Inferentialist Resource Semantics (Extended Abstract)[⋆,⋆⋆]

Alexander V. Gheorghiu[a,1] Tao Gu[a,2] David J. Pym[a,b,c,3]

[a] *Department of Computer Science*
University College London
London WC1E 6BT, UK

[b] *Department of Philosophy*
University College London
London WC1E 6BT, UK

[c] *Institute of Philosophy*
University of London
London WC1H 0AR, UK

Abstract

In systems modelling, a *system* typically comprises located resources relative to which processes execute. One important use of logic in informatics is in modelling such systems for the purpose of reasoning (perhaps automated) about their behaviour and properties. To this end, one requires an interpretation of logical formulae in terms of the resources and states of the system; such an interpretation is called a *resource semantics* of the logic. This paper shows how *inferentialism* — the view that meaning is given in terms of inferential behaviour — enables a versatile and expressive framework for resource semantics. Specifically, how inferentialism seamlessly incorporates the assertion-based approach of the logic of Bunched Implications, foundational in program verification (e.g., as the basis of Separation Logic), and the renowned number-of-uses reading of Linear Logic. This integration enables reasoning about shared and separated resources in intuitive and familiar ways, as well as about the composition and interfacing of system components.

Keywords: inferentialism, proof-theoretic semantics, resource semantics, linear logic, bunched logic, systems modelling

1 Introduction

Within informatics, perhaps the most important systems concept is that of a *distributed system* [13,58]. From the systems modelling perspective, with a little abstraction, one can think of a distributed system as comprising delimited collections of interconnected component systems [33,61,11,12,3,18]. These systems

⋆ This work has been partially supported by the UK EPSRC grants EP/S013008/1 and EP/R006865/1 and by the EU MOSAIC MCSA-RISE project.

⋆⋆We are grateful to Gabriele Brancati, Tim Button, Yll Buzoku, Tristan Caulfield, Diana Costa, Timo Eckhardt, Didier Galmiche, Timo Lang, Sonia Marin, Peter O'Hearn, Elaine Pimentel, Eike Ritter, Edmund Robinson, and Will Venters for discussions of various aspects of this work.

[1] Email: alexander.gheorghiu.19@ucl.ac.uk

[2] Email: tao.gu.18@ucl.ac.uk

[3] Email: d.pym@ucl.ac.uk

ultimately comprise *locations*, at which are situated *resources* relative to which *processes* execute, consuming, creating, moving, combining, and otherwise manipulating resources as they evolve, so delivering services.

One primary application of logic in informatics is for representing, understanding, and reasoning about systems; this determines the field of *logical systems modelling*. In this context, the 'modelling' is used both in the general and mathematical logical sense. The goal is to utilize logic to represent, analyze, and simulate systems by interpreting logical structures and relationships in terms of concepts relevant to the model in question. We discuss several examples below.

In the field of logical systems modelling, *substructural logics* are useful because of their *resource interpretations*. The study of such interpretations of logics, especially in the context of systems modelling, is called *resource semantics* — see Section 3. Perhaps the most celebrated examples of resource semantics are the 'number-of uses' reading of *Linear Logic* (LL) [23] and the 'sharing/separation' reading of *the logic of Bunched Implications* (BI) [35]. While both are applicable to systems modelling, these two readings work in different ways and are related to different ways of doing the modelling as we discuss presently.

The *number-of-uses reading* of LL (e.g., [32,24,1,28]) proceeds through LL's proof theory and is not reflected in its truth-functional semantics (e.g., [24,2,14]). Heuristically, it concerns the dynamics of the system: formulae denote processes and resources themselves. It is conveniently illustrated by the (by now familiar) *Vending Machine Model*:

> *Having a chocolate bar is denoted by C, and having one euro by $1€$. The formula $1€ \to C$ (using material implication) is intended to denote '$1€$ buys one chocolate bar', since (by* modus ponens*) $1€$ together with $1€ \to C$ yield C. However this probably doesn't model the economy we meant as it also follows that $1€ \to C \wedge C$ (i.e., '$1€$ buys two chocolate bars')! The point is that a euro is a resource that should be* consumed *in a transaction, yielding only one chocolate bar. We can model things more carefully using linear implication and the tensor product: from $1€ \multimap C$, we infer $1€ \otimes 1€ \multimap C \otimes C$, but not $1€ \multimap C \otimes C$.*

Observe that we have not given a model in the formal sense of an algebraic structure, but rather have given a reading of the logical structures and relationships within LL in terms of some concepts pertinent to vending machines.

In contrast to the number-of-uses reading of LL, the *sharing/separation* reading of BI (e.g., [35,44,3,46]) proceeds through its (ordered-monoidal/relational) model-theoretic semantics [20]. In this case, one has models in the general sense represented by models in the algebraic sense; for example, beginning with the stack-heap model of computer memory and using the *sharing interpretation* of BI, Ishtiaq and O'Hearn [29] (following Reynolds [49]) invented the *stack-heap* model of (usually Boolean) BI on which Separation Logic [50] rests. The details of this and its efficacy has been discussed by Pym et al. [44,46,17]. Returning to the *Vending Machine Model* above:

> *We use the ordered monoid of the natural numbers $\langle \mathbb{N}, \leq, +, 0 \rangle$ to model euros. Suppose chocolate bar A costs $2€$ and chocolate bar B costs $3€$: we write $3 \models A \wedge B$ to say that $3€$ suffice for* both *chocolates in the sense that one could purchase either one; and we write $5 \models A * B$ to denote that $5€$ may be split/shared to purchase A and B* separately*. Note that, in the first example ($\wedge$), we use* persistence*, in the sense that as $2€$ suffice for A and $3 > 2$, so do $3€$.*

There are some important facts about the number-of-uses and sharing/separation readings and their use in systems modelling that need to be emphasised:

- firstly, while both readings are useful, they are individually limited in the context of systems modelling: sharing/separation expresses the *structure* of distributed systems and number-of-uses expresses the *dynamics* of the resources involved;
- secondly, the two readings operate in completely different paradigms: number-of-uses proceeds from the proof theory of LL, while sharing/separation proceeds from the model theory of BI.

We propose and discuss the use of a unifying framework for the two resource interpretations given above: *proof-theoretic semantics* (P-tS). The semantic paradigm supporting P-tS is *inferentialism* [6] — the view that meaning (or validity) arises from inference. Therefore, as a point of departure for this work, we adopt an inferentialist view of systems modelling in which the basis for reasoning about a system and its properties should be based on inferences over the underlying specification of the system. More precisely,

in distributed systems *policies* are used to determine behaviours. From the inferentialist point of view, these behaviours can be interpreted as (collectively) giving the *meaning* of the distributed system [30]. In this paper, we give an account of such 'inferential' models of distributed systems using recent advances in the proof-theoretic semantics of substructural logics. For example, an implication $\varphi \to \psi$ denotes a policy for an action that moves the system from a state that satisfies policy φ to a state that satisfies policy ψ: instantiated to the Vending Machine Model, policy φ is the state of having 1€ and policy ψ is the state of having a chocolate bar, our model of the policy should make the exchange of resource linear in the sense that, having executed it, one no longer has the resource 1€. Details are given in Section 3.

Having given the relevant background in proof-theoretic semantics in Section 2, the main contribution of the paper begins in Section 3 with a general definition of *resource semantics*. We explain how the recent P-tS of LL and of BI, respectively, correspond to their number-of-uses and sharing/separation interpretations; briefly, they both proceed by a semantic judgment relation called *support*, $\Gamma \Vdash^R_{\mathscr{B}} \varphi$. In Section 3, we give an interpretation of each component of support judgments in the context of systems modelling: intuitively, Γ describes a system policy, $\mathscr{B}$ represents an inferential model of a system policy, R constitutes a collection of available resources, and φ signifies an assertion regarding the executed system according to the policy of Γ and $\mathscr{B}$ and collection resources R. In Section 4, we offer an illustration of this interpretation by modelling the departure security infrastructure of an airport and multi-factor authentication. These examples are intended to be both familiar and evidently generic in that it should be clear how other examples would map onto them quite naturally. In Section 5, we discuss informally how our inferentialist resource semantics applies to distributed systems in general.

We include a brief glossary for the various logics discussed in this paper. We have several subsystems of *Linear Logic* (LL): *intuitionistic* LL is abbreviated ILL, ILL without ! is abbreviated IMALL, IMALL without $\&, \oplus, 1, \bot$ is abbreviated IMLL. The *logic of Bunched Implications* is abbreviated BI. Finally, *Intuitionistic Propositional Logic* is abbreviated IPL.

2 Proof-theoretic Semantics: Base-extension Semantics

In model-theoretic semantics (M-tS), logical consequence is defined in terms of truth in models — understood as abstract mathematical structures. In the standard reading of Tarski [59,60], a propositional formula φ follows (model-theoretically) from a context Γ iff every model of Γ is a model of φ. Therefore, consequence is understood as the *transmission of truth*. From this perspective, *meaning* and *validity* are characterized in terms of *truth*.

Meanwhile, in proof-theoretic semantics (P-tS) [39,40,15,16,62,22], meaning and validity are characterized in terms of *proofs* — understood as objects denoting collections of acceptable inferences from accepted premisses — and *provability*. To emphasize: it is not that one provides a proof system for the logic, but rather one explicates the meaning of the connectives *in terms of* proof systems. Indeed, as Schroeder-Heister [55] observes, since no formal system is fixed (only notions of inference) the relationship between semantics and provability remains the same as it has always been: soundness and completeness are desirable features of formal systems. Essentially, *proof* in P-tS plays the part of *truth* in M-tS.

The field of P-tS is wide and encompasses several distinct approaches. In this paper, we restrict attention to *base-extension semantics* (B-eS), which is a particular approach to P-tS. A B-eS is a characterization of a logic by a judgment relation called *support* that defines the validity of formulae. It is inductively defined according to the syntax of the logic. It is analogous to the *satisfaction* judgment in M-tS (cf. [31]). Crucially, the base case is given by 'derivability in a base', which is regarded as a *logic-free* notion of proof; that is, bases are proof systems restricted to atoms. Despite being structurally similar to M-tS, the subtle differences in the setup have significant consequences (discussed below).

In this paper, we follow the approach to B-eS by Sandqvist [51,52,53] and Piecha et al. [37,36,38]. While there are other versions than that presented here (cf. [25,56]), they are not directly relevant for this work. In this section, we present three B-eS on which we base the inferentialist account of resource semantics. First, in Section 2.1, we discuss the B-eS of *intuitionistic propositional logic* (IPL) as this provides a good background and example of how B-eS works. Second, in Section 2.2, we discuss the B-eS of (inutitionistic) LL, which arises through modifications on the B-eS of LL by accounting for some notion of atomic 'resource'. Finally, in Section 2.3, we discuss the B-eS of BI, which enriches the work on LL with the more delicate structure of bunches.

$\Vdash_{\mathscr{B}} p$	iff	$\vdash_{\mathscr{B}} p$	(At)
$\Vdash_{\mathscr{B}} \varphi \to \psi$	iff	$\varphi \Vdash_{\mathscr{B}} \psi$	($\to$)
$\Vdash_{\mathscr{B}} \varphi \wedge \psi$	iff	$\Vdash_{\mathscr{B}} \varphi$ and $\Vdash_{\mathscr{B}} \psi$	($\wedge$)
$\Vdash_{\mathscr{B}} \varphi \vee \psi$	iff	$\forall \mathscr{C} \supseteq \mathscr{B} \forall p \in \mathbb{A}$, if $\varphi \Vdash_{\mathscr{C}} p$ and $\psi \Vdash_{\mathscr{C}} p$, then $\Vdash_{\mathscr{C}} p$	($\vee$)
$\Vdash_{\mathscr{B}} \bot$	iff	$\Vdash_{\mathscr{B}} p$ for any $p \in \mathbb{A}$	($\bot$)
$\Vdash_{\mathscr{B}} \Gamma$	iff	$\Vdash_{\mathscr{B}} \gamma$ for any $\gamma \in \Gamma$	
$\Gamma \Vdash_{\mathscr{B}} \varphi$	iff	$\forall \mathscr{C} \supseteq \mathscr{B}$, if $\Vdash_{\mathscr{C}} \Gamma$, then $\Vdash_{\mathscr{C}} \varphi$	(Inf)
$\Gamma \Vdash \varphi$	iff	$\Gamma \Vdash_{\mathscr{B}} \varphi$ for all bases $\mathscr{B}$	

Fig. 1. Base-extension Semantics for IPL

2.1 Intuitionistic Propositional Logic

In this section, we introduce the techniques of B-eS through a treatment, due to Sandqvist [53], of IPL. Fix a (denumerable) set of atomic propositions $\mathbb{A}$. Here (and in the sequel) lower-case Romans denote atoms and upper-case Romans denote sets (and, later, multisets and bunches) of atoms; lower-case Greeks denote formulae and upper-case Greeks denote sets (and, later, multisets and bunches) of formulae.

The B-eS for IPL begins by defining atomic rules. An *atomic rule* is a natural deduction rule of the following form, in which $p, p_1, ..., p_n$ are atoms and P_1,...,P_n are (possibly empty) sets of atoms:

$$\frac{}{p}\ \mathsf{A} \qquad \frac{\begin{array}{ccc}[P_1] & & [P_n] \\ p_1 & \ldots & p_n\end{array}}{p}\ \mathsf{R}$$

A *base* is a set of such atomic rules. We write $\mathscr{B}, \mathscr{C}, \ldots$ to denote bases, and $\varnothing$ to denote the empty base (i.e., the base with no rules). We say $\mathscr{C}$ is an *extension* of $\mathscr{B}$ if $\mathscr{C}$ is a superset of $\mathscr{B}$, denoted $\mathscr{C} \supseteq \mathscr{B}$.

Importantly, atomic rules are taken *per se* and not closed under substitution when creating derivations.

Definition 2.1 (Derivability in a Base) Derivability in a base $\mathscr{B}$ *is the smallest relation* $\vdash_{\mathscr{B}}$ *satisfying the following:*

- REF: $P, p \vdash_{\mathscr{B}} p$, *for any* $p \in \mathbb{A}$
- APP$_1$: *If* $\mathsf{A} \in \mathscr{B}$, *then* $P \vdash_{\mathscr{B}} p$
- APP$_2$: *If* $\mathsf{R} \in \mathscr{B}$ *and* $P, P_1 \vdash_{\mathscr{B}} p_1, \ldots, P, P_n \vdash_{\mathscr{B}} p_n$, *then* $P \vdash_{\mathscr{B}} p$.

Derivability in a base ($\vdash_{\mathscr{B}}$) is the base case of *support in a base* ($\Vdash_{\mathscr{B}}$) that gives the B-eS — that is, for any $p \in \mathbb{A}$, $\Vdash_{\mathscr{B}} p$ iff $\vdash_{\mathscr{B}} p$. The inductive case is given analogously to the definition of satisfaction in M-tS.

Definition 2.2 (Support for IPL) *Support is the smallest relation* $\Vdash$ *satisfying the clauses of Figure 1.*

Theorem 2.3 (Sandqvist [54]) $\Gamma \vdash_{\mathrm{IPL}} \varphi$ *iff* $\Gamma \Vdash \varphi$.

Soundness — that is, $\Gamma \vdash_{\mathrm{IPL}} \varphi$ implies $\Gamma \Vdash \varphi$ — follows from showing that support admits all the rules of Gentzen's NJ [57]. Completeness — that is, $\Gamma \Vdash \varphi$ implies $\Gamma \vdash_{\mathrm{IPL}} \varphi$ — is more subtle. In this case one constructs a special base $\mathscr{N}$ that 'simulates' Gentzen's NJ [57] and uses the clauses of the semantics to show that support in $\mathscr{N}$ is indeed equivalent to provability in NJ.

Before concluding this section, there are a couple of important features of the semantics to note:

Remark 2.4 The clauses for various connectives in a B-eS can be substantially different from their treatments in other semantics. In particular, the B-eS for IPL has the following clauses for disjunction ($\vee$):

$$\Vdash_{\mathscr{B}} \varphi \vee \psi \qquad \text{iff} \qquad \forall \mathscr{C} \supseteq \mathscr{B} \forall p \in \mathbb{A}, \text{ if } \varphi \Vdash_{\mathscr{C}} p \text{ and } \psi \Vdash_{\mathscr{C}} p, \text{ then } \Vdash_{\mathscr{C}} p$$

This is one significant consequence of the inferentialist setup: Piecha et al. [37,36,38] have shown that the standard meta-level 'or' clause — that is, $\Vdash_{\mathscr{B}} \varphi \vee \psi$ iff $\Vdash_{\mathscr{B}} \varphi$ or $\Vdash_{\mathscr{B}} \psi$ — yields incompleteness. Note that

the clause is closely related to the 'second-order' definition of disjunction (see Prawitz [41]) — that is,

$$U + V = \forall X\,((U \to X) \to (V \to X) \to X)$$

This adumbrates the categorical perspective on B-eS for IPL given by Pym et al. [45,42]

Remark 2.5 The treatment of non-empty context is given by a *base-extension*,

$$\Gamma \Vdash_{\mathscr{B}} \varphi \qquad \text{iff} \qquad \forall \mathscr{X} \supseteq \mathscr{B}, \text{ if } \Vdash_{\mathscr{X}} \Gamma, \text{ then } \Vdash_{\mathscr{X}} \varphi$$

While *prima facie* this use of base-extension appears to be related to pre-order in the M-tS of IPL (cf. [31]), that remains to be determined (though see [25,56]).

This brief introduction suffices to highlight some important features of B-eS. The remainder will now give the detail in the context of substructural logic.

2.2 Linear Logic

The challenges and motivations involved in producing a B-eS for a substructural logic are discussed in detail by the authors [21,26]. In this section, we present the treatment of LL provided by the authors [21] (with the additives given by Buzoku [10,9]).

It is instructive to begin with the semantics for IMLL. This simplifies the problem to the following: *How does one make support for IPL linear?* Looking at (Inf) in Figure 1, we expect that φ being supported in a base $\mathscr{B}$ relative to some multiset of formulas Γ means that the 'resources' garnered by Γ suffice to produce φ. We express this by enriching the notion of support with multisets of resources P and U combined with multiset union — denoted $,$ — as follows:

$$\Gamma \Vdash^{S}_{\mathscr{B}} \varphi \quad \text{iff} \quad \text{for any } \mathscr{X} \supseteq \mathscr{B} \text{ and any } U, \text{ if } \Vdash^{U}_{\mathscr{X}} \Gamma, \text{ then } \Vdash^{S,U}_{\mathscr{B}} \varphi$$

where

$$\Vdash^{U}_{\mathscr{B}} \Gamma_1,\Gamma_2 \quad \text{iff} \quad \text{there are } U_1 \text{ and } U_2 \text{ such that } U = (U_1,U_2),\ \Vdash^{U_1}_{\mathscr{X}} \Gamma_1, \text{ and } \Vdash^{U_2}_{\mathscr{X}} \Gamma_2$$

The 'resource' reading here is indeed inspired by the number-of-uses reading of LL. In this way, this paper presents a semantics of ILL in which this celebrated reading is, by design, present.

We now continue to give the technical details of semantics. Again, fix a (denumerable) set of atoms $\mathbb{A}$. Recall that we use lower-case Roman letters $p, \ldots$ to denote elements of $\mathbb{A}$, upper-case Roman letters $P, Q, S, U, V \ldots$ to denote (finite) *multisets* of elements of $\mathbb{A}$, lower-case Greek letters $\varphi, \psi, \chi, \ldots$ to denote formulae, and upper-case Greek letters $\Gamma, \Delta, \ldots$ to denote (finite) *multisets* of formulae. More generally, for a set X, we write $\mathbb{M}(X)$ to denote the set of (finite) *multisets* of elements of X; we use $,$ to denote multiset union, and $\varnothing$ to denote the empty multiset.

For ILL, atomic rules are taken in the format used by Bierman [5] and Negri [34]; that is, rules have premisses with (possibly empty) multiset of hypothesises that are divided into sets,

$$\frac{}{p}\ \mathsf{A} \qquad \frac{\left\{\begin{array}{c}[P_1^{(1)}]\\ p_1^{(1)}\end{array} \ \ldots\ \begin{array}{c}[P_{n_1}^{(1)}]\\ p_{n_1}^{(1)}\end{array}\right\} \ \cdots\ \left\{\begin{array}{c}[P_1^{(k)}]\\ p_1^{(k)}\end{array} \ \ldots\ \begin{array}{c}[P_{n_k}^{(k)}]\\ p_{n_k}^{(k)}\end{array}\right\}}{p}\ \mathsf{R}$$

A base $\mathscr{B}$ is a set of such rules; we use the same conventions as in Section 2.1.

Definition 2.6 (Derivability in a Base) *Derivability in a base $\mathscr{B}$ is the smallest relation $\vdash_{\mathscr{B}}$ satisfying the following:*

- REF: *$p \vdash_{\mathscr{B}} p$, for all $p \in \mathbb{A}$*
- APP$_1$: *If $\mathsf{A} \in \mathscr{B}$, then $\varnothing \vdash_{\mathscr{B}} p$.*
- APP$_2$: *If $\mathsf{R} \in \mathscr{B}$ and there are $C_1, \ldots, C_k \in \mathbb{M}(\mathbb{A})$ s.t. $C_i\,, P_j^{(i)} \vdash_{\mathscr{B}} p_j^{(i)}$ for every $i = 1 \ldots k$ and $j = 1 \ldots n_i$, then $C_1, \ldots, C_k \vdash_{\mathscr{B}} p$.*

$\Vdash^S_{\mathscr{B}} p$	iff	$S \vdash_{\mathscr{B}} p$	(At)
$\Vdash^S_{\mathscr{B}} \varphi \multimap \psi$	iff	$\varphi \Vdash^S_{\mathscr{B}} \psi$	($\multimap$)
$\Vdash^S_{\mathscr{B}} \varphi \otimes \psi$	iff	$\forall \mathscr{X} \supseteq \mathscr{B}$, $\forall U$, $\forall p$, if $\varphi, \psi \Vdash^U_{\mathscr{X}} p$, then $\Vdash^{S,U}_{\mathscr{B}} p$	($\otimes$)
$\Vdash^S_{\mathscr{B}} \varphi \oplus \psi$	iff	$\forall \mathscr{X} \supseteq \mathscr{B}$, $\forall U$, $\forall p$, if $\varphi \Vdash^U_{\mathscr{X}} p$ and $\psi \Vdash^U_{\mathscr{X}} p$, then $\Vdash^{S,U}_{\mathscr{X}} p$	($\oplus$)
$\Vdash^S_{\mathscr{B}} \varphi \,\&\, \psi$	iff	$\Vdash^S_{\mathscr{B}} \varphi$ and $\Vdash^S_{\mathscr{B}} \psi$	(&)
$\Vdash^S_{\mathscr{B}} 1$	iff	$\forall \mathscr{X} \supseteq \mathscr{B}$, $\forall U$, $\forall p \in \mathbb{A}$, if $\Vdash^U_{\mathscr{X}} p$, then $\Vdash^{S,U}_{\mathscr{X}} p$	(1)
$\Vdash^S_{\mathscr{B}} 0$	iff	$S \vdash_{\mathscr{B}} p$ for any $p \in \mathbb{A}$	(0)
$\Vdash^S_{\mathscr{B}} \Gamma, \Delta$	iff	$\exists U, V$ s.t. $S = (U, V)$, $\Vdash^U_{\mathscr{B}} \Gamma$, and $\Vdash^V_{\mathscr{B}} \Delta$	(,)
$\Gamma \Vdash^S_{\mathscr{B}} \varphi$	iff	$\forall \mathscr{X} \supseteq \mathscr{B}$ and $\forall U$, if $\Vdash^U_{\mathscr{X}} \Gamma$, then $\Vdash^{S,U}_{\mathscr{X}} \varphi$	(Inf)
$\Gamma \Vdash \varphi$	iff	$\Gamma \Vdash^{\varnothing}_{\mathscr{B}} \varphi$ for any base $\mathscr{B}$	

Fig. 2. Base-extension Semantics for IMALL

Remark 2.7 We make Definition 2.1 substructural to yield Definition 2.6 by restricting the assumption to be the atom itself in REF and by combining assumptions for each collection of premisses in APP.

Relative to this notion of derivability in a base, the definition of *support* for ILL without ! (i.e., IMALL) follows as before:

Definition 2.8 (Support for IMALL) *Support is the smallest relation* $\Vdash$ *satisfying the clauses of Figure 2.*

Theorem 2.9 (Gheorghiu, Gu, and Pym [21], Buzoku [10]) $\Gamma \vdash_{\mathrm{IMALL}} \varphi$ *iff* $\Gamma \Vdash \varphi$.

Soundness and completeness are provided in the same way as for IPL in Section 2.1; that is, soundness follows by the admissibility of the rules from a natural deduction system for ILL with respect to support, and completeness by 'simulating' a natural deduction system for ILL by using the clauses of the semantics.

Remark 2.10 Intuitively, the indexed multiset of atoms denotes available *resources*. The Vending Machine Model discussed in Section 1 is easily expressed:

> Let Γ be a theory defining the machine's behaviour (e.g., Γ is the multiset of just $1 \in \multimap C$) and let $\mathscr{V}$ be any base supporting that theory (i.e., $\Vdash^{\varnothing}_{\mathscr{V}} \Gamma$), then the situation 'one euro will buy one chocolate bar' corresponds to the judgement $\Vdash^{1\in}_{\mathscr{V}} C$.

This stands in contrast to M-tS of linear logics in which the number-of-uses reading is not clearly expressed.

What about the exponential, '!' ? Intuitively, its resource reading is that the resource to which it is applied is *unbounded* — zero or arbitrarily many instances of it may be used. This can be expressed in the presented set-up, but requires a lot of delicate overhead, as given in [10,9], that is not essential for presenting our basic account of resource semantics and its use in modelling distributed systems.

2.3 The Logic of Bunched Implications

In this section, we recall the B-eS for BI by the authors [26]. Essentially, BI is the *free* combination of IPL — with connective $\top, \wedge, \vee, \to$ — and IMLL — with connective $\top^*, *, \mathrel{-\!\!*}$. Intuitively, therefore, the B-eS is a free combination of the semantics for IPL in Section 2.1 and for IMLL in Section 2.2. While this intuition is fulfilled, there are several subtle technical details that require careful handling.

A principal challenge of BI is that, as a consequence of having two conjunctions and two (primitive) implications, contexts are not one of the standard data structures (e.g., sets, multiset, lists), but rather *bunches*, a term that derives from the relevance logic literature — see, for example, Read [47]. The subtlety is the algebraic properties of these bunches: in BI, part of the bunch admits the structural rules of weakening and contraction, and other parts do not. The B-eS of BI has to precisely express these algebraic properties. Furthermore, while our previous work has primarily utilized the 'natural deduction'

$$
\begin{array}{lll}
\Vdash^{S}_{\mathscr{B}} p & \text{iff} & S \vdash_{\mathscr{B}} p \qquad \text{(At)} \\
\Vdash^{S}_{\mathscr{B}} \varphi \wedge \psi & \text{iff} & \forall \mathscr{X} \supseteq \mathscr{B}, \forall U(\cdot) \in \dot{\mathbb{B}}(\mathbb{A}), \forall p \in \mathbb{A}, \text{ if } \varphi \fatsemi \psi \Vdash^{U(\cdot)}_{\mathscr{X}} p, \text{ then } \Vdash^{U(S)}_{\mathscr{X}} p \qquad (\wedge) \\
\Vdash^{S}_{\mathscr{B}} \varphi * \psi & \text{iff} & \forall \mathscr{X} \supseteq \mathscr{B}, \forall U(\cdot) \in \dot{\mathbb{B}}(\mathbb{A}), \forall p \in \mathbb{A}, \text{ if } \varphi \mathbin{,} \psi \Vdash^{U(\cdot)}_{\mathscr{X}} p, \text{ then } \Vdash^{U(S)}_{\mathscr{X}} p \qquad (*) \\
\Vdash^{S}_{\mathscr{B}} \varphi \vee \psi & \text{iff} & \forall \mathscr{X} \supseteq \mathscr{B}, \forall U(\cdot) \in \dot{\mathbb{B}}(\mathbb{A}), \forall p \in \mathbb{A}, \text{if } \varphi \Vdash^{U(\cdot)}_{\mathscr{X}} p \text{ and } \psi \Vdash^{U(\cdot)}_{\mathscr{X}} p, \text{ then } \Vdash^{U(S)}_{\mathscr{X}} p \qquad (\vee) \\
\Vdash^{S}_{\mathscr{B}} \varphi \to \psi & \text{iff} & \varphi \Vdash^{S \fatsemi (\cdot)}_{\mathscr{B}} \psi \qquad (\to) \\
\Vdash^{S}_{\mathscr{B}} \varphi \mathrel{-\!\!*} \psi & \text{iff} & \varphi \Vdash^{S \mathbin{,} (\cdot)}_{\mathscr{B}} \psi \qquad (\mathrel{-\!\!*}) \\
\Vdash^{S}_{\mathscr{B}} \bot & \text{iff} & \forall U(\cdot) \in \dot{\mathbb{B}}(\mathbb{A}), \forall p \in \mathbb{A}, \Vdash^{U(S)}_{\mathscr{B}} p \qquad (\bot) \\
\Vdash^{S}_{\mathscr{B}} \top & \text{iff} & \forall \mathscr{X} \supseteq \mathscr{B}, \forall U(\cdot) \in \dot{\mathbb{B}}(\mathbb{A}), \forall p \in \mathbb{A}, \text{ if } \Vdash^{U(\varnothing_+)}_{\mathscr{X}} p, \text{ then } \Vdash^{U(S)}_{\mathscr{X}} p \qquad (\top) \\
\Vdash^{S}_{\mathscr{B}} \top^{*} & \text{iff} & \forall \mathscr{X} \supseteq \mathscr{B}, \forall U(\cdot) \in \dot{\mathbb{B}}(\mathbb{A}), \forall p \in \mathbb{A}, \text{ if } \Vdash^{U(\varnothing_\times)}_{\mathscr{X}} p, \text{ then } \Vdash^{U(S)}_{\mathscr{X}} p \qquad (\top^*) \\
\Vdash^{S}_{\mathscr{B}} \Gamma \mathbin{,} \Delta & \text{iff} & \exists Q_1, Q_2 \in \mathbb{B}(\mathbb{A}) \text{ such that } S \succeq Q_1 \mathbin{,} Q_2, \Vdash^{Q_1}_{\mathscr{B}} \Gamma, \text{ and } \Vdash^{Q_2}_{\mathscr{B}} \Delta \qquad (\mathbin{,}) \\
\Vdash^{S}_{\mathscr{B}} \Gamma \fatsemi \Delta & \text{iff} & \exists Q_1, Q_2 \in \mathbb{B}(\mathbb{A}) \text{ such that } S \succeq Q_1, S \succeq Q_2, \Vdash^{Q_1}_{\mathscr{B}} \Gamma, \text{ and } \Vdash^{Q_2}_{\mathscr{B}} \Delta \qquad (\fatsemi) \\
\Gamma \Vdash^{R(\cdot)}_{\mathscr{B}} \psi & \text{iff} & \forall \mathscr{X} \supseteq \mathscr{B}, \forall U \in \mathbb{B}(\mathbb{A}), \text{ if } \Vdash^{U}_{\mathscr{X}} \Gamma, \text{ then } \Vdash^{R(U)}_{\mathscr{X}} \psi \qquad \text{(Inf)} \\
\Gamma \Vdash \varphi & \text{iff} & \Gamma \Vdash^{(\cdot)}_{\mathscr{B}} \psi \text{ for any base } \mathscr{B}
\end{array}
$$

Fig. 3. Base-extension Semantics for BI

proof formalism, BI is more suitably expressed in sequent calculus form (cf. [4]), necessitating adjustments to the entire framework of atomic systems.

Notwithstanding the challenges posed by bunches, BI is a well-established logic in systems modelling, as discussed in Section 1. In Section 3, we see that the treatment of bunches in the B-eS facilitates the expression of certain important features of distributed systems. We now proceed by providing the technical background for BI and defining its B-eS. Recall our previous specification of notation (in Section 2.1).

Bunches. Essentially, bunches in BI are the free combination of the context constructors of IPL — denoted $\varnothing_+$ and $\fatsemi$ — and IMLL — denoted $\varnothing_\times$ and $\mathbin{,}$.

Definition 2.11 (Bunch) Bunches *over some set* $\mathbb{X}$ *are defined as:*

$$X ::= x \in \mathbb{X} \mid \varnothing_+ \mid \varnothing_\times \mid X \fatsemi X \mid X \mathbin{,} X$$

The set of all bunches over $\mathbb{X}$ *is* $\mathbb{B}(\mathbb{X})$

If $\mathbb{F}$ is the set of formulas (over atoms $\mathbb{A}$) in BI, then $\mathbb{B}(\mathbb{F})$ is the set of bunches of formulas. A *sequent* in BI is a pair $\Gamma \triangleright \varphi$ in which $\Gamma \in \mathbb{B}(\mathbb{F})$ and $\varphi \in \mathbb{F}$.

Definition 2.12 (Sub-bunch) *If a bunch* Δ *is a sub-tree of a bunch* Γ*, then* Δ *is a* sub-bunch *of* Γ.

Remark 2.13 Importantly, sub-bunches are sensitive to occurrence — for example, in the bunch $\Delta \mathbin{,} \Delta$, each occurrence of Δ is a different sub-bunch.

We may write $\Gamma(\Delta)$ to express that Δ is a sub-bunch of Γ. This notation is traditional for BI [35], but requires careful handling. To avoid confusion, we never write Γ and later $\Gamma(\Delta)$ to denote the same bunch, but rather maintain one presentation. The substitution of Δ for Δ' in Γ is denoted $\Gamma(\Delta)[\Delta \mapsto \Delta']$. This is not a universal substitution, but a substitution of the particular occurrence of Δ meant in writing $\Gamma(\Delta)$. The bunch $\Gamma(\Delta)[\Delta \mapsto \Delta']$ may be denoted $\Gamma(\Delta')$ when no confusion arises.

Since bunches are perhaps not as standardly known as other data structures, we give some detail about their meta-theory that is essential to understanding BI. We require the notion of a *contextual bunch*, which we think of a contextual bunch as a *bunch with a hole* — cf. *nests with a hole* in work by Brünnler [8]. This is already suggested by the substitution notation, but it is worth giving an explicit mathematical treatment to avoid confusion.

Definition 2.14 (Contextual Bunch) *A* contextual bunch *(over* $\mathbb{X}$*) is a function* $b : \mathbb{B}(\mathbb{X}) \to \mathbb{B}(\mathbb{X})$ *such that there is* $\Gamma(\Delta) \in \mathbb{B}(\mathbb{X})$ *and* $b(\Sigma) = \Gamma(\Sigma)$ *for any* $\Sigma \in \mathbb{B}(\mathbb{X})$*. The identity on* $\mathbb{B}(\mathbb{X})$ *is a contextual bunch,*

it is denoted $(\cdot)$. *The set of all contextual bunches (over* $\mathbb{X}$*) is* $\dot{\mathbb{B}}(\mathbb{X})$.

Observe that $\dot{\mathbb{B}}(\mathbb{X})$ can be identified as the subset of $\mathbb{B}(\mathbb{X} \cup \{\circ\})$, where $\circ \notin \mathbb{X}$, in which bunches contain a single occurrence of $\circ$. Specifically, if $b(x) = \Gamma(x)$, then identify b with $\Gamma(\circ) \in \mathbb{B}(\mathbb{X} \cup \{\circ\})$. We write $\Gamma(\cdot)$ for the contextual bunch identified with $\Gamma(\circ)$. We require a notion of equivalence on bunches.

Definition 2.15 (Coherent Equivalence) *Bunches* $\Gamma, \Gamma' \in \mathbb{B}(\mathbb{X})$ *are* coherently equivalent *when* $\Gamma \equiv \Gamma'$*, where* $\equiv$ *is the least relation satisfying: commutative monoid equations for* $;$ *with unit* $\varnothing_+$ *and for* $,$ *with unit* $\varnothing_\times$*, and congruence (i.e., if* $\Delta \equiv \Delta'$ *then* $\Gamma(\Delta) \equiv \Gamma(\Delta')$*).*

In practice, bunches are regarded as the syntactic constructions in $\mathbb{B}(\mathbb{X})$ modulo coherent equivalence.

We require a way to express the structurality of the additive context-former.

Definition 2.16 (Bunch-extension) *The* bunch-extension *relation* $\succeq$ *is the least relation satisfying:*

- *(*weakening*) if* $\Gamma \equiv \Gamma'[\Delta \mapsto \Delta ; \Delta']$*, then* $\Gamma \succeq \Gamma'$
- *(*transitivity*) if* $\Gamma \succeq \Gamma'$ *and* $\Gamma' \succeq \Gamma''$*, then* $\Gamma \succeq \Gamma''$*.*

Note that (reflexivity) $\Gamma \succeq \Gamma$ is derivable here.

Base-extension Semantics. The notion of *bases* used for BI diverges from that of Section 2.1 and Section 2.2 as the natural deduction format is insufficiently expressive to make all the requisite distinctions about multiplicative and additive structures in bunches. Accordingly, we move to a sequent calculus format.

An *atomic sequent* is a pair $P \triangleright p$ where $P \in \mathbb{B}(\mathbb{A})$, $p \in \mathbb{A}$. An atomic rule is a rule over atomic sequents,

$$\frac{}{P \triangleright p}\ \mathsf{A} \qquad \frac{P_1 \triangleright p_1 \quad \dots \quad P_n \triangleright p_n}{P \triangleright p}\ \mathsf{R}$$

A base is a set of atomic rules; we use the same same conventions as in Section 2.1 and Section 2.2. They behave as sequent calculus systems.

Definition 2.17 (Derivability in a Base) *Derivability in a base* $\mathscr{B}$ *is the smallest relation* $\vdash \mathscr{B}$ *satisfying the following:*

- taut*:* $p \vdash_{\mathscr{B}} p$*, for all* $p \in \mathbb{A}$
- initial*: If* $\mathsf{A} \in \mathscr{B}$*, then* $P \vdash_{\mathscr{B}} p$
- rule*: If* $\mathsf{R} \in \mathscr{B}$ *and* $P_1 \vdash_{\mathscr{B}} p_1, \dots, P_n \vdash_{\mathscr{B}} p_n$*, then* $P \vdash_{\mathscr{B}} p$
- weak*: If* $P(Q) \vdash_{\mathscr{B}} p$*, then* $P(Q ; Q') \vdash_{\mathscr{B}} p$*, for any* $Q' \in \mathbb{B}(\mathbb{A})$
- cont*: If* $P(Q ; Q) \vdash_{\mathscr{B}} p$*, then* $P(Q) \vdash_{\mathscr{B}} p$
- exch*: If* $P(Q) \vdash_{\mathscr{B}} p$ *and* $Q \equiv Q'$*, then* $P(Q') \vdash_{\mathscr{B}} p$
- cut*: If* $T \vdash_{\mathscr{B}} q$ *and* $S(q) \vdash_{\mathscr{B}} p$*, then* $S(T) \vdash_{\mathscr{B}} p$*.*

Remark 2.18 While Definition 2.17 appears more complex than its analogues for IPL (Definition 2.1) and LL (Definition 2.6), this is caused by the change of formalism; in particular, taut corresponds to ref, initial to app_1, rule to app_2, and cut to the composition natural deduction proofs. The remaining conditions — namely, weak, cont, and exch — simply express the algebraic properties of bunches.

Definition 2.19 (Support for BI) *Support is the smallest relation* $\Vdash$ *satisfying the clauses in Figure 3, in which* $S \in \mathbb{B}(\mathbb{A})$*,* $R(\cdot) \in \dot{\mathbb{B}}(\mathbb{A})$*,* $\Gamma, \Delta \in \mathbb{B}(\mathbb{F})$*.*

Theorem 2.20 (Gu, Gheorghiu, and Pym [26]) $\Gamma \vdash_{\text{BI}} \varphi$ *iff* $\Gamma \Vdash \varphi$*.*

The proof of this again follows the techniques used by Sandqvist [53]; in particular, completeness follows from simulating a natural deduction proof systems for BI expressed in sequent calculus form (see, e.g., [35,43,7,27,19,26]). Moreover, using the correspondences in Remark 2.18, it is straightforward to express the B-eS of IPL and IMLL in Section 2.2 as a restriction of the B-eS of BI herein. Unlike in Section 2.2, in the B-eS of BI presented here, these fragments are freely combined.

There are a few things to remark upon about the support relation.

Remark 2.21 In contrast to the treatment of IPL [53] discussed in Section 1, support is now also parametrized by a bunch of atoms. In Section 3, we see that these may usefully be thought of as *resources*. The role of a contextual bunch can be seen particularly clearly in the (Inf) clause in Figure 3: the

resources required for the sequent $\Gamma \triangleright \varphi$ are combined with those required for Γ in order to deliver those required for φ.

Remark 2.22 The treatment of the support of a combination of contexts $\Gamma \, , \Delta$ follows the naïve Kripke-style interpretation of multiplicative conjunction, corresponding to an introduction rule in natural deduction, but the support of the tensor product $\varphi \otimes \psi$ follows the form of a natural deduction elimination rule. The significance of this is explored in Section 3.

Remark 2.23 The Vending Machine Model is expressed exactly as in Section 2.2 (Remark 2.10). This stands in stark contrast to the situation in Section 1 where the extant resource semantics of the logics require fundamentally different approaches to model the same system.

We conclude the technical background required for this paper: we have provided inferentialist semantics for IPL, IMALL, and BI. It remains to give a resource interpretation of these semantics.

3 Inferentialist Resource Semantics

We give a systematic account of how B-eS can be used to give an account of the resource semantics in which formulae are interpreted as assertions about the flow of resources in a distributed system, expressing both the sharing/separation interpretation of BI [44] and the number-of-uses reading of LL. It is also related to the syntactic resource semantics of Pfenning and Reed [48]. After the conceptual understanding in this section, we then give detailed examples in Section 4.

In the context of models of distributed systems that are formulated in terms of locations, resources, and processes, we begin with a conceptual definition of *resource semantics*, as follows:

> *A resource semantics for a system of logic is an interpretation of its formulae as assertions about states of processes and is expressed in terms of the resources that are manipulated by those processes.*

This definition requires a few notes: we intend no restriction on the assertions that are to be included in the scope of this definition; assertions may refer not only to ground states but also 'higher-order' assertions about state transitions. Moreover, we intend the manipulation of resources by the system's processes to include, *inter alia*, consuming/using resources, creating resources, copying/deleting resources, and moving resources between locations. Furthermore, we require that an adequate resource semantics should be able to provide accounts of the following concepts: counting of resources, composition of resources, comparison of resources, sharing of resources, and separation of resources.

A system's policy is what determines the system's processes and how they manipulate resources — see Section 1. In an *inferential* account of resource semantics, those behaviours determine the meaning of the system — see Section 2. We now show how B-eS is used to model policies directly, and then illustrate this account through examples in Section 4.

In the B-eS we have presented in Section 2, it is the (Inf) clause that articulates the consequence relation free of the structure of the connectives. Indeed, whilst the semantic clauses for connectives vary in the B-eS for LL and BI, their (Inf) clauses follow a similar rationale. This is no coincidence: $\Gamma \Vdash \varphi$ is about the transmission of validity, thus its definition reflects the semantic paradigm rather than of the specific logic system. Indeed, analogously in M-tS, $\Gamma \Vdash \varphi$ expresses that 'for any model $\mathfrak{M}$, if Γ is true in $\mathfrak{M}$, then so is φ', regardless of the specific logic and semantics of interest. The power of B-eS to unify the aforementioned resource interpretations can be summarized in the following general pattern of support judgement and its semantics clauses for LL and BI — (Inf) is the key clause that connects the left- and right-sides of the support judgement, free of reference to connectives:

$$\Gamma \Vdash^{S(\cdot)}_{\mathscr{B}} \varphi \quad \text{iff} \quad \forall \mathscr{C} \supseteq \mathscr{B}, \forall U \in \mathbb{R}(\mathbb{A}), \text{ if } \Vdash^{U}_{\mathscr{C}} \Gamma, \text{ then } \Vdash^{S(U)}_{\mathscr{C}} \varphi \qquad \text{(Gen-Inf)}$$

Let us spell out each ingredient of (Gen-Inf) and their roles in the resource semantics, which describes the execution of certain policy in a system:

- φ is a formula of the chosen logic. It is an assertion describing (a possible state of) the system
- Γ is a collection (e.g., multiset, bunch, etc.) of formulas. It specifies a policy describing the executions of a system's processes

- $\mathbb{R}(\mathbb{A})$ is some choice of atomic resource (e.g., multisets of atoms in ILL, bunches of atoms in BI)
- $S(\cdot)$ is some contextual atomic resource, such that whenever combined with some atomic resource in $\mathbb{R}(\mathbb{A})$, it returns a 'richer' atomic resource (e.g., multisets of atoms in IMLL, contextual bunches in BI). It specifies the resources that are available for the system model's processes to execute according to the given policy
- $\mathscr{B}, \mathscr{C}$ are bases of one's choice. They are models, and $\Vdash^{U}_{\mathscr{C}} \Gamma$ reads as $\mathscr{C}$ is a model of policy Γ when supplied with resource U.

Putting all these together, the support judgement $\Gamma \Vdash^{S(\cdot)}_{\mathscr{B}} \varphi$ says that, if policy Γ were to be executed with contextual resource $S(\cdot)$ based on the model $\mathscr{B}$, then the result state would satisfy φ. (Gen-Inf) explains how such execution is triggered: for arbitrary model $\mathscr{C}$ that extends $\mathscr{B}$, if policy Γ is met in model $\mathscr{C}$ using some resource U, then Γ could be executed, and the resulting state — which consumes U in the available contextual resource $S(\cdot)$ — satisfies φ.

We demonstrate how this general resource interpretation retrieves the number-of-uses reading and the sharing/separation semantics when instantiated in ILL (in particular, IMALL) and BI, respectively.

3.1 Linear Logic

Following Section 2.2, we restrict ourselves to IMALL. In this context, a multiset of atoms P denotes a contextual resource $S(\cdot)$ through multiset union — that is, $S(\cdot) : U \mapsto P \fatsemi U$.

The number-of-uses interpretation of LL instantiates the definition of resource semantics above as follows: a formula φ asserts that the system has some resources 'φ' (discussed below), and an implication $\varphi \multimap \psi$ asserts that the system has a process that uses the resources 'φ' and yields resources 'ψ'. What resources are 'φ' and 'ψ'? That depends on the policy $\mathscr{B}$ governing the system; for example, when the policy is empty ($\mathscr{B} = \varnothing$) and φ and ψ are atoms p and q, respectively, then 'φ' is p and 'ψ' is q — we give further examples below. Understanding that this interpretation is relativized to a policy is essential; for example, when $\mathscr{B}$ contains an axiom A, then one has an indefinite amount of the resource c available. Importantly, this handling of resource is the idea driving the 'simulation' proof of the completeness of the semantics — see Gheorghiu et al. [21] — discussed in Section 2.2.

The clauses in Figure 2 enable this way of reading number-of-uses as a resource semantics; that is, they explicate inductively how a formula is interpreted as a collection of resources relative to a policy — from $\Vdash^{P}_{\mathscr{B}} \varphi \multimap \psi$, one recovers a judgement $\Vdash^{P \fatsemi U}_{\mathscr{B} \cup \mathscr{C}} \psi$, in which U is 'φ' as determined by $\mathscr{C}$. We illustrate for some of connectives to show how their number-of-uses readings manifest:

- $\&$: This clause very clearly says that $\varphi \,\&\, \psi$ denotes that φ and ψ are available processes that each require the same resources; in other words, both process φ and ψ *may* be done, but only one *will* be.
- $\fatsemi$: Similar to the above, both process φ and ψ may be done, but this time both *must* be done, so the resources must be divided in some suitable way.
- $\otimes$: One may expect this clause to take precisely the same form as ($\fatsemi$), but taking the elimination form instead ensures a certain coherence in the system [21]. Indeed, taking this form, the clause allows us to see precisely how a formula denotes a collection of resources relative to a policy.

  Let us return to $\varphi \multimap \psi$. We already considered the case in which both φ and ψ are atoms. Suppose now that φ is a *tensor* of atoms (i.e., $\varphi := p_1 \otimes \ldots \otimes p_n$). Again, choosing the simplest possible $\mathscr{B}$ to model this process and make 'φ' as plain as possible, the judgement $\Vdash \varphi \multimap \psi$ can be reduced to $p_1 \fatsemi \ldots \fatsemi p_n \vdash_{\mathscr{B}} q$, in which $\mathscr{B}$ is simply a policy for a process that *uses resources φ and yields the resources ψ*.

  In general, one has 'φ' $\vdash_{\mathscr{B} \cup \mathscr{C}} q$, in which $\mathscr{C}$ ensures that 'φ' *behaves* as $p_1 \otimes \cdots \otimes p_n$. For example, '$\varphi$' can be a single atom and $\mathscr{C}$ may simulate the appropriate introduction and elimination rules,

$$\frac{\{p_1\} \quad \ldots \quad \{p_n\}}{\text{`}\varphi\text{'}} \qquad \frac{\{\text{`}\varphi\text{'}\} \quad \left\{ \begin{array}{c} [p_1 \fatsemi \ldots \fatsemi p_n] \\ x \end{array} \right\}}{x}$$

  A similar analysis can be made for when ψ is a tensor-formula.
- $\oplus$: This connective denotes non-deterministic choice: one of φ or ψ will fire, but we do not know

which. The elimination-rule format of the clause expresses precisely this: what the process φ and ψ can both yields with resources U, is yielded by $\varphi \oplus \psi$ with resources U.

This suffices to illustrate how the number-of-uses reading of LL manifests in its B-eS; the remaining connectives follow similarly. In the treatment of $\otimes$, we have concentrated on it as a pre-/post- condition of $\multimap$ (as opposed to the combination of two process that both execute as in $,$). This is to explicate how the format of the clause, which indeed passes through $,$, supports the interpretation of formulae as assertions about resources.

3.2 Bunched Implications

Briefly, a distributed system is comprised of components that exchange and process resources (see Section 5 for more on this). Two components *share* resources when they use the available resource at the same time and *separate* resources when the available resources must be divided between components. The sharing/separation interpretation of BI (see Pym [44]) pertains to modelling this aspect of the architecture of distributed systems. We illustrate how this manifests in BI's B-eS relative to the notation in (Gen-Inf).

First, the bunch of atoms U denotes the available resources. If V and W are collections of resources, then $V \between W$ denotes a collection of resources such that each sub-collection may be shared — that is, components that share resources may use both V and W simultaneously, just V, just W, or one component uses V while the other uses W — and V , W denotes a collection of resource such that each sub-collection must be used separately. This follows from the $\between$- and $,$-clauses, observing also the role played by bunch-extension in those clauses.

Second, the bunch Γ describes the architecture of the system. The two context-formers provide the sharing/separation reading: if Γ and Δ describe system policies for two components, then $\Gamma \between \Delta$ describes a system policy in which those components share resources, and Γ , Δ describes a system policy in which they separate resources. Again, this is immediately expressed by the $\between$- and $,$-clauses as the available resources are copied or divided between the sub-bunches, respectively. Intuitive readings of the associated units follow similarly.

Third, φ is an assertion about the state of the system in terms of its sharing/separation architecture; in particular, considering some of the connectives,

- $*$: denotes the combined effect of two separating components of the system taken as a single component
- $\wedge$: denotes the combined effect of two sharing components of the system taken as a single component
- $\mathrel{-\!\!*}$: denotes that, if the state of the system satisfies the antecent assertion, then the system can be modified to satisfy the consequent assertion
- $\rightarrow$: denotes that, if the state of the system satisfies the antecedent, then the (unmodified) system also satisfies the consequent.

Presented individually in this way, the significance of these readings is, perhaps, obscured. In the base case, an atom p may be thought of as the assertion that the system has a certain resource; thus, a formula $p \mathrel{-\!\!*} (q_1 * q_2)$ asserts that the system can move from a state in which a component may use a resource p to a state in which two separate components may use q_1 and q_2. This is made more concrete in Section 4 with examples given in which these readings are instantiated in the setting of a relatable distributed systems.

Observe that restricting to the multiplicative fragment of BI one also recovers IMLL, hence one has the number-of-uses interpretation described for LL above. However, this reading actually extends to the whole of BI. The sharing/separation reading arises from the interpretation of the meta-connectives (i.e., the context-formers and their units), but one can consistently keep a number-of-uses interpretation of formulae as the way in which they make assertions about the system (i.e., they express how resources behave within the system).

The implications assert that there is a *process* that transitions the system from a state satisfying the antecedent to one satisfying the consequent; they are distinguished by whether or not the processes *modify the system* when executed. What do we mean by 'modify'? In the basic case, we simply mean the consuming of resources; for example, both $p \mathrel{-\!\!*} q$ and $p \rightarrow q$ refers to a process in which a system moves from a state in which it has a resource p to one in which it has a q, but the former does it while consuming p (in the sense of the number-of-uses reading) and the latter only requires that p is available. In the general case, we must account for the fact that the precondition of the process may itself be a process; for example, $(\varphi \mathrel{-\!\!*} \psi) \mathrel{-\!\!*} \chi$ denotes a process that modifies the system from the state $\varphi \mathrel{-\!\!*} \psi$ (i.e., 'there

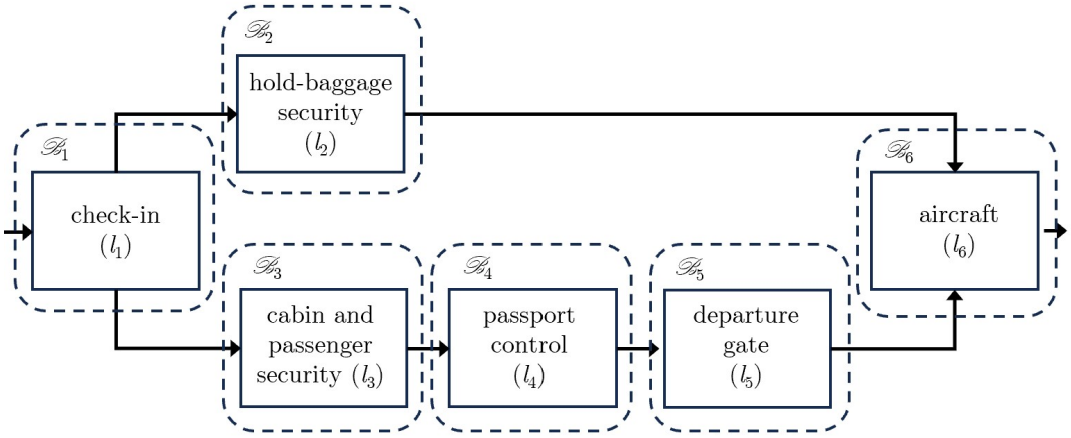

Fig. 4. The Airport Security Architecture

is a process ...') to the state χ.

Note, since $-\!\!*$ modifies the system, its required resources must be private (not shared with other processes) so it uses separated resources. Since $\rightarrow$ does not modify the system its required resources are the kind that may be shared.

3.3 Thesis

We have described how a general view of B-eS can be instantiated to explicate the resource semantics both of logics that exhibit the sharing/separation semantics of propositions and logics that support the number-of-uses readings of propositions.

Recall from Section 1 and our discussion of (Gen-Inf) that an inferentialist account of resource semantics is one in which policies determine behaviours and these behaviours collectively give the meaning of the system. Thus we conclude this section by asserting the thesis of this paper:

> *The paradigm of base-extension semantics provides an inferentialist account of resource semantics that uniformly encompasses both the number-of-uses readings — as found in the family of linear logics — and the sharing/separation semantics — as found in bunched logics, such as BI and relevance logics.*

Observe that this uniformity stands in contrast to extant accounts of these readings of linear and bunches logic that proceed through different frameworks — namely, proof theory and model theory.

We now illustrate this thesis, in Section 4, with two familiar yet evidently generic examples of distributed systems and, in Section 5, we describe the realization of the thesis for distributed systems more generally.

4 Generic Examples: Airport Security and MFA

Having developed (a sketch of) an inferentialist resource semantics, we now illustrate the ideas by exploring some substantial examples. These examples are intended to be familiar and relatable settings in which policies are applied to located resources (e.g., [12,11,3]). Despite being specific in their details, they are both structurally and conceptually quite generic. Since the resource semantics of BI provided by the B-eS expresses both sharing/separation and number-of-uses in systems modelling, we concentrate on this logic.

4.1 Airport Security Processes

A model of the departure security process at an airport shown in Figure 4. There are six locations — l_1, ..., l_6 — each of which has an associated *policy*, modelled by a base ($\mathscr{B}_i$ at l_i). There are lots of natural notions of *resource* in this setting (e.g., machines, baggage, passengers, etc), but to keep things simple and focused on *security*, we shall restrict attention to documentation (i.e., passports, tickets, bar-codes, etc).

You arrive at *check-in* (l_1). You show your passport, receive your boarding-card, and drop your hold-baggage. This situation is described by $\Gamma_1 := p \mathrel{-\!\!*} ((p \wedge t) * h)$ — atom p denotes your passport, t denotes the boarding-card (ticket), and h denotes the baggage-label. The $*$ is used because the system bifurcates at this point and the resources $p \wedge t$ and h go to *separating* components, and the $\mathrel{-\!\!*}$ is used because the system is modified: the state of passport p is changed as it only goes down one branch (the same as your ticket, t) and is no longer globally available. We explain the use of $*$ at l_1 in more detail below. An inferential model is given by a base $\mathscr{B}_1$ supporting Γ_1.

Next, two processes occur in parallel, one through l_2 and one through l_3, l_4, l_5. They are described by Δ_1 and Δ_2, respectively. Since they occur separately — that is, without sharing resources, this portion of the system is described by the bunch $\Gamma_2 := \Delta_1 \, , \Delta_2$. Indeed, the $,$ at l_2 matches the $*$ used in l_1 above.

The top path of Figure 4 passes through *hold-baggage security* (l_2). Here, the label h on your baggage is verified (and the baggage itself is checked). That h validates is denoted by $\Delta_1 := h \mathrel{-\!\!*} s_{\text{hold}}$.

The bottom path of Figure 4 passes through l_3, l_4, and l_5, modelled by χ_1, χ_2, and χ_3, respectively. This branch does not consume resources, so each is modelled using $\to$ (as opposed to $\mathrel{-\!\!*}$). They share resources (e.g., passport), so $\Delta_2 := \chi_1 \fatsemi \chi_2 \fatsemi \chi_3$. Their ordering is enforced by the security tokens (see below), otherwise $\to$ may be used in place of $\fatsemi$. We consider each location separately.

- You arrive at *security* (l_3). You must present a valid ticket to pass through the security gates. We denote this situation $\chi_1 := t \to s_{\text{cab}}$.
- You enter *passport control* (l_4). Your passport is validated and you are granted access to the gates. We denote this situation $\chi_2 := (s_{\text{cab}} \wedge p) \to s_{\text{pass}}$.
- You arrive at *the gate* (l_5). Your passport and ticket are checked and you are granted access. We denote this situation $\chi_3 := (s_{\text{pass}} \wedge p \wedge t) \to s_{\text{gate}}$.

The two separate processes (i.e., l_2 and l_3 to l_5) come together at *the aircraft* (l_6). Here the ground-crew and the air-crew have check-lists that need to be processed and combined, but that is all hidden from you. From your perspective, both you and your hold-baggage must have been authorized to board and then, assuming clearance, the plane takes off (f — flight). We model this situation by $\Gamma_3 := (s_{\text{gate}} * s_{\text{hold}}) \mathrel{-\!\!*} f$.

We have modelled airport security in three parts, producing Γ_1, Γ_2, and Γ_3. How should they be composed into a single theory describing the whole system at once? Observe that overall Figure 4 describes a *single process* in which a passport p yields flight f. Hence, it is modelled by an implication. The details of the process are what are described by Γ_1, Γ_2 (Δ_1 and Δ_2), and Γ_3, so we take their formula translations φ_1, φ_2 (ψ_1 and ψ_2), and φ_3, respectively (i.e., replace $\fatsemi$ with $\wedge$, and $,$ with $*$); that is,

$$\Gamma := \varphi_1 \mathrel{-\!\!*} ((\psi_1 * \psi_2) \mathrel{-\!\!*} \varphi_3)$$

The $\mathrel{-\!\!*}$ (rather than $\to$) is appropriate because the system only flows one way; for example, having passed l_1, one cannot return to it later, the system has changed. What process do the implications explicit above represent? They are the *interfacing* between the various sections of system; this is part of the system and are modelled by $\mathscr{C}_1$ and $\mathscr{C}_2$, respectively.

What $\mathscr{B}$ models the policy described by Γ? The compositional approach by which we described the model is entirely suitable, $\mathscr{B} := \mathscr{B}_1 \cup \ldots \cup \mathscr{B}_6 \cup \mathscr{C}_1 \cup \mathscr{C}_2$ suffices. This describes how the overall system is modelled, but it is instructive to look at bases in more detail to see how modelling of the components in this compositional approach is done.

We require $\mathscr{B}_1$ to support the formula $p \mathrel{-\!\!*} (h * (t \wedge p))$; that is, we require $\Vdash^{\varnothing_\times}_{\mathscr{B}_1} p \mathrel{-\!\!*} ((p \wedge t) * h)$ to hold. Recall the clause for $*$,

$$\Vdash^{p}_{\mathscr{B}_1} (p \wedge t) * h \quad \text{iff} \quad \forall \mathscr{X} \supseteq \mathscr{B}_1 \, \forall x \in \mathbb{A}, \text{ if } (p \fatsemi t) \, , h \Vdash^{U(\cdot)}_{\mathscr{X}} x, \text{ then } \Vdash^{U(p)}_{\mathscr{X}} x$$

This means that in any situation in which one can use the collection of resources $h \, , (t \fatsemi p)$, it suffices to use the resource p. So, in the simplest case, it suffices for the base $\mathscr{B}_1$ to contain the following rules: for any $U(\cdot) \in \dot{\mathbb{B}}(\mathbb{A})$ and $x \in \mathbb{A}$,

$$\frac{U(h \, , t) \rhd x}{U(p) \rhd x}$$

This is the coarsest possible approach. In practice, *check-in* is a whole system unto itself (see the discussion on *substitution* in Section 5) and there are many internal processes that run. For example, check-in may proceed as follows: the system extracts from the passport three different parts, the name p_{name}, the date of birth p_{dob}, and the passport-number p_{num}; uses the passport number to issue the ticket; and uses the name on the passport and the ticket together to issue the hold-baggage label. In this case, $\mathscr{B}_1$ may have the following rules:

$$\frac{}{p \rhd p_{\text{name}}} \qquad \frac{}{p \rhd p_{\text{dob}}} \qquad \frac{}{p \rhd p_{\text{num}}} \qquad \frac{p_{\text{num}} \rhd t \quad p_{\text{name}} \mathbin{,} t \rhd h \quad U(h \mathbin{,} t) \rhd x}{U(p) \rhd x} \qquad \ldots$$

where $\ldots$ denotes the part of the base that models the way in which the passport number (p_{num}) is used to issue the ticket (t) and the name and ticket together ($p_{\text{name}} \mathbin{,} t$) are used to issue the hold-baggage label (h). The same kind of flexibility in the modelling occurs at each location.

4.2 Multi-factor Authentication (MFA)

To illustrate how generic the notion of system used really is, we give an immediate translation to the setting of information security. The goal of MFA is to increase the overall security of systems by requiring multiple forms of verification before granting access; for example, login systems on a banking app. Such verification forms are called *authentication factors*, and typical examples include, *inter alia*, knowledge factors (e.g., passwords, pin codes, etc.), possession factors (e.g., verification code via text message or email), inherent factors (e.g., biometrics — fingerprints, voice/face recognition, etc.). We model a simple MFA example of user-login for an account at a fictional bank called *Bunched Money*, which abstracts banking apps quite generally.

To access their Bunched Money account, a user needs two out of the three possible authentication factors: a password p, a one-time passcode o, or a hardware fob f. Access is denoted by a security token s_{acc}. This policy can be expressed by a theory Γ that consists of exactly one formula:

$$\Gamma := ((p * o) \vee (p * f) \vee (o * f)) \mathrel{-\!\!*} s_{\text{acc}}$$

The $*$ (as opposed to $\wedge$) because it is crucial that the two authentication factors (e.g., p and o) are from *separate* sources. The separation is essential, as otherwise these two factors would not enhance the security compared to a single factor — for example, two passwords are not substantially stronger than one.

Let $\mathscr{B}$ be a model of the policy described by Γ — that is, $\mathscr{B}$ is defined by the property $\Vdash^{\varnothing_\times}_{\mathscr{B}} \Gamma$. Let R be an appropriate collection of verification factors — for example, $R := p \mathbin{,} o$. That access is granted under these conditions is expressed by $\Vdash^{R}_{\mathscr{B}} s_{\text{acc}}$, which indeed obtains.

5 Discussion: The Generality of the Approach

In Section 1, we discussed an abstract view of (distributed) systems based in the concepts of *location*, *resource*, and *process*: processes manipulate resources that reside at locations. A system is located within an *environment*, itself a system. In general, one can sketch a system with diagrams as in Figure 5; for example, the modelling of airport security in Section 4 begins with the sketch in Figure 4.

What we have offered in Section 3 is an interpretation of the B-eS of ILL and BI in terms of systems concepts. We now explain how it applies for modelling systems described by a diagram as in Figure 5.

Relative to a diagram such as Figure 5, we can explain quite generally how the inferentialist resource semantics applies to systems modelling. A *distributed* system D comprises distinct components (or subsystems) $C_1, \ldots, C_n$, each of which has its own policy which may interface with each other. Using the resource interpretation in Section 3, the policy at the C_i are individually described by formulae φ_i and the policy of the distributed system D is described by a bunch Γ over these descriptions. An inferential *model* of the policy of D is given by a base $\mathscr{B}$ supporting Γ.

Remark 5.1 Following the resource semantics of bunches, one can model each component's policy φ_i individually by a base $\mathscr{B}_i$, and construct a model $\mathscr{D}$ of the overall policy for D by taking the union of

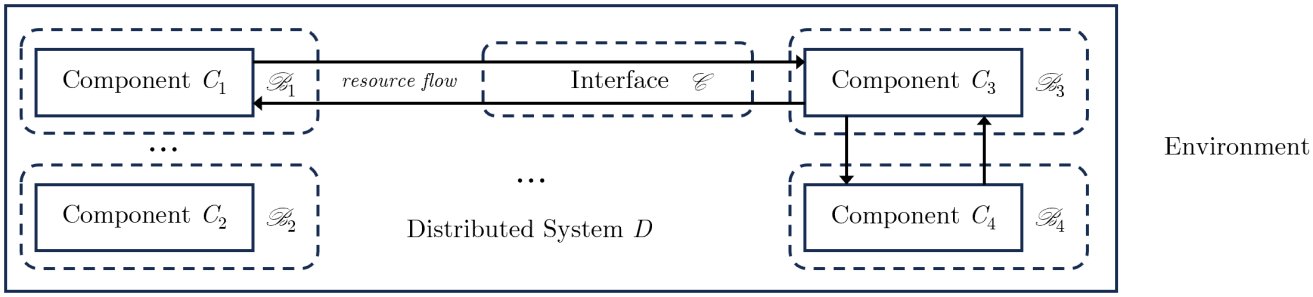

Fig. 5. General Distributed System Architecture

those bases together with some rules $\mathscr{C}$ governing their interfacing, $\mathscr{D} := \mathscr{B}_1 \cup \ldots \cup \mathscr{B}_n \cup \mathscr{C}$. In this sense, the resource semantics given by B-eS is *compositional*.

Remark 5.2 The inferential models of policies can be given at various degrees of refinement. In modelling distributed systems, this corresponds to the fact that each component C_i of D may, if useful, be thought of as a distributed system in itself; that is, one *substitutes* C_i for a more detailed set of systems $C_1^{(i)},\ldots,C_k^{(i)}$, which interface to yield the overall effect of C_i. The base modelling the policy of $C_1^{(i)},\ldots,C_k^{(i)}$ still models the policy of C_i. No change is needed.

This account implicitly assumes a formal model of the compositional structure of a distributed system as sketched in Figure 5. There are of course many ways of establishing such a model. For our current purposes, we consider, at an appropriate level of abstraction, a conceptualization of distributed systems in terms of locations, resources, and processes, as introduced in Section 1 [12,3,18], as follows:

- the distributed system D is based on a directed graph (or a similar topological structure) with certain sub-graphs denoting the components C_i;
- the vertices of the graph denote locations and the edges give the connectivity between the components;
- each component system has some collection of resources that carries some algebraic structure (e.g., as a bunch, or an ordered monoid, and so on) that is coherent with its sub-graph structure; and
- processes, perhaps represented using a resource-process algebra (see, for example, [12,3]) or a primitive notion of behaviour (see, for example, [18]), describe the manipulation of resources and hence the delivery of services by the system.

Relative to such architecture — in which the states of a system are described by processes executing relative to resources that reside at locations — logical formulae represent the policy of the system; that is, given a certain distribution of resources across the locations, what subsequent distributions are possible. The details of these readings are given in Section 3; for example, the formula $\varphi \vee \psi$ represents the policy that the system may behave as any state satisfying φ or ψ ambiguously — that is, it may move into a state satisfying policy χ if it is the case that were it in a state φ it could move into a state χ and were it in a state ψ it could move into a state χ.

Within Figure 5, we suggest two possible situations in describing the resource flow between components: first, resources may be subject to compliance with policy, as between C_1 and C_3, and so we also model the *interface* between components; second, resource flow may not be subject to compliance with policy, as between C_3 and C_4, but must sill be compliant with the policy of the components themselves. These situations can be seen to arise in the example depicted in Figure 4.

In the first case, a passenger may move from ground-side to air-side provided they are compliant with the airport's security policy: in Figure 4, l_3 can be seen as such an interface between (models of) ground-side and air-side, consisting of policies that determine whether the passenger after check-in (i.e., component l_1) could continue to passport control (i.e., component l_4). In the second case, the departure gate implements a check of boarding cards and passports, and perhaps also compliance with cabin-baggage policy, but the details of these are not specified as an interface component of the model.

In general, the permitted manipulations of resources are determined by the system's policies and these policies can be described as logical formulae that are interpreted according to the principles of the resource semantics that we have described in general in Section 3 and illustrated by examples in Section 4.

Further work, beyond our present scope, is to develop conceptual and modelling tools for this general set-up more formally, perhaps starting from a 'minimalistic' approach to systems modelling [18].

6 Conclusion

The concept of a *distributed system* is foundational in informatics, characterizing the architecture of both physical and abstract infrastructures that are an integral part of modern life. Logic serves as a vital tool for representing, understanding, and reasoning about such systems. One common approach is to give 'resource semantics' of logics; that is, interpretations of logical structures and relations in terms of system concepts. Notable examples include the number-of-uses interpretation for linear logics and the sharing/separation interpretation for bunched logics.

Despite the distinct nature of these two resource semantics, this paper presents a unified definition of 'resource semantics' that encompasses both. Furthermore, it offers a consistent interpretation of the inferentialist perspective across these logics, specifically through their *base-extension semantics* that uniformly recovers their established resource readings. This underscores the paper's thesis, as articulated in Section 3.3, asserting inferentialism as an intuitive and useful framework for logical systems modelling.

The thesis is illustrated through the modelling of airport security architecture and multi-factor authentication systems, which demonstrate the generic applicability of the approach. Future research, discussed in Section 5, includes formalizing the resource interpretation by establishing a conceptual model of distributed systems and verifying the faithfulness and adequacy of the resource interpretation within that model. From a logical point of view, there is room to explore the base-extension semantics of logic models that precisely capture desired properties of distributed systems; for example, potentially incorporating action and knowledge modalities to refine the understanding of resource movement and policy across distributed systems.

In summary, this paper provides a conceptually and technically well-grounded starting point for developing a general, systematic, logic-based inferentialist semantics for systems modelling.

References

[1] Abramsky, S., *Computational interpretations of linear logic*, Theoretical Computer Science **111**, pages 3–57 (1993).

[2] Allwein, G. and J. M. Dunn, *Kripke Models for Linear Logic*, The Journal of Symbolic Logic **58**, pages 514–545 (1993), ISSN 00224812.

[3] Anderson, G. and D. Pym, *A calculus and logic of bunched resources and processes*, Theoretical Computer Science **614**, pages 63–96 (2016).

[4] Bean, J. M. L., *Ribbon Proofs — A Proof System for the Logic of Bunched Implications*, Ph.D. thesis, Queen Mary University of London (2006).

[5] Bierman, G. M., *On Intuitionistic Linear Logic*, Ph.D. thesis, University of Cambridge (1994). Available as Computer Laboratory Technical Report 346.

[6] Brandom, R., *Articulating Reasons: An Introduction to Inferentialism*, Harvard University Press (2000).

[7] Brotherston, J., *Bunched Logics Displayed*, Studio Logica **100**, pages 1223–1254 (2012). https://doi.org/10.1007/s11225-012-9449-0

[8] Brünnler, K., *Deep Sequent Systems for Modal Logic*, Archive for Mathematical Logic **48**, pages 551–577 (2009). https://doi.org/10.1007/s00153-009-0137-3

[9] Buzoku, Y., *Presentation at 'Proof-theoretic Semantics and Truth'*. University of Bristol, December 2023. https://sites.google.com/view/pts-and-truth/home.

[10] Buzoku, Y., *A Proof-theoretic Semantics for Intuitionistic Linear Logic*, arXiv:2402.01982 (2024). Submitted.

[11] Caulfield, T. and D. Pym, *Modelling and simulating systems security policy*, in: *Proceedings of the 8th International Conference on Simulation Tools and Techniques*, SIMUTools '15, page 9–18, ICST (2015). https://doi.org/10.4108/eai.24-8-2015.2260765

[12] Collinson, M., B. Monahan and D. Pym, *A Discipline of Mathematical Systems Modelling*, College Publications (2012).

[13] Coulouris, G., J. Dollimore, T. Kindberg and G. Blair, *Distributed Systems: Concepts and Design*, Addison-Wesley Publishing Company, USA, 5th edition (2011), ISBN 0132143011.

[14] Coumans, D., M. Gehrke and L. van Rooijen, *Relational semantics for full linear logic*, Journal of Applied Logic **12**, pages 50–66 (2014), ISSN 1570-8683. https://doi.org/10.1016/j.jal.2013.07.005

[15] Dummett, M., *The Logical Basis of Metaphysics*, Harvard University Press (1991).

[16] Francez, N., *Proof-theoretic Semantics*, College Publications (2015).

[17] Galmiche, D., P. Kimmel and D. J. Pym, *A substructural epistemic resource logic: Theory and modelling applications*, J. Log. Comput. **29**, pages 1251–1287 (2019). https://api.semanticscholar.org/CorpusID:202577454

[18] Galmiche, D., T. Lang and D. Pym, *Minimalistic system modelling: Behaviours, interfaces, and local reasoning*, arXiv:2401.16109 (2024). Accessed 20 March 2024.

[19] Galmiche, D., D. Méry and D. Pym, *The Semantics of BI and Resource Tableaux*, Mathematical Structures in Computer Science **15**, page 1033–1088 (2005), ISSN 0960-1295. https://doi.org/10.1017/S0960129505004858

[20] Gheorghiu, A. and D. Pym, *Semantical analysis of the logic of bunched implications*, Studia Logica **111**, pages 525–571 (2023). https://doi.org/https://doi.org/10.1007/s11225-022-10028-z

[21] Gheorghiu, A. V., T. Gu and D. J. Pym, *Proof-theoretic semantics for intuitionistic multiplicative linear logic*, in: R. Ramanayake and J. Urban, editors, *Automated Reasoning with Analytic Tableaux and Related Methods — TABLEAUX*, pages 367–385, Springer (2023), ISBN 978-3-031-43513-3.

[22] Gheorghiu, A. V. and D. J. Pym, *From Proof-theoretic Validity to Base-extension Semantics for Intuitionistic Propositional Logic*, arXiv:2210.05344 (2024). Submitted.

[23] Girard, J.-Y., *Linear logic*, Theoretical computer science **50**, pages 1–101 (1987).

[24] Girard, J.-Y., *Linear logic: its syntax and semantics*, in: *Advances in Linear Logic*, London Mathematical Society Lecture Note Series, page 1–42, Cambridge University Press (1995).

[25] Goldfarb, W., *On Dummett's "Proof-theoretic Justifications of Logical Laws"*, in: T. Piecha and P. Schroeder-Heister, editors, *Advances in Proof-theoretic Semantics*, pages 195–210, Springer (2016).

[26] Gu, T., A. V. Gheorghiu and D. J. Pym, *Proof-theoretic-semantics for the Logic of Bunched Implications*, arXiv:2311.16719 (2024). Accessed December 2023.

[27] Harland, J. and D. Pym, *Resource-distribution via Boolean Constraints*, ACM Transactions on Computational Logic **4(1)**, pages 56–90 (2003). https://doi.org/https://doi.org/10.1145/601775.601778

[28] Hoare, C. A. R., *Communicating Sequential Processes*, Prentice-Hall International (1985), ISBN 0-13-153271-5.

[29] Ishtiaq, S. S. and P. W. O'Hearn, *BI as an Assertion Language for Mutable Data Structures*, in: C. Hankin and D. Schmidt, editors, *Symposium on Principles of Programming Languages — POPL*, pages 14–26, Association for Computing Machinery (ACM) (2001). https://doi.org/10.1145/360204.375719

[30] Jaakko Kuorikoski, S. R., *Making It Count: An Inferentialist Account of Computer Simulation*, https://osf.io/preprints/socarxiv/v9bmr (2022). Accessed January 2023. https://doi.org/10.31235/osf.io/v9bmr

[31] Kripke, S. A., *Semantical Analysis of Intuitionistic Logic I*, Studies in Logic and the Foundations of Mathematics **40**, pages 92–130 (1965).

[32] Lafont, Y., *Introduction to linear logic* (1993). Lecture notes from TEMPUS Summer School on Algebraic and Categorical Methods in Computer Science, Brno, Czech Republic.

[33] Milner, R., *Bigraphs as a model for mobile interaction*, in: *ICGT 2002, LNCS 2505*, pages 8–13, Springer (2002).

[34] Negri, S., *A normalizing system of natural deduction for intuitionistic linear logic*, Archive for Mathematical Logic **41**, pages 789–810 (2002).
https://doi.org/10.1007/s001530100136

[35] O'Hearn, P. W. and D. J. Pym, *The Logic of Bunched Implications*, Bulletin of Symbolic Logic **5**, pages 215–244 (1999).
https://doi.org/10.2307/421090

[36] Piecha, T., *Completeness in Proof-theoretic Semantics*, in: T. Piecha and P. Schroeder-Heister, editors, *Advances in Proof-theoretic Semantics*, pages 231–251, Springer (2016).

[37] Piecha, T., W. de Campos Sanz and P. Schroeder-Heister, *Failure of Completeness in Proof-theoretic Semantics*, Journal of Philosophical Logic **44**, pages 321–335 (2015).
https://doi.org/10.1007/s10992-014-9322-x

[38] Piecha, T. and P. Schroeder-Heister, *Incompleteness of Intuitionistic Propositional Logic with Respect to Proof-theoretic Semantics*, Studia Logica **107**, pages 233–246 (2019).
https://doi.org/10.1007/s11225-018-9823-7

[39] Prawitz, D., *Ideas and Results in Proof Theory*, in: J. Fenstad, editor, *Studies in Logic and the Foundations of Mathematics*, volume 63, pages 235–307, Elsevier (1971).
https://doi.org/10.1016/S0049-237X(08)70849-8

[40] Prawitz, D., *Towards a Foundation of a General Proof Theory*, in: *Studies in Logic and the Foundations of Mathematics*, volume 74, pages 225–250, Elsevier (1973).

[41] Prawitz, D., *Natural Deduction: A Proof-theoretical Study*, Courier Dover Publications (2006 [1965]).
https://doi.org/10.2307/2271676

[42] Pym, D., E. Ritter and E. Robinson, *Categorical Proof-theoretic Semantics*, Studia Logica (2024).
https://doi.org/https://doi.org/10.1007/s11225-024-10101-9

[43] Pym, D. J., *The Semantics and Proof Theory of the Logic of Bunched Implications*, volume 26 of *Applied Logic Series*, Springer (2002), ISBN 978-90-481-6072-3.
https://doi.org/10.1007/978-94-017-0091-7

[44] Pym, D. J., *Resource Semantics: Logic as a Modelling Technology*, ACM SIGLOG News **6**, pages 5–41 (2019).
https://doi.org/10.1145/3326938.3326940

[45] Pym, D. J., E. Ritter and E. Robinson, *Proof-theoretic Semantics in Sheaves (Extended Abstract)*, in: Å. Hirvonen and F. Velázquez-Quesada, editors, *Proceedings of the Eleventh Scandinavian Logic Symposium — SLSS 11*, Scandinvavian Logic Society (2022).

[46] Pym, D. J., J. M. Spring and P. O'Hearn, *Why Separation Logic Works*, Philosophy & Technology **32**, pages 483–516 (2019).
https://doi.org/10.1007/s13347-018-0312-8

[47] Read, S., *Relevant Logic*, Basil Blackwell (1988).
https://doi.org/10.2307/2219818

[48] Reed, J., *A Hybrid Logical Framework*, Ph.D. thesis, Carnegie Mellon University (2009). Available as CMU-CS-09-155.

[49] Reynolds, J. C., *Intuitionistic reasoning about shared mutable data structure*, in: *Millennial Perspectives in Computer Science*, volume 2, pages 303–321, Macmillan Education (2000). Proceedings of the 1999 Oxford-Microsoft Symposium in Honour of Sir Tony Hoare.

[50] Reynolds, J. C., *Separation Logic: A Logic for Shared Mutable Data Structures*, in: G. Plotkin, editor, *Logic in Computer Science — LICS*, pages 55–74, IEEE (2002).
https://doi.org/10.1109/LICS.2002.1029817

[51] Sandqvist, T., *An Inferentialist Interpretation of Classical Logic*, Ph.D. thesis, Uppsala University (2005).

[52] Sandqvist, T., *Classical Logic without Bivalence*, Analysis **69**, pages 211–218 (2009).

[53] Sandqvist, T., *Base-extension Semantics for Intuitionistic Sentential Logic*, Logic Journal of the IGPL **23**, pages 719–731 (2015).
https://doi.org/10.1093/jigpal/jzv021

[54] Sandqvist, T., *Hypothesis-discharging Rules in Atomic Bases*, in: H. Wansing, editor, *Dag Prawitz on Proofs and Meaning*, pages 313–328, Springer (2015).

[55] Schroeder-Heister, P., *Proof-Theoretic versus Model-Theoretic Consequence*, in: M. Pelis, editor, *The Logica Yearbook 2007*, Filosofia (2008).
https://doi.org/95b360ffe9ad174fd305539813800ea23fec33de

[56] Stafford, W. and V. Nascimento, *Following All the Rules: Intuitionistic Completeness for Generalized Proof-Theoretic Validity*, Analysis (forthcoming).
https://doi.org/10.1093/analys/anac100

[57] Szabo, M. E., editor, *The Collected Papers of Gerhard Gentzen*, North-Holland Publishing Company (1969).
https://doi.org/10.2307/2272429

[58] Tanenbaum, A. S. and M. V. Steen, *Distributed Systems: Principles and Paradigms*, Prentice Hall PTR, USA, 1st edition (2001), ISBN 0130888931.

[59] Tarski, A., *O pojęciu wynikania logicznego*, Przegląd Filozoficzny **39** (1936).

[60] Tarski, A., *On the concept of following logically*, History and Philosophy of Logic **23**, pages 155–196 (2002).
https://doi.org/10.1080/0144534021000036683

[61] Tripakis, S., B. Lickly, T. Henzinger and E. Lee, *A theory of synchronous relational interfaces*, ACM Transactions on Programming Languages and Systems **33**, pages 1–41 (2011).

[62] Wansing, H., *The idea of a proof-theoretic semantics and the meaning of the logical operations*, Studia Logica **64**, pages 3–20 (2000).