Part I : Space

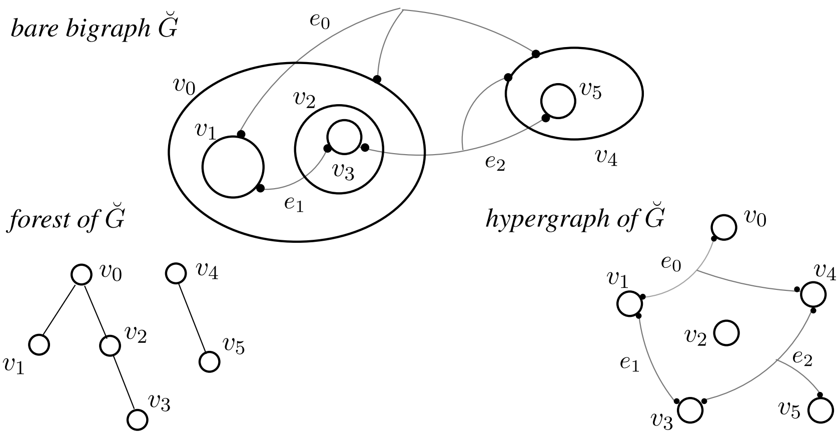
1
The idea of bigraphs

In this chapter we develop the notion of a *bigraph* from the simple idea that it consists of two independent structures on the same set of nodes.

To prepare for the formal Definitions 1.1–2.7, we start informally from two well-known concepts: a *forest* is a set of rooted trees; and a *hypergraph* consists of a set of nodes, together with a set of edges each linking any number of nodes.

Idea *A bigraph with nodes V and edges E has a forest whose nodes are V; it also has a hypergraph with nodes V and edges E.*

Let us call an entity with this structure a *bare bigraph*. We shall use $\breve{F}$, $\breve{G}$ to stand for bare bigraphs. Here is a bare bigraph $\breve{G}$ having nodes $V = \{v_0, \ldots, v_5\}$ and edges $E = \{e_0, e_1, e_2\}$, with its forest and hypergraph:

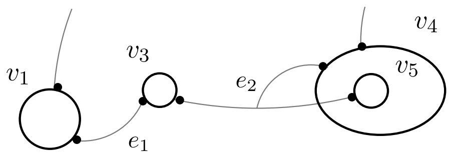


The upper diagram presents both the forest and the hypergraph; it depicts the forest by nesting. The lower two diagrams represent the two structures separately, in a conventional manner. The *children* of each node are the nodes immediately below it in the forest (i.e. immediately within it, in the upper diagram). Thus v_1 and v_2 are children of v_0, which is their *parent*.

An edge is represented by connected thin lines; $\breve{G}$ has two edges that each connect three nodes, and one that connects two nodes. The points at which an edge impinges on its nodes are called *ports*, shown as black blobs.[1]

We now add further structure to a bare bigraph. It will allow bigraphs to be composed, and will allow one bigraph to be considered as a component of another. Here is $\breve{F}$, informally a 'part' of $\breve{G}$, having only some of its nodes and with one hyperlink broken. Can we call it a component of $\breve{G}$?

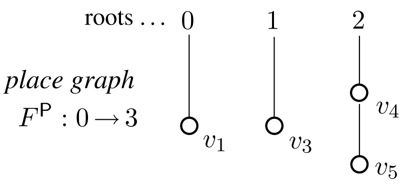
bare bigraph $\breve{F}$

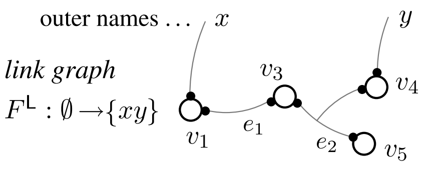


To make it so, we add *interfaces* to bare bigraphs, thus extending $\breve{F}$ and $\breve{G}$ into bigraphs F and G. This will allow us to represent the occurrence of F as a component of G by an equation $G = H \circ F$, where H is some 'host' or contextual bigraph. We do this extension independently for forests and hypergraphs; a forest with interfaces will be called a *place graph*, and a hypergraph with interfaces will be called a *link graph*.

Let us illustrate with the bare bigraph $\breve{F}$. A place graph interface will be a natural number n, which we shall treat as a finite ordinal, the set $n = \{0, 1, \ldots, n–1\}$ whose members are all preceding ordinals. A place graph's *outer* and *inner* interfaces – or *faces* as we shall call them – index respectively its *roots* and its *sites*. For the forest of $\breve{F}$ we choose the outer face $3 = \{0, 1, 2\}$, providing distinct roots as parents for the nodes v_1, v_3 and v_4. For the inner face of $\breve{F}$ we choose 0, i.e. it has no sites. This extends the forest to a place graph $F^{\mathsf{P}} : 0 \to 3$, an arrow in a precategory[2] whose objects are natural numbers. It is shown at the left of the diagram below.

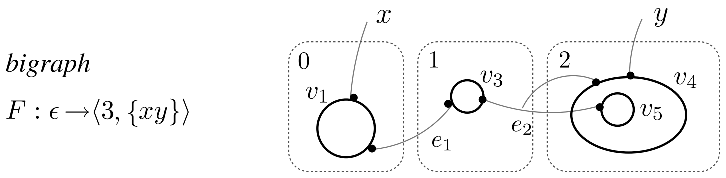
[1] By making ports explicit we permit distinct roles to be played by the edges impinging on a given node, just as each argument of a given mathematical function plays a distinct role.
[2] We shall define precategories in Chapter 2. For now, it is enough to know that a precategory has two kinds of entity, *objects* and *arrows*; that each arrow goes from a tail to a head, both of which are objects; and that these entities behave nicely together. Both objects and arrows may have all kinds of structure.





The outer and inner faces of a link graph are *name-sets*: respectively, its *outer* and *inner names*. For the hypergraph of $\breve{F}$ we choose outer face $\{xy\}$, thus naming the parts of the broken hyperlink, and inner face $\emptyset$.[3] This extends the hypergraph to a link graph $F^{\mathsf{L}} : \emptyset \to \{xy\}$, an arrow in a precategory whose objects are finite name-sets. Names are drawn from a countably infinite vocabulary $\mathcal{X}$.

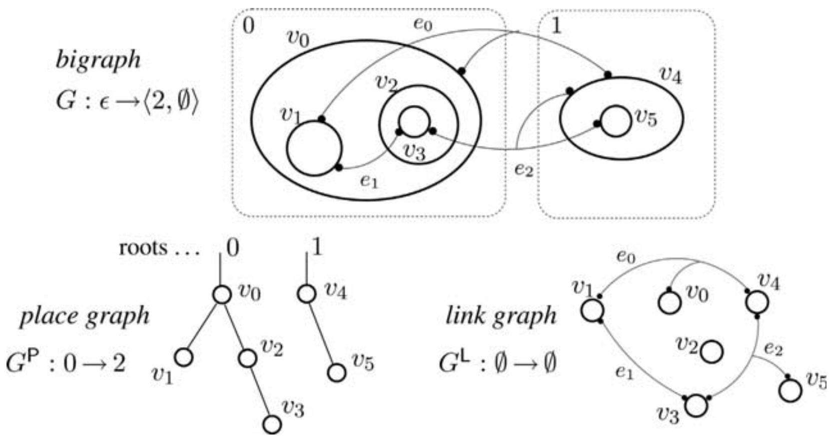
Finally, a *bigraph* is a pair $B = \langle B^{\mathsf{P}}, B^{\mathsf{L}} \rangle$ of a place graph and a link graph; these are its *constituents*. Its outer face is a pair $\langle n, Y \rangle$, where n and Y are the outer faces of B^{P} and B^{L} respectively. Similarly for its inner face $\langle m, X \rangle$. For our example $F = \langle F^{\mathsf{P}}, F^{\mathsf{L}} \rangle$ these pairs are $\langle 3, \{xy\} \rangle$ and $\langle 0, \emptyset \rangle$ respectively. We call the trivial interface $\epsilon \overset{\text{def}}{=} \langle 0, \emptyset \rangle$ the *origin*. Thus $\breve{F}$ is extended to an arrow $F : \epsilon \to \langle 3, \{xy\} \rangle$ in a precategory whose objects are such paired interfaces. F will be drawn as follows:



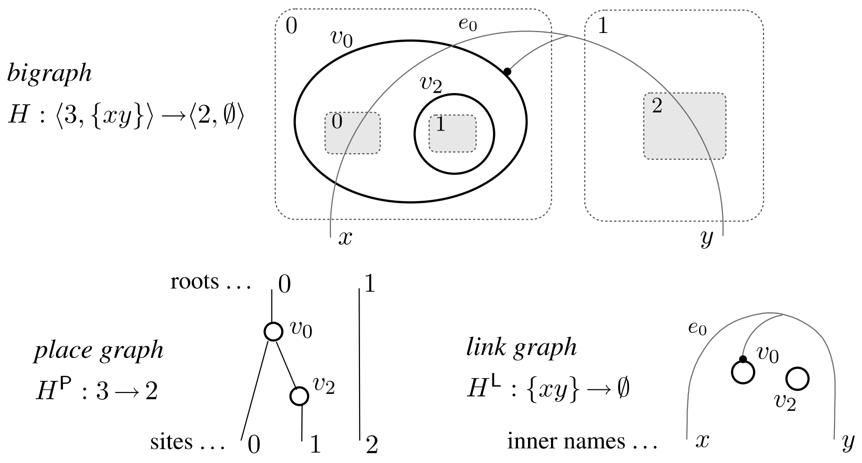
The rectangles in F – sometimes called *regions* – are just a way of drawing its roots, seen also in F^{P}. The link graph F^{L} has four *links*. Two of these are the edges e_1 and e_2, also called *closed* links; the other two are named x and y, and are called *open* links.

Let us now add interfaces to the bare bigraph $\breve{G}$, extending it into a bigraph G. It has no open links, i.e. all its links are edges, so the name-set in its outer face will be empty. Let us give it two roots; then, if G is placed in some larger context, v_0 and v_4 may be in distinct places – i.e. may have distinct parents. The diagram below shows G and its constituents. Note that there is no significance in where a link 'crosses' the boundary of a node or region in a bigraph; this is because the forest and hypergraph structures are independent.

[3] We use single letters for names, so we shall often write a set $\{x, y, \ldots\}$ of names as $\{xy \cdots\}$, or even as $xy \cdots$, when there is no ambiguity.


bigraph
$G : \epsilon \to \langle 2, \emptyset \rangle$
0
1
v_0
v_1
v_2
v_3
v_4
v_5
e_0
e_1
e_2
roots . . .
place graph
$G^{\mathsf{P}} : 0 \to 2$
link graph
$G^{\mathsf{L}} : \emptyset \to \emptyset$

We are now ready to construct a bigraph H such that $G = H \circ F$, illustrating composition, which will later be defined formally. The inner face of H must be $\langle 3, \{xy\} \rangle$, the outer face of F; to achieve this, H must have three *sites* $0, 1$ and 2, and inner names x and y. Here are H and its constituents, with sites shown as shaded rectangles:


bigraph
$H : \langle 3, \{xy\} \rangle \to \langle 2, \emptyset \rangle$
0
1
2
v_0
v_2
e_0
x
y
roots . . .
sites . . .
place graph
$H^{\mathsf{P}} : 3 \to 2$
link graph
$H^{\mathsf{L}} : \{xy\} \to \emptyset$
inner names . . .

In the place graph, each site and node has a parent, a node or root; in the link graph, each inner name and port belongs to a link, closed or open. Just as it is insignificant where links 'cross' node or root boundaries, so it is insignificant where they 'cross' a site. We draw inner names below the bigraph and outer names above it; this is merely a convention to indicate their status as inner or outer. A name may be both inner and outer, whether or not in the same link.

In general, let $F : I \to J$ and $H : J \to K$ be two bigraphs with disjoint nodes and edges, where $I = \langle k, X\rangle$, $J = \langle m, Y\rangle$ and $K = \langle n, Z\rangle$. Then the composite bigraph $H \circ F : I \to K$ is just the pair of composites $\langle H^{\mathsf{P}} \circ F^{\mathsf{P}}, H^{\mathsf{L}} \circ F^{\mathsf{L}}\rangle$, whose constituents are constructed as follows (informally):

(i) To form the place graph $H^{\mathsf{P}} \circ F^{\mathsf{P}} : k \to n$, for each $i \in m$ join the ith root of F^{P} with the ith site of H^{P};
(ii) To form the link graph $H^{\mathsf{L}} \circ F^{\mathsf{L}} : X \to Z$, for each $y \in Y$ join the link of F^{L} having the outer name y with the link of H^{L} having the inner name y.

Thus H and F are joined at every place or link in their common face J, which ceases to exist. The reader may like to check these constructions for H and F as in our example.

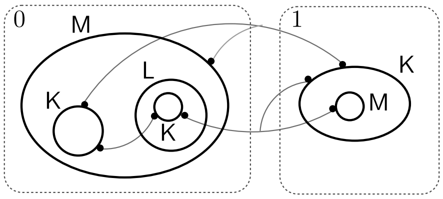
In our formal treatment, operations on bigraphs will be defined in terms of their constituent place and link graphs. But it is convenient, and even necessary for practical purposes, to have diagrams not only for the constituents but for the bigraphs themselves, such as for F, G and H in the example above. Such a diagram must be to some extent arbitrary, because we are trying to represent placing and linking, which are independent, in two dimensions! In particular, note that we have drawn outer names above the picture (in F and G for example), and we have drawn inner names below the picture (in H for example). Other conventions are possible.

It will be helpful to look now at Figure 1.2, at the end of this chapter, showing the anatomical elements of bigraphs that will later be defined formally. In the present chapter we give only one formal definition, which determines how to introduce different kinds of node for different applications.

Definition 1.1 (basic signature) A *basic signature* takes the form $(\mathcal{K}, ar)$. It has a set $\mathcal{K}$ whose elements are kinds of node called *controls*, and a map $ar : \mathcal{K} \to \mathbb{N}$ assigning an *arity*, a natural number, to each control. The signature is denoted by $\mathcal{K}$ when the arity is understood. A bigraph over $\mathcal{K}$ assigns to each node a control, whose arity indexes the *ports* of a node, where links may be connected. □

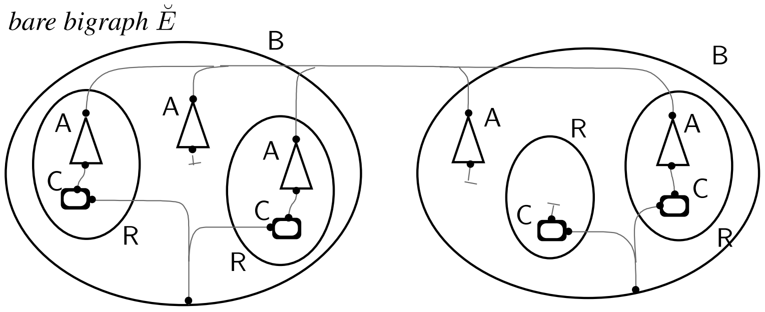
A signature suitable for our example is $\mathcal{K} = \{\mathsf{K}:2,\ \mathsf{L}:0,\ \mathsf{M}:1\}$. (Thus arities are made explicit.) Here is our bigraph $G : \epsilon \to \langle 2, \emptyset\rangle$, with controls assigned to the nodes:

bigraph G
with controls



We have omitted node- and edge-identifiers, as we often shall when they are irrelevant. To end this chapter, let us look at a realistic (but simplified) example, which indicates that bigraphs can go beyond the usual topics for process calculi.

Example 1.2 (a built environment) The next diagram shows a bare bigraph $\breve{E}$ over the signature $\mathcal{K} = \{\mathsf{A}:2,\ \mathsf{B}:1,\ \mathsf{C}:2,\ \mathsf{R}:0\}$, which classifies nodes as agents, buildings, computers and rooms. The node-shapes are not significant, except to indicate the purpose of each port. The figure represents a state which may change because of the movement of agents, and perhaps other movements. Think of the five agents as conducting a conference call (the long link). An agent in a room may also be logged in (the short links) to a computer in the room, and the computers in a building are linked to form a local area network. □



Bearing in mind our earlier example, the following exercise will be instructive.

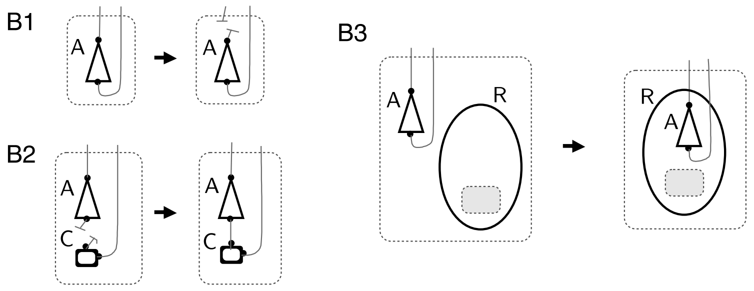
EXERCISE 1.1

(1) Draw a bare bigraph $\breve{D}$ representing the three agents that are inside rooms. Make this into a bigraph D by defining its outer face.

(2) Propose an outer face that makes $\breve{E}$ into a bigraph E, allowing the possibility that the two buildings may be situated in different cities. Draw the bigraph C, with sites, such that $C \circ D = E$. □

Although the detailed study of dynamics is deferred to Part II, let us now illustrate how bigraphs can reconfigure themselves. We are free to define different reconfigurations for each application. This is done by *reaction rules* each consisting of a *redex* (the pattern to be changed) and a *reactum* (the changed pattern). Part of the idea of bigraphs is that these changes may involve both placing and linking.

The redex and reactum of a rule are themselves bigraphs, and may match any part of a larger bigraph. (This remark will be made precise in Part II.) Here are three possible rules for built environments, such as the system E:



Rule B1 is the simplest: an agent can leave a conference call. The redex – the left-hand pattern – can match any agent; the out-pointing links mean that either of her ports may at first be linked to *zero or more* other ports, in the same place or elsewhere. If she is linked in a conference call to other agents, perhaps in other buildings, the reaction by B1 will unlink her; any link to a computer is retained.

Rule B2 shows a computer connecting to an agent in the same place (presumably a room). The redex insists that at first the agent is linked to no computer and the computer is linked to no agent. Rules B1 and B2 change only the linking – not the placing – in a bigraph, though the redex of B2 does insist on juxtaposition.

Rule B3, by contrast, changes the placing; an agent enters a room. Again, the rule requires the agent and the room to be in the same place (presumably a building). The site (shaded) allows the room to contain other occupants, e.g. a computer and other agents. The matching discipline allows these occupants to be linked anywhere, either to each other or to nodes lying outside the room.

Another feature of B3 is that its redex allows the lower port of the agent to be already linked to a computer somewhere, perhaps in another room. B3 retains any such link. Equally, there may be no such link – the context in which the rule is applied may close it off. Thus B3 can be applied to the system represented by $\breve{E}$, or E, allowing an agent in the right-hand building to enter a room.

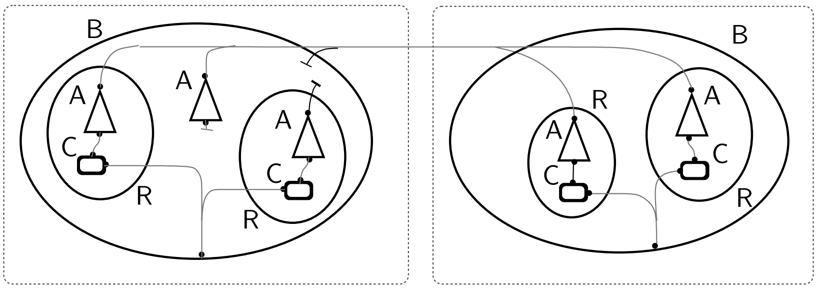
Taking this a step further, observe that in E an agent and a computer are linked only when they occupy the same room. Moreover, starting from E, our rules B1–B3 will preserve this property, since only B3 creates such links, and only within a room. We therefore call the property an *invariant* for E in the system with this rule-set. We now briefly discuss invariants.

Given a rule-set, we refer to the configurations that a system may adopt as *states*. The rule-set determines a reaction relation $\longrightarrow\!\!\triangleright$ between states. The diagram below shows the state E_3 adopted by E after three reactions

$$E \longrightarrow\!\!\triangleright E_1 \longrightarrow\!\!\triangleright E_2 \longrightarrow\!\!\triangleright E_3 \;;$$

in the first, B1 is applied to the third agent from the left; in the second, B3 is applied to the fourth agent; this enables B2 to be applied to that agent in the third reaction.

bigraph E_3



We say that a property of states is (an) *invariant* for E (under a given rule-set) if it holds for all states reachable from E via reactions permitted by the rule-set, i.e. it holds for all E' such that $E \longrightarrow\!\!\triangleright \cdots \longrightarrow\!\!\triangleright E'$. For example, under the rule-set B1–B3, the property 'there are exactly five agents' is invariant for E.

Of course, our present rule-set is very limited. The following exercise suggests how to enrich this rule-set a little, and explores what invariants may then hold.

EXERCISE 1.2

(1) Add a rule B4 to enable an agent linked with a computer to sever this link, and another rule B5 to allow an agent unlinked to a computer to leave a room. Give a few examples of invariants for E under the rule-set B1–B5.

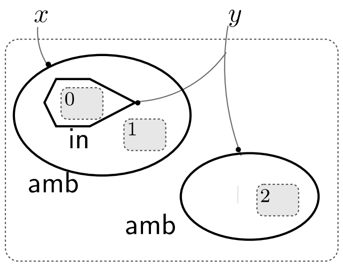
(2) Instead of B4 and B5, design a single rule B6 that allows an agent to leave a room, simultaneously severing any link with the computer. How does this change affect your invariants? □

Our behavioural model of the occupants of a building is crude, of course. But reaction rules of this kind, hardly more complex, are beginning to find realistic application in biological modelling. A crucial refinement is to add stochastic information that determines which reactions are more likely to occur, and therefore to preempt others. In the built environment, an interesting refinement is to allow agents to discover who is where, and record this information via the computers; these stories can then be combined so that the system becomes *reflective*, meaning that it can represent (part of) itself, and answer questions such as 'where is agent X?'.[4]

In another direction, bigraphs can model process calculi. In this case, the controls of a bigraph represent the constructors of the calculus. As an example, we

[4] These experimental applications are discussed, with citations, in Chapter 12.

take the calculus of mobile ambients, which partly inspired the bigraph model. In mobile ambients the main constructor is ‘amb’ with arity 1, representing an *ambient* – a region within which activity may occur; its single port allows an ambient to be named. Other constructors represent commands, or capabilities.



The above diagram shows two ambients, each with arbitrary content represented by the sites; one ambient also contains an ‘in’ capability, which refers to the other ambient by name. Let us use this example to illustrate the algebraic language for bigraphs, which we shall develop in later chapters. Here is the algebraic term for the above system:

$$\mathsf{amb}_x.(\mathsf{in}_y.d_0 \mid d_1) \mid \mathsf{amb}_y.d_2\ .$$

The combinator ‘|’ represents juxtaposition, and is commutative and associative; the combinator ‘.’ denotes nesting. We shall see in Chapter 3 that both combinators are derived from the categorical operations of composition and tensor product. The metavariables d_0, d_1 and d_2 stand for parameters, i.e. arbitrary occupants of the sites.

Let us now look at the dynamics of ambients. The above bigraph is, in fact, the redex of one of the reaction rules for mobile ambients, three of which are shown in Figure 1.1. In the first rule, the ‘in’ command causes its parent ambient named x, together with all its other contents, to move inside the ambient named y. The ‘in’ command, having done its job, vanishes; this exposes its contents to reactions with the ambient’s other occupants. Note that reconfiguration is permitted within an ‘amb’ node, but not within an ‘in’ node; the occupant of an ‘in’ node has a potential for interaction, which becomes actual only when the node itself has vanished.

In the second rule, conversely, the ‘out’ command causes the exit of its parent ambient from its own parent. These two rules provide our first example of moving sub-bigraphs from one region to another.

Finally, in the third rule the ‘open’ command causes an ambient node to vanish, exposing its contents to interactions in a wider region.

EXERCISE 1.3 Modify rule A3 to use a ‘send’ command instead of ‘open’. It