}))^\perp \otimes (?^l\, \mathtt{root}(\mathtt{id1}))^\perp \otimes \bot \mid l \in L\right\}$ is derivable. We will henceforth denote this defined set of formulas as $\mathtt{PF}(\mathcal{B}_1, \mathcal{B}_2)$. A focused derivation on each of formulas in $\mathtt{PF}(\mathcal{B}_1, \mathcal{B}_2)$ will behave similarly to the case of `CF`: three premises will be generated.

$$P_1 : \vdash_\Sigma \Gamma, l{:}(\mathtt{site}(\mathtt{id2}))^\perp \qquad P_2 : \vdash_\Sigma \Gamma', l{:}(\mathtt{root}(\mathtt{id1}))^\perp \qquad P_3 : \vdash_\Sigma \Gamma'', \bot$$

And in order to prove P_1 and P_2 it is necessary that $\Gamma = \{l{:}\mathtt{site}(\mathtt{id2})\}$ and $\Gamma' = \{r{:}\mathtt{root}(\mathtt{id1})\}$. Since r is a variable, we obtain the substitution $[l/r]$ when proving P_2 which is propagated to Γ''. The variable r represented a root in $\mathcal{B}_1$, while l is the place where a site is located in $\mathcal{B}_2$. Note that Γ'' might have occurrences of r as the label of other formulas, and by replacing it with l, we are in fact changing the location of some formulas. They are moved from the root r in $\mathcal{B}_1$ to a location l in $\mathcal{B}_2$, which is exactly what happens during composition of places. The subexponential signature also needs to be changed accordingly, and this is done by accumulating this substitution in Σ's σ.

Note that, since `PF` is a linear set, the connection of roots to nodes will only be complete once all formulas in the set are derived. Of course this will be composed of many focused phases.

We can now combine the operations of linking and placement to obtain a sequent whose derivation will result on a premise that encodes the composition of two bigraphs. We will require that such derivation is *exhaustive*, meaning that the formulas in ∞ are focused on until they can no longer be derived and there are no positive formulas in ε.

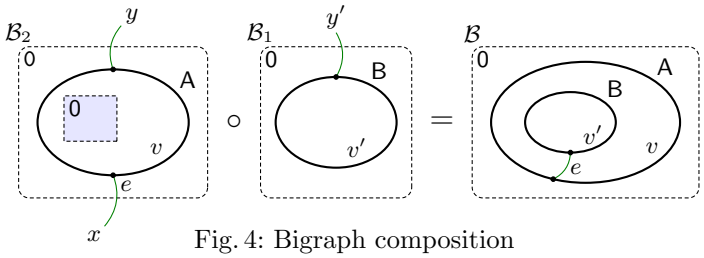


Fig. 4: Bigraph composition

Theorem 7 (Adequacy of bigraph composition). *Let $\mathcal{B}_1$ and $\mathcal{B}_2$ be two bigraphs such that their composition $\mathcal{B}_2 \circ \mathcal{B}_1 = \mathcal{B}$ is well defined. Let $E(\mathcal{B}_1) = \vdash_{\Sigma_1} \Gamma_1$, $E(\mathcal{B}_2) = \vdash_{\Sigma_2} \Gamma_2$ and $\Sigma = \Sigma_2 \circ \Sigma_1$. Then the exhaustive focused derivation of the sequent $\vdash_\Sigma \Gamma_1, \Gamma_2, \mathtt{CF}(\mathcal{B}_1, \mathcal{B}_1), \mathtt{PF}(\mathcal{B}_1, \mathcal{B}_2)$ will have exactly one open premise $\vdash_\Sigma \Gamma, \infty{:}\Delta$ with $D(\vdash_\Sigma \Gamma) = \mathcal{B}$.*

Proof (sketch). After exhaustive proof search, the failure to derive $\mathtt{CF}(\mathcal{B}_1, \mathcal{B}_2)$ and the derivation of all formulas in $\mathtt{PF}(\mathcal{B}_1, \mathcal{B}_2)$ means that there are no more $\mathtt{in}(x, e, \mathtt{id2})$, $\mathtt{out}(y, \mathtt{id1})$, $\mathtt{site}(\mathtt{id2})$ and $\mathtt{root}(\mathtt{id1})$ in the sequent. The fact that the bigraphs were compatible for composition in the first place, and the unique identifier in the atoms, guarantees that each of the predicates occurring in the sequent have the same multiplicity (number of inner links and sites of $\mathcal{B}_2$ must be the same as the number of outer links and roots of $\mathcal{B}_1$, respectively). The unifications during the derivations of $\mathtt{CF}$'s will ensure that outer links in $\mathcal{B}_1$ are renamed to the proper edges or outer links of $\mathcal{B}_2$. The unifications during the derivations of $\mathtt{PF}$'s will ensure that all elements inside of a root in $\mathcal{B}_1$ are correctly placed in the parent of the respective site in $\mathcal{B}_2$. Therefore, decoding the sequent without the ∞-formulas produces a bigraph with the desired structure. □

As an example, consider the composition depicted in Figure 4. The encoding of each operand will be:

$$
\begin{aligned}
E(\mathcal{B}_1): \quad & \Gamma_1 = \mathtt{out}(y', \mathtt{id1}), r_0'{:}\mathtt{root}(\mathtt{id1}), r_0'{:}\mathtt{control}(\mathtt{B}, \mathtt{v'}, [y']) \\
& \Sigma_1 = (\{r_0', \mathtt{v'}, \infty\}, \{\infty\}, \{r_0' \leq_1 \mathtt{v'}\}) \\
E(\mathcal{B}_2): \quad & \Gamma_2 = \mathtt{in}(x, \mathtt{e}, \mathtt{id2}), \mathtt{out}(y, \mathtt{id2}), r_0{:}\mathtt{root}(\mathtt{id2}), \\
& \qquad \mathtt{v}{:}\mathtt{site}(\mathtt{id2}), r_0{:}\mathtt{control}(\mathtt{A}, \mathtt{v}, [y, \mathtt{e}]) \\
& \Sigma_2 = (\{r_0, \mathtt{v}, 0, \infty\}, \{\infty\}, \{r_0 \leq_2 \mathtt{v} \leq_2 0\})
\end{aligned}
$$

The formulas encoding the operation are:

$$
\begin{aligned}
\mathtt{CF}(\mathcal{B}_1, \mathcal{B}_2): \quad & \infty{:}\exists e.\, \mathtt{in}(e, e, \mathtt{id2})^\perp \otimes \mathtt{out}(e, \mathtt{id1})^\perp \otimes \perp \\
\mathtt{PF}(\mathcal{B}_1, \mathcal{B}_2): \quad & \left\{\varepsilon{:}(?^v\, \mathtt{site}(\mathtt{id2}))^\perp \otimes (?^v\, \mathtt{root}(\mathtt{id1}))^\perp \otimes \perp\right\}
\end{aligned}
$$

The formula $\mathtt{CF}$ can only be derived once, indeed there is only one formula of each kind: $\mathtt{in}(x, \mathtt{e}, \mathtt{id2})$ and $\mathtt{out}(y', \mathtt{id1})$. As a result of this derivation, we will obtain the following substitution: $[\mathtt{e}/x, y']$. The derivation of the only formula in $\mathtt{PF}$ will provide the substitution $\sigma = [\mathtt{v}/r_0']$ for the subexponentials and formula labels. The resulting open sequent will be $\vdash_\Sigma \Gamma$, where:

$$\Gamma = \varepsilon : \mathtt{out}(y, \mathtt{id2}),\ r_0 : \mathtt{root}(\mathtt{id2}),\ r_0 : \mathtt{control}(\mathtt{A}, \mathtt{v}, [y, \mathtt{e}]),\ \mathtt{v} : \mathtt{control}(\mathtt{B}, \mathtt{v'}, [\mathtt{e}])$$

$\Sigma = (\{r_0, \mathtt{v}, 0, \mathtt{v'}, \infty\}, \{\infty\}, \{r_0 \leq \mathtt{v} \leq \mathtt{v'}\})$.

4 Conclusions, Related Work, and Perspectives

Conclusions. Using the expressive power of subexponentials, we have given an adequate encoding of pure bigraphs in *SEL*. We were able to reason about the static structure of bigraphs with only the subexponential signature generated from the encoding. With the full sequent encoding of two bigraphs, we can easily simulate juxtaposition and we can emulate composition using a (focused) derivation in *SELF*.

Sorting. In [10, Chapter 6], Milner shows how to add sorts to links and places, and to define formation rules that will restrict the possible structures of a bigraph. Likewise the place and link graphs of a bigraph, the place and link sorts (and their formation rules) are orthogonal. In our encoding, we would use a multi-sorted version of *SEL* and assign the correct sorts for the variables and constants. The formation rules, by definition, are preserved under composition and juxtaposition, and as our encoding is adequate, they will also be preserved when operating on the meta-level.

Link sorting assigns sorts for each member of the arity of the controls and additionally to the inner and outer edges. The formation rules dictate what sorts of links a hyperedge may have in its extremities. In our encoding, we could transform each control `K` into a function symbol taking a list of arguments of appropriate type. In this case, the predicates $\mathtt{control}(_, \mathtt{K}, [l_1, \ldots, l_n])$ become $\mathtt{control}(_, \mathtt{K}(l_1, \ldots, l_n))$. Since `K` is always a constant in the encoding, this change is harmless. As for the inner and outer edges, we simply need to assign the correct types to the x and y variables. Unification will guarantee that the correct type is used on the connections when operating on the encoded bigraphs.

Place sorting assigns sorts for each control K and formation rules determine what sorts can be contained in others. In our encoding, controls are assigned to vertices via the `control` predicate, but the containment relation is modeled by the relation between subexponentials. For this reason, adding sorts to places is a little more tricky then link sorting, but still possible. Sorts now have to be assigned to the `v` constants, representing nodes and, consistently, to the `v` subexponentials. Note that we bypass the sorts of controls and deal directly with the nodes (except sites and roots).

Bigraphs with Sharing. The pure bigraphs (with or without sorting) presented so far are sometimes not expressive enough; in particular, the tree-nature of the place graph is often too restrictive. In [16], the place graph was generalized from trees to arbitrary DAGs, allowing it to model an arbitrary Boolean combination of places. For our encoding, supporting DAGs is no harder than trees: our $\leq$ relation simply needs to be generalized to partial orders. This is easy: the meta-theory of *SEL* only requires $\leq$ to be a pre-order, so arbitrary directed graphs (acyclic or not) can be supported. The rest of the encoding remains unchanged.

Directed Bigraphs. A comparatively non-trivial extension is to move from an arbitrary (possibly sorted) hypergraph for the link graph to a directed link graph with link names as nodes [7]. While such a change of perspective does not need a

modification of the encoding of bigraphs in this paper, adequacy is a bit more delicate. Specifically, the decoding function D needs to know the exact signature of link names since the encoding represents link names as constants and not as linear resources. If we were to change the encoding to track the link names linearly, then the composition operation will need to be extended with new rules for pairing up the link names.

Perspectives. The most interesting followup to this work would be to develop the necessary reasoning on *SEL* encodings in order to get a bisimulation of bigraph *reactions*. In this way, we would have a full syntactical description of bigraphs with the advantage of having available automated proof theory tools.

Acknowledgment: This work was partially supported by the ERC Advanced Grant *ProofCert*.

References

1. M. Beauquier and C. Schürmann. A bigraph relational model. In *LFMTP*, volume 71 of *EPTCS*, pages 14–28, 2011.
2. L. Cardelli. Brane calculi. In *CMSB*, volume 3082 of *LNCS*, pages 257–278. Springer, 2004.
3. L. Cardelli and A. Gordon. Mobile ambients. In *FOSSACS'98*, volume 1378 of *LNCS*. Springer, 1998.
4. K. Chaudhuri. Classical and intuitionistic subexponential logics are equally expressive. In *CSL*, volume 6247 of *LNCS*, pages 185–199. Springer, Aug. 2010.
5. K. Chaudhuri. Undecidability of multiplicative subexponential logic. In *3rd LINEARITY*, volume 176 of *EPTCS*, pages 1–8, July 2014.
6. V. Danos, J.-B. Joinet, and H. Schellinx. The structure of exponentials: Uncovering the dynamics of linear logic proofs. In *Kurt Gödel Colloquium*, volume 713 of *LNCS*, pages 159–171. Springer, 1993.
7. D. Grohmann and M. Miculan. Directed bigraphs. *ENTCS*, 173:121–137, 2007.
8. O. H. Jensen and R. Milner. Bigraphs and mobile processes (revised). Technical Report UCAM-CL-TR-580, University of Cambridge, Feb. 2004.
9. R. Milner. Pure bigraphs: Structure and dynamics. *Information and Computation*, 204(1):60 – 122, 2006.
10. R. Milner. *The Space and Motion of Communicating Agents*. Cambridge University Press, 2009.
11. V. Nigam. *Exploiting non-canonicity in the sequent calculus*. PhD thesis, Ecole Polytechnique, Sept. 2009.
12. V. Nigam and D. Miller. Algorithmic specifications in linear logic with subexponentials. In *PPDP*, pages 129–140, 2009.
13. V. Nigam, C. Olarte, and E. Pimentel. On subexponentials, focusing and modalities in concurrent systems. Draft Manuscript submitted for publication, 2015.
14. V. Nigam, E. Pimentel, and G. Reis. An extended framework for specifying and reasoning about proof systems. *J. of Logic and Computation*, 2014.
15. G. Păun. Membrane computing. In *Handbook of Natural Computing*, pages 1355–1377. Springer, 2012.
16. M. Sevegnani and M. Calder. Bigraphs with sharing. *Theoretical Computer Science*, 577:43–73, 2015.