

# Abstraction in Reasoning about Higraph-Based Systems

John Power and Konstantinos Tourlas\*

Division of Informatics, The University of Edinburgh, King's Buildings, EH9 3JZ,  
United Kingdom  
{ajp,kxt}@inf.ed.ac.uk

**Abstract.** Higraphs, a kind of hierarchical graph, underlie a number of sophisticated diagrammatic formalisms, including Statecharts. Naturally arising from hierarchy in higraphs is an abstraction operation known as *zooming out*, which is of profound importance to reasoning about higraph-based systems. We motivate how, in general, the use of zooming in reasoning requires sophisticated extensions to the basic notion of higraph and a careful definition of higraph dynamics (i.e. semantics), which we contribute. Our main results characterise zooming by means of a universal property and establish a precise relationship between the dynamics of a higraph and that of its zoom-out.

## 1 Introduction

Recent years have witnessed a rapid, ongoing popularisation of diagrammatic notations in the specification, modelling and programming of computing systems. Most notable among them are Statecharts [5], a notation for modelling reactive systems, and the Unified Modelling Language (UML), a family of diagrammatic notations for object-based modelling. Being spatial rather than linear representations of computing systems, diagrams lend themselves to a variety of intuitive structural manipulations, common among which are those implementing filtering and abstraction operations to control the level of detail [10].

Often, such manipulations are employed to assist in the visualisation process [10], as diagrams may grow impractically large even in economic and compact notations. Of particular importance also are uses of filtering and abstraction in the course of reasoning about the represented system. In that case, the user attempts to simplify the reasoning task by considering a higher-level, more abstract diagrammatic representation of the system in question, obtained by discarding detail which is deemed irrelevant to the reasoning argument. Thus, a precise relationship between the form (syntax) and meaning (semantics) of a diagram resulting from filtering or abstraction and those of the original one is pre-requisite to ensuring soundness of reasoning.

---

\* Both authors wish to acknowledge the support of a British Council grant and of AIST Japan. John Power also acknowledges the support of EPSRC grant no. M566333, and Konstantinos Tourlas the support of EPSRC grant no. GR/N12480.

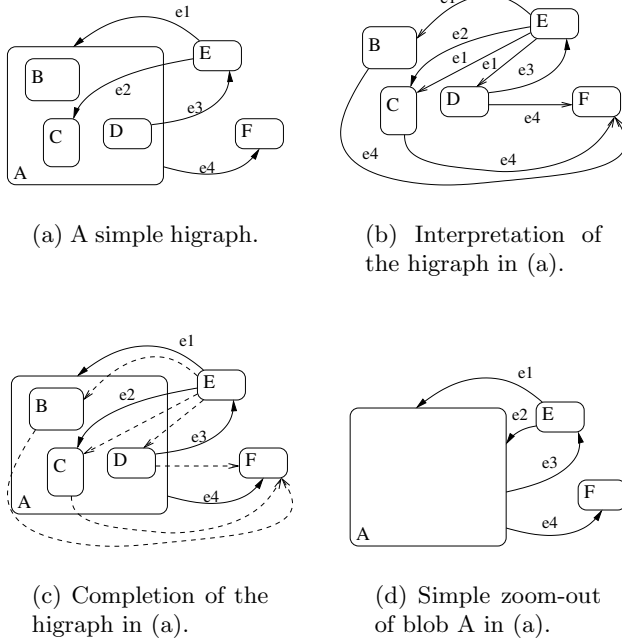
Starting with *higraphs*, which are structures underlying a variety of sophisticated diagrammatic formalisms in computing, we provide such a precise relationship for a practically important filtering operation known as *zooming out*. Doing so, we argue, requires sophisticated extensions to the basic notion of higraph and a careful definition of zooming and of higraph dynamics (i.e. semantics), all of which we contribute. So we give precise definitions of a *meet higraphs*, the dynamics of meet-higraphs, and zooming-out of a meet-higraph. We further contribute two main results. The first characterises one’s informal understanding of zooming by means of the universal property of an adjunction. The other establishes how the dynamics of a higraph is reflected in the dynamics of its zoom-out, as required for supporting the use of zooming in reasoning about higraph-based systems.

The work in [4, 1], concerned with compositionality and modularity questions, also addresses some similar semantic issues. It does, however, require that hierarchical graphs are extended to include, for each node, a set of “interface points.” Here, we do not wish to rely on this kind of non-standard extension. Instead we develop, explain and formalise our solution in terms only of the devices, known as “stubs” or “loose edges”, which are already used in practice and have been adopted in UML and other applications.

Thus, our work relates closely to Harel’s original formulation of higraphs [6] and to the notions of “zooming out” and “loose higraph” which were briefly and informally introduced therein. We have formalised these structures in [11, 2, 12]. After recalling our previous analyses and results from [12], our work here extends our theory and develops new results. This is necessary, as we argue below, in order to address important practical applications of higraphs which lie beyond the scope of Harel’s original brief treatment. The latter focused zooming on only a limited class of higraphs (those arising from Statecharts), but which are clearly too restricted for other important applications, including the hierarchical entity-relationship (ER) modelling of complex databases, also from [6].

The utility, nonetheless, of the framework of concepts and techniques which we develop here extends beyond higraphs. The concept of *hulls*, for instance, which is introduced and studied below, naturally pertains, more generally than higraphs, to other graph-based notations which feature intersecting vertices. There has been increasing computational interest in such notations, one recent example being the Constraint diagrams [3] for use alongside UML.

In the following section we recall the most basic notions of higraph and zooming, and we motivate the need for developing more subtle higraph structures so as to support uses of zooming in reasoning. *Meet higraphs*, such suitably sophisticated structures, are developed in Section 3. *Hulls*, a related but more widely applicable concept, is independently developed in Section 4. In Section 5, we define a zooming operation appropriate to meet-higraphs and prove the main theorem asserting its intuitive universal property. A notion of dynamics for our higraphs and the main theorem relating the dynamics of a meet higraph and those of its zoom-out is the subject of Section 6. Finally, in the concluding section we remark on how our work can benefit the development of software



**Fig. 1.** Interpretation, completion and zooming on a higraph.

tools for higraph-based, as well as other diagrammatic notations. Owing to space considerations, some proofs and lemmas are delegated to an Appendix.

## 2 Higraphs, Zooming and Reasoning

Higraphs are structures extending graphs by permitting spatial containment among the nodes. Since their introduction by Harel [6] as a foundation for Statecharts [5], higraphs have rapidly become prominent structures in computing. Beyond Statecharts and UML, their diverse applications include notations for the entity-relationship diagrams [6] of database theory and knowledge representation, and for reasoning with temporal logics [8], as well as programming languages, such Argos [9].

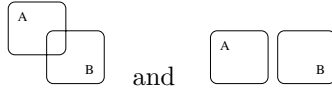
We recall from [11] the definition of an (ordinary) higraph:

**Definition 1.** A higraph is a quadruple  $(B, E; s, t : E \rightarrow B)$ , where  $B$  and  $E$  are posets and  $s, t$  are monotone functions. □

We refer to the elements of  $B$  as *blobs* and to those of  $E$  as *edges*, while the functions provide for each  $e \in E$  its *source*  $s(e)$  and *target*  $t(e)$ . The partial order on  $B$  captures containment, and thus  $b \leq b'$  is interpreted as asserting the

containment of  $b$  in  $b'$ , which trivially includes the case of  $b$  and  $b'$  being the same blob. The partial order on the edges is of lesser importance for the purposes of the present paper. More broadly, however, it is a justified generalisation over pictorial intuition and, as detailed in [11], also a necessary device in studying completion (Fig. 1(c)) and “conflicts” among the edges. We write  $b < b'$  to mean ( $b \leq b'$  and  $b \neq b'$ ). Therefore,  $b \not< b'$  holds when  $b \geq b'$  or when  $b$  and  $b'$  are unrelated by the partial order. The spatial containment of blobs is commonly referred to as *depth*.

*Example 1.* Fig. 1(a) may be seen as the pictorial representation of the higraph with blobs  $\{A, B, C, D, E, F\}$  where  $B, C, D < A$ ; edges  $\{e_1, e_2, e_3, e_4\}$  where  $e_2 < e_1$ ;  $s(e_1) = E$ ,  $t(e_1) = A$ ,  $s(e_2) = E$ ,  $t(e_2) = C$ , and so on.  $\square$



Note, however, that both pictorial representations of the *same* higraph. That is, the definition of higraph does not account for non-trivial intersections [6]. (We say that two blobs  $b_1$  and  $b_2$  in a higraph intersect *trivially* whenever there is a third blob  $b_3$  contained in both  $b_1$  and  $b_2$ .) Yet, the need for intersections often arises most naturally in the diagrammatic modelling of computing structures, and particularly of complex and interrelated datasets, as is evident in ER modelling with higraphs [6] and in constraint modelling [3].

Also in computing, higraphs are employed as uncluttered representations of (the graphs which underlie) complex state-transition systems. The basic idea behind such efficient, higraph-based representations is to generalise the notion of transition from individual states to whole *collections of states*. Each such collection, concretely represented as a blob which in turn contains other blobs, corresponds to a (conceptual or actual) *sub-system*.

*Example 2.* A higher-level edge such as the one from A to F in Fig. 1(a) is understood as implying lower-level edges, in this case from each of the blobs B, C and D contained in the sub-system represented by A, to the target blob F. Thus, the higraph of Fig. 1(a) concisely represents the transition system in Fig. 1(b).  $\square$

In general, this relationship of the underlying transition system to its representation as a higraph is understood in terms of *completing* a higraph with the addition of all implied edges. This completion operation, which we characterised mathematically in [11], is illustrated in Fig. 1(c) (in which added edges are shown dashed).

We therefore seek a notion of *dynamics* (i.e. of state-transition behaviour) for higraphs which implicitly contains the behaviour of the represented transition system. Central to the dynamics, as well as to more “static” interpretations, is a notion of a *path* appropriate for higraphs:

**Definition 2.** A path in a higraph is a finite sequence  $\langle e_0, \dots, e_{n-1} \rangle$  of edges such that  $t(e_i) \leq s(e_{i+1})$  or  $s(e_{i+1}) \leq t(e_i)$ , i.e. the target  $t(e_i)$  of  $e_i$  is contained in the source  $s(e_{i+1})$  of  $e_{i+1}$ , or vice versa.  $\square$

One of the most fundamental operations on higraphs is that of zooming out [6]. The idea is that one often wants to eliminate all structure contained within a given (i.e. selected) blob in a higraph. More precisely, zooming results in the selected blob becoming *atomic* (or *minimal*) by identifying with it all the blobs it contains. Fig. 1(d) illustrates the result of zooming out of A in the higraph of Fig. 1(a). Because abstraction of subsystems is such an essential device in reasoning about complex systems, and zooming is such a natural way of effecting this kind of abstraction on higraphs, the practical importance of zooming in reasoning is profound.

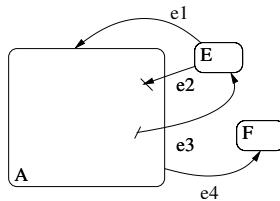
In order, however, to *soundly* infer results about the dynamics of a higraph from the dynamics of its zoom-out, one must know precisely which paths in latter higraph also exist in former. In other words, it is imperative that any reasonable operation of zooming-out does not introduce paths in a way which may lead to false inferences. So, in Section 6, once the terms in its statement have been precisely defined, we prove a generalisation of the following

**Theorem:** Whenever a higraph  $\mu$ , in which blobs may intersect, zooms out to a higraph  $\mu'$ , then every (must-)path in  $\mu'$  is *reflected* (i.e. is the image of) a (must-)path in  $\mu$ .

Unfortunately, this requirement is failed by the simple, almost naive notion of zooming-out which we have so far outlined:

*Example 3.* Observe that zooming has created the path  $\langle e_2, e_3 \rangle$  from E to itself in Fig. 1(d) which does not exist in the original higraph of Fig. 1(a). □

In response to this problem, we require a more subtle notion of higraph which permits edges to be loosely attached to blobs. Zooming out of blob A in Fig. 1(a) now results in the *loose higraph* of Fig. 2. Here, the understanding is that the



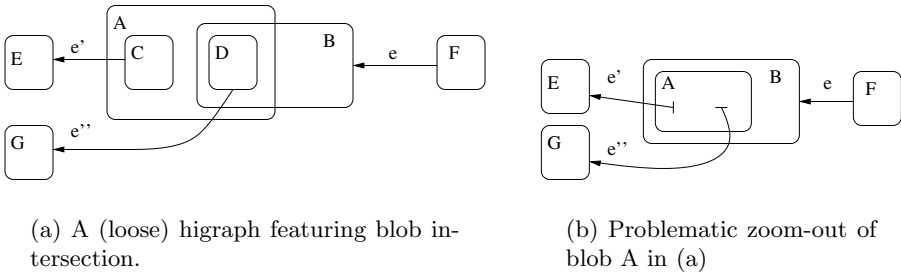
**Fig. 2.** Zoom-out of the higraph in Fig. 1(a) as a loose higraph.

suppressed target of  $e_2$ , indicated as having been some blob originally contained in A, may have not been the same as the suppressed source of  $e_3$ . (We note in passing that, should an edge had existed between, say, B and C in Fig.1(a), it would have appeared as a loose edge from A to itself in Fig.2. For the purposes of our semantic analysis, as well as mathematically natural reasons, we retain such completely loose edges in our definitions. To remove them, as implementations of zooming typically do, a further operation may easily be defined and which

composes with our basic notion of zooming.) Thus, in contrast with ordinary higraphs where only one notion of path exists, the introduction of loose edges deliberately creates a distinction between two notions of path: *must-paths* (i.e. certain paths), such as  $\langle e_1, e_4 \rangle$  in Fig. 2, and *may-paths*, such as  $\langle e_2, e_3 \rangle$ . Observe also that  $\langle e_1, e_3 \rangle$  is a must-path, as  $e_1$  implies edges to all blobs contained in A, including the suppressed source of  $e_3$ . In terms of the above distinction, we consequently demand at least that the theorem above holds of loose higraphs and must-paths.

To obtain this important result in general, a further extension to the notion of (loose) higraph is still required. Problematic situations, which are not resolved by the introduction of loose edges alone, still arise when blobs intersect by means of having common sub-blobs.

*Example 4.* Fig. 3(a) illustrates a (loose) higraph, where blobs A and B intersect by means of containing D. Zooming out of A in that loose higraph produces,



**Fig. 3.** The problem of intersecting blobs.

rather unexpectedly, the result of Fig. 3(b), in which a must-path  $\langle e, e' \rangle$  has been created that did not previously exist.  $\square$

The problem in the preceding example is caused by the inclusion of A into B in Fig. 3(b), which occurs as a side-effect of identifying D with A, owing to the intuitive coherence requirement that zooming must also respect the original containment of D in B. Thus, any reasonable notion of zooming here must not identify D with A (or with B, in the case of zooming out of B). Intuitively, resolving the situation requires the identification of D with a new entity representing the *intersection* of A with B in the picture of Fig. 4. Thus, we have argued that, generally, zooming necessitates an extension to higraphs which substantiates blob intersections.

### 3 Meet Higraphs

In response, the present paper develops the notion of a *higraph with meets*, and its “loose” variant, together with an appropriate operation of zooming-out. To

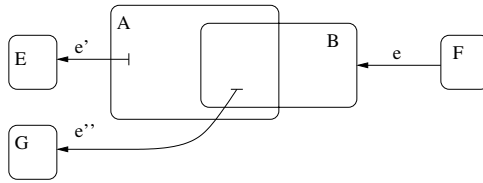


Fig. 4. Intuitive zoom-out of Fig. 3(a).

explicitly account for a notion of intersection among blobs (which, in particular, includes trivial intersections), we first endow the poset  $B$  in Definition 1 with meet semi-lattice structure, thus resulting in the following:

**Definition 3.** A meet higraph consists of: a poset  $E$ ; a finite,  $\wedge$ -semi-lattice  $B$ ; and monotone functions  $s, t : E \rightarrow B$ . □

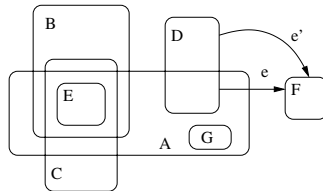
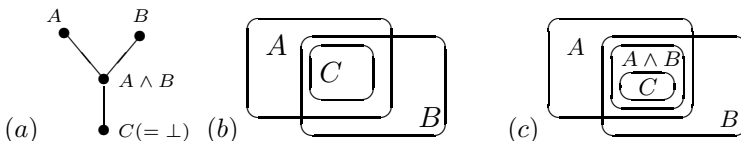


Fig. 5. Meet higraph in which blobs A and D intersect trivially.

*Example 5.* Fig. 5 represents a meet higraph with underlying  $\wedge$ -semi-lattice having elements  $\{A, B, C, D, E, F, G, A \wedge B, B \wedge C, A \wedge B \wedge C, A \wedge D, \perp\}$ , and where  $G < A$ ,  $E < A \wedge B \wedge C$ ,  $A \wedge D < A$ , and so on;  $\perp$  is the least element (which is, by convention, not pictured);  $A \wedge G = G$ ,  $F \wedge E = \dots = F \wedge D \wedge E = \dots = \perp$  and so on;  $s(e) = A \wedge D$  and  $s(e') = D$ . □

To illustrate the subtle difference between Definitions 1 and 3, the following example compares and contrasts the relation of the two notions of higraph wrt. the usual pictorial representation and intuition:

*Example 6.* Consider the four-point  $\wedge$ -semi-lattice shown as a Hasse diagram in (a) below. Regarded as a meet higraph with no edges, it corresponds to the picture shown in (b) below. By contrast, when regarded as an ordinary higraph without edges it corresponds to the picture (c):



Further, to capture the notion of selecting a blob (or, more generally, intersection region) in a meet higraph we introduce the following:

**Definition 4.** A pointed meet higraph is a meet higraph  $(B, E; s, t : E \rightarrow B)$  together with a distinguished element  $p$  of  $B$ , called the point.  $\square$

We have informally argued in Sec. 2 that the effect of zooming on a pointed meet higraph should be to identify each blob strictly contained in the point with the intersection of certain blobs that contain it. In zooming out of  $A$  in Fig. 5, for instance, one intuitively expects that  $G$  becomes identified with  $A$ , and  $E$  with  $A \wedge B \wedge C$ . To make this idea precise, we introduce and study in the next section the salient and topologically natural notion of *hull*.

### 4 Hulls

**Definition 5.** Let  $X$  be finite,  $\wedge$ -semi-lattice and  $x_0 \in X$ . For every  $x \leq x_0$ , the hull of  $x$  relative to  $x_0$  is the meet of all elements not strictly less than  $x_0$  which are greater than or equal to  $x$ :

$$hull_{x_0}(x) \stackrel{\text{def}}{=} \bigwedge \{x' \mid x' \not\prec x_0 \text{ and } x \leq x'\} \quad \square$$

*Example 7.* Taking  $X$  to be the  $\wedge$ -semi-lattice underlying the example of Fig. 5 and  $x_0$  to be the blob labelled ‘A’, one has  $hull_A(G) = A$ ,  $hull_A(E) = A \wedge B \wedge C$ ,  $hull_A(A) = A$ ,  $hull_A(A \wedge D) = A \wedge D$  and so on.  $\square$

To establish some of the intuition behind Def. 5 and facilitate the understanding of subsequent technical proofs, we record in the following Lemma and its corollary three basic properties of hulls (while deferring their proof to the Appendix).

**Lemma 1.** For every finite  $\wedge$ -semi-lattice  $X$  and  $x, x_0 \in X$  such that  $x \leq x_0$ :

1.  $x \leq hull_{x_0}(x) \leq x_0$
2.  $hull_{x_0}(hull_{x_0}(x)) \leq hull_{x_0}(x)$ .  $\square$

**Corollary 1.** For all  $x \leq x_0$ ,  $hull_{x_0}(hull_{x_0}(x)) = hull_{x_0}(x)$ .  $\square$

The concept of hull is not only central to the more sophisticated notion of zooming which is sought here. Being a topologically natural concept, it is a particularly natural notion to associate not only with meet higraphs, but also with many other notions of graph which feature intersections.

To formalise the connection between hulls and the sought notion of zooming which emerged from the analysis in the previous section, we consider first the simple case of a pointed meet higraph with no edges. Every such meet higraph is, in essence, a pointed, finite,  $\wedge$ -semi-lattice  $\langle B, p \rangle$ . Then, in this degenerate case of no edges, the required operation of zooming out of  $p$  should have precisely the effect of identifying each  $b \leq p$  with  $hull_p(b)$ . (This may be seen by recalling the transition from Fig. 3(a) to Fig. 4 while ignoring all edges in the two pictures.) The following definition makes this construction explicit:



**Definition 6.** Given a pointed, finite  $\wedge$ -semi-lattice  $\langle X, x_0 \rangle$ , define a function  $\zeta_{\langle X, x_0 \rangle} : \langle X, x_0 \rangle \rightarrow \langle X, x_0 \rangle$  as follows:

$$\zeta_{\langle X, x_0 \rangle}(x) \stackrel{\text{def}}{=} \begin{cases} \text{hull}_{x_0}(x), & \text{if } x \leq x_0 \\ x, & \text{otherwise} \end{cases}$$

Beyond being monotone (Lemma 3, Appendix), each function  $\zeta_{\langle X, x_0 \rangle}$  inherits, as a corollary of Lemma 1 above, properties which intuitively include idempotency:  $\zeta_{\langle X, x_0 \rangle} \circ \zeta_{\langle X, x_0 \rangle} = \zeta_{\langle X, x_0 \rangle}$ .

Regarding  $\langle B, p \rangle$  as a pointed meet higraph without edges we may now precisely, with the aid of the function  $\zeta_{\langle B, p \rangle}$ , capture its “zoom-out”  $\zeta(\langle B, p \rangle)$ , as in the following:

**Definition 7.** For every pointed, finite  $\wedge$ -semi-lattice  $\langle X, x_0 \rangle$  define  $\zeta(\langle X, x_0 \rangle)$  to be the pair  $\langle X_\zeta, x_0 \rangle$ , where  $X_\zeta$  is the sub-poset of  $X$  with elements all those  $x \in X$  such that  $\zeta_{\langle X, x_0 \rangle}(x) = x$ . □

*Example 8.* For  $\langle X, x_0 \rangle$  as in Example 7,  $X_\zeta = X \setminus \{E, G\}$ . □

For this construction to be meaningful in our context one must establish that  $X_\zeta$  is a sub- $\wedge$ -semi-lattice of  $X$  and that  $x_0 \in X_\zeta$ , thus making each  $\zeta(\langle X, x_0 \rangle)$  a finite, pointed  $\wedge$ -semi-lattice. Indeed, it is not hard to prove the following

**Proposition 1.** For every finite, pointed  $\wedge$ -semi-lattice  $\langle X, x_0 \rangle$ , the poset  $X_\zeta$  of Definition 7 is a sub- $\wedge$ -semi-lattice of  $X$ . Moreover,  $x_0 \in X_\zeta$ , and for all  $x \leq x_0$  in  $X_\zeta$ ,  $\text{hull}_{x_0}(x) = x$ . □

The statement  $\text{hull}_{x_0}(x) = x$  in the preceding proposition asserts, in agreement with one’s intuition about zooming, that the point of  $\zeta(\langle B, p \rangle)$  is *minimal*, in the specific sense that all hulls relative to it are trivial. To make this notion of minimality precise before discussing its intuitive appeal, we introduce the following

**Definition 8.** The point  $x_0$  in a pointed, finite,  $\wedge$ -semi-lattice  $\langle X, x_0 \rangle$  is minimal (with respect to hulls) if  $\text{hull}_{x_0}(x) = x$  for all  $x \leq x_0$ . □

The reader is urged to observe the difference between this hull-specific notion of minimality and the usual order-theoretic one ( $x \in X$  is order-theoretically minimal in  $X$  whenever  $x' \leq x \implies x' = x$  for all  $x' \in X$ ). In particular, minimality wrt. hulls does not imply minimality in the order-theoretic sense. Instead, the notion of minimality in Def. 8 alludes an intuitive notion of “pictorial minimality” expressed in terms of the spatial containment of contours representing blobs in pictures of meet higraphs.

*Example 9.* Consider again the  $\wedge$ -semi-lattice underlying the example of Fig. 5. The point corresponding to the blob labelled ‘A’ in the same figure is *not* minimal wrt. hulls, as  $\text{hull}_A(E) = A \wedge B \wedge C$ . By contrast, the point labelled ‘D’ is minimal wrt. hulls, as  $D \wedge A$  is the sole element less than D and  $\text{hull}_D(D \wedge A) = D \wedge A$ . Neither A nor D, however, are minimal in the usual order-theoretic sense. Yet, the contour labelled D in the figure is “pictorially minimal” in the intuitive sense of not wholly containing the contour of any other blob.

Before proceeding to considering edges, we explicate (Theorem 1, below) the universal property associated with Def. 7. It arises from the minimality (wrt. hulls) property of the point in each  $\zeta(\langle X, x_0 \rangle)$ , and will be much useful later.

**Definition 9.** Let  $\mathbf{SL}_*^{fin,\wedge}$  denote the category having objects all finite, pointed  $\wedge$ -semi-lattices, and arrows  $f : \langle X, x_0 \rangle \rightarrow \langle Y, y_0 \rangle$  all monotone functions  $f : X \rightarrow Y$  which, in addition, preserve points and hulls: i.e.,  $f(x_0) = y_0$  and  $\forall x. x \leq x_0 \implies f(\text{hull}_{x_0}(x)) = \text{hull}_{y_0}(f(x))$ . Further, let  $\mathbf{SL}_{*,min}^{fin,\wedge}$  denote the full subcategory of  $\mathbf{SL}_*^{fin,\wedge}$  consisting of all those objects  $\langle X, x_0 \rangle$  in which the point  $x_0$  is minimal in the sense of Def. 8. We write  $J$  for the full and faithful functor including  $\mathbf{SL}_{*,min}^{fin,\wedge}$  into  $\mathbf{SL}_*^{fin,\wedge}$ .  $\square$

Noteworthy here for its central role in subsequent development, and for explaining how the elements of each  $\langle B, p \rangle$  map to elements in its “zoom-out”  $\zeta(\langle B, p \rangle)$ , is the following family of morphisms:

**Definition 10.** For each finite,  $\wedge$ -semi-lattice  $\langle X, x_0 \rangle$  let  $\eta_{\langle X, x_0 \rangle} : \langle X, x_0 \rangle \rightarrow J(\zeta(\langle X, x_0 \rangle))$  be the morphism in  $\mathbf{SL}_*^{fin,\wedge}$  defined by  $x \in X \mapsto \zeta_{\langle X, x_0 \rangle}(x)$ .  $\square$

The statement of the following Theorem is now the expression, in very precise terms, of the intuitive understanding of  $\zeta(\langle X, x_0 \rangle)$  as obtained from  $\langle X, x_0 \rangle$  by making the point  $x_0$  minimal wrt. hulls, but without otherwise disturbing the structure of  $\langle X, x_0 \rangle$ :

**Theorem 1.** The function  $\zeta : \text{Obj}(\mathbf{SL}_*^{fin,\wedge}) \rightarrow \text{Obj}(\mathbf{SL}_{*,min}^{fin,\wedge})$  of Def. 7 extends to a functor  $\zeta : \mathbf{SL}_*^{fin,\wedge} \rightarrow \mathbf{SL}_{*,min}^{fin,\wedge}$  which is left adjoint to  $J$ . The unit  $\eta$  of the adjunction has components the morphism of Def. 10.

*Proof.* Consider any arrow  $f : \langle X, x_0 \rangle \rightarrow J(\langle Y, y_0 \rangle)$  in  $\mathbf{SL}_*^{fin,\wedge}$  and any  $x \in X$  such that  $x \leq x_0$ . Since all hulls in  $\langle Y, y_0 \rangle$  are trivial (wrt.  $y_0$ ), it follows from  $f(x) \leq f(x_0) = y_0$  that  $\text{hull}_{y_0}(f(x)) = f(x)$ . Preservation of hulls by  $f$  now yields:  $f(\text{hull}_{x_0}(x)) = \text{hull}_{y_0}(f(x)) = f(x)$ . Thus, we have shown that  $\forall x. x \leq x_0 \implies f(\text{hull}_{x_0}(x)) = f(x)$ . From this it follows easily that there exists morphism  $f^b : \zeta(\langle X, x_0 \rangle) \rightarrow \langle Y, y_0 \rangle$  in  $\mathbf{SL}_{*,min}^{fin,\wedge}$  such that  $f = J(f^b) \circ \eta_{\langle X, x_0 \rangle}$ . Since each  $\eta_{\langle X, x_0 \rangle}$  is epi, this factorisation is unique.  $\square$

## 5 Loose Meet Higraphs and Zooming-Out

Having introduced hulls and the mechanics of zooming-out in the restricted case of meet higraphs with no edges, we proceed to treat the general case. While, as we argued in Sec. 2, the addition of “loose edges” to ordinary higraphs is in itself insufficient, loose edges are still a necessary device in developing a full solution to the problem of reasoning with the aid of zooming-out abstraction. We therefore begin by allowing edges in our meet higraphs to be loosely attached, thus resulting in a corresponding notion of *loose meet higraph*.

Consider again the loose higraph of Fig. 2. We recall from [2, 12] that every such loose higraph with blobs  $B$  can be formally cast as an ordinary higraph

having the same edges but containing two distinct copies  $\langle 0, b \rangle$  and  $\langle 1, b \rangle$  of each  $b \in B$ , tagged with 0's and 1's. In the pictorial representation of loose higraps the convention is that blobs tagged with 0 are not shown at all and that, for instance, an edge with target of the form  $\langle 0, b \rangle$ , such as  $e_2$  in Fig. 2 with target  $\langle 0, A \rangle$ , has its endpoint lying *inside* the contour picturing  $b$ . Formally,

**Definition 11.** Given any poset  $B$  define poset  $B^\sharp$  as having underlying set  $\{0, 1\} \times B$ , and partial order generated by the following two rules:

- $\langle 0, b \rangle < \langle 1, b \rangle$  for all  $b \in B$ ; and
- $\langle 1, b \rangle \leq \langle 1, b' \rangle$  whenever  $b \leq_B b'$  in  $B$ .

Further, define the “projection”  $\pi_B : B^\sharp \rightarrow B$  as mapping both  $\langle 0, b \rangle$  and  $\langle 1, b \rangle$  to  $b$ , for each  $b \in B$ . The function  $(-)^{\sharp}$  extends to an endofunctor on **Poset**, by taking  $(f : A \rightarrow B)^{\sharp}$  to be the monotone map sending  $\langle i, a \rangle$  to  $\langle i, f(a) \rangle$ .  $\square$

We shall use  $b, b', b_1, \dots$  to range over the poset  $B$  and  $v, v', v_1, \dots$  to range over  $B^\sharp$ . Using this auxiliary poset structure we can now make precise the definition of a meet higraph with loosely attached edges:

**Definition 12.** A meet higraph with loosely attached edges (or loose meet higraph for short) is a quadruple  $(B, E; s, t : E \rightarrow B^\sharp)$  where:  $E$  is a poset;  $B$  is a finite,  $\wedge$ -semi-lattice; and  $s, t : E \rightarrow B^\sharp$  are functions making both  $\pi_B \circ s$  and  $\pi_B \circ t$  monotone. A pointed, loose meet higraph  $\mu_* = \langle \mu, p \rangle$  consists of a loose meet higraph  $\mu = (B, E; s, t : E \rightarrow B^\sharp)$  together with a distinguished element  $p \in B$  called the point.  $\square$

*Example 10.* In the loose meet higraph of Fig. 4, one has  $s(e') = \langle 0, A \rangle$ ,  $s(e'') = \langle 0, A \wedge B \rangle$ ,  $t(e) = \langle 1, B \rangle$ , and so on.  $\square$

We now seek a notion of morphism for pointed loose meet higraps which smoothly extends the morphisms of Def. 10 to the new setting. In addition to components  $f_E, f_B$  which map the elements (edges and blobs) that are visible in pictures, such a morphism  $f$  must also have a component  $f_{B^\sharp}$  which also maps the invisible elements. While  $f_{B^\sharp}$  cannot, in general, be monotone (as  $\langle 0, b \rangle \not\leq \langle 0, b' \rangle$  even when  $b \leq b'$ ), it must at least be consistent with  $f_B$  and well-behaved with respect to hulls. Making this precise, we have:

**Definition 13.** A morphism  $f : \langle \mu_0, p_0 \rangle \rightarrow \langle \mu_1, p_1 \rangle$  of pointed loose meet higraps, where  $\mu_0 = (B_0, E_0; s_0, t_0 : E_0 \rightarrow B_0^\sharp)$  and  $\mu_1 = (B_1, E_1; s_1, t_1 : E_1 \rightarrow B_1^\sharp)$ , consists of: a monotone function  $f_E : E_0 \rightarrow E_1$ ; a monotone and hull-preserving function  $f_B : B_0 \rightarrow B_1$ ; and a function  $f_{B^\sharp} : B_0^\sharp \rightarrow B_1^\sharp$ . These data are subject to the following conditions:

- sources and targets of edges are preserved: e.g.,  $s_1 \circ f_E = f_{B^\sharp} \circ s_0$ ;
- $f_B$  preserves the point and  $f_{B^\sharp}$  is consistent with  $f_B$ :  $f_B \circ \pi_{B_0} = \pi_{B_1} \circ f_{B^\sharp}$ ;
- $f_{B^\sharp}$  reflects hulls, in the sense that  $f_{B^\sharp}(v) = \langle 1, \text{hull}_{p_1}(f_B(b)) \rangle$  implies  $v = \langle 1, \text{hull}_{p_0}(b) \rangle$  for all  $b \leq p_0$  in  $B_0$  and  $v \in B_0^\sharp$ .  $\square$

Morphisms of pointed, loose higraphs compose component-wise and have obvious identities. Thus one has a category  $\mathcal{LMH}_*$  of pointed loose meet higraphs.

In the last section we described how zooming out of a pointed meet higraph without edges makes the point in  $\langle B, p \rangle$  minimal wrt. hulls in  $\zeta(\langle B, p \rangle)$ , subject to the universal property of Thm. 1. A contrasting look at figures 3(a) and 4 confirms one’s intuition that a similar universal property must hold in the general case of zooming out of an arbitrary loose meet higraph  $\mu_*$ . This is what we prove in the remainder of this section, starting with defining how the sources and targets of edges in  $\mu_*$  are appropriately fixed in the zoom-out  $Z(\mu_*)$ :

**Definition 14.** *Let  $\mu_* = \langle \mu, p \rangle$  be a pointed loose meet higraph with  $\mu$  being  $(B, E; s, t : E \rightarrow B^\sharp)$ . Formally,  $Z(\mu_*)$  is the pointed loose meet higraph given by  $(B_\zeta, E; \eta_{B^\sharp} \circ s, \eta_{B^\sharp} \circ t : E \rightarrow B_\zeta^\sharp)$  and point  $p \in B_\zeta$ , where  $\eta_{B^\sharp} : B^\sharp \rightarrow B_\zeta^\sharp$  is the function sending each  $\langle i, b \rangle \in B^\sharp$  such that  $b \notin B_\zeta$  to  $\langle 0, \eta_{\langle B, p \rangle}(b) \rangle$ ; and to  $\langle i, \eta_{\langle B, p \rangle}(b) \rangle$  otherwise. (Recall that  $\eta_{\langle B, p \rangle}$ , regarded here as a monotone function from  $B$  to  $B_\zeta$ , is defined in Def 10 and that  $B_\zeta$  is the  $\wedge$ -semi-lattice underlying  $\zeta(\langle B, p \rangle)$  in Def 7.)  $\square$*

*Example 11.* The preceding definition formalises how the source of an edge such as, say,  $e''$  in Fig. 3(a) is fixed to  $\langle 0, A \wedge B \rangle$  in Fig. 4, after its original source  $\langle 1, D \rangle$  becomes identified with  $\langle 0, \text{hull}_A(D) \rangle = \langle 0, A \wedge B \rangle$  as a result of zooming out of blob  $A$ .  $\square$

With  $\mathcal{LMH}_{*,min}$  we shall denote the (full) subcategory of  $\mathcal{LMH}_*$  consisting of all pointed, loose meet higraphs in which the point is minimal in the sense of Def. 8. Thus the function  $Z$  in Def. 14, being a function from the objects of  $\mathcal{LMH}_*$  to those of  $\mathcal{LMH}_{*,min}$ , formalises the intuitive understanding of zooming out as reducing the point of  $\mu_*$  to a minimal (wrt. hulls) point in  $Z(\mu_*)$ . Moreover, it does so without otherwise disturbing the structure of  $\mu_*$ . Before proving this universal property of  $Z$  in Thm. 2 below, we need an intuitive mapping (i.e. morphism) from each  $\mu_*$  to its zoom-out:

**Definition 15.** *Let  $I$  be the (fully faithful) inclusion functor  $\mathcal{LMH}_{*,min} \rightarrow \mathcal{LMH}_*$ . For each object  $\mu_* = \langle \mu, p \rangle$  of  $\mathcal{LMH}_*$ , where  $\mu = (B, E; s, t : E \rightarrow B^\sharp)$ , define a morphism  $\eta_{\mu_*} : \mu_* \rightarrow I(Z(\mu_*))$  with the following components:  $(\eta_{\mu_*})_E = \text{id}_E : E \rightarrow E$ ;  $(\eta_{\mu_*})_B = \eta_{\langle B, p \rangle}$  (where  $\eta_{\langle B, p \rangle}$  is the morphism of Def 10 regarded here as a monotone, hull preserving function from  $B$  to  $B_\zeta$ ); and  $(\eta_{\mu_*})_{B^\sharp}$  is the function  $\eta_{B^\sharp}$  in Def. 14.  $\square$*

**Theorem 2.** *The function  $Z : \text{Obj}(\mathcal{LMH}_*) \rightarrow \text{Obj}(\mathcal{LMH}_{*,min})$  extends to a functor which is left adjoint to  $I$ , with unit being the morphisms  $\eta_{\mu_*}$  of Def. 15.*

*Proof. (Sketch)* Given any morphism  $f : \mu_* \rightarrow I(\mu'_*)$ , where  $\mu_*$  and  $\mu'_*$  are objects of  $\mathcal{LMH}_*$  and  $\mathcal{LMH}_{*,min}$  respectively, define a morphism  $f^b : Z(\mu_*) \rightarrow \mu'_*$  in  $\mathcal{LMH}_{*,min}$  as having the following components:  $f^b_E$  maps each  $e \in E$  to  $f_E(e)$ ;  $f^b_B$  is the induced by the adjunction in Thm. 1, unique morphism  $f^b_{\langle B, p \rangle} : B \rightarrow B'$

such that  $f_B^b \circ (\eta_{\mu_*})_B = f_B$ ; and  $f_{B^\#}^b = f_{B^\#} \circ \iota_{\langle B, p \rangle}$  where  $\iota_{\langle B, p \rangle}$  is the inclusion of  $\zeta(\langle B, p \rangle) = \langle B_\zeta, p \rangle$  into  $\langle B, p \rangle$ .

In addition to  $f_E^b \circ (\eta_{\mu_*})_E = f_E$  and  $f_B^b \circ (\eta_{\mu_*})_B = f_B$ , which hold straightforwardly, one also has (Lemma 5, Appendix) that  $f_{B^\#}^b \circ (\eta_{\mu_*})_{B^\#} = f_{B^\#}$ . Thus,  $f = I(f^b) \circ \eta_{\mu_*}$ . As each component of  $\eta_{\mu_*}$  is epi,  $f^b$  is moreover the unique morphism in  $\mathcal{LMH}_{*,min}$  with this property, and the proof concludes with an appeal to Theorem IV.2(ii) of [7]. □

The preceding result directly parallels the corresponding theorem in [12] for (non-meet) loose higraphs. What, however, the new notion of zooming on loose meet higraphs makes possible, over and above previous work in [12], is the ability to fully relate the semantics before and after a zoom-out, as we show next.

## 6 Dynamics of Loose Meet Higraphs

In this section we make precise the dynamics, i.e. transition semantics, of loose meet higraphs by introducing a notion of *run* akin to similar notions in use with ordinary (“flat”) transition systems. In an ordinary higraph, or a meet higraph, a run is essentially a sequence of edges (transitions) together with the blobs (or *states*, in the popular jargon of most applications) which are traversed by performing the transitions:

**Definition 16.** A run through a meet higraph  $(B, E; s, t : E \rightarrow B)$  is a finite sequence of the form  $b_0 \xrightarrow{e_1} b_1 \xrightarrow{e_2} \dots \xrightarrow{e_n} b_n$  where, for all  $1 \leq i \leq n$ ,  $b_{i-1} \leq_B s(e_i)$  and  $b_i \leq_B t(e_i)$ . □

*Example 12.* In Fig. 1(a) (regarded trivially as a meet higraph),  $E \xrightarrow{e_1} A \xrightarrow{e_4} F$  is clearly a run, as is  $E \xrightarrow{e_1} D \xrightarrow{e_3} E$  because  $s(e_3) = D < A = t(e_1)$ . Similarly,  $E \xrightarrow{e_2} C \xrightarrow{e_4} F$  is also a run because  $t(e_2) = C < A = s(e_4)$ . □

In ordinary transition systems, the sequence of states traversed is implicit in the notion of path (i.e. connected sequence of transitions). In higraphs and meet higraphs, where higher-level edges are taken to imply lower-level ones, as is implicit in the joining condition  $b_{i-1} \leq_B s(e_i)$  and  $b_i \leq_B t(e_i)$  of Def. 16 above, the notion of path no longer provides adequate state information, hence the need for a notion of “run”. Suggestive of dynamics though its name might be, we wish to stress that the mathematical structure of runs also pertains, by subsuming the notion of path, to more static interpretations of higraphs.

Recall (Section 2) how the introduction of loosely attached edges incurs a distinction between *must-* and *may-paths*. Intuitively, the idea is that a may-path may perish, in the sense of becoming disconnected, when one makes explicit the suppressed end-points of every loose edge in the path. In Fig. 2, for instance, one may introduce two new, distinct blobs C and D contained in A, so as to make  $t(e_2) = \langle 1, C \rangle$  and  $s(e_3) = \langle 1, D \rangle$ , thereby making  $\langle e_2, e_3 \rangle$  a non-path. By

contrast, a must-path must persist, no matter how one attaches its loose edges to newly introduced blobs.

A similar distinction between *must-runs* and *may-runs* through a loose meet higraph is therefore necessary. Here we concentrate on must-runs only:

**Definition 17.** A must-run through a pointed, loose meet higraph  $\mu_*$ , where  $\mu$  is  $(B, E; s, t : E \rightarrow B^\sharp)$ , is a sequence of the form  $\langle i_0, b_0 \rangle \xrightarrow{e_1} \langle i_1, b_1 \rangle \xrightarrow{e_2} \dots \xrightarrow{e_n} \langle i_n, b_n \rangle$ , where  $\langle i_j, b_j \rangle \in B^\sharp$  and  $e_j \in E$ , subject to the following conditions:

1. for all  $1 \leq j \leq n$ ,  $\langle i_j, b_j \rangle \leq t(e_j)$  and  $\langle i_{j-1}, b_{j-1} \rangle \leq s(e_j)$ ; and
2.  $\pi_0(t(e_j)) = 1$  or  $\pi_0(s(e_{j+1})) = 1$  for all  $1 \leq j < n$ , where  $\pi_0$  is the projection mapping each  $\langle i, b \rangle \in B^\sharp$  to  $i$ .  $\square$

*Example 13.* Consider Fig. 2, this time as representing a (pointed) loose meet higraph. While  $\langle 1, E \rangle \xrightarrow{e_1} \langle 1, A \rangle \xrightarrow{e_4} \langle 1, F \rangle$  is a run, the sequence  $\langle 1, E \rangle \xrightarrow{e_1} \langle 1, A \rangle \xrightarrow{e_3} \langle 1, E \rangle$  isn't because it violates the first condition in Def. 17. However  $\langle 1, E \rangle \xrightarrow{e_1} \langle 0, A \rangle \xrightarrow{e_3} \langle 1, E \rangle$  is a run, as is  $\langle 1, E \rangle \xrightarrow{e_1} \langle 0, A \rangle \xrightarrow{e_4} \langle 1, F \rangle$ . But  $\langle 1, E \rangle \xrightarrow{e_2} \langle 0, A \rangle \xrightarrow{e_3} \langle 1, E \rangle$  is not, as it violates the second condition.  $\square$

The following theorem now establishes precisely how each must-run through  $Z(\mu_*)$  is the image of a corresponding must-run through  $\mu_*$ :

**Theorem 3.** Every must-run through  $I(Z(\mu_*))$ , where  $\mu$  is  $(B, E; s, t : E \rightarrow B^\sharp)$ , is of the form  $\eta_{\mu_*}(v_0) \xrightarrow{e_1} \eta_{\mu_*}(v_1) \xrightarrow{e_2} \dots \xrightarrow{e_n} \eta_{\mu_*}(v_n)$ , for some run  $v_0 \xrightarrow{e_1} v_1 \xrightarrow{e_2} \dots \xrightarrow{e_n} v_n$  through  $\mu_*$ , where  $\eta_{\mu_*}$  is the component at  $\mu_*$  of the unit of the adjunction of Theorem 2.

*Proof. (Sketch)* Let  $\hat{v}_0 \xrightarrow{e_1} \hat{v}_1 \xrightarrow{e_2} \dots \xrightarrow{e_n} \hat{v}_n$ , where  $\hat{v}_j \in B_\zeta^\sharp$ , be an arbitrary run through  $Z(\mu_*)$ . Take  $v_0 = \langle i_0, \pi_B(s(e_1)) \rangle$ . Calculation establishes  $v_0 \leq s(e_1)$  and  $\eta_{\mu_*}(v_0) = \hat{v}_0$ . Similarly take  $v_n = \langle i_n, \pi_B(t(e_n)) \rangle$ . For  $1 \leq j < n$ , each  $v_j$  is obtained by an application of Lemma 6 in the Appendix.  $\square$

## 7 Conclusions

Of particular importance to practitioners is the kind of semi-formal, tool-assisted reasoning which consists of the progressive simplification of a diagram, by repeatedly using abstraction, until either a counter-example is reached or the property can easily be proven. The potential for error in this style of reasoning is great, and so supporting tools must intervene to prohibit any unsound steps or inferences. To do so, we have argued using higraphs as concrete examples, tools must often maintain internal representations (such as meet higraphs) which are more sophisticated than the user's notation. Also, such tools must know how the semantics of a zoom-out relates to the semantics of the original diagram. Here we have developed such a semantics and precise relationship based on, but not limited to, the common interpretation of higraphs as transition systems. More generally, we have argued, our work also applies to other notations, particularly those which feature intersections among vertices.

## References

- [1] R. Alur and R. Grosu. Modular refinement of hierarchic reactive machines. In *Symposium on Principles of Programming Languages*, pages 390–402, 2000.
- [2] Stuart Anderson, John Power, and Konstantinos Tourlas. Reasoning in higraphs with loose edges. In *Proceedings of the 2001 IEEE Symposia on Human-Centric Computing Languages and Environments*, pages 23–29. IEEE Computer Society Press, September 2001.
- [3] J. Gil, J. Howse, and S. Kent. Towards a formalisation of constraint diagrams. In *Proceedings of the 2001 IEEE Symposia on Human-Centric Computing Languages and Environments*, pages 72–79. IEEE Computer Society Press, September 2001.
- [4] R. Grosu, Gh. Stănescu, and M. Broy. Visual formalisms revisited. In *Proc. IEEE Int. Conf. Application of Concurrency to System Design, CSD*, 1998.
- [5] David Harel. Statecharts: A visual approach to complex systems. *Science of Computer Programming*, 8(3):231–275, 1987.
- [6] David Harel. On visual formalisms. *Communications of the ACM*, 31(5):514–530, 1988.
- [7] Saunders MacLane. *Categories for the Working Mathematician*, volume 5 of *Graduate Texts in Mathematics*. Springer-Verlag, 1971.
- [8] Z. Manna and A. Pnueli. Temporal verification diagrams. In *Proceedings of TACS*, volume 789 of *Lecture Notes in Computer Science*, pages 726–765. Springer-Verlag, 1994.
- [9] F. Maraninchi. The Argos language: Graphical representation of automata and description of reactive systems. In *Proceedings of the IEEE Workshop on Visual Languages*, 1991.
- [10] Bonnie M. Nardi. *A Small Matter of Programming: Perspectives on End-User Computing*. MIT Press, 1993.
- [11] John Power and Konstantinos Tourlas. An algebraic foundation for higraphs. In L. Fribourg, editor, *Proceedings of the 15th Annual Conference of the European Association for Computer Science Logic (CSL)*, volume 2142 of *Lecture Notes in Computer Science*, pages 145–159. Springer-Verlag, September 2001.
- [12] John Power Stuart Anderson and Konstantinos Tourlas. Zooming-out on higraph-based diagrams: syntactic and semantic issues. In James Harland, editor, *Electronic Notes in Theoretical Computer Science*, volume 61. Elsevier Science Publishers, 2002.

## Appendix: Technical Lemmas and Proofs

### Meet Higraphs and Hulls

The following are simple properties arising from the definitions of hull (Def. 5) and of the functions  $\zeta_{\langle X, x_0 \rangle}$  (Def. 6):

**Lemma 1.** For every finite  $\wedge$ -semi-lattice  $X$  and  $x, x_0 \in X$  such that  $x \leq x_0$ , the following hold:

1.  $x \leq \text{hull}_{x_0}(x) \leq x_0$
2.  $\text{hull}_{x_0}(\text{hull}_{x_0}(x)) \leq \text{hull}_{x_0}(x)$ .

*Proof.* Fix arbitrary  $x \leq x_0$ .

- To show (1), observe that  $x \leq \bigwedge \{x' \mid x' \not\leq x_0 \text{ and } x \leq x'\}$  because  $x$  is a lower bound of that set. Moreover, one clearly has  $x_0 \in \{x' \mid x' \not\leq x_0 \text{ and } x \leq x'\}$ , and thus  $\text{hull}_{x_0}(x) \leq x_0$ .
- For showing (2) observe that, for all  $x'$  such that  $x' \not\leq x_0$  and  $x \leq x'$ , one has  $\text{hull}_{x_0}(x) \leq x'$  (by the very definition of  $\text{hull}_{x_0}(x)$  as a meet). Hence, the following inclusion of sets holds:  $\{x' \mid x' \not\leq x_0 \text{ and } x \leq x'\} \subseteq \{x' \mid x' \not\leq x_0 \text{ and } \text{hull}_{x_0}(x) \leq x'\}$ , from which one obtains  $\text{hull}_{x_0}(\text{hull}_{x_0}(x)) \leq \text{hull}_{x_0}(x)$ .

**Lemma 2.** *Given any elements  $x_1, x_2, x_0$  in a finite  $\wedge$ -semi-lattice such that  $x_1 \leq x_2 \leq x_0$ , one has  $\text{hull}_{x_0}(x_1) \leq \text{hull}_{x_0}(x_2)$ .*

**Lemma 3.** *Each function  $\zeta_{\langle X, x_0 \rangle}$  is monotone.*

*Proof.* Arbitrarily fix  $x_1, x_2 \in X$  such that  $x_1 \leq x_2$ . As one always has  $x_2 \leq \zeta_{\langle X, x_0 \rangle}(x_2)$  (by Lemma 1(1) and Definition 6), we show that  $\zeta_{\langle X, x_0 \rangle}(x_1) \leq x_2$  by case analysis:

1.  $x_1, x_2 \leq x_0$ . Monotonicity follows from Lemma 2 above.
2.  $(x_1 \leq x_0 \text{ and } x_2 \not\leq x_0)$ . In this case  $\text{hull}_{x_0}(x_1) \leq x_2$  because  $x_2$  is an element of the set  $\{x' \mid x' \not\leq x_0 \text{ and } x_1 \leq x'\}$ . It follows from Definition 6 that  $\zeta_{\langle X, x_0 \rangle}(x_1) = \text{hull}_{x_0}(x_1) \leq x_2$ .
3.  $x_1 \not\leq x_0 \text{ and } x_2 \not\leq x_0$ . In this case, trivially,  $\zeta_{\langle X, x_0 \rangle}(x_1) = x_1 \leq x_2$ .

**Lemmas Used in the Proof of Theorem 2**

**Lemma 4.** *Let  $f : \langle \mu_0, p_0 \rangle \rightarrow \langle \mu_1, p_1 \rangle$  be a morphism in  $\mathcal{LMH}_*$ , where  $\langle \mu_1, p_1 \rangle$  is minimally pointed wrt. hulls. Then, for all  $b \leq p_0$  the following two hold:*

1.  $f_{B^\sharp}(\langle 0, b \rangle) = \langle 0, f_B(b) \rangle$ ; and
2.  $b \notin B_\zeta \implies f_{B^\sharp}(\langle 1, b \rangle) = \langle 0, f_B(b) \rangle$ .

*Proof.* From  $b \leq p_0$ , monotonicity of  $f_B$  and preservation of points by  $f_B$  it follows that  $f_B(b) \leq p_1$ , whereby, from the minimality (wrt. hulls) of  $p_1$ , one has

$$f_B(b) = \text{hull}_{p_1}(f_B(b)) \quad (1)$$

To show (1) assume  $f_{B^\sharp}(\langle 0, b \rangle) \neq \langle 0, f_B(b) \rangle$  and derive a contradiction. The only possibility is  $f_{B^\sharp}(\langle 0, b \rangle) = \langle 1, f_B(b) \rangle$  whereby, from equation (1) and condition (3) of Def. 13, it follows that  $\langle 0, b \rangle = \langle 1, \text{hull}_{p_0}(b) \rangle$ : a contradiction, as  $0 \neq 1$ .

To show the contrapositive of (2) assume  $f_{B^\sharp}(\langle 1, b \rangle) \neq \langle 0, f_B(b) \rangle$ . The only possibility is  $f_{B^\sharp}(\langle 1, b \rangle) = \langle 1, f_B(b) \rangle$  whereby, from equation (1) and condition (3) of Def. 13, it follows that  $\langle 1, b \rangle = \langle 1, \text{hull}_{p_0}(b) \rangle$ , equivalently  $b = \text{hull}_{p_0}(b)$ , equivalently  $b \in B_\zeta$  by Def 7.



**Lemma 5.** *Let  $f_{B^\sharp}^b$ ,  $(\eta\mu_\star)_{B^\sharp}$  and  $f_{B^\sharp}$  be as in the sketched proof of Thm 2. Then  $f_{B^\sharp}^b \circ (\eta\mu_\star)_{B^\sharp} = f_{B^\sharp}$ .*

*Proof.* For notational simplicity, we abbreviate  $\eta_{\mu_\star}$  to just  $\eta$  in the course of this proof. Consider now arbitrary  $\langle i, b \rangle \in B^\sharp$ .

- In the case of  $b \in B_C$ , the equation  $(f_{B^\sharp}^b \circ \eta_{B^\sharp})(\langle i, b \rangle) = f_{B^\sharp}(\langle i, b \rangle)$  is established by straightforward calculation from the definitions of  $f, \eta, \iota$  and  $f^b$ .
- Assume now that  $b \notin B_C$ . Then, by the definition of  $\eta$  (recall also Def. 6), it must be that  $b \leq p$ , where  $p$  the point of  $\mu_\star$  in Thm. 2. One now has  $(f_{B^\sharp}^b \circ \eta_{B^\sharp})(\langle i, b \rangle) = f_{B^\sharp}^b(\eta_{B^\sharp}(\langle i, b \rangle)) = f^b(\langle 0, \eta_B(b) \rangle)$  by definition of  $\eta$ . Further, by two applications of Lemma 4 above (to  $f^b$  and  $f$  respectively), it follows that  $f_{B^\sharp}(\langle 0, b \rangle) = \langle 0, f_B(b) \rangle = \langle 0, f_B(\eta_B(b)) \rangle = f_{B^\sharp}^b(\langle 0, \eta_B(b) \rangle)$  and thus that

$$(f_{B^\sharp}^b \circ \eta_{B^\sharp})(\langle i, b \rangle) = f_{B^\sharp}(\langle 0, b \rangle) . \tag{2}$$

In the case of  $i = 0$ , equation (2) immediately yields the desired result. When  $i = 1$ , on the other hand,  $b \notin B_C$  and Lemma 4(2) above yield  $f_{B^\sharp}(\langle 1, b \rangle) = \langle 0, f_B(b) \rangle = (f_{B^\sharp}^b \circ \eta_{B^\sharp})(\langle 1, b \rangle)$  because of equation (2) above.

**Lemmas Used in the Proof of Theorem 3**

**Lemma 6.** *Let  $\mu_\star$  be  $(B, E; s, t : E \rightarrow B^\sharp)$  with point  $p$ ;  $e_1, e_2 \in E$  and  $\hat{v} \in B_C^\sharp$ . If  $\xrightarrow{e_1} \hat{v} \xrightarrow{e_2}$  is part of a must-run through  $Z(\mu_\star)$  (i.e.  $\hat{v}$ , the target  $t'(e_1)$  of  $e_1$  in  $Z(\mu_\star)$ , and the source  $s'(e_2)$  of  $e_2$  in  $Z(\mu_\star)$  satisfy the conditions in Def. 17), then there exists  $v \in B^\sharp$  such that  $\eta_{\mu_\star}(v) = \hat{v}$  and  $\xrightarrow{e_1} v \xrightarrow{e_2}$  is part of a must-run through  $\mu_\star$ .*

*Proof.* Having arbitrarily fixed  $\mu_\star$ , we abbreviate as just  $\eta$  the morphism  $\eta_{\mu_\star}$  of Def. 15. We also write, for brevity,  $t_B$  and  $s_B$  for the composites  $\pi_B \circ t$  and  $\pi_B \circ s$  respectively. Recalling from the definition of  $Z(\mu_\star)$  (Def. 14) that  $s' = \eta_{B^\sharp} \circ s$  and  $t' = \eta_{B^\sharp} \circ t$ , we proceed by case analysis:

1. Case  $\pi_0(t'(e_1)) = 1$ . Then, by the definitions of  $t'$  and  $\eta_{B^\sharp}$ , one must have  $t(e_1) = t'(e_1)$ . If also  $\pi_0(s'(e_2)) = 1$  then, similarly,  $s(e_2) = s'(e_2)$  and one can simply take  $v = \hat{v}$ . If, on the other hand, one assumes  $\pi_0(s'(e_2)) = 0$  then, from condition  $\hat{v} \leq s'(e_2)$  and Def. 11, it follows that one must also have  $\hat{v} = s'(e_2)$ . Now, from condition  $\hat{v} \leq t'(e_1)$ , it follows that one must have  $s'(e_2) \leq t'(e_1)$  which, in turn, implies  $\eta_B(s_B(e_2)) \leq \eta_B(t_B(e_1))$ . Taking  $v = \langle 0, s_B(e_2) \rangle$  one therefore immediately has  $v \leq s(e_2)$ , as required. Further,  $v \leq \langle 1, s_B(e_2) \rangle \leq \langle 1, \eta_B(s_B(e_2)) \rangle \leq \langle 1, \eta_B(t_B(e_1)) \rangle = t'(e_1) = t(e_1)$  because of the definition of  $\eta_B$  (Def 10), and Def. 6. Finally,  $\eta_{B^\sharp}(v) = \eta_{B^\sharp}(\langle 0, s_B(e_2) \rangle) = \langle 0, \eta_B(s_B(e_2)) \rangle = s'(e_2) = \hat{v}$ .
2. Case  $\pi_0(t'(e_1)) = 0$ . It follows from Def. 17 that one must have  $\pi_0(s'(e_2)) = 1$ . In that case one takes  $v = \langle 0, t_B(e_1) \rangle$  and the proof proceeds in a manner similar to that of the previous case above.