Logical Methods in Computer Science
Vol. 7 (1:7) 2011, pp. 1–35
www.lmcs-online.org

Submitted Jul. 30, 2009
Published Mar. 23, 2011

A CALCULUS FOR COSTED COMPUTATIONS

MATTHEW HENNESSY

Department of Computer Science, Trinity College Dublin, Ireland
e-mail address: matthew.hennessy@cs.tcd.ie

Abstract. We develop a version of the picalculus Picost where channels are interpreted as *resources* which have costs associated with them. Code runs under the financial responsibility of *owners*; they must pay to *use* resources, but may profit by *providing* them.

We provide a proof methodology for processes described in Picost based on bisimulations. The underlying behavioural theory is justified via a contextual characterisation. We also demonstrate its usefulness via examples.

1. Introduction

The purpose of this paper is to develop a behavioural theory of processes, in which computations depend on the ability to fund the resources involved. The theory will be based on the well-known concept of bisimulations, [Mil99], which automatically gives a powerful co-inductive proof methodology for establishing properties of processes; here these properties will include the cost of behaviour.

We take as a starting point the well-known picalculus, [SW01, Mil99], a language for describing mobile processes which has a well-developed behavioural theory. In the picalculus a process is described in terms of its ability to input and output on *communication channels*. Here we interpret these channels as *resources*, or services, as for example in [CGP08]. So input along a channel, written as $c?(x)\,.P$ in the picalculus, is now interpreted as *providing* the service c, while output, written $c!\langle v\rangle.P$, is interpreted as a request to *use* the service c. A process is now determined by the manner in which it *provides* services and *uses* them.

Viewed from this perspective, we extend the picalculus in two ways. Firstly we associate a *cost* with resources; specifically for each resource we assume that a certain amount of funds k_u is charged to *use* it, and an amount k_p is also required to *provide* it. Secondly we introduce *principals* or *owners* who provide the funds necessary for the functioning of resources. The novel construct in the language is $[P]_{\mathsf{o}}$, representing the (picalculus) process P running under the financial responsibility of o. For example in $[c!\langle v\rangle.Q]_{\mathsf{o}}$ the use of the resource c is only possible if o can fund the charges. Similarly with $[c?(x)\,.Q]_{\mathsf{o}}$, but

1998 ACM Subject Classification: F.3.1, F.3.2, F.3.3.
Key words and phrases: resources, cost, picalculus, bisimulations, amortisation.
The financial support of SFI is gratefully acknowledged.

DOI:10.2168/LMCS-7 (1:7) 2011

here there is also the potential for gain for owner o; in our formulation o profits from any difference between the cost in providing the resource and the charge made to use it.

Our language Picost is presented in Section 2, and is essentially a variation on Dpi, a typed distributed version of the picalculus, [Hen07]. The reduction semantics is given in terms of judgements of the form

$$(\Gamma \rhd M) \longrightarrow (\Delta \rhd N)$$

where Γ, Δ are *cost environments*. These have a static component, giving the costs associated with resources, and a dynamic part, which gives the funds available to owners and also records expenditure. The usefulness of the language is demonstrated by a series of simple examples.

But the main achievement of the paper is a behavioural theory, expressed as judgements

$$(\Gamma \rhd M) \sqsubseteq_{\text{awgt}} (\Delta \rhd N) \tag{1.1}$$

indicating that, informally speaking,

(i) the process M running relative to the cost environment Γ is bisimilar, in the standard sense [Mil89], with process N running relative to Δ
(ii) the costs associated with $(\Delta \rhd N)$ are no more, and possibly less, than those associated with $(\Gamma \rhd M)$.

Influenced by [KAK05] we first develop a general framework of *weighted labelled transition systems* or *wLTSs*, in which actions, including internal actions, may have multiple weights associated with them. We then define a notion of *amortised weighted bisimulations* between their states, giving rise to a preorder $s \sqsubseteq_{\text{awgt}} t$, meaning that s, t are bisimilar but in some sense the behaviours of t are *lighter* than those of s. From this we obtain, in the standard manner, a co-inductive proof methodology for proving that two systems are related; it is sufficient to find, or construct, a particular amortised weighted bisimulation containing the pair (s, t).

This proof methodology is applied to Picost by first interpreting the language as an LTS, in agreement with the reduction semantics, and then interpreting this LTS as a wLTS, giving rise to (parametrised versions of) the judgements (1.1) above. But as we will see these judgements can be interpreted in two ways. If the recorded expenditure represents *costs* then $(\Delta \rhd N)$ can be considered an improvement on $(\Gamma \rhd M)$. On the other hand if it represents *profits* then we have the reverse; $(\Gamma \rhd M)$ is an improvement on $(\Delta \rhd N)$ as it has the potential to be heavier.

The details of this theory are given in Section 3, and the resulting proof methodology is illustrated by examples. However in Section 4 we re-examine this proof methodology, in the light of reasonable properties we would expect of it; and these are found wanting. It turns out that the manner in which we generate the wLTS for Picost from its operational semantics is too coarse. We show how to generate a somewhat more abstract wLTS, and prove that the resulting proof methodology is satisfactory, in a precise technical sense, by adapting the notion of reduction barbed congruence, [HT92, SW01, HR04, Hen07].

2. The language Picost

2.1. **Syntax:** We assume a set of *channel* or *resource* names Chan, ranged over by $a, b, c, \ldots$, $r, \ldots$ whose use requires some cost, a distinct set of (value) variables Var, ranged over by

$M,\ N$	$::=$	
	$[T]_{\mathsf{o}}$	Owned code
	$M \mid N$	Composition
	$(\mathsf{new}\, r{:}\mathsf{R})M$	Scoped resource
	$\mathsf{0}$	Identity

$T,\ U$	$::=$	
	$u?(x)\,.T$	Provide resource u
	$u!\langle v\rangle.T$	Use resource u
	$\mathsf{if}\ v = v\ \mathsf{then}\ T\ \mathsf{else}\ U$	Matching
	$(\mathsf{new}\, r{:}\mathsf{R})T$	Resource creation
	$T \mid U$	Concurrency
	$\mathsf{rec}\ X.\ T$	Recursion
	X	Recursion variable
	stop	Termination

Figure 1: Syntax of Picost

$x, y, \ldots$, and a further distinct set of recursion variables, $X, Y, \ldots$; u ranges over *identifiers*, which may be either resource names or (value) variables. We also assume a set of *principals* or *owners* Own containing at least two elements, ranged over by $\mathsf{o}, \mathsf{u}, \mathsf{p}$, who are implicitly registered for these resources and who finance their provision and use. The syntax of Picost is then given in Figure 1, and is essentially a very minor variation on Dpi, [Hen07]. The main syntactic category represents code running under responsibility, with $[P]_{\mathsf{o}}$ being the novel construct. As explained in the Introduction this represents the code P running under the responsibility of the owner o; intuitively o is financially responsible for the computation P. Thus in general a system is simply a collection of computation threads each running under the responsibility of an explicit owner, which may share private resources. The syntax for these threads is a version of the well-known picalculus, [SW01].

The type R of a resource describes the costs associated with that resource. There is a cost associated with *using* a resource, and a cost associated with *providing* it; therefore types take the form $\langle k_u, k_p\rangle$ where $k_u,\ k_p$ are elements from some *cost domain* K. Here we take K simply to be $\mathbb{N}$ ordered in the standard manner, but most of our results apply equally well to variations.

We employ the standard abbreviations associated with the picalculus, and associated terminology. In particular we assume *Barendregt's convention*, which implies that bound variables used in terms or definitions are distinct, and different from any free variables in use in the current context. In Figure 1 meta-variable v range over *value expressions*, whose specification we omit; but they include at least resource names $a \in \mathsf{Chan}$, variables x from Var, and elements of K. As usual we omit every occurrence of a trailing stop and abbreviate $u?()\,.T$, $u!\langle\rangle.T$ to $u?.T$, $u!.T$ respectively. We are only interested in *closed code terms*, those which contain no free occurrences of variables, which are ranged over by $P,\ Q, \ldots$; we use $\mathsf{fn}(P)$ to denote the set of names from Chan which occur freely in P. In the sequel we assume all terms are closed.

2.2. **Cost environments:** Since computations have financial implications, the execution of processes is now relative to a *cost environment* Γ. This records the financial resources available to principals, and the cost of providing and using resources; in order to be able to compare the cost of computations we also assume a component which records the expenditure as a computation proceeds. Thus judgements of the reduction semantics take the form

$$\Gamma \rhd M \longrightarrow \Delta \rhd N$$

where Γ, Δ are cost environments.

There are many possibilities for *cost environments*; see [HG08] for an example which directly associates funds with resources. In the present paper we define them in such a way that the owners retain total control over their own funds.

Definition 2.1 (Cost environments)**.** A *cost environment* Γ consists of a 4-tuple $\langle \Gamma^o, \Gamma^u, \Gamma^p, \Gamma^{\mathsf{rec}} \rangle$ where

- $\Gamma^u : \mathsf{Chan} \rightharpoonup K$
  $\Gamma^u(a)$ records the cost of *using* resource a; this is a *static* component, and will not vary during computations
- $\Gamma^p : \mathsf{Chan} \rightharpoonup K$
  $\Gamma^p(a)$ records the cost of *providing* resource a; again this is a static component
- $\Gamma^o : \mathsf{Own} \rightharpoonup K$
  $\Gamma^o(\mathsf{o})$ records the funds available to owner o; this will vary as computations proceed, as owners will need to fund their interactions with resources
- $\Gamma^{\mathsf{rec}} \in K$
  Γ^{rec} keeps an account of the expenditure occurred during a computation; of course this also will vary as a computation proceeds.

We assume that both functions Γ^u, Γ^p have the same finite domain, but not necessarily that $\Gamma^u(a) \geq \Gamma^p(a)$ whenever these are defined. ■

We now define some operations on cost environments which will enable us to reflect their impact on the semantics of our language. The most important is a partial function, $\Gamma \xrightarrow{(\mathsf{u},a,\mathsf{p})} \Delta$, which informally means that in Γ owner u has sufficient funds to cover the cost of using resource a and owner p has sufficient funds to provide it. Then Δ records the result of the expenditure of both o and p of those funds. There is also considerable scope as to what happens to these funds, and how their expenditure is recorded. Here we take the view that the provider p gains the cost which the user expends, to offset p's cost in providing the resource.

Definition 2.2 (Resource charging)**.** Let $\xrightarrow{(\mathsf{u},a,\mathsf{p})}$ be the partial function over cost environments defined as follows: $\Gamma \xrightarrow{(\mathsf{u},a,\mathsf{p})} \Delta$ if

(i) $\Gamma^o(\mathsf{u}) \geq \Gamma^u(a)$ and $\Gamma^o(\mathsf{p}) \geq \Gamma^p(a)$
(ii) Δ is the cost environment obtained from Γ by
  (a) decreasing $\Gamma^o(\mathsf{u})$ by the amount $\Gamma^u(a)$
  (b) increasing $\Gamma^o(\mathsf{p})$ by the amount $\Gamma^u(a) - \Gamma^p(a)$, which may of course be negative
(iii) Finally there is considerable flexibility in how this resource expenditure is recorded in Δ^{rec}. We call resource charging for *a standard* when this is set to $\Gamma^{\mathsf{rec}} + \Gamma^u(a) - \Gamma^p(a)$; that is we add to the record the gain obtained in using resource a. But in general

we allow functions $\mathsf{rec}_a(-,-)$, for each resource a, in which case we define Δ^{rec} to be $\Gamma^{\mathsf{rec}} + \mathsf{rec}_a(\Gamma^u(a), \Gamma^p(a))$. ■

In general we allow the owners u and p in this definition to coincide. So, for example if $\Gamma \xrightarrow{(\mathsf{o},a,\mathsf{o})} \Delta$, then the effect of performing (a) above, followed by (b), is that $\Delta^o(\mathsf{o})$ is set to $\Gamma^o(\mathsf{o}) - \Gamma^p(a)$.

The use of two independent charges for each resource, Γ^u and Γ^p, may seem overly complex. A simpler model can be obtained by having only one combined charge; effectively we could assume $\Gamma^p(a)$ to be 0 for every a, and so resource charging simply transfers the appropriate amount of funds from the user to the provider; this could be achieved by restricting attention to *simple types*, resource types R of the form $\langle k_u, 0\rangle$. Indeed this simplification will be quite useful in order to achieve some theoretical properties of our proof methodology; see Definition 4.18 and Section 4.2. Nevertheless the use of the two independent charges $\Gamma^p(-)$ and $\Gamma^u(-)$ allows scope for more interesting examples. In particular it provides considerable scope for variation in the manner in which resource expenditure is recorded in the component Γ^{rec}; see Example 2.8 for an instance.
We also need to extend cost environments with new resources.

Definition 2.3 (Resource registration)**.** The cost environment $\Gamma, a{:}\mathsf{R}$, is only defined if a is *fresh* to Γ, that is, if a is neither in $\mathrm{dom}(\Gamma^u)$ nor in $\mathrm{dom}(\Gamma^p)$. In this case it gives the new cost environment Δ obtained by adding the new resource, with the capabilities determined by R. Formally the dynamic components of Δ, namely Δ^o and Δ^{rec}, are inherited directly from Γ, while the static components have the obvious definition; for example if R is the type $\langle k_u, k_p\rangle$ then Δ^u is given by

$$\Delta^u(x) = \begin{cases} k_u & \text{if } x = a \\ \Gamma^u(x) & \text{otherwise} \end{cases}$$

We also assume that the resource charging for a in $(\Gamma, a{:}\mathsf{R})$ is always standard. ■

Note that every cost environment may be written in the form

$$\Gamma_{dyn}, a_1{:}\mathsf{R}_1, \ldots a_n{:}\mathsf{R}_n$$

where Γ_{dyn} is a *basic* environment; that is the static components Γ^u_{dyn} and Γ^p_{dyn} are both empty, and so it only contains non-trivial dynamic components.

2.3. **Reduction semantics:** The pair $(\Gamma \rhd M)$ is called a *configuration* provided that $\mathsf{fn}(M) \subseteq \mathrm{dom}(\Gamma^u) = \mathrm{dom}(\Gamma^p)$, that is every free resource name in M is known to the *cost environment* Γ. The reduction semantics for Picost is then defined as the least relation over configurations which satisfies the rules in Figure 2. The majority of the rules come directly from the reduction semantics of Dpi, [Hen07], and are housekeeping in nature. The only rule of interest is (R-COMM), representing the communication along the channel a, or in Picost the *use* of the resource a by owner u which is *provided* by owner p. However this reduction is only possible whenever the premise $\Gamma \xrightarrow{(\mathsf{u},a,\mathsf{p})} \Delta$ is satisfied. As we have seen, this means that in Γ owner u has sufficient funds to cover the cost of using resource a and owner p has sufficient funds to provide it; and further Δ records the result of the expenditure of both u and p of those funds.

The remainder of the rules are borrowed directly from the standard reduction semantics of Dpi; note that (R-STRUCT) requires a structural equivalence between terms; this again is

$$\frac{\Gamma \xrightarrow{(\mathsf{u},a,\mathsf{p})} \Delta}{\Gamma \rhd [a!\langle v\rangle.Q]_{\mathsf{u}} \mid [a?(x)\,.P]_{\mathsf{p}} \longrightarrow \Delta \rhd [Q \mid P\{\!\{{}^{v}\!/\!_{x}\}\!\}]_{\mathsf{p}}} \quad \text{(R-COMM)}$$

$$\Gamma \rhd [M \mid N]_{\mathsf{o}} \longrightarrow \Gamma \rhd [M]_{\mathsf{o}} \mid [N]_{\mathsf{o}} \quad \text{(R-SPLIT)}$$

$$\Gamma \rhd [(\mathsf{new}\, r{:}\mathsf{R})P]_{\mathsf{o}} \longrightarrow \Gamma \rhd (\mathsf{new}\, r{:}\mathsf{R})[P]_{\mathsf{o}} \quad \text{(R-EXPORT)}$$

$$\Gamma \rhd [\mathsf{rec}\ x.\ T]_{\mathsf{o}} \longrightarrow \Gamma \rhd [T\{\!\{{}^{\mathsf{rec}\ x.\ T}\!/\!_{x}\}\!\}]_{\mathsf{o}} \quad \text{(R-UNWIND)}$$

$$\Gamma \rhd [\mathsf{if}\ a = a\ \mathsf{then}\ P\ \mathsf{else}\ Q]_{\mathsf{o}} \longrightarrow \Gamma \rhd [P]_{\mathsf{o}} \quad \text{(R-MATCH)}$$

$$\Gamma \rhd [\mathsf{if}\ a = b\ \mathsf{then}\ P\ \mathsf{else}\ Q]_{\mathsf{o}} \longrightarrow \Gamma \rhd [Q]_{\mathsf{o}} \quad a \neq b \quad \text{(R-MISMATCH)}$$

$$\frac{M \equiv M',\ \Gamma \rhd M \longrightarrow \Delta \rhd N,\ N \equiv N'}{\Gamma \rhd M' \longrightarrow \Delta \rhd N'} \quad \text{(R-STRUCT)}$$

$$\frac{\Gamma \rhd M \longrightarrow \Delta \rhd M'}{\Gamma \rhd M \mid N \longrightarrow \Delta \rhd M' \mid N} \quad \text{(R-CNTX)}$$

$$\frac{\Gamma, b{:}\mathsf{R} \rhd M \longrightarrow \Delta, b{:}\mathsf{R} \rhd N}{\Gamma \rhd (\mathsf{new}\, b{:}\mathsf{R})M \longrightarrow \Delta \rhd (\mathsf{new}\, b{:}\mathsf{R})N} \quad \text{(R-NEW)}$$

Figure 2: Reduction semantics

$$\begin{array}{lrcl}
\text{(S-EXTR)} & (\mathsf{new}\, r{:}\mathsf{R})(M \mid N) & \equiv & M \mid (\mathsf{new}\, r{:}\mathsf{R})N,\ \text{if } r \notin \mathsf{fn}(M) \\
\text{(S-COM)} & M \mid N & \equiv & N \mid M \\
\text{(S-ASSOC)} & (M \mid N) \mid O & \equiv & M \mid (N \mid O) \\
\text{(S-ZERO)} & M \mid \mathsf{0} & \equiv & M \\
 & [\mathsf{stop}]_{\mathsf{o}} & \equiv & \mathsf{0} \\
\text{(S-FLIP)} & (\mathsf{new}\, r{:}\mathsf{R})(\mathsf{new}\, r'{:}\mathsf{R}')M & \equiv & (\mathsf{new}\, r'{:}\mathsf{R}')(\mathsf{new}\, r{:}\mathsf{R})M
\end{array}$$

Figure 3: Structural equivalence of Picost

the standard one from Dpi, the definition of which is given in Figure 3. Also the final rule (R-NEW) uses the registration operation on cost environments, given in Definition 2.3.

Proposition 2.4. *If* $(\Gamma_1 \rhd M_1)$ *is a configuration and* $(\Gamma_1 \rhd M_1) \longrightarrow (\Gamma_2 \rhd M_2)$ *then* $(\Gamma_2 \rhd M_2)$ *is also a configuration.*

Proof. Straightforward, by induction on the proof that $(\Gamma_1 \rhd M_1) \longrightarrow (\Gamma_2 \rhd M_2)$. When handling the rule (R-STRUCT) it uses the obvious fact that $M \equiv N$ implies that M and N have the same set of free names; this in turn means that $M \equiv N$ implies $\Gamma \rhd M$ is a configuration if and only if $\Gamma \rhd N$ is. □

The reductions of a configuration affect its cost environment, and as a sanity check we can describe precisely the kinds of changes which are possible:

$$
\begin{aligned}
\mathsf{Sys} &\Leftarrow (\ [\mathsf{Reader}]_{\mathsf{pub}} \ \mid\ [\mathsf{Library} \mid \mathsf{Store}]_{\mathsf{lib}}\ ) \\
&\text{where} \\
\mathsf{Reader} &\Leftarrow \mathsf{rec}\ R.\ \mathsf{goLib}?(\mathsf{name})\,.(\mathsf{new}r)\ \mathsf{reqR}!\langle r, \mathsf{name}\rangle. \\
&\qquad\qquad r?(b)\,.\mathsf{goHome}!\langle b\rangle.R \\
\mathsf{Library} &\Leftarrow \mathsf{rec}\ L.\ \mathsf{reqR}?(y,z)\,.\ \ y!\langle \mathsf{book}(z)\rangle.L \\
&\qquad\qquad \oplus\ (\mathsf{new}r)\ \mathsf{reqS}!\langle r, z\rangle.r?(b)\,.y!\langle b\rangle.L \\
\mathsf{Store} &\Leftarrow \mathsf{rec}\ S.\ \mathsf{reqS}?(y,z)\,.y!\langle \mathsf{book}(z)\rangle.S
\end{aligned}
$$

Figure 4: Running a library

Proposition 2.5. *Suppose* $(\Gamma_1 \rhd M_1) \longrightarrow (\Gamma_2 \rhd M_2)$. *Then*

(i) $\Gamma_1 = \Gamma_2$, *and* $(\Delta \rhd M_1) \longrightarrow (\Delta \rhd M_2)$ *whenever* $(\Delta \rhd M_1)$ *is a configuration*

(ii) *or* $\Gamma_1 \xrightarrow{(\mathsf{u},a,\mathsf{p})} \Gamma_2$, *for some resource* a *and owners* u, p, *and whenever* $(\Delta \rhd M_1)$ *is a configuration* $\Delta \xrightarrow{(\mathsf{u},a,\mathsf{p})} \Delta'$ *implies* $(\Delta \rhd M_1) \longrightarrow (\Delta' \rhd M_2)$

(iii) *or* $\Gamma_1, a{:}\mathsf{R} \xrightarrow{(\mathsf{u},a,\mathsf{p})} \Gamma_2, a{:}\mathsf{R}$, *for some (fresh) resource* a, *resource type* R *and owners* u, p, *and whenever* $(\Delta \rhd M_1)$ *is a configuration* $\Delta, a{:}\mathsf{R} \xrightarrow{(\mathsf{u},a,\mathsf{p})} \Delta', a{:}\mathsf{R}$ *implies* $(\Delta \rhd M_1) \longrightarrow (\Delta' \rhd M_2)$

Proof. Again this is a simple proof by rule induction on the premise $(\Gamma_1 \rhd M_1) \longrightarrow (\Gamma_2 \rhd M_2)$. Intuitively possibility (i) corresponds to a move where no communication occurs, (ii) is when the move is a communication along a channel a known to Γ_1, and (iii) when the communication is along a private internal channel. □

2.4. **Examples:** Formally Picost has only unary communication, but in these examples we will informally allow the communication of tuples along channels. In addition we will use the standard abbreviations associated with the picalculus. We also omit types for channels when they are not relevant; in such cases we assume that they cost nothing to provide, and that there is no charge for using them. It will be convenient to have an *internal choice* operator, with $P \oplus Q$ representing an internal choice between P and Q. This can be taken to be short-hand notation for $(\mathsf{new}c)(c!\langle\rangle \mid c?()\,.P \mid c?()\,.Q)$, where c is a fresh channel.

Example 2.6 (Running a library)**.** Consider the system Sys from Figure 4, which consists of three recursive components, a library user Reader, running under the responsibility of the principal pub, standing for public, a library interface Library and an auxiliary book depository Store, both running under some other principal lib.

The programming of these components involves the systematic generation of *reply channels*. Thus for example the Reader gets the name of a book with which to go to the library, generates a new reply channel r and submits this together with the name of the book via reqR; it awaits the book and then returns home. The Store is also very simple; it recursively awaits a request on reqS, consisting of a reply channel and a name and returns the appropriate book on the channel. Finally the Library service requests at reqR consisting of

a reply channel and a name. The book may be immediately available, in which case it is returned, or it may be necessary to send a request to the Store.

Let us now consider the behaviour of these systems relative to two cost environments Γ_{local}, $\Gamma_{\mathsf{central}}$ representing two different strategies for providing library services. To focus on the relative cost of providing these services let us assume that their use is free, that is $\Gamma^u_*(a) = 0$ for every resource a, where $*$ ranges over local, central, and that the amount of funds available is not an issue, that is $\Gamma^o_*(\mathsf{pub}) = \Gamma^o_*(\mathsf{lib}) = \infty$. The cost of providing the services, Γ^p_* is given in the table below, reflecting on the one hand the relative convenience to the Reader of the local services, and on the other the relative convenience to the authorities in providing central services.

	local	central
goLib	1	5
goHome	1	5
reqR	3	1
reqS	5	1

Finally let us take the counters Γ^{rec}_* to be initially set to 0. Note that Γ_{local} can be written as

$$\Gamma_{dyn},\ \mathsf{goLib}{:}\mathsf{R}^g_l,\ \mathsf{goHome}{:}\mathsf{R}^h_l,\ \mathsf{reqR}{:}\mathsf{R}^r_l,\ \mathsf{reqS}{:}\mathsf{R}^s_l$$

where $\mathsf{R}^g_l, \mathsf{R}^h_l, \mathsf{R}^r_l, \mathsf{R}^s_l$ are the types $\langle 0,1\rangle, \langle 0,1\rangle, \langle 0,3\rangle, \langle 0,5\rangle$ respectively, and Γ_{dyn} is a basic environment; $\Gamma_{\mathsf{central}}$ has a similar representation, with a slightly different sequence of types.

To exercise the system we use

$$\mathsf{Book} \Leftarrow [\mathsf{goLib}!\langle str\rangle.\mathsf{goHome}?(x)\,.\,\mathsf{stop}]_{\mathsf{pub}}$$

to prod the Reader into action, where *str* is the name of some book. Consider the configuration

$$\mathcal{C}_1 = \Gamma_{\mathsf{local}} \rhd (\mathsf{Book} \mid \mathsf{Sys}),$$

and let us ignore the computation steps involved in generating reply channels, and general housekeeping such as the unwinding of recursive definitions, which in any event cost nothing. Because of the internal non-determinism in the library service there are essentially two computations from $\mathcal{C}_1$. If the Store is not used then after three computation steps which require funds it is in the state $\Delta_{\mathsf{local}} \rhd \mathsf{Sys}$, where $\Delta^{\mathsf{rec}}_{\mathsf{local}} = 5$. This represents the overall cost of this transaction, 2 of which is paid by pub and 3 by lib.

On the other hand if the Store is used, then there are four computation steps which require funding, after which the state $\Theta_{\mathsf{local}} \rhd \mathsf{Sys}$ is reached, where $\Theta^{\mathsf{rec}}_{\mathsf{local}} = 10$. However using the central cost environment $\Gamma_{\mathsf{central}}$ the two possibilities are $\Delta^{\mathsf{rec}}_{\mathsf{central}} = 11$ and $\Theta^{\mathsf{rec}}_{\mathsf{central}} = 12$ respectively. In each eventuality the local implementation is more efficient, in the sense that the costs are systematically lower. ■

The charging regime for resources is such that their use effectively means a transfer of funds to the *provider* from the *user*, provided the cost of providing the resource is less than the charge for its use. This enables us to implement a systematic way of transferring funds between owners.

$$
\begin{array}{rl}
& \mathsf{Sys} \Leftarrow [P]_{\mathsf{p}} \mid [N]_{\mathsf{n}} \mid [A]_{\mathsf{a}} \mid [R]_{\mathsf{r}} \\
\text{where} & \\
& P \Leftarrow \mathsf{rec}\ P.\ (\mathsf{new}r_1)\mathsf{news!}\langle r_1\rangle.(\mathsf{new}r_2)\mathsf{adv!}\langle r_2\rangle. \\
& \qquad\qquad r_1?(n)\,.r_2?(d)\,.\mathsf{publish}?(z)\,.z!\langle n,d\rangle.P \\
& N \Leftarrow \mathsf{rec}\ N.\ \mathsf{news}?(r)\,(\mathsf{new}n)r!\langle n\rangle.N \\
& A \Leftarrow \mathsf{rec}\ A.\ \mathsf{adv}?(r)\,.(\mathsf{new}d)r!\langle d\rangle.A \\
& R \Leftarrow \mathsf{rec}\ R.\ (\mathsf{new}r)\mathsf{publish!}\langle r\rangle.r?(n,d)\,.R
\end{array}
$$

Figure 5: Publishing

Example 2.7 (Fund transfer)**.** Consider the systems defined as follows:

$$
\begin{array}{rl}
& \mathsf{Sys} \Leftarrow [D]_{\mathsf{dad}} \mid [K]_{\mathsf{kate}} \\
\text{where} & \\
& D \Leftarrow \mathsf{req}?(x)\,.(\mathsf{new}\ s{:}\mathsf{R}_s)x!\langle s\rangle.s!.S \\
& K \Leftarrow (\mathsf{new}r)\mathsf{req!}\langle r\rangle.r?(y)\,.y?.H
\end{array}
$$

The size of the transfer from dad to kate depends on the type R_s at which the new channel s is declared. Suppose this type is $\langle 0, k\rangle$, and let Γ be a cost environment in which $\Gamma^o(\mathsf{dad})$ is at least k. Then there is a computation

$$(\Gamma \rhd \mathsf{Sys}) \longrightarrow^* (\Delta \rhd [S]_{\mathsf{dad}} \mid [H]_{\mathsf{kate}})$$

in which $\Delta^o(\mathsf{dad}) = \Gamma^o(\mathsf{dad}) - k$ and $\Delta^o(\mathsf{kate}) = \Gamma^o(\mathsf{kate}) + k$. ■

Example 2.8 (Publishing)**.** Consider the system Sys in Figure 5, which has four components:

(a) publisher: *uses* a news service via the resource news, *uses* an advertising agency via the resource adv and *provides* the resource publish
(b) news service: *provides* a service via news
(c) ad agency: *provides* a service via adv
(d) reader: *uses* the resource publish

The viability of publishing depends of course on the cost associated with these resources. As an example consider an environment Γ_{327}, of the form $\Gamma_{dyn}, \mathsf{news}{:}\mathsf{R}_n, \mathsf{adv}{:}\mathsf{R}_a, \mathsf{publish}{:}\mathsf{R}_p$, where these types are $\langle 3,1\rangle, \langle 2,0\rangle, \langle 7,1\rangle$ respectively, and let us assume $\Gamma^{\mathsf{rec}}_{327}$ is initialised to 0. Furthermore, since we are concentrating on the publisher, let us assume that the resource charging is defined so that only the effect on the owner p is recorded. Refering to Definition 2.2 this means that resource charging is standard for publish but we need to set $\mathsf{rec}_a(k_u, k_p)$ to be $-k_u$, if a is either news or adv.

Now consider a computation from the configuration $\Gamma_{317} \rhd \mathsf{Sys}$. Provided the owners have sufficient funds, specifically $\Gamma^o(\mathsf{p}), \Gamma^o(\mathsf{n})$ and $\Gamma^o(\mathsf{r})$ must be at least $5, 1, 7$ respectively, then we have a computation

$$(\Gamma_{317} \rhd \mathsf{Sys}) \longrightarrow^* (\Delta_1 \rhd \mathsf{Sys})$$

where $\Delta_1^{\mathsf{rec}} = 1$; the record part of the initial environment was set to 0, during the computation it was set to -3 after the publisher uses the news resource, then to -5 after using

adv; finally, when the reader uses the publish resource, this is increased by $(7-1)$ to give 1. Because we have defined expenditure recording to reflect the point of view of the publisher, this represents the fact that the publisher has made a profit of 1 as a result of this sequence of transactions. Note also that at this point $\Delta_1^o(\mathsf{p})$ is $\Gamma_{327}^o(\mathsf{p})+1$.

We can also see what happens when the costs of using resources is changed. Let Γ_{216} be the environment in which the cost of all three resources are decreased by 1. Then we have the computation

$$(\Gamma_{216} \rhd \mathsf{Sys}) \longrightarrow^* (\Delta_2 \rhd \mathsf{Sys})$$

where now $\Delta_2^{\mathsf{rec}} = 2$; this represents an increase in profits for the publisher. ■

Example 2.9 (Kickbacks)**.** Suppose in Figure 5 we change the situation so that the publisher obtains a kickback from the ad agency when an ad is downloaded. The modified code is given by

$$\begin{aligned} P_K \Leftarrow \mathsf{rec}\ P.\ (\mathsf{new} r_1)\mathsf{news!}\langle r_1\rangle.(\mathsf{new} r_2)(\mathsf{new}\ k{:}\mathsf{K})\mathsf{adv!}\langle k, r_2\rangle.\\ r_1?(n)\ .r_2?(d)\ .\mathsf{publish}?(z)\ .k?.z!\langle n,d\rangle.P\\ A_K \Leftarrow \mathsf{rec}\ A.\ \mathsf{adv}?(k,r)\ .(\mathsf{new} d)r!\langle d\rangle.(A \mid k!) \end{aligned}$$

and let Sys_K denote the revised system. The size of the kickback depends on the parameters in the type K. In Sys the ad agency receives the benefit 2 for supplying the ad; if we set K to be $\langle 1, 0\rangle$ then in Sys_K this benefit is split equally with the publisher. Under the same assumptions as in Example 2.8 we have the computations

$$(\Gamma_{327} \rhd \mathsf{Sys}_K) \longrightarrow^* (\Phi_1 \rhd \mathsf{Sys}_K) \qquad \text{and} \qquad (\Gamma_{216} \rhd \mathsf{Sys}_K) \longrightarrow^* (\Phi_2 \rhd \mathsf{Sys}_K)$$

where now $\Phi_1^{\mathsf{rec}}, \Phi_2^{\mathsf{rec}}$ are $2, 3$ respectively, indicating more profit in each case for the publisher. ■

3. Compositional reasoning

The aim of this section is to develop a proof methodology for Picost. The idea is to define a *behavioural preorder*

$$(\Gamma \rhd M) \sqsubseteq (\Delta \rhd N), \tag{3.1}$$

meaning that in some sense $(\Gamma \rhd M)$ and $(\Delta \rhd N)$ offer the same behaviour, but the latter is at least as efficient as the former, and possibly more. We follow the standard approach of defining the preorder (3.1) as the largest relation between Picost configurations satisfying a transfer property, associated with the ability of processes to interact with their peers. We thereby automatically get a co-inductive proof methodology for establishing relationships between configurations.

In fact, referring to (3.1), it is better to move away from terminology such as *efficiency* as the interpretation depends very much on the nature of the units being recorded. In Example 2.6 these are *costs* and in such a scenario it is reasonable to interpret (3.1) as saying $(\Delta \rhd N)$ is an improvement on $(\Gamma \rhd M)$ as it potentially involves less cost. On the other hand in Example 2.8 the units are *profit* (for the publisher), and here $(\Gamma \rhd M)$ would be considered to be an improvement on $(\Delta \rhd N)$, as there is potential for more profit (for the publisher).

We therefore move to the more neutral terminology of *weights*. However we can not simply base the formulation of (3.1) on the relative weight associated with each individual action, as the following example shows.

Example 3.1 (Amortising costs)**.** Consider the simple system

$$\mathsf{UD} \Leftarrow [\mathsf{rec}\ x.\ \mathsf{up!.down!}.x]_{\mathsf{o}}$$

and let Γ_{25} be an environment in which the unique owner o has unlimited funds, the use of up costs 2 and the use of down costs 5. If we compare $(\Gamma_{25} \rhd \mathsf{UD})$ with $(\Gamma_{42} \rhd \mathsf{UD})$, where Γ_{42} is defined analogously, then intuitively the latter is more efficient than the former, despite the fact that in the latter the action up is more expensive; this is compensated for by the relative costs of the other action down. ■

The remainder of this section is divided into three subsections. In the first we present a theory of amortised weighted bisimulations, based on so-called *weighted labelled transition systems*, wLTSs. This gives rise to a parametrised behavioural preorder, which we call the *amortised weighted bisimulation preorder*. The aim is to apply this theory to Picost; with this in mind, in the second subsection we present a (detailed) labelled transition semantics for Picost, and show that it is in agreement with the reduction semantics given in Figure 2. In the third section we show how this automatically generates a wLTS, which in turn gives us an amortised weighted bisimulation preorder between Picost configurations. We demonstrate the usefulness of the resulting proof methodology by re-examining the examples from Section 2.4.

3.1. **Amortised weighted bisimulations:** Here we generalise the concepts of [KAK05]; our aim is to apply them to Picost but our formulation is at a more abstract level.

Definition 3.2 (Weighted labelled transition systems)**.** An *weighted labelled transition system* or wLTS is a 4-tuple $\langle S, \mathsf{Act}_\tau, W, \longrightarrow \rangle$ where S is a set of states, W set of weights, and $\longrightarrow \ \subseteq S \times \mathsf{Act}_\tau \times W \times S$. Here Act_τ denotes a set of action names Act to which is added an extra distinct name τ which will represent internal action. We normally write $s \xrightarrow{\mu}_w s'$ to mean $(s, \mu, w, s') \in \longrightarrow$. As a default we take the set of weights to be $\mathbb{Z}$, the set of integers, both negative and positive. ■

A wLTS is called *standard* whenever there is a cost function $\mathsf{weight} : \mathsf{Act} \to W$ with the property that $s \xrightarrow{a}_w s'$ if and only if $w = \mathsf{weight}(a)$ for every $a \in \mathsf{Act}$. So in a standard wLTS there is a unique weight associated with external actions, although internal actions may have multiple possible associated weights, reflecting the different ways in which these actions may be generated from external moves. The wLTS which we will (eventually) generate for Picost will be standard, but the development below will not require that we are working with standard wLTSs.

Relative to a given wLTS *weak moves* are generated in the standard manner, although the associated weights need to be accumulated: $s \stackrel{\mu}{\Longrightarrow}_w s'$ is the least relation satisfying:

- $s \xrightarrow{\mu}_w s'$ implies $s \stackrel{\mu}{\Longrightarrow}_w s'$
- $s \stackrel{\mu}{\Longrightarrow}_w s''$, $s'' \xrightarrow{\tau}_v s'$ implies $s \stackrel{\mu}{\Longrightarrow}_{(w+v)} s'$
- $s \xrightarrow{\tau}_w s''$, $s'' \stackrel{\mu}{\Longrightarrow}_v s'$ implies $s \stackrel{\mu}{\Longrightarrow}_{(w+v)} s'$

We also use a variation on the standard notation $s \stackrel{\hat{\mu}}{\Longrightarrow}_w t$ from [Mil89]; when μ is any action other than τ this denotes $s \stackrel{\mu}{\Longrightarrow}_w t$, but when it is τ it means either that $s \stackrel{\tau}{\Longrightarrow}_w t$ or that s is t and $w = 0$.

Definition 3.3 (Amortised weighted bisimulations)**.** A family of relations $\{ \mathcal{R}^n \mid n \in \mathbb{N} \}$ over the states in a wLTS is called an *amortised weighted bisimulation* whenever $s \, \mathcal{R}^n \, t$:

(i) $s \stackrel{\mu}{\longrightarrow}_v s'$ implies $t \stackrel{\hat{\mu}}{\Longrightarrow}_w t'$ for some t', w such that $s' \, \mathcal{R}^{(n+v-w)} \, t'$

(ii) conversely, $t \stackrel{\mu}{\longrightarrow}_w t'$ implies $s \stackrel{\hat{\mu}}{\Longrightarrow}_v s'$ for some s', v such that $s' \, \mathcal{R}^{(n+v-w)} \, t'$ ■

Here the parametrisation with respect to $\mathbb{N}$ puts an extra requirement on the standard *transfer properties* associated with bisimulations. In (i) and (ii) above the index $(n+v-w)$ must be in $\mathbb{N}$, that is must be non-negative. So for example if the amortisation n is 0 then v, the weight of the left hand action, must be greater than or equal to w, the weight of the right hand action. For this reason a standard bisimulation, which ignores the weights, may not be an amortised weighted bisimulation. But the more general effect of the parameter n in the definition is to allow a relaxation in the comparison between the actual weights of the actions in the processes being compared; this point is explained in detail in Example 3.6.

We can mimic the standard development of bisimulations and write $s \sqsubseteq^m_{\mathrm{wgt}} s'$ to say that there is some amortised bisimulation $\{ \mathcal{R}^n \mid n \in \mathbb{N} \}$ such that $s \, \mathcal{R}^m \, s'$. Weighted bisimulations are (point-wise) closed under unions, and therefore we can mimic the standard development of *bisimulation equivalence*, [Mil89], to obtain the following:

Proposition 3.4.

(a) *The family of relations* $\{ \sqsubseteq^n_{\mathrm{wgt}} \mid n \in \mathbb{N} \}$ *is an amortised weighted bisimulation.*

(b) *This family is the largest (point-wise) amortised weighed bisimulation.*

(c) *If* $s \sqsubseteq^m_{\mathrm{wgt}} t$ *and* $s \stackrel{\mu}{\Longrightarrow}_v s'$ *then* $t \stackrel{\hat{\mu}}{\Longrightarrow}_w t'$ *for some* t', v *such that* $s' \sqsubseteq^{(m+v-w)}_{\mathrm{wgt}} t'$.

Proof. Straightforward, using standard techniques. □

When we are uninterested in the exact amortisation used we write simply $s \sqsubseteq_{\mathrm{wgt}} t$, meaning that there is some $k \geq 0$ such that $s \sqsubseteq^k_{\mathrm{wgt}} t$, and we refer to this preorder as the *amortised weighted bisimulation preorder*.

Proposition 3.5.

(a) *The relations* $\sqsubseteq^n_{\mathrm{wgt}}$ *are reflexive*

(b) $s_1 \sqsubseteq^m_{\mathrm{wgt}} s_2$, $s_2 \sqsubseteq^n_{\mathrm{wgt}} s_3$ *implies* $s_1 \sqsubseteq^{(m+n)}_{\mathrm{wgt}} s_3$

(c) $\sqsubseteq^m_{\mathrm{wgt}} \subseteq \sqsubseteq^n_{\mathrm{wgt}}$ *whenever* $m \leq n$.

Proof. In each case it is sufficient to exhibit a suitable amortised weighted bisimulation, that is a suitable family of relations over states. For example to prove (b) we let $\mathcal{R}^k$, for $k \geq 0$, be the set of pairs $\langle s_1, s_2 \rangle$ such that $s_1 \sqsubseteq^n_{\mathrm{wgt}} s_3$ and $s_3 \sqsubseteq^m_{\mathrm{wgt}} s_2$ for some state s_3 and some numbers n, m such that $k = n + m$.

To show $\{ \mathcal{R}^k \mid k \in \mathbb{N} \}$ is an amortised weighted bisimulation let us suppose $s_1 \, \mathcal{R}^k \, s_2$ and $s_1 \stackrel{\mu}{\longrightarrow}_v s_1'$; we have to prove

$$s_2 \stackrel{\hat{\mu}}{\Longrightarrow}_w s_2' \text{ for some } s_2' \text{ satisfying } s_1' \, \mathcal{R}^{(k+v-w)} \, s_2' \tag{3.2}$$

(The proof of the symmetric requirement is similar.)

(i) From $s_1 \sqsubseteq^n_{\mathrm{wgt}} s_3$ we know $s_3 \stackrel{\hat{\mu}}{\Longrightarrow}_u s_3'$ such that $s_1' \sqsubseteq^{(n+v-u)}_{\mathrm{wgt}} s_3'$

(ii) From $s_3 \sqsubseteq^m_{\mathrm{wgt}} s_2$, and the final part of the previous Proposition, we know $s_2 \stackrel{\hat{\mu}}{\Longrightarrow}_w s'_2$ such that $s'_3 \sqsubseteq^{(m+u-w)}_{\mathrm{wgt}} s'_2$.

But since $(n+v-u)+(m+u-w) = (k+v-w)$ we have $s'_1 \,\mathcal{R}^{(k+v-w)}\, s'_2$ and the requirement (3.2) follows.

The proof of part (c) is similar using the family of relations $\{\, \mathcal{R}^n \mid n \in \mathbb{N} \,\}$, where $s \,\mathcal{R}^n\, t$ whenever $s \sqsubseteq^m_{\mathrm{wgt}} t$ for some $m \leq n$, while the proof of part (a) uses the family where each $\mathcal{R}^n$ is the identity relation.

□

Example 3.6 (Amortising costs continued)**.** Here we continue with Example 3.1. Shortly we will see a systematic way of associating weights with actions in Picost. But informally we can simply say

$$\mathcal{C}_{25} \;\stackrel{\mathsf{up!}}{\longrightarrow}_2\; \mathcal{D}_{25} \;\stackrel{\mathsf{down!}}{\longrightarrow}_5\; \mathcal{C}_{25}$$

where $\mathcal{C}_{25}$ and $\mathcal{D}_{25}$ are abbreviations for the configurations $(\Gamma_{25} \rhd \mathsf{UD})$, respectively, $(\Gamma_{25} \rhd [\mathsf{down!.rec}\ x.\ \mathsf{up!.down!}.x]_{\mathsf{o}})$, and analogously for $(\Gamma_{42} \rhd \mathsf{UD})$. Then relative to this induced wLTS we can show that the following is a weighted bisimulation:

$$\mathcal{R}^n = \{\langle \mathcal{D}_{25}, \mathcal{D}_{42}\rangle\} \cup \{\, \langle \mathcal{C}_{25}, \mathcal{C}_{42}\rangle \mid n \geq 2 \,\}$$

It follows that

$$(\Gamma_{25} \rhd \mathsf{UD}) \sqsubseteq^2_{\mathrm{wgt}} (\Gamma_{42} \rhd \mathsf{UD})$$

However $(\Gamma_{42} \rhd \mathsf{UD}) \not\sqsubseteq^k_{\mathrm{wgt}} (\Gamma_{25} \rhd \mathsf{UD})$ for any k. To see this suppose $\{\, \mathcal{R}^n \mid n \geq 0 \,\}$ is a weighted bisimulation; we prove by induction on k that

$$\begin{aligned} \langle \mathcal{D}_{42}, \mathcal{D}_{25}\rangle &\notin \mathcal{R}^{(k+2)} \\ \langle \mathcal{C}_{42}, \mathcal{C}_{25}\rangle &\notin \mathcal{R}^{k} \end{aligned} \tag{3.3}$$

First notice that the pair $\langle \mathcal{D}_{42}, \mathcal{D}_{25}\rangle$ can not be in $\mathcal{R}^2$; this is because the move $\mathcal{D}_{42} \stackrel{\mathsf{down!}}{\longrightarrow}_2 \mathcal{C}_{42}$ can not be matched by a move $\mathcal{D}_{42} \stackrel{\mathsf{down!}}{\Longrightarrow}_w \mathcal{C}_{42}$ such that $\mathcal{C}_{42}\, \mathcal{R}^{(2+2-w)}\, \mathcal{C}_{25}$. The only only possible candidate is the move $\mathcal{D}_{42} \stackrel{\mathsf{down!}}{\Longrightarrow}_5 \mathcal{C}_{42}$ and $\mathcal{R}^{-1}$ does not exist.

From this fact it follows immediately that the pair $\langle \mathcal{C}_{42}, \mathcal{C}_{25}\rangle$ can not be in $\mathcal{R}^0$; for matching the move $\mathcal{C}_{42} \stackrel{\mathsf{up!}}{\longrightarrow}_4 \mathcal{D}_{42}$ would require the impossible, that $\langle \mathcal{D}_{42}, \mathcal{D}_{25}\rangle$ be $\mathcal{R}^2$. In other words we have shown (3.3) in the case when $k = 0$.

Suppose it is true for k; the proof that it follows for $(k+1)$ is also straightforward. This is because

- for $\langle \mathcal{D}_{42}, \mathcal{D}_{25}\rangle$ to be in $\mathcal{R}^{(k+3)}$ we would require that $\langle \mathcal{C}_{42}, \mathcal{C}_{25}\rangle$ be in $\mathcal{R}^{(k+3+2-5)}$ which contradicts the induction hypothesis
- for $\langle \mathcal{C}_{42}, \mathcal{C}_{25}\rangle$ to be in $\mathcal{R}^{(k+1)}$ we would require $\langle \mathcal{D}_{42}, \mathcal{D}_{25}\rangle$ to be in $\mathcal{R}^{(k+3)}$, which we have just shown not to be possible.

It is important that the set of natural numbers $\mathbb{N}$ is used in Definition 3.3, or at least that the family of relations be parametrised relative to a well-founded order. If instead we allowed families of relations $\{\, R^z \mid z \in \mathbb{Z} \,\}$, where $\mathbb{Z}$ is the set of all integers, positive and negative, then $(\Gamma_{42} \rhd \mathsf{UD}) \sqsubseteq^0_{\mathrm{wgt}} (\Gamma_{25} \rhd \mathsf{UD})$ would follow. Simply letting $\mathcal{R}^z = \{\langle \mathcal{C}_{42}, \mathcal{C}_{25}\rangle, \langle \mathcal{D}_{42}, \mathcal{D}_{25}\rangle\}$ for every $z \in \mathbb{Z}$, we would obtain an extended family of relations trivially satisfying the requirements in Definition 3.3. Indeed in general, using $\mathbb{Z}$ in place of $\mathbb{N}$, there would be no

difference between amortised weighted bisimulations and standard bisimulations (where all weights are ignored). ■

3.2. **An operational semantics for Picost.** As a first step in applying the theory of *amortised weighted bisimulations* to Picost we give an operational semantics for the language in terms of a (standard) LTS.

In Figure 6 and Figure 7 we give a set of rules for deriving judgements of the form

$$(\Gamma \rhd M) \stackrel{\lambda}{\mapsto} (\Delta \rhd N),$$

where λ can take one of the forms

(i) internal action, τ
(ii) input, $(\mathsf{u}, (\tilde{r}{:}\tilde{\mathsf{R}})a?v, \mathsf{p})$: input by resource a of a known or fresh name, or value, where p is the provider of the resource and u the user
(iii) output: $(\mathsf{u}, (\tilde{r}{:}\tilde{\mathsf{R}})a!v, \mathsf{p})$: delivery of a known or fresh name, to resource a, where again p is the provider of the resource and u the user.

We restrict attention to well-formed λ, that is, in the input and output actions each r_i must occur somewhere in v, and applications of the rules must preserve well-formedness. However note that because Picost only uses unary communication the vectors $(\tilde{r}), (\tilde{b})$ will have length either 0 or 1.

The rules are inherited directly from the corresponding ones for Dpi, [Hen07], and for the sake of clarity obvious symmetric rules, such as for (L-COMM) and (L-CNTX), are omitted; *Barendregt's convention* is also liberally applied, for example in omitting side-conditions to (L-CNTX). The only point of interest is the use of the preconditions $\Gamma \xrightarrow{(\mathsf{o}_1,a,\mathsf{o}_2)} \Delta$ in (L-IN) and (L-OUT); communication is only deemed to be possible if it can be paid for in some manner. Note that u in (L-IN), and p in (L-OUT) are free meta-variables. So for example the simple process $[a!\langle v\rangle.P]_{\mathsf{o}}$ can perform the actions $[a!\langle v\rangle.P]_{\mathsf{o}} \stackrel{(\mathsf{o},a!v,\mathsf{o}')}{\mapsto} \Delta \rhd [P]_{\mathsf{o}}$ for every owner $\mathsf{o}' \in \mathsf{Own}$ such that $\Gamma \xrightarrow{(\mathsf{o},a,\mathsf{o}')} \Delta$. Also in the communication rule (L-COMM) any new resources used in the communication, $\tilde{r} : \tilde{\mathsf{R}}$ remain private but in general the resulting cost environment Δ will be different from Γ; the internal communication involves the use of a resource, and the change from Γ to Δ will reflect the associated costs.

We can perform a number of sanity checks on these rules. For example one can show that if $(\Gamma_1 \rhd P_1) \stackrel{(b:\mathsf{R})\alpha}{\mapsto} (\Gamma_2 \rhd P_2)$ then $\Gamma_2 = \Delta, b{:}\mathsf{R}$ for some Δ such that $\Gamma_1 \xrightarrow{(\mathsf{u},a,\mathsf{p})} \Delta$, for some u, p, where a is the channel used in α; a more detailed analysis of the possible judgements is given in the two lemmas below. The actions also preserve configurations:

Proposition 3.7. *If* $(\Gamma_1 \rhd M_1)$ *is a configuration and* $(\Gamma_1 \rhd M_1) \stackrel{\lambda}{\mapsto} (\Gamma_2 \rhd M_2)$ *then* $(\Gamma_2 \rhd M_2)$ *is also a configuration.*

Proof. A straightforward induction on the inference of the judgements. □

We also have a consistency check with respect to the reduction semantics of Section 2, stated in the theorem below; the proof requires two technical lemmas.

Lemma 3.8 (Deriv-output)**.** *Suppose* $\Gamma \rhd M \stackrel{(\mathsf{u},(\tilde{r}:\tilde{\mathsf{R}})a!v,\mathsf{p})}{\mapsto} \Delta \rhd N$. *Then*

(i) $\Delta = (\Gamma', \tilde{r}{:}\tilde{\mathsf{R}})$ *for some* Γ'

(L-IN)
$$\frac{\Gamma \xrightarrow{(\mathsf{u},a,\mathsf{o})} \Delta}{\Gamma \rhd [a?(x)\,.P]_{\mathsf{o}} \overset{(\mathsf{u},a?v,\mathsf{o})}{\mapsto} \Delta \rhd [P\{\!\{v/x\}\!\}]_{\mathsf{o}}} \quad v \in \mathrm{dom}(\Gamma^u) \text{ or } v \text{ not a channel}$$

$$\frac{\Gamma \xrightarrow{(\mathsf{u},a,\mathsf{o})} \Delta}{\Gamma \rhd [a?(x)\,.P]_{\mathsf{o}} \overset{(\mathsf{u},(b:\mathsf{R})a?b,\mathsf{o})}{\mapsto} \Delta, b:\mathsf{R} \rhd [P\{\!\{b/x\}\!\}]_{\mathsf{o}}} \quad b \notin \mathrm{dom}(\Gamma^u)$$

(L-OUT)
$$\frac{\Gamma \xrightarrow{(\mathsf{o},a,\mathsf{p})} \Delta}{\Gamma \rhd [a!\langle v\rangle.P]_{\mathsf{o}} \overset{(\mathsf{o},a!v,\mathsf{p})}{\mapsto} \Delta \rhd [P]_{\mathsf{o}}}$$

(L-COMM)
$$\frac{\Gamma \rhd M \overset{(\mathsf{u},(\tilde{r}:\tilde{\mathsf{R}})a?v,\mathsf{p})}{\mapsto} \Delta, \tilde{r}:\tilde{\mathsf{R}} \rhd M', \quad \Gamma \rhd N \overset{(\mathsf{u},(\tilde{r}:\tilde{\mathsf{R}})a!v,\mathsf{p})}{\mapsto} \Delta, \tilde{r}:\tilde{\mathsf{R}} \rhd N'}{\Gamma \rhd M \mid N \overset{\tau}{\mapsto} \Delta \rhd (\mathsf{new}\,\widetilde{r}:\widetilde{\mathsf{R}})(M' \mid N')}$$

Figure 6: An action semantics for Picost: main rules

(L-OPEN)
$$\frac{\Gamma, b:\mathsf{R} \rhd M \overset{(\mathsf{u},a!b,\mathsf{p})}{\mapsto} \Gamma' \rhd M'}{\Gamma \rhd (\mathsf{new}\,b:\mathsf{R})M \overset{(\mathsf{u},(b:\mathsf{R})a!b,\mathsf{p})}{\mapsto} \Gamma' \rhd M'} \quad a \neq b$$

(L-EXPORT)
$$\Gamma \rhd [(\mathsf{new}\,r:\mathsf{R})P]_{\mathsf{o}} \overset{\tau}{\mapsto} \Gamma \rhd (\mathsf{new}\,r:\mathsf{R})[P]_{\mathsf{o}}$$

(L-SPLIT)
$$\Gamma \rhd [M \mid N]_{\mathsf{o}} \overset{\tau}{\mapsto} \Gamma \rhd [M]_{\mathsf{o}} \mid [N]_{\mathsf{o}}$$

(L-UNWIND)
$$\Gamma \rhd [\mathsf{rec}\ x.\ T]_{\mathsf{o}} \overset{\tau}{\mapsto} \Gamma \rhd [T\{\!\{^{\mathsf{rec}\ x.\ T}/x\}\!\}]_{\mathsf{o}}$$

(L-MATCH)
$$\frac{}{\Gamma \rhd [\mathsf{if}\ a = a\ \mathsf{then}\ P\ \mathsf{else}\ Q]_{\mathsf{o}} \overset{\tau}{\mapsto} \Gamma \rhd [P]_{\mathsf{o}}}$$

(L-MISMATCH)
$$\frac{}{\Gamma \rhd [\mathsf{if}\ a = b\ \mathsf{then}\ P\ \mathsf{else}\ Q]_{\mathsf{o}} \overset{\tau}{\mapsto} \Gamma \rhd [Q]_{\mathsf{o}}} \quad a \neq b$$

(L-CNTX)
$$\frac{\Gamma \rhd M \overset{\lambda}{\mapsto} \Gamma' \rhd M'}{\Gamma \rhd M \mid N \overset{\lambda}{\mapsto} \Gamma' \rhd M' \mid N}$$

(L-CNTX)
$$\frac{\Gamma, b:\mathsf{R} \rhd M \overset{\lambda}{\mapsto} \Gamma', b:\mathsf{R} \rhd M'}{\Gamma \rhd (\mathsf{new}\,b:\mathsf{R})M \overset{\lambda}{\mapsto} \Gamma' \rhd (\mathsf{new}\,b:\mathsf{R})M'} \quad b \notin \mathsf{n}(\lambda)$$

Figure 7: An action semantics for Picost: more rules

(ii) $\Gamma \xrightarrow{(\mathsf{u},a,\mathsf{p})} \Gamma'$
(iii) $M \equiv (\mathsf{new}\,\tilde{r}:\tilde{\mathsf{R}})(M' \mid [a!\langle v\rangle.Q]_{\mathsf{u}})$
(iv) $N \equiv (M' \mid [Q]_{\mathsf{u}})$
(v) $\Theta \rhd M \overset{(\mathsf{u},(\tilde{r}:\tilde{\mathsf{R}})\alpha,\mathsf{p}')}{\mapsto} \Theta', \tilde{r}:\tilde{\mathsf{R}} \rhd N$ *whenever* $\Theta \xrightarrow{(\mathsf{u},a,\mathsf{p}')} \Theta'$, *for any owner* p'.

Proof. By induction on the derivation of $\Gamma \rhd M \overset{(\mathsf{u},(\tilde{r}:\tilde{\mathsf{R}})a!v,\mathsf{p})}{\mapsto} \Delta \rhd N$. □

Lemma 3.9 (Deriv-input)**.** *Suppose* $\Gamma \rhd M \overset{(\mathsf{u},(\tilde{r}:\tilde{\mathsf{R}})a?v,\mathsf{p})}{\mapsto} \Delta \rhd N$. *Then*

(i) $\Delta = (\Gamma', \tilde{r}:\tilde{\mathsf{R}})$ *for some* Γ'
(ii) $\Gamma \xrightarrow{(\mathsf{u},a,\mathsf{p})} \Gamma'$
(iii) $M \equiv (\mathsf{new}\,\tilde{c}:\mathsf{C})([a?(x).T]_{\mathsf{p}} \mid M')$
(iv) $N \equiv (\mathsf{new}\,\tilde{c}:\mathsf{C})([T\{\!|{}^{v}\!/_{x}|\!\}]_{\mathsf{p}} \mid M')$
(v) $\Theta \rhd M \overset{(\mathsf{u}',(\tilde{r}:\tilde{\mathsf{R}}')\alpha,\mathsf{p})}{\mapsto} \Theta', \tilde{r}:\tilde{\mathsf{R}}' \rhd N$ *whenever* $\Theta \xrightarrow{(\mathsf{u}',a,\mathsf{p})} \Theta'$, *for any owner* u', *and types* $(\tilde{\mathsf{R}}')$.

Proof. Again a straightforward induction on the derivation $\Gamma \rhd M \overset{(\mathsf{u},(\tilde{r}:\tilde{\mathsf{R}})a?v,\mathsf{p})}{\mapsto} \Delta \rhd N$. Note that in part (v) arbitrary types $(\tilde{\mathsf{R}}')$ can be used because there is no restriction on the type R in the second part of the rule (L-IN) in Figure 6. □

Theorem 3.10. $\Gamma \rhd M \longrightarrow \Delta \rhd N$ *if and only if* $\Gamma \rhd M \overset{\tau}{\mapsto} \Delta \rhd N'$ *for some* N' *such that* $N \equiv N'$.

(Outline). First we need to show the auxiliary result that structural equivalence is preserved by actions. That is $\Gamma \rhd M \overset{\lambda}{\mapsto} \Delta \rhd M'$ and $M \equiv N$ implies $\Gamma \rhd N \overset{\lambda}{\mapsto} \Delta \rhd N'$ for some N' such that $M' \equiv N'$; this is proved by induction on the proof of the fact that $M \equiv N$ from the rules in Figure 3. Then a straightforward proof by induction on the derivation of $\Gamma \rhd M \longrightarrow \Delta \rhd N$ from the rules in Figure 2 will show that this implies $\Gamma \rhd M \overset{\tau}{\mapsto} \Delta \rhd N'$ with $N \equiv N'$; the auxiliary result is required when considering the rule (R-STRUCT).

To prove the converse we also employ the two previous lemmas, giving the structure of input and output actions. Suppose $\Gamma \rhd M \overset{\tau}{\mapsto} \Delta \rhd N$; we prove by rule induction that $\Gamma \rhd M \longrightarrow \Delta \rhd N$. The only non-trivial case is when this judgement is inferred using the rule (L-COMM), or its dual. So without loss of generality we know

- $M = M_1 \mid M_2$
- $N = (\mathsf{new}\,\tilde{r}:\tilde{\mathsf{R}})(N_1 \mid N_2)$
- $\Gamma \rhd M_1 \overset{(\mathsf{u},(\tilde{r}:\tilde{\mathsf{R}})a?v,\mathsf{p})}{\mapsto} \Delta, \tilde{r}:\tilde{\mathsf{R}} \rhd N_1$
- $\Gamma \rhd M_2 \overset{(\mathsf{u},(\tilde{r}:\tilde{\mathsf{R}})a!v,\mathsf{p})}{\mapsto} \Delta, \tilde{r}:\tilde{\mathsf{R}} \rhd N_2$

The previous two lemmas can now be applied to obtain the structure of M_1, M_2, N_1 and N_2, up to structural equivalence; by rearranging $M_1 \mid M_2$, again using the structural equivalence rules, an application of (R-COMM) followed by one of (R-STRUCT) gives the required $\Gamma \rhd M \longrightarrow \Delta \rhd N$. □

3.3. **A proof methodology for** Picost**.** The operational semantics given in the previous subsection can be used in a straightforward way to obtain a wLTS for Picost configurations. It suffices to attach a weight to the actions, which can be done in a systematic manner: we write

$$(\Gamma \rhd M) \xrightarrow{\mu}_w (\Delta \rhd N)$$

whenever

- $(\Gamma \rhd M) \overset{\mu}{\mapsto} (\Delta \rhd N)$ can be deduced from the rules in Figure 6 and Figure 7

- $w = (\Delta^{\mathsf{rec}} - \Gamma^{\mathsf{rec}})$

Note that the weight associated with an action is ultimately determined by the manner in which expenditure is recorded in the cost environments; this may reflect the cost of providing the resource in question, as in Example 2.6, the profit to be gained by a particular owner in the use of the resource, as in Example 2.8, or combinations of such concerns.

We can now apply Definition 3.3 to this wLTS to obtain a family of preorders

$$(\Gamma \rhd M) \sqsubseteq^{n}_{\mathrm{wgt}} (\Delta \rhd N) \tag{3.4}$$

between Picost configurations. However we must be somewhat careful here, as some of the actions used involve bound names; but by a systematic application of *Barendregt's convention*, mentioned on page 3, confusions between these and free names can be avoided.

As is well-known, the relations (3.4) come equipped with a powerful co-inductive proof methodology. In order to prove $(\Gamma \rhd M) \sqsubseteq^{k}_{\mathrm{wgt}} (\Delta \rhd N)$ for a particular k it is sufficient to exhibit a family of relations $\{\, \mathcal{R}^n \mid n \in \mathbb{N} \,\}$ which satisfy the transfer properties of Definition 3.3, such that $\mathcal{R}^k$ contains the pair $(\Gamma \rhd M, \Delta \rhd N)$. In the remainder of this section we apply this proof methodology to the examples in Section 2. This allows us to now reason about the behaviour of systems, how they interact with other systems, rather than reason simply about their computation runs.

Example 3.11 (Running a library, revisited)**.** Refering to the definitions in Example 2.6, by exhibiting a witness weighted bisimulation it is possible to show

$$(\Gamma_{\mathsf{central}} \rhd [\mathsf{Reader}]_{\mathsf{pub}}) \sqsubseteq^{0}_{\mathrm{wgt}} (\Gamma_{\mathsf{local}} \rhd [\mathsf{Reader}]_{\mathsf{pub}})$$

This is despite the fact that the local use of the service reqR is more expensive than the central use; this is compensated for by the fact that both goLib and goHome are less expensive locally. It is also worth noting that although the *use* of resources in both $\Gamma_{\mathsf{central}}$ and Γ_{local} is free, in the generated wLTS the output actions actually have non-zero weights associated with them. For example, a typical run in this wLTS from $(\Gamma_{\mathsf{central}} \rhd [\mathsf{Reader}]_{\mathsf{pub}})$ takes the form

$$(\Gamma_{\mathsf{central}} \rhd [\mathsf{Reader}]_{\mathsf{pub}}) \xrightarrow{\mathsf{goLib}?n}_{5} \cdots \xrightarrow{(r)\mathsf{reqR}!(r,n)}_{1} \cdots \xrightarrow{\mathsf{goHome}!b}_{5} \cdots$$

whereas the corresponding local run is

$$(\Gamma_{\mathsf{local}} \rhd [\mathsf{Reader}]_{\mathsf{pub}}) \xrightarrow{\mathsf{goLib}?n}_{1} \cdots \xrightarrow{(r)\mathsf{reqR}!(r,n)}_{3} \cdots \xrightarrow{\mathsf{goHome}!b}_{1} \cdots$$

To compare the efficiency of the library service itself we consider the following definitions

$$\mathsf{Lib}_{\mathsf{local}} \Leftarrow (\mathsf{new}\ \mathsf{reqS}{:}\mathsf{R}^{l}_{s})([\mathsf{Library} \mid \mathsf{Store}]_{\mathsf{lib}})$$
$$\mathsf{Lib}_{\mathsf{central}} \Leftarrow (\mathsf{new}\ \mathsf{reqS}{:}\mathsf{R}^{c}_{s})([\mathsf{Library} \mid \mathsf{Store}]_{\mathsf{lib}})$$

where, as explained in Example 2.6, R^{s}_{l}, R^{s}_{c}, are the types $\langle 0, 5\rangle$, $\langle 0, 1\rangle$ respectively; here the interaction between the library and the store has been internalised, with types reflecting the relative cost of local and central access. Both these configurations simply *provide* the service reqR, and viewed in isolation the local service is not more efficient than the central one; no matter what n we choose, we have

$$(\Gamma_{\mathsf{central}} \rhd \mathsf{Lib}_{\mathsf{central}}) \not\sqsubseteq^{n}_{\mathrm{wgt}} (\Gamma_{\mathsf{local}} \rhd \mathsf{Lib}_{\mathsf{local}}) \tag{3.5}$$

However if we combine the library service with the reader then the overall systems is locally more efficient than the centralised one:

$$(\Gamma_{\mathsf{central}} \rhd \mathsf{Sys}_{\mathsf{central}}) \sqsubseteq^{2}_{\mathrm{wgt}} (\Gamma_{\mathsf{local}} \rhd \mathsf{Sys}_{\mathsf{local}}) \tag{3.6}$$

where

$$\mathsf{Sys}_{\mathsf{local}} \Leftarrow (\mathsf{new}\ \mathsf{reqR}{:}\mathsf{R}^{l}_{r})([\mathsf{Reader}]_{\mathsf{pub}} \mid \mathsf{Lib}_{\mathsf{local}})$$
$$\mathsf{Sys}_{\mathsf{central}} \Leftarrow (\mathsf{new}\ \mathsf{reqR}{:}\mathsf{R}^{c}_{r})([\mathsf{Reader}]_{\mathsf{pub}} \mid \mathsf{Lib}_{\mathsf{central}})$$

We should point out that in (3.5) and (3.6) we have used the full cost environments Γ_{local}, $\Gamma_{\mathsf{central}}$, despite the fact that some of the resources have been restricted in the systems; this is simply in order to avoid the definition of even more environments.

As an example of how such statements can be proved see the Section A.1 in the appendix for a witness bisimulation which establishes (3.6). ■

4. Contextual characterisation

In the previous section we have demonstrated that the preorders $\sqsubseteq^{n}_{\mathrm{wgt}}$ provide a useful co-inductive methodology for comparing the behaviour of processes, relative to resource costs. In this section we critically review its formulation, revealing some significant inadequacies, and offer a revised version where these are addressed.

Informally we would expect at least the following two properties of a proof methodology:

(a) It should support *compositional reasoning*, whereby the analysis of process behaviour can be carried out structurally.
(b) *Soundness:* Any relationship established between the behaviour of processes using the proof methodology should be justifiable in some independent manner.

Further we could hope for:

(c) *Completeness:* any pair of processes which are intuitively behaviourally related, should be provably related using our methodology.

Relative to our language Picost the first criteria, (a), is straightforward to formalise, as a property of the preorders $\sqsubseteq^{n}_{\mathrm{wgt}}$.

Definition 4.1 (Compositional)**.** A relation $\mathcal{R}$ over Picost configurations is said to be *compositional* whenever $(\Gamma \rhd M)\ \mathcal{R}\ (\Delta \rhd N)$ implies

(i) $(\Gamma \rhd M \,|\, O)\ \mathcal{R}^{m}\ (\Delta \rhd N \,|\, O)$, provided $(\Gamma \rhd M \,|\, O)$ and $(\Delta \rhd N \,|\, O)$ are configurations
(ii) $(\Gamma, r{:}\mathsf{R} \rhd M)\ \mathcal{R}^{m}\ (\Delta, r{:}\mathsf{R} \rhd N)$. ■

We could of course demand that the relation $\mathcal{R}$ should be preserved by all the operators in the language, but for the purposes of the discussion to follow it is sufficient to concentrate on the two most important ones.

Our first remark is that the relations $\sqsubseteq^{n}_{\mathrm{wgt}}$ are not compositional, and therefore our proposed proof methodology does not support compositional reasoning.

Example 4.2 (Non-compositionality)**.** Let Γ be a cost environment with two owners o, p and two resources a, b. Suppose further that $\Gamma^{o}(\mathsf{o}) = \Gamma^{o}(\mathsf{p}) = \infty$, while $\Gamma^{u}(a) = 20$, $\Gamma^{u}(b) = 10$; the remaining fields in Γ are unimportant, but to be definite let us say that $\Gamma^{p}(a) = \Gamma^{p}(b) = 0$. Let Δ be another cost environment with the same resources, with both usage

costs being 10, and the same owners, but with the difference that $\Delta^o(\mathsf{o}) = 10$. Then it is easy to check that

$$\Gamma \rhd [a!]_{\mathsf{o}} \sqsubseteq^0_{\mathrm{wgt}} \Delta \rhd [a!]_{\mathsf{o}}$$

However one can also show that

$$\Gamma \rhd [a!]_{\mathsf{o}} \mid [b!]_{\mathsf{o}} \not\sqsubseteq^0_{\mathrm{wgt}} \Delta \rhd [a!]_{\mathsf{o}} \mid [b!]_{\mathsf{o}}$$

The problem occurs when we consider the action $(\Gamma \rhd [a!]_{\mathsf{o}} \mid [b!]_{\mathsf{o}}) \overset{(\mathsf{o},b!,\mathsf{p})}{\mapsto}_{10} (\Gamma_1 \rhd [a!]_{\mathsf{o}} \mid [\mathsf{stop}]_{\mathsf{o}})$. This can be matched by the action $(\Delta \rhd [a!]_{\mathsf{o}} \mid [b!]_{\mathsf{o}}) \overset{(\mathsf{o},b!,\mathsf{p})}{\mapsto}_{10} (\Delta_1 \rhd [a!]_{\mathsf{o}} \mid [\mathsf{stop}]_{\mathsf{o}})$ but at the expense of exhausting all of o's funds. $\Delta^o_1(\mathsf{o})$ is now set to 0 and therefore the action $(\Gamma_1 \rhd [a!]_{\mathsf{o}} \mid [\mathsf{stop}]_{\mathsf{o}}) \overset{(\mathsf{o},a!,\mathsf{p})}{\mapsto}_{20} (\Gamma_1 \rhd [\mathsf{stop}]_{\mathsf{o}} \mid [\mathsf{stop}]_{\mathsf{o}})$ can not be matched by any action from $(\Delta_1 \rhd [a!]_{\mathsf{o}} \mid [\mathsf{stop}]_{\mathsf{o}})$.

The other criteria, (b) and (c) above, are more difficult to formalise. But even in the absence of a precise formalisation we can also show that our proof methodology runs into difficulties with them, by considering a proposed touchstone family of preorders $\sqsubseteq^n_{\mathrm{behav}}$, $n \geq 0$, which incorporate some intuitive properties which we would expect. First an easy example, essentially taken from [HR04].

Example 4.3 (Problem with output types)**.** Consider the two configurations $\mathcal{C}$ and $\mathcal{D}$, denoted by

$$\Gamma \rhd (\mathsf{new}\, r{:}\mathsf{R}_1)([a!\langle r\rangle.\, \mathsf{stop}]_{\mathsf{o}}), \qquad \Gamma \rhd (\mathsf{new}\, r{:}\mathsf{R}_2)([a!\langle r\rangle.\, \mathsf{stop}]_{\mathsf{o}})$$

respectively, where the types R_1, R_2 are different, and Γ has sufficient resources for a to be exercised; that is $\Gamma \xrightarrow{(\mathsf{o},a,\mathsf{p})} \Gamma'$ for some owner p and some Γ'.

Then it is easy to see that $\mathcal{C} \not\sqsubseteq^k_{\mathrm{wgt}} \mathcal{D}$ for any k because the only actions which the configurations can perform are different; they are labelled $(\mathsf{p}, (r{:}\mathsf{R}_1)a!r, \mathsf{o})$ and $(\mathsf{p}, (r{:}\mathsf{R}_2)a!r, \mathsf{o})$ respectively.

However it is difficult to envisage any context in which these two configurations can be distinguished; for any reasonable definition of the touchstone relations we would expect $\mathcal{C} \sqsubseteq^k_{\mathrm{behav}} \mathcal{D}$ to be true. Thus our proof methodology will not be *complete*. ■

Our next example focuses on some of the novel features of Picost.

Example 4.4 (Problem with owner identification)**.** Let $\mathcal{C}$, $\mathcal{D}$ denote the configurations

$$\Gamma \rhd [a!]_{\mathsf{o}_1}, \qquad \Gamma \rhd [a!]_{\mathsf{o}_2}$$

respectively, where o_1, o_2 are two different owners, and $\Gamma^o(\mathsf{o}_1) = \Gamma^o(\mathsf{o}_2)$.

Here again we would expect $\mathcal{C} \sqsubseteq^k_{\mathrm{behav}} \mathcal{D}$ to be true because there is no mechanism in Picost which would enable an observer to discover who was funding the use of the resource a. However assuming some owner p has sufficient funds in Γ to provide the resource a, we have $\mathcal{C} \not\sqsubseteq^0_{\mathrm{wgt}} \mathcal{D}$ again because the configurations perform different actions, labelled $(\mathsf{o}_1, a!, \mathsf{p})$ and $(\mathsf{o}_2, a!, \mathsf{p})$ respectively. ■

4.1. **Behavioural preorders.** In order to address the inadequacies with our proof methodology let us first give one possible formalisation of the touchstone family of behavioural preorders which we have been refering to as $\sqsubseteq^n_{\mathrm{behav}}, n \geq 0$; we adapt the theory of *reduction barbed congruences*, [HT92, SW01, HR04] to Picost, often refered to informally as *contextual equivalences*. For simplicity we assume that resource charging is always standard, and that the only values used are channel/resource names.

We first need to introduce into the reduction semantics some record of the costs being expended. Let us write $\Gamma \rhd M \longrightarrow_c \Delta \rhd N$ whenever $\Gamma \rhd M \longrightarrow \Delta \rhd N$ can be deduced from the reduction rules, in Figure 2, and $(\Delta^{\mathsf{rec}} - \Gamma^{\mathsf{rec}}) = c$. This is generalised in the obvious manner to $\Gamma \rhd M \longrightarrow^*_d \Delta \rhd N$ by the accumulation of costs.

Definition 4.5 (Cost improving)**.** We say that the family of relations $\{\, \mathcal{R}^n \mid n \in \mathbb{N} \,\}$ over configurations is *cost improving* whenever $\mathcal{C}\ \mathcal{R}^m\ \mathcal{D}$ for any m, then

(i) $\mathcal{C} \longrightarrow_c \mathcal{C}'$ implies $\mathcal{D} \longrightarrow^*_d \mathcal{D}'$ such that $\mathcal{C}'\ \mathcal{R}^{(m+c-d)}\ \mathcal{D}'$

(ii) conversely, $\mathcal{D} \longrightarrow_d \mathcal{D}'$ implies $\mathcal{C} \longrightarrow^*_c \mathcal{C}'$ such that $\mathcal{C}'\ \mathcal{R}^{(m+c-d)}\ \mathcal{D}'$. ■

This is a natural generalisation of the notion of *reduction closure* or *reduction bisimulation* from LTSs to weighted LTSs; for a justification of its use in defining behavioural preorders see Chapter 2 of [SW01].

Definition 4.6 (Observations)**.** Let us write $(\Gamma \rhd M) \Downarrow a?$ whenever $(\Gamma \rhd M) \longrightarrow^* (\Delta \rhd N)$ where for some owner o

(i) $N \equiv (\mathsf{new}\tilde{c})([a?(x)\,.T]_{\mathsf{o}} \mid N')$, and a does not occur in $(\tilde{c})$

(ii) $\Delta \xrightarrow{(\mathsf{u},a,\mathsf{o})} \Delta'$ for some u and Δ'.

The predicate $(\Gamma \rhd M) \Downarrow a!$ is defined in an analogous manner. Note that here the owner o has to be able to pay the appropriate costs for the barb.

Then we say that the family of relations $\{\, \mathcal{R}^n \mid n \in \mathbb{N} \,\}$ over configurations *preserves observations* whenever, for any n, $\mathcal{C}_1\ \mathcal{R}^n\ \mathcal{C}_2$ $\mathcal{C}_1 \Downarrow o$ if and only if $\mathcal{C}_2 \Downarrow o$. ■

Note that unlike [HG08] we do not record the cost of making observations; nor do we observe the owner responsible for the observation. This means that our notion of barb is more elementary.

Example 4.2 demonstrates that demanding a behavioural preorder to be compositional, in particular that it be preserved by arbitrary parallel contexts, is very problematic as intuitively it gives observers or external users of a system access to all the funds available to owners of the system. Here we address this issue by defining a relativised version of compositionality, relativised to the set of owners whose funds are available to external users.

Definition 4.7 (O-contextual)**.** Let O be a subset of the owners Own. A relation $\mathcal{R}$ over Picost configurations is said to be O-*contextual* whenever $(\Gamma \rhd M)\ \mathcal{R}\ (\Delta \rhd N)$ implies

(i) $(\Gamma \rhd M \mid [P]_{\mathsf{o}})\ \mathcal{R}\ (\Delta \rhd N \mid [P]_{\mathsf{o}})$ for every $\mathsf{o} \in \mathsf{O}$, provided $(\Gamma \rhd M \mid [P]_{\mathsf{o}})$ and $(\Delta \rhd N \mid [P]_{\mathsf{o}})$ are configurations.

(ii) $(\Gamma, r{:}\mathsf{R} \rhd M)\ \mathcal{R}\ (\Delta, r{:}\mathsf{R} \rhd N)$. ■

Combining these three properties we obtain:

Definition 4.8 (The contextual improvement preorder)**.** Let $\{\, \sqsubseteq^n_{\mathsf{O}:\mathrm{cxt}} \mid\ n \in \mathbb{N} \,\}$ be the largest family (point-wise) of O-contextual relations over configurations which preserves observations, and is cost improving. ■

The idea here is that we only consider the behaviour of systems relative to contexts in which observers, or users of the systems, can use code running under the financial authority of the owners in O. At one extreme we can take O to be the entire set of owners Own and then observers have access to all owners, and their funds; this gives Compositionality, as expressed in Definition 4.1. The other extreme is when observers have access to none of the owners users in the systems under observation; in this case the observers have to provide their own funds, to support observations.

We now set ourselves the task of modifying the proof methodology of Section 3.3 so that the informal properties (a), (b), and (c) are enforced, relative to the touchstone preorders $\sqsubseteq^n_{\mathsf{O}:\mathrm{cxt}}$. First note that Example 4.4 and Example 4.3 still apply when the informal relations $\sqsubseteq^n_{\mathrm{behav}}$ are instantiated by the formal $\sqsubseteq^n_{\mathsf{O}:\mathrm{cxt}}$. But the problems presented in Example 4.2 depend on the choice of observers O:

Example 4.9 (Unsoundness)**.** Let Γ, Δ be as defined in Example 4.2. Then we have already argued that $\Gamma \rhd [a!]_{\mathsf{o}} \sqsubseteq^0_{\mathrm{wgt}} \Delta \rhd [a!]_{\mathsf{o}}$. Here we argue that $\Gamma \rhd [a!]_{\mathsf{o}} \not\sqsubseteq^0_{\mathsf{O}:\mathrm{cxt}} \Delta \rhd [a!]_{\mathsf{o}}$. whenever $\mathsf{o} \in \mathsf{O}$. For otherwise, this would imply

$$\Gamma \rhd [a!]_{\mathsf{o}} \mid [P]_{\mathsf{o}} \sqsubseteq^0_{\mathsf{O}:\mathrm{cxt}} \Delta \rhd [a!]_{\mathsf{o}} \mid [P]_{\mathsf{o}}$$

for any process P which ensures that the configurations are still well-formed.

However for a contradiction take P to be $a?.(b! \mid b?.\omega!)$ where ω is some cost-free fresh channel. Then we can make the observation $\omega!$ on the left hand configuration but not on the right hand one. ■

This example shows that in general O-observers can deplete the resources of any owner in O, which is important if those owners have only finite funds. A significant consequence is given in the next proposition, which limits the applicability of this behavioural preorder for arbitrary O.

Proposition 4.10. *If* $(\Gamma \rhd M) \sqsubseteq^n_{\mathsf{O}:\mathrm{cxt}} (\Delta \rhd N)$ *for any* n*, then* $\Gamma^o(\mathsf{o}) = \Delta^o(\mathsf{o})$ *for every* o *in* O*.*

Proof. Suppose $(\Gamma \rhd M) \sqsubseteq^n_{\mathsf{O}:\mathrm{cxt}} (\Delta \rhd N)$ for some n, with o an owner in O. We prove that $k \leq \Gamma^o(\mathsf{o})$ if and only if $k \leq \Delta^o(\mathsf{o})$.

Consider the process $O = [(\mathsf{new}\, r\!:\!\mathsf{R})r! \mid r?.\omega!\langle\rangle]_{\mathsf{o}}$, where ω is a fresh cost-free channel, where R is the resource type $\langle k, 0\rangle$; so r costs k to use but is free to provide. Then by compositionality we know

$$\Gamma, \omega : \mathsf{E} \rhd M \mid O \;\sqsubseteq^n_{\mathsf{O}:\mathrm{cxt}}\; \Delta, \omega : \mathsf{E} \rhd N \mid O$$

where E denotes the trivial type $\langle 0, 0\rangle$.

If $k \leq \Gamma^o(\mathsf{o})$, we have $\Gamma, \omega : \mathsf{E} \rhd M \mid O \Downarrow \omega!$ and therefore, by the preservation of observations, $\Delta, \omega : \mathsf{E} \rhd N \mid O \Downarrow \omega!$. But this is only possible if $k \leq \Delta^o(\mathsf{o})$.

The converse argument is similar. □

In effect this means that the behavioural preorders $\sqsubseteq^n_{\mathsf{O}:\mathrm{cxt}}$ can not be used to differentiate between configurations in which owners from O accrue different levels of funds; a typical case in point occurs with the systems in Example 2.9. For this reason we are primarily interested in the extreme case, when the observers have no access to the funds of the owners in the systems under investigation. Let us introduce some special notation for these situations.

Let e denote some arbitrary owner, intuitively taken to be external to the systems under observation. For an arbitrary cost environment Γ we use Γ^{e} to denote the extended

cost environment obtained by adding e to the domain of Γ^o and setting $\Gamma^o(\mathsf{e})$ to be ∞; in particular Γ^{e} is only defined whenever e is new to the domain of Γ^o. Finally we use the notation

$$\Gamma \rhd M \sqsubseteq^n_{\mathsf{ecxt}} \Delta \rhd N$$

as an abbreviation for

$$\Gamma^{\mathsf{e}} \rhd M \sqsubseteq^n_{\{\mathsf{e}\}:\mathrm{cxt}} \Delta^{\mathsf{e}} \rhd N$$

Here the observer has no access to the owners' resources used in the configurations $\mathcal{C}$, $\mathcal{D}$ but has an infinite amount of resources with which to run experiments.

Our revised proof methodology is based on endowing Picost with the structure of a different, more abstract, wLTS, which takes into account the set of owners whose funds are available to observers, and employing Definition 3.3 to obtain a more abstract family of co-inductive preorders. In order to obtain our more abstract wLTS we forget some of the details in the labels of the actions of the operational semantics for Picost, given in Figure 6 and Figure 7, so that they reflect not what processes can do, but rather what external observers with access to the funds in O can observe them doing. This leads to *abstract labels* of the following form, ranged over by μ:

(a) internal label τ as before
(b) input label $(\mathsf{u}, (\tilde{r} : \tilde{\mathsf{R}})a?v)$
(c) output label $((\tilde{r})a!v, \mathsf{p})$

Here only one owner is recorded in the external actions; for input we note the user of the resource u while for output it is the producer p.

Definition 4.11 (O-actions)**.** For each abstract label μ let the corresponding O-action $\mathcal{C} \xrightarrow{\mu}{}^{\mathsf{O}}_{w} \mathcal{D}$ be defined by

(a) $(\Gamma_1 \rhd M) \xrightarrow{\tau}{}^{\mathsf{O}}_{w} (\Gamma_2 \rhd N)$ whenever $(\Gamma_1 \rhd M) \overset{\tau}{\mapsto} (\Gamma_2 \rhd N)$ can be deduced from the rules, where $(\Gamma_2^{\mathsf{rec}} - \Gamma_1^{\mathsf{rec}}) = w$.
(b) $(\Gamma_1 \rhd M) \xrightarrow{((\tilde{r})a!b,\mathsf{p})}{}^{\mathsf{O}}_{w} (\Gamma_2 \rhd N)$ whenever $\mathsf{p} \in \mathsf{O}$ and $(\Gamma_1 \rhd M) \overset{(\mathsf{u},(\tilde{r}:\tilde{\mathsf{R}})a!b,\mathsf{p})}{\mapsto} (\Gamma_2 \rhd N)$ can be deduced from the rules for some $(\tilde{\mathsf{R}})$, and some owner u, where $(\Gamma_2^{\mathsf{rec}} - \Gamma_1^{\mathsf{rec}}) = w$.
(c) $(\Gamma_1 \rhd M) \xrightarrow{(\mathsf{u},(\tilde{r}:\tilde{\mathsf{R}})a?b)}{}^{\mathsf{O}}_{w} (\Gamma_2 \rhd N)$ whenever $\mathsf{u} \in \mathsf{O}$ and $(\Gamma_1 \rhd M) \overset{(\mathsf{u},(\tilde{r}:\tilde{\mathsf{R}})a?b)}{\mapsto} (\Gamma_2 \rhd N)$ can be deduced from the rules for some owner p, where $(\Gamma_2^{\mathsf{rec}} - \Gamma_1^{\mathsf{rec}}) = w$.

Note that in (a) the set of owners O plays no role, but we leave it there for the sake of uniformity. ■

This endows Picost configurations with the structure of a more abstract wLTS, whose actions depend on the set of owners O. We refer to this as the O-wLTS and we write $\mathcal{C} \sqsubseteq^n_{\mathrm{Owgt}} \mathcal{D}$ whenever there is an amortised weighted bisimulation $\{\, \mathcal{R}^n \mid n \in \mathbb{N} \,\}$ in this O-wLTS such that $\mathcal{C}\, \mathcal{R}^n\, \mathcal{D}$. When O is the singleton set $\{\mathsf{e}\}$ where the owner e is fresh, that is external to the configurations being compared, we abbreviate this to $\mathcal{C} \sqsubseteq^n_{\mathsf{ewgt}} \mathcal{D}$.

Example 4.12 (Publishing, revisited)**.** Here we use the notation and definitions from Example 2.8 and Example 2.9.

First we can compare the profits gained by running the publishing system in different cost environments. As before let Γ_{327} represent any cost environment of the form $\Gamma_{dyn}, \mathsf{news}{:}\mathsf{R}_n, \mathsf{adv}{:}\mathsf{R}_a, \mathsf{publish}{:}\mathsf{R}_p$, where these types are $\langle 3, 1\rangle, \langle 2, 0\rangle, \langle 7, 1\rangle$ respectively, and

let Γ_{216} be the same environment but with these types changed to $\langle 2,1\rangle, \langle 1,0\rangle, \langle 6,1\rangle$. Then it is straightforward to exhibit a witness bisimulation to establish

$$(\Gamma_{216} \rhd [P]_{\mathsf{p}}) \sqsubseteq^0_{\mathsf{ewgt}} (\Gamma_{327} \rhd [P]_{\mathsf{p}})$$

Recall from Example 2.8 that in these cost environments we record the costs of the actions relative to their effect on the funds of p the publisher. So this means that that more profit can be gained by the publisher p by using the cost regime underlying the environment Γ_{216}.

To investigate the effect of implementing the *kickback* we consider the two systems

$$\mathrm{PA} \Leftarrow (\mathsf{new}\,\mathsf{adv}{:}\mathsf{R}_a)([P]_{\mathsf{p}} \mid [A]_{\mathsf{a}})$$
$$\mathrm{PA}_K \Leftarrow (\mathsf{new}\,\mathsf{adv}{:}\mathsf{R}_a)([P_K]_{\mathsf{p}} \mid [A_K]_{\mathsf{a}})$$

Both these systems *use* the news resource and *provide* the $\mathsf{publish}$ resource. Here we can show, for example, that

$$(\Gamma_{327} \rhd \mathrm{PA}_K) \sqsubseteq^0_{\mathsf{ewgt}} (\Gamma_{327} \rhd \mathrm{PA})$$

provided $\Gamma^o_{327}(\mathsf{p})$ is at least 5. See Section A.2 of the appendix for a description of a witness bisimulation. Again because of the way in which we have set up the accounting in the cost environments this means that the code PA_K is more profitable for the publisher than PA. ■

The abstract O-wLTS has precisely enough information about actions to characterise the touchstone contextual behavioural preorder, at least in the extreme case of $\mathsf{O} = \{\mathsf{e}\}$.

Theorem 4.13 (Full-abstraction, external case)**.** *For every $n \in \mathbb{N}$, $(\Gamma \rhd M) \sqsubseteq^n_{\mathsf{ecxt}} (\Delta \rhd N)$ if and only if $(\Gamma \rhd M) \sqsubseteq^n_{\mathsf{ewgt}} (\Delta \rhd N)$.*

Proof. This will follow from the more general full-abstraction result, given in Theorem 4.19. □

Unfortunately this result is not true for an arbitrary set of external owners O. Example 4.9 can be used to show that the O-wLTS has not taken into account the fact that observers have access to the funds of arbitrary owners in O.

Example 4.14. We use the notation from Example 4.9, which in turn is inherited from Example 4.2. Let O be a set of owners which includes o and the fresh e. Then it is easy to check that $\Gamma \rhd [a!]_{\mathsf{o}} \sqsubseteq^0_{\mathrm{Owgt}} \Delta \rhd [a!]_{\mathsf{o}}$. But we have already argued in Example 4.9 that $\Gamma \rhd [a!]_{\mathsf{o}} \not\sqsubseteq^0_{\mathsf{O:cxt}} \Delta \rhd [a!]_{\mathsf{o}}$. ■

So we have to revise the O-wLTS to take into account the access which observers may have to funds being used by the systems under investigation.

Definition 4.15 (Fund transfer)**.** For every $k \in \mathbb{N}$ let $\xrightarrow{(\mathsf{u},k,\mathsf{p})}$ be the partial function over cost environments defined by letting $\Gamma \xrightarrow{(\mathsf{u},k,\mathsf{p})} \Delta$ whenever Δ can be obtained from Γ by transferring k funds from owner u to owner p. Formally this partial function is only defined when $\Gamma^o(\mathsf{u}) \geq k$, in which case $\Delta^o(\mathsf{u}) = \Gamma^o(\mathsf{u}) - k$, $\Delta^o(\mathsf{p}) = \Gamma^o(\mathsf{p}) + k$, when $\mathsf{p} \neq \mathsf{u}$ and all other components of Δ are inherited directly from Γ; when $\mathsf{p} = \mathsf{u}$ the operation leaves Δ unchanged. This leads to a new action over configurations, with a new abstract label $\mathsf{ext}(\mathsf{u},k,\mathsf{p})$: we let

$$(\Gamma_1 \rhd M) \xrightarrow{\mathsf{ext}(\mathsf{u},k,\mathsf{p})}{}^{\mathsf{o}}_{w} (\Gamma_2 \rhd M)$$

whenever $\Gamma_1 \xrightarrow{(\mathsf{u},k,\mathsf{p})} \Gamma_2$, and u,p are owners in O, where $w = (\Gamma_2^{\mathsf{rec}} - \Gamma_1^{\mathsf{rec}})$. ■

This gives rise to yet another LTS whose states are Picost configurations, which we refer to as O-awLTS, which induces another bisimulation preorder. But we also need to take Proposition 4.10 into account.

Definition 4.16 (Abstract weighted bisimulation preorder)**.** A family of relations over Picost configurations $\{\,\mathcal{R}^n \mid n \in \mathbb{N}\,\}$ is said to be a O-*abstract amortised weighted bisimulation* whenever

(i) $\Gamma \rhd M \;\; \mathcal{R}^n \;\; \Delta \rhd M'$ implies $\Gamma^o(\mathsf{o}) = \Delta^o(\mathsf{o})$ for every o in O

(ii) $\{\,\mathcal{R}^n \mid n \in \mathbb{N}\,\}$ is an amortised weighted bisimulation in O-awLTS.

We write $\mathcal{C} \sqsubseteq^n_{\mathsf{O}\text{awgt}} \mathcal{D}$ to denote the maximal family of such relations. ■

Note that these relations $\{\,\sqsubseteq^n_{\mathsf{O}\text{awgt}} \mid \; n \in \mathbb{N}\,\}$ actually coincide with $\{\,\sqsubseteq^n_{\mathsf{e}\text{wgt}} \mid \; n \in \mathbb{N}\,\}$ when O is the singleton external observer $\{\mathsf{e}\}$; this follows because the extra fund transfer actions have no effect: $(\Gamma^{\mathsf{e}}_1 \rhd M) \xrightarrow{\mathsf{ext}(\mathsf{u},k,\mathsf{p})}{}^{\{\mathsf{e}\}}_{w} (\Gamma^{\mathsf{e}}_2 \rhd M)$ if and only if $\Gamma^{\mathsf{e}}_1 = \Gamma^{\mathsf{e}}_2$.

It also coincides with the preorders used in Section 3.3, under certain conditions.

Proposition 4.17. *Let* O *be the set of owners used in the two configurations* Γ *and* Δ *and suppose that all owners in* O *have indefinite funds; that is* $\Gamma(o) = \Delta(o) = \infty$ *for every owner* $\mathsf{o} \in \mathsf{O}$*. Then* $\Gamma \rhd M \sqsubseteq^n_{\mathsf{wgt}} \Delta \rhd N$ *implies* $\Gamma \rhd M \sqsubseteq^n_{\mathsf{O}\text{awgt}} \Delta \rhd N$*.*

Proof. Straightforward. When funds are unlimited the constraint (i) in Definition 4.16 is vacuous, as is the requirement to match the fund actions labelled $\mathsf{ext}(\mathsf{u},k,\mathsf{p})$. The result now follows because every concrete action in the wLTS used in Section 3.3 is automatically also an abstract action in O-awLTS. □

It follows that the work of Section 3.3 has not been in vain; the proofs in the examples can be taken to be about the more abstract preorders $\sqsubseteq^n_{\mathsf{O}\text{awgt}}$.

The remainder of this section is devoted to showing that, subject to a minor restriction, the co-inductive proof methodology based on $\{\,\sqsubseteq^n_{\mathsf{O}\text{awgt}} \mid \; n \in \mathbb{N}\,\}$ satisfies the informal criteria (a), (b), and (c) set out at the begining of this section. It has certain advantages over that used in Section 3.3; in matching input and output moves the principles involved do not have to match up exactly. However in the general case it also has a disadvantage with cost environments in which certain owners have finite funds. If the observer has access to such owners then is necessary to establish that the proposed relations between configurations are invariant under the transfer of funds between them. Of course in the particular case of a purely external observer, where O is taken to be $\{\mathsf{e}\}$, which is possibly the most interesting case, then this requirement is vacuous.

Definition 4.18 (Simple types)**.** The type $\mathsf{R} = \langle k_u, k_p\rangle$ is *simple* whenever $k_p = 0$, meaning that resources of type R cost nothing to provide. A cost environment is called *simple* whenever it can be written as $\Gamma_{dyn}, a_1{:}\mathsf{R}_1, \ldots a_n{:}\mathsf{R}_n$ where Γ_{dyn} is a basic environment and all R_i are simple.

Restricting attention to simple types we know that for every resource name a there is some $k \in \mathbb{N}$ such that $\Gamma \xrightarrow{(\mathsf{u},a,\mathsf{p})} \Delta$ if and only if $\Gamma \xrightarrow{(\mathsf{u},k,\mathsf{p})} \Delta$. ■

Theorem 4.19 (Full-abstraction)**.** *Assuming simple cost environments, for every set of observers* O *and every* $n \in \mathbb{N}$*,* $(\Gamma \rhd M) \sqsubseteq^n_{\mathsf{O}:\mathrm{cxt}} (\Delta \rhd N)$ *if and only if* $(\Gamma \rhd M) \sqsubseteq^n_{\mathsf{O}\text{awgt}} (\Delta \rhd N)$*.*

The proof of this result is the subject of the remainder of this section; we will also see how the restriction to simple types can be lifted, at the expense of a generalisation of the fund action from Definition 4.15.

4.2. **Full abstraction.** First let us consider criteria (a) above, Compositionality. In fact we now have a parametrised version of this, O-contextuality from Definition 4.7, which we tackle in two steps. First we require a lemma.

Lemma 4.20.

(i) *Suppose* $\Gamma \rhd M \overset{\lambda}{\mapsto} \Delta \rhd N$. *Then* $\Gamma, r{:}\mathsf{R} \rhd M \overset{\lambda}{\mapsto} \Delta, r{:}\mathsf{R} \rhd N$.

(ii) *Conversely, suppose* $\Gamma, r{:}\mathsf{R} \rhd M \overset{\lambda}{\mapsto} \Delta, r{:}\mathsf{R} \rhd N$, *where the label* λ *does not describe a communication along the channel* r. *Then*

(a) $\Gamma \rhd M \overset{\lambda}{\mapsto} \Delta \rhd N$

(b) *or the concrete action label* λ *is of the form* $(\mathsf{u}, a?r, \mathsf{p})$, *in which case* $\Gamma \rhd M \overset{(\mathsf{u},(r:\mathsf{R})a?r,\mathsf{p})}{\mapsto} \Delta, r{:}\mathsf{R} \rhd N$.

(iii) $\Gamma \rhd M \overset{(\mathsf{u},(r:\mathsf{R})a?r,\mathsf{p})}{\mapsto} \Delta, r{:}\mathsf{R} \rhd N$ *implies* $\Gamma, r{:}\mathsf{R} \rhd M \overset{(\mathsf{u},a?r,\mathsf{p})}{\mapsto} \Delta, r{:}\mathsf{R} \rhd N$

Proof. Each statement is proved by induction on the derivation of the judgement. Note that for any a in the domain of Γ, $\Gamma \xrightarrow{(\mathsf{u},a,\mathsf{p})} \Delta$ if and only if $\Gamma, r{:}\mathsf{R} \xrightarrow{(\mathsf{u},a,\mathsf{p})} \Delta, r{:}\mathsf{R}$. □

Proposition 4.21 (O-contextual)**.** $(\Gamma \rhd M) \sqsubseteq^n_{\mathsf{O}\text{awgt}} (\Delta \rhd N)$ *implies* $(\Gamma, r{:}\mathsf{R} \rhd M) \sqsubseteq^n_{\mathsf{O}\text{awgt}} (\Delta, r{:}\mathsf{R} \rhd N)$.

Proof. Let $\{\, \mathcal{R}^n \mid n \in \mathbb{N} \,\}$ be the family of relations over Picost configurations defined by letting $(\Gamma, r{:}\mathsf{R} \rhd M)\, \mathcal{R}^n\, (\Delta, r{:}\mathsf{R} \rhd N)$ whenever

(i) either $(\Gamma \rhd M) \sqsubseteq^n_{\mathsf{O}\text{awgt}} (\Delta \rhd N)$

(ii) or $(\Gamma, r{:}\mathsf{R} \rhd M) \sqsubseteq^n_{\mathsf{O}\text{awgt}} (\Delta, r{:}\mathsf{R} \rhd N)$.

It is sufficient to show that this satisfies the conditions in Definition 4.16. Note that condition (i) of this definition is trivial.

So suppose $(\Gamma, r{:}\mathsf{R} \rhd M)\, \mathcal{R}^n\, (\Delta, r{:}\mathsf{R} \rhd N)$ and $(\Gamma, r{:}\mathsf{R} \rhd M) \xrightarrow{\mu}{}^{\mathsf{o}}_{v} (\Gamma', r{:}\mathsf{R} \rhd M')$ is an abstract action. We have to find a matching abstract move $(\Delta, r{:}\mathsf{R} \rhd N) \overset{\hat{\mu}}{\Longrightarrow}{}^{\mathsf{o}}_{w} (\Delta', r{:}\mathsf{R} \rhd N')$.

Let us look at the concrete action underlying this abstract action, $(\Gamma, r{:}\mathsf{R} \rhd M) \overset{\lambda}{\mapsto} (\Gamma', r{:}\mathsf{R} \rhd M')$. Since we know $(\Gamma \rhd M)$ is a configuration λ can not describe a communication along r, and so we can apply part (2) of the previous lemma, to obtain two cases:

(a) $\Gamma \rhd M \overset{\lambda}{\mapsto} \Gamma' \rhd M'$. In this case the required matching move can be obtained using the fact that $(\Gamma \rhd M) \sqsubseteq^n_{\mathsf{O}\text{awgt}} (\Delta \rhd N)$, together with an application of part (1) of Lemma 4.20.

(b) λ is the input action $(\mathsf{u}, a?r, \mathsf{p})$, and $\Gamma \rhd M \overset{(\mathsf{u},(r:\mathsf{R})a?r,\mathsf{p})}{\mapsto} \Gamma', r{:}\mathsf{R} \rhd N$. Here we again use the fact that $(\Gamma \rhd M) \sqsubseteq^n_{\mathsf{O}\text{awgt}} (\Delta \rhd N)$ to find a matching weak concrete move from $(\Delta \rhd N)$ labelled $(\mathsf{u}, (r : \mathsf{R})a?r, \mathsf{p}')$ for some owner p'. Part (3) of Lemma 4.20 can now be used to transform this into a required matching move from $(\Delta, r{:}\mathsf{R} \rhd N)$. In this case the matching will be because of clause (ii) in the definition of the family $\mathcal{R}^n$. □

Theorem 4.22 (O-contextual)**.** *Suppose* $(\Gamma \rhd M \,|\, [P]_{\mathsf{o}})$ *and* $(\Delta \rhd N \,|\, [P]_{\mathsf{o}})$ *are both configurations, where* $\mathsf{o} \in \mathsf{O}$. *Then* $(\Gamma \rhd M) \sqsubseteq^k_{\mathsf{O}\text{awgt}} (\Delta \rhd N)$ *implies* $(\Gamma \rhd M \,|\, [P]_{\mathsf{o}}) \sqsubseteq^k_{\mathsf{O}\text{awgt}} (\Delta \rhd N \,|\, [P]_{\mathsf{o}})$.

Proof. We follow the standard proof structure, see Section 2.3 of [SW01], Proposition 6.4 of [HR04], Proposition 2.21 of [Hen07]; however the precise details are somewhat different. Let $\{\, \mathcal{R}^n \mid n \in \mathbb{N} \,\}$ be the smallest family of relations which satisfies:

(i) $\Gamma \rhd M \sqsubseteq^n_{\text{awgt}} \Delta \rhd N$ implies $\Gamma \rhd M\, \mathcal{R}^n\, \Delta \rhd N$

(ii) $\Gamma \rhd M \,\mathcal{R}^n\, \Delta \rhd N$ implies $(\Gamma \rhd M \mid [P]_{\mathsf{o}})\,\mathcal{R}^n\,(\Delta \rhd N \mid [P]_{\mathsf{o}})$, whenever $\mathsf{o} \in \mathsf{O}$ and both $(\Gamma \rhd M \mid [P]_{\mathsf{o}})$ and $(\Delta \rhd N \mid [P]_{\mathsf{o}})$ are configurations

(iii) $\Gamma, r : \mathsf{R}_1 \rhd M \,\mathcal{R}^n\, \Delta, r : \mathsf{R}_2 \rhd N$ implies $\Gamma \rhd (\mathsf{new}\, r{:}\mathsf{R}_1)M \,\mathcal{R}^n\, \Delta \rhd (\mathsf{new}\, r{:}\mathsf{R}_2)N$.

We show that this family satisfies the requirements of Definition 4.16, up to structural equivalence, from which the result will follow.

First note that for any n,

$$\Gamma \rhd M \,\mathcal{R}^n\, \Delta \rhd N \;\text{ implies }\; \Gamma, r{:}\mathsf{R} \rhd M \,\mathcal{R}^n\, \Delta, r{:}\mathsf{R} \rhd N \tag{4.1}$$

This can be proved by induction on why $\Gamma \rhd M \,\mathcal{R}^n\, \Delta \rhd N$, with the base case being provided by Proposition 4.21.

So suppose $\Gamma \rhd M \;\mathcal{R}^n\; \Delta \rhd N$ and $\Gamma \rhd M \xrightarrow{\mu}{}^{\mathsf{o}}_{v} \Gamma' \rhd M_d$; we have to find a matching abstract move $\Delta \rhd N \overset{\hat{\mu}}{\Longrightarrow}{}^{\mathsf{o}}_{w} \Delta' \rhd N_d$ such that $\Gamma' \rhd M_d \,\mathcal{R}^{(n+v-w)}\, \Delta' \rhd N_d$; the symmetric requirement, of matching a move from $\Delta \rhd N$ by a corresponding one from $\Gamma \rhd M$, is treated in an analogous fashion.

We proceed by induction on why $\Gamma \rhd M \;\; \mathcal{R}^n \;\; \Delta \rhd N$, there being three cases, (i), (ii) and (iii) above, to consider. In the first case the requirement comes from Proposition 3.4. We concentrate on case (ii), where we know M, N have the form $(M' \mid [P]_{\mathsf{o}})$, $(N' \mid [P]_{\mathsf{o}})$ respectively, where $\mathsf{o} \in \mathsf{O}$ and we know by induction that $\Gamma \rhd M' \mathcal{R}^n \Delta \rhd N'$. We now examine why $\Gamma \rhd M' | [P]_{\mathsf{o}} \xrightarrow{\mu}{}^{\mathsf{o}}_{v} \Gamma' \rhd M_d$, and to start let us assume that μ is the label $\mathsf{ext}(\mathsf{u}, k, \mathsf{p})$, where the reasoning is straightforward. This means, by definition, that M_d is $M \,|\, [P]_{\mathsf{o}}$, u, p are in O and $\Gamma \xrightarrow{(\mathsf{u},k,\mathsf{p})} \Gamma'$, which in turn implies $\Gamma \rhd M' \xrightarrow{\mathsf{ext}(\mathsf{u},k,\mathsf{p})}{}^{\mathsf{o}}_{v} \Gamma \rhd M'$; moreover incidently k and v must coincide, although this fact is not required here. By induction this can be matched by an action $\Delta \rhd N' \overset{\mathsf{ext}(\mathsf{u},k,\mathsf{p})}{\Longrightarrow}{}^{\mathsf{o}}_{w} \Delta' \rhd N''$ such that $(\Gamma \rhd M')\,\mathcal{R}^{(n+v-w)}\,(\Delta \rhd N'')$. This matching action can now be transformed into an action of the form $\Delta \rhd N' | [P]_{\mathsf{o}} \overset{\mathsf{ext}(\mathsf{u},k,\mathsf{p})}{\Longrightarrow}{}^{\mathsf{o}}_{w} \Delta' \rhd N'' | [P]_{\mathsf{o}}$ which is easily seen to be the required matching abstract move.

Having disposed of this simple case we now know that there is a derivation using the rules from Figure 6, Figure 7 of the underlying action

$$\Gamma \rhd M' \mid [P]_{\mathsf{o}} \overset{\lambda}{\mapsto} \Gamma' \rhd M_d, \tag{4.2}$$

where $v = (\Gamma'^{\mathsf{rec}} - \Gamma^{\mathsf{rec}})$, and λ is the more concrete version of the label μ. If M' is responsible for the concrete action (4.2), then a straightforward application of the induction hypothesis will provide the required corresponding move. Suppose instead that $[P]_{\mathsf{o}}$ is responsible, that is (4.2) takes the form

$$\Gamma \rhd M' \mid [P]_{\mathsf{o}} \overset{\lambda}{\mapsto} \Gamma' \rhd M' \mid [P']_{\mathsf{o}} \tag{4.3}$$

because $\Gamma \rhd [P]_{\mathsf{o}} \overset{\lambda}{\mapsto} \Gamma' \rhd [P']_{\mathsf{o}}$; here the reasoning needs to be more involved.

(a) First suppose this move is external, say an output with label λ being $(\mathsf{o}, (\tilde{r}{:}\tilde{\mathsf{R}})a!v, \mathsf{p})$ for some owner p. Because we are actually matching O-actions we know that this p is actually in O.

Applying Lemma 3.8 we know that Γ' has the form $\Gamma'', \tilde{r}{:}\tilde{\mathsf{R}}$, where $\Gamma \xrightarrow{(\mathsf{u},a,\mathsf{p})} \Gamma''$. The use of simple types means that $\Gamma^p(a) = 0$ and $\Gamma^u(a) = k$ for some k, and standard resource charging implies that this k is actually v. Thus we have the external move $\Gamma \rhd M' \xrightarrow{\mathsf{ext}(\mathsf{u},k,\mathsf{p})}{}^{\mathsf{o}}_{v} \Gamma'' \rhd M'$ and we know by induction this move can be matched by

some $\Delta \rhd N' \stackrel{\mathsf{ext}(\mathsf{u},k,\mathsf{p})}{\Longrightarrow}{}^{\mathsf{o}}_{w} \Delta'' \rhd N'$ such that $\Gamma'' \rhd M' \,\mathcal{R}^{(n+v-w)}\, \Delta'' \rhd N'$. This matching move actually has the form

$$\Delta \rhd N' \stackrel{\tau}{\Longrightarrow}{}^{\mathsf{o}}_{w_1} \Delta_1 \rhd N_1' \stackrel{\mathsf{ext}(\mathsf{u},k,\mathsf{p})}{\longrightarrow}{}^{\mathsf{o}}_{k} \Delta_2 \rhd N_1' \stackrel{\tau}{\Longrightarrow}{}^{\mathsf{o}}_{w_3} \Delta'' \rhd N' \tag{4.4}$$

with $w = w_1 + k + w_3$.

An application of part (iv) of Lemma 3.9 or Lemma 3.8 gives the move $\Delta_1 \rhd [P]_{\mathsf{o}} \stackrel{(\mathsf{u},(\tilde{r}:\tilde{\mathsf{R}})\alpha,\mathsf{p})}{\mapsto} \Delta_2, \tilde{r}{:}\tilde{\mathsf{R}} \rhd [P']_{\mathsf{o}}$ which can be combined with the pre- and post- τ moves in (4.4) to give $\Delta \rhd N' \,|\, [P]_{\mathsf{o}} \stackrel{\lambda}{\Longrightarrow}_w \Delta'', \tilde{r}{:}\tilde{\mathsf{R}} \rhd N' \,|\, [P']_{\mathsf{o}}$. This is the required matching move since we know $\Gamma'' \rhd M' \,\mathcal{R}^{(n+v-w)}\, \Delta'' \rhd N'$, from which $\Gamma'', \tilde{r}{:}\tilde{\mathsf{R}} \rhd M' \,|\, [P']_{\mathsf{o}}\ \mathcal{R}^{(n+v-w)}\ \Delta'', \tilde{r}{:}\tilde{\mathsf{R}} \rhd N' \,|\, [P']_{\mathsf{o}}$ follows by the remark (4.1) above and the definition of the family $\{\, \mathcal{R}^k \mid k \geq 0 \,\}$.

When the label λ in the move (4.3) above is an input the argument is very much the same but with an application of Lemma 3.9 in place of Lemma 3.8; it is therefore omitted.

(b) Now suppose the move from $[P]_{\mathsf{o}}$ we are examining is an internal move, taking the form $\Gamma \rhd [P]_{\mathsf{o}} \stackrel{\tau}{\mapsto} \Gamma' \rhd [P']_{\mathsf{o}}$. Here we apply Theorem 3.10 and Proposition 2.5, which tell us that there are in principle three possibilities, (i), (ii) or (iii). But an analysis of the proof will show that for processes of the form $[P]_{\mathsf{o}}$ case (i) is actually the only possibility. Here Γ' coincides with Γ, implying incidently that $v = 0$. As we know $\Delta \rhd [P]_{\mathsf{o}}$ is a configuration we also get $\Delta \rhd [P]_{\mathsf{o}} \stackrel{\tau}{\mapsto} \Delta \rhd [P']_{\mathsf{o}}$ and therefore that $\Delta \rhd N' \,|\, [P]_{\mathsf{o}} \stackrel{\tau}{\longrightarrow}_0 \Delta \rhd N' \,|\, [P']_{\mathsf{o}}$. It is easy to now check that this is the required matching move, since by definition $\Gamma \rhd M \,|\, [P]_{\mathsf{o}}\, \mathcal{R}^n\, \Delta \rhd N \,|\, [P']_{\mathsf{o}}$.

We are left with the possibility that the underlying action to be matched, (4.2) above, involves communication and therefore takes the form

$$\Gamma \rhd M' \,|\, [P]_{\mathsf{o}} \stackrel{\tau}{\mapsto} \Gamma' \rhd (\mathsf{new}\, \tilde{r}{:}\tilde{\mathsf{R}})(M'' \,|\, [P']_{\mathsf{o}})$$

There are two cases, depending on whether M' performs an input or an output. Let us consider the latter, the former being similar but slightly easier. So we have

$$\begin{aligned} &\Gamma \rhd M' \stackrel{\lambda}{\mapsto} \Gamma', \tilde{r}{:}\tilde{\mathsf{R}} \rhd M'' \\ &\Gamma \rhd [P]_{\mathsf{o}} \stackrel{\overline{\lambda}}{\mapsto} \Gamma', \tilde{r}{:}\tilde{\mathsf{R}} \rhd [P']_{\mathsf{o}} \end{aligned} \tag{4.5}$$

with $\lambda, \overline{\lambda}$ taking the forms $(\mathsf{u}, (\tilde{r}{:}\tilde{\mathsf{R}})a!v, \mathsf{o})$, $(\mathsf{u}, (\tilde{r}{:}\tilde{\mathsf{R}})a?v, \mathsf{o})$ respectively, for some owner u. By induction the first move, or rather its abstract version, can be matched because o is an owner in O, giving

$$\Delta \rhd N' \stackrel{\tau}{\mapsto}{}^* \Delta_1 \rhd N_1' \stackrel{(\mathsf{u}',(\tilde{r}:\tilde{\mathsf{R}}')a!v,\mathsf{o})}{\mapsto} \Delta_2, \tilde{r}{:}\tilde{\mathsf{R}}' \rhd N_2' \stackrel{\tau}{\mapsto}{}^* \Delta', \tilde{r}{:}\tilde{\mathsf{R}}' \rhd N'' \tag{4.6}$$

for some owner u', such that $(\Gamma', \tilde{r}{:}\tilde{\mathsf{R}} \rhd M'')\ \ \mathcal{R}^{(n+v-w)}\, (\Delta', \tilde{r}{:}\tilde{\mathsf{R}}' \rhd N'')$, where $w = (\Delta'^{\mathsf{rec}} - \Delta'^{\mathsf{rec}})$. Note that the type of the extruded names, $\tilde{\mathsf{R}}'$, may in general be different than the types at which they were extruded by M', and the owner u' may also be different, thereby a priori complicating matters when we try to combine this action with that from $[P]_{\mathsf{o}}$, in (4.5) above.

However an application of part (ii) of Lemma 3.8, gives $\Delta_1 \stackrel{(\mathsf{u}',a,\mathsf{o})}{\longrightarrow} \Delta_2$, and therefore from (4.5) and part (v) of Lemma 3.9 we get $\Delta_1 \rhd [P]_{\mathsf{o}} \stackrel{(\mathsf{u}',(\tilde{r}:\tilde{\mathsf{R}}')a?v,\mathsf{o})}{\mapsto} \Delta_2, \tilde{r}{:}\tilde{\mathsf{R}}' \rhd [P']_{\mathsf{o}}$. This

concrete move can now be combined with the concrete move (4.6) to give the required matching abstract move $\Delta \rhd N' \mid [P]_{\mathsf{o}} \stackrel{\tau}{\Longrightarrow}_w (\mathsf{new}\,\tilde{r}{:}\tilde{\mathsf{R}}')(N' \mid [P']_{\mathsf{o}})$. □

The attentive reader will have noticed that the restriction to *simple* types was necessary in order to be able to model the use of a resource by the observers using actions based on the transfer function $\Gamma \xrightarrow{\mathsf{ext}(\mathsf{u},k,\mathsf{p})} \Delta$, which records the transfer of k funds, the cost of using the resource, from the user to the provider. If we drop the restriction to *simple types*, then the effect of using a resource is more complicated; a certain amount will be debited to the user, while another amount, possibly negative, will be credited to the user. This can be accommodated by a more general transfer function $\Gamma \xrightarrow{\mathsf{ext}(\mathsf{u},(k_1,k_2),\mathsf{p})} \Delta$, leading in turn to a more general abstract arrow in part (d) of Definition 4.11. With this adjustment compositionality can also be established for arbitrary types.

This contextual results leads in a straightforward manner to establishing the second informal criteria, (b):

Theorem 4.23 (Soundness)**.** *For every $n \in \mathbb{N}$ and every set of owners O, $(\Gamma \rhd M) \sqsubseteq^n_{\mathsf{O}\mathrm{awgt}} (\Delta \rhd N)$ implies $(\Gamma \rhd M) \sqsubseteq^n_{\mathsf{O}:\mathrm{cxt}} (\Delta \rhd N)$.*

Proof. (Outline) It is sufficient to show that the family of relations $\{ \sqsubseteq^n_{\mathrm{awgt}} \mid n \in \mathbb{N} \}$ satisfies the three defining properties of the family of contextual equivalences. *Cost improving* follows by definition, at least up to structural induction, in view of Theorem 3.10, and the two preceding results establish O-contextuality. The final property, *Preservation of observations*, is also straightforward, since, for example, the ability to observe $a!$ from a configuration coincides with its ability to perform some output action on the resource a. □

The final criteria (c), *Completeness*, depends as usual on the ability to define contexts which capture the effect of each of the abstract O-actions described in Definition 4.11. We first make this precise.

We use two fresh cost-free resources, succ, fail to record the success or failure of tests, and a third req for housekeeping purposes. For any Γ we use Γ^t to denote the cost environment obtained by adding on these resources. Now let μ be an abstract action which uses the bound names $(\tilde{r})$. Then we say μ is definable relative to O if for every finite set of names F there exists a system T^F_μ using only the owners from O such that

(i) if $\mathrm{dom}(\Gamma^u) \subseteq F$ and $\Gamma \rhd M \xrightarrow{\mu}{}^{\mathsf{o}}_w \Delta, \tilde{r}{:}\tilde{\mathsf{R}} \rhd N$ then

$$\Gamma^t \rhd M \mid T^F_\mu \stackrel{\tau}{\Longrightarrow}{}^{\mathsf{o}}_w \Delta^t \rhd (\mathsf{new}\, r{:}\mathsf{R})(\mathsf{succ}!\langle \tilde{r} \rangle \mid N)$$

where $M' \Downarrow \mathsf{succ}!$ and $R \not\Downarrow \mathsf{fail}!$

(ii) conversely, $\Gamma^t \rhd M \mid T_\mu \stackrel{\tau}{\Longrightarrow}{}^{\mathsf{o}}_w \Delta^t \rhd M'$ where $M' \Downarrow \mathsf{succ}!$ and $M' \not\Downarrow \mathsf{fail}!$ implies $M' \equiv (\mathsf{new}\,\tilde{r}{:}\tilde{\mathsf{R}})(\mathsf{succ}!\langle \tilde{r} \rangle \mid N)$, where $\Gamma \rhd M \stackrel{\mu}{\Longrightarrow}{}^{\mathsf{o}}_w \Delta, \tilde{r}{:}\tilde{\mathsf{R}} \rhd N$, whenever $\mathrm{dom}(\Gamma^u) \subseteq F$.

Theorem 4.24 (Definability)**.** *All input, output and external actions are definable.*

Proof. (Outline) Let us look at two examples. First suppose that μ is the label $\mathsf{ext}(\mathsf{u}, k, \mathsf{p})$ where u and p are both in O; here $(\tilde{r})$ is empty and the set of names F plays no role. The definition of T^F_μ uses a variation on Example 2.7. We use

$$[\mathsf{fail}! \mid (\mathsf{new}\, r{:}\mathsf{R}_k)\mathsf{req}!\langle r \rangle.r!.\,\mathsf{stop}]_{\mathsf{u}} \;\mid\; [\mathsf{req}?(x)\,.y?.\mathsf{fail}?.\mathsf{succ}!]_{\mathsf{p}}$$

where R_k is the type $(k, 0)$. This ensures that whenever $(\Gamma \rhd M \mid T_\mu^F)$ evolves at cost w to a configuration $\mathcal{C}$ such that $\mathcal{C} \Downarrow \mathsf{succ}$ but $\mathcal{C} \not\Downarrow \mathsf{fail}$ then the newly generated resource r must have been used by u and provided by p. This is only possible if $\Gamma^t \rhd M$ can evolve to a configuration in which a transfer of k can be made from u to p; that is a configuration $\Gamma^{t'} \rhd M'$ such that $\Gamma^{t'} \xrightarrow{(\mathsf{u},k,\mathsf{p})} \Gamma^{t''}$. This in turns implies that we must have $\Gamma \rhd M \xRightarrow{\mathsf{ext}(\mathsf{u},k,\mathsf{p})}{}^{\mathsf{o}}_{w} \Delta \rhd N$ for some configuration $\Delta \rhd N$. Note the cost here is w because all of the resources used by the test T_μ^F are cost-free.

For the second example consider the abstract output action label $((r)a!r, \mathsf{p})$, where we know p is in O. Here we let T_μ^F be

$$[\mathsf{fail}! \mid a?(x)\,.\mathsf{if}\ x \in F\ \mathsf{then}\ \ \mathsf{stop}\ \ \mathsf{else}\ \mathsf{fail}?.\mathsf{succ}!]_{\mathsf{p}}$$

where $x \in F$ is an abbreviation for a series of tests deciding whether or not x is in the finite set of names F. Intuitively whenever this is used in a cost environment Γ satisfying $\mathrm{dom}(\Gamma^u) \subseteq F$ this test will fail only when x is instantiated by a fresh name.

Once more it is easy to say that the ability of $\Gamma^t \rhd M \mid T_m^F$ to evolve to a configuration $\mathcal{C}$ satisfying $\mathcal{C} \Downarrow \mathsf{succ}$ but $\mathcal{C} \not\Downarrow \mathsf{fail}$ coincides with the ability of $\Gamma \rhd M$ to do a weak concrete move labelled $(\mathsf{u}, (r : \mathsf{R})a!r, \mathsf{p})$ for some owner u and type R. Moreover the cost of this weak concrete action will be exactly the same as the evolution from $\Gamma^t \rhd M \mid T_m^F$, because the interactions with the test T_μ^F is free. □

Theorem 4.25 (Completeness)**.** *For every $n \in N$ and every set of owners O, $(\Gamma \rhd M) \sqsubseteq^n_{\mathsf{O}:\mathrm{cxt}} (\Delta \rhd N)$ implies $(\Gamma \rhd M) \sqsubseteq^n_{\mathsf{O}\mathrm{awgt}} (\Delta \rhd N)$.*

Proof. (Outline) It suffices to show that the family $\{ \sqsubseteq^n_{\mathsf{O}:\mathrm{cxt}} \mid\ n \in \mathbb{N} \}$ satisfies the conditions in Definition 4.16. Note that condition (i) is already established by Proposition 4.10. Now suppose $\Gamma \rhd M \sqsubseteq^n_{\mathsf{O}:\mathrm{cxt}} \Delta \rhd N$ and $\Gamma \rhd M \xrightarrow{\mu}{}^{\mathsf{o}}_{v} \Gamma' \rhd M'$. We have to find a matching move from $\Delta \rhd N$, which is relatively straightforward because of Theorem 4.24. As an example suppose μ is the output label $((r)a!r, \mathsf{p})$, and so Γ' has the structure $\Gamma'', r{:}\mathsf{R}$ for some R. Because of Compositionality we know $\Gamma^t \rhd M \mid T_\mu \sqsubseteq^n_{\mathrm{cxt}} \Delta^t \rhd N \mid T_\mu$. Using the first part of the Definability Theorem we know that, up to structural equivalence,

$$\Gamma^t \rhd M \mid T_\mu^F \longrightarrow_v^* \Gamma^{t''} \rhd (\mathsf{new}\, r{:}\mathsf{R})(\mathsf{succ}!\langle r\rangle \mid M').$$

Using the properties of the family $\{ \sqsubseteq^n_{\mathsf{O}:\mathrm{cxt}} \mid\ n \in \mathbb{N} \}$ this move must be matched by move

$$\Delta^t \rhd N \mid T_\mu^F \longrightarrow_w^* \Delta^{t''} \rhd N''$$

where

$$\Gamma^{t''} \rhd (\mathsf{new}\, r{:}\mathsf{R})(\mathsf{succ}!\langle r\rangle \mid M') \ \sqsubseteq^{(n+v-w)}_{\mathrm{cxt}}\ \Delta^{t''} \rhd N'' \tag{4.7}$$

Moreover we know $N'' \Downarrow \mathsf{succ}!$ and $N'' \not\Downarrow \mathsf{fail}!$ and so the Definability theorem tells us that $N'' \equiv (\mathsf{new}\, r{:}\mathsf{R}')(\mathsf{succ}!\langle r\rangle \mid N')$ where

$$\Delta \rhd N \xRightarrow{\mu}{}^{\mathsf{o}}_{w} \Delta'', r{:}\mathsf{R}' \rhd N'$$

This would be the required matching move, if we had

$$\Gamma'', r{:}\mathsf{R} \rhd M' \ \sqsubseteq^{(n+v-w)}_{\mathsf{O}:\mathrm{cxt}}\ \Delta'', r{:}\mathsf{R} \rhd N' \tag{4.8}$$

whereas (4.7) only gives us, up to structural equivalence,

$$\Gamma^{t''} \rhd (\mathsf{new}\, r{:}\mathsf{R})(\mathsf{succ}!\langle r\rangle \mid M') \ \sqsubseteq^{(n+v-w)}_{\mathrm{cxt}}\ \Delta^{t''} \rhd (\mathsf{new}\, r{:}\mathsf{R}')(\mathsf{succ}!\langle r\rangle \mid N') \tag{4.9}$$

However the so-called *Extrusion Lemma*, see Proposition 6.7 of [HR04] and Lemma 2.38 of [Hen07], can easily be adapted to Picost, to show that the required (4.8) does indeed follow from (4.9) □

5. Conclusion

In this paper we have developed a behavioural theory based on bisimulations for a version of the picalculus, Picost, in which

- resources have costs associated with them
- code runs under the financial responsibility of owners, or principals
- code can only be executed if the owner responsible for it can finance the available transactions.

The behavioural theory gives rise to a co-inductive proof methodology for comparing the costed behaviour of systems. We have demonstrated the usefulness of the methodology by treating some examples, and we have offered at least a preliminary justification for the theory in terms of contextual requirements, parametrised on sets of owners. We have provided some evidence that the most appropriate theory emerges when this set of observers is taken to be some single external observer, external to the owners funding the systems being investigated. In particular with this particular set of observers there is no need to consider the extra actions $\mathsf{ext}(\mathsf{u}, k, \mathsf{p})$ when establishing bisimulations.

The language could be extended in many ways without unduely affecting the underlying theory. Perhaps the most obvious extension would be the introduction of *ownership types*, to control which owners can use which resources; this would help in the modularisation of systems. One could also introduce a scoping mechanism for owners, limiting the range within systems of their financial responsibility. One effect of such extensions would be that owners would play a much more significant role in the (abstract) actions on which bisimulations are based. Such investigations we leave for future work.

The language could also be extended with mechanisms whereby processes could be aware of which owners are funding which resources, and more importantly base their behaviour on such knowledge. More ambitiously the semantics of the language could be generalised so that behaviour is now dependent on some *dynamic cost model*. There is considerable scope here for inventing more realistic cost models, whereby for example costs associated with producing/consuming resources could vary according to *market dynamics*. It is likely that a probabilistic setting would be most appropriate for developing such models.

The underlying theory of *weighted bisimulations* also deserves attention. For example it is not clear if the theory is decidable, even for finite-state systems. More generally it would be interesting to have techniques which would calculate the costs necessary to assign to actions in order to ensure the equivalence of two systems. There is already an extensive literature on *weighted automata* [DKV09] and decidability issues concerned with them, which may help in this regard.

Related work: The research reported in the current paper grew out of preliminary work reported in [HG08]. There a language π_{cost} was defined and also given a semantics relative to *cost environments*. But there are significant differences. At the language level the construct central to Picost, $[P]_{\mathsf{o}}$, is absent in π_{cost}; indeed in the latter there is no representation of owners being responsible for specific computations. The cost environments used are

Reader:

$$R_1 \Leftarrow \mathsf{goLib}?(\mathsf{name})\,.(\mathsf{new}r)\ R_2(r,\mathsf{name})$$
$$R_2(r,\mathsf{name}) \Leftarrow \mathsf{reqR}!\langle r,\mathsf{name}\rangle.R_3(r)$$
$$R_3(r) \Leftarrow r?(b)\,.R_4(b)$$
$$R_4(b) \Leftarrow \mathsf{goHome}!\langle b\rangle.R_1$$

Library:

$$L_1 \Leftarrow \mathsf{reqR}?(y,z)\,.L_2(y,z)$$
$$L_2(y,z) \Leftarrow L_3(y,z) \oplus (\mathsf{new}r)L_4(r,y,z)$$
$$L_3(y,z) \Leftarrow y!\langle\mathsf{book}(z)\rangle.L_1$$
$$L_4(r,y,z) \Leftarrow \mathsf{reqS}!\langle r,z\rangle.L_5(y)$$
$$L_5(y) \Leftarrow r?(b)\,.L_6(y,b)$$
$$L_6(y,b) \Leftarrow y!\langle b\rangle.L_1$$

Store:

$$S_1 \Leftarrow \mathsf{reqS}?(y,z)\,.S_2(y,z)$$
$$S_2(y,z) \Leftarrow y!\langle\mathsf{book}(z)\rangle.S_1$$

Figure 8: Notation for library code

also quite different; in π_{cost} funds are associated directly with resources, which complicates considerably the reduction semantics as the resource types need to be dynamic. Here all funds are retained by owners, which simplifies matters considerably, and this facilities the introduction of *charges* for resource usage and *benefits* for resource provision. Finally the behavioural theories are different. The concept of *weighted bisimulation* is considerably more flexible than the *cost bisimulations* of [HG08], as the latter simply compares the relative cost of performing each particular action.

Weighted bisimulations are a direct generalisation of the notion of *amortised bisimulations* from [KAK05]; these were originally defined for a version of CCS, [Mil89], in which only external actions have associated with them a cost. Nevertheless we believe that our generalisation is significant, at least in that it will make the concepts more generally applicable. However similar ideas have a long history in the field of *timed* process calculi; see for example [Tof94]. A good survey of the use of *amortisation* for timed processes can be found in [LV06].

Other resource-aware calculi have already appeared in the literature. A typical example is the variant of *mobile ambients* [CG00] from [BBDCS03] in which the resource in question is *space*, and the processes in the calculi have a *bounded capacity* to host incoming ambients. Another interesting example may be found in [Tel04], and related publications, which develops a version of the picalculus in which unused resources/channels may be garbage collected. Of particular interest to us is the general theory of *resource-based* computation being developed in [CP07], and related publications. In future work we hope to adapt their resource-based modal logic to Picost.

Appendix A. Some witness bisimulations

$$
\begin{aligned}
N_1 &\Leftarrow (\mathsf{new\ reqR}{:}\mathsf{R}^c_r)([R_1]_{\mathsf{pub}} \mid (\mathsf{new\ reqS}{:}\mathsf{R}^c_s)([L_1]_{\mathsf{lib}} \mid [S_1]_{\mathsf{lib}}))\\
N_2(n) &\Leftarrow (\mathsf{newreqR}, r)([R_2(r,n)]_{\mathsf{pub}} \mid (\mathsf{newreqS})([L_1]_{\mathsf{lib}} \mid [S_1]_{\mathsf{lib}}))\\
N_3(n) &\Leftarrow (\mathsf{newreqR}, r)([R_3(r)]_{\mathsf{pub}} \mid (\mathsf{newreqS})([L_2(r,n)]_{\mathsf{lib}} \mid [S_1]_{\mathsf{lib}}))\\
N_{41}(n) &\Leftarrow (\mathsf{newreqR}, r)([R_3(r)]_{\mathsf{pub}} \mid (\mathsf{newreqS})([L_3(r,n)]_{\mathsf{lib}} \mid [S_1]_{\mathsf{lib}}))\\
N_{51}(b) &\Leftarrow (\mathsf{newreqR}, r)([R_4(b)]_{\mathsf{pub}} \mid (\mathsf{newreqS})([L_1]_{\mathsf{lib}} \mid [S_1]_{\mathsf{lib}}))\\
N_{42}(n) &\Leftarrow (\mathsf{newreqR}, r, r')([R_3(r)]_{\mathsf{pub}} \mid (\mathsf{newreqS})([L_4(r,r',n)]_{\mathsf{lib}} \mid [S_1]_{\mathsf{lib}}))\\
N_{52}(n) &\Leftarrow (\mathsf{newreqR}, r, r')([R_3(r)]_{\mathsf{pub}} \mid (\mathsf{newreqS})([L_5(r,r')]_{\mathsf{lib}} \mid [S_2(r',n)]_{\mathsf{lib}}))\\
N_{53}(b) &\Leftarrow (\mathsf{newreqR}, r, r')([R_3(r)]_{\mathsf{pub}} \mid (\mathsf{newreqS})([L_6(r,b')]_{\mathsf{lib}} \mid [S_1]_{\mathsf{lib}}))
\end{aligned}
$$

Figure 9: Library systems

A.1. **The library.** Here we revisit the example on running a library, discussed in Example 2.6 and Example 3.11, and prove

$$(\Gamma_{\mathsf{central}} \rhd \mathsf{Sys}_{\mathsf{central}}) \sqsubseteq^2_{\mathrm{wgt}} (\Gamma_{\mathsf{local}} \rhd \mathsf{Sys}_{\mathsf{local}}) \tag{A.1}$$

by exhibiting a witness bisimulation. For convenience we work up to *structural equivalence* and modulo β-moves; essentially these are moves which have no effect on the overall behaviour of systems; see [Hen07, GS96] for details. In Picost these include the actions generated by the rules (L-EXPORT), (L-UNWIND), (L-SPLIT), (L-MATCH), (L-MISMATCH). Let us assume a set of book names BN, ranged over by n and a set of books BK, ranged over by b.

Let us write $\Gamma \sim \Delta$ whenever

(a) Γ has the form Γ_{dyn}, $\mathsf{goLib}{:}\langle 0,5\rangle$, $\mathsf{goHome}{:}\langle 0,5\rangle$, $\mathsf{reqR}{:}\langle 0,1\rangle$, $\mathsf{reqS}{:}\langle 0,1\rangle$ for some basic environment Γ_{dyn}

(b) Δ has the form Δ_{dyn}, $\mathsf{goLib}{:}\langle 0,1\rangle$, $\mathsf{goHome}{:}\langle 0,1\rangle$, $\mathsf{reqR}{:}\langle 0,3\rangle$, $\mathsf{reqS}{:}\langle 0,5\rangle$ where again Δ_{dyn} is some basic environment.

(c) $\mathrm{dom}(\Gamma^o) = \mathrm{dom}(\Delta^o) = \{\mathsf{pub}, \mathsf{lib}\}$, with $\Gamma^o(\alpha) = \Delta^o(\alpha) = \infty$, for every α in its domain.

So effectively Γ must be like $\Gamma_{\mathsf{central}}$ with perhaps a different record filed Γ^{rec}, and Δ must be like Γ_{local}. Our witness bisimulation will contain pairs of the form

$$\Gamma \rhd N \leftrightarrow \Delta \rhd M \qquad\qquad \text{where } \Gamma \sim \Delta$$

The allowed forms of N are described in Figure 9, where for convenience we have omitted the explicit occurrence of the local types R^c_r, R^c_s after the first line. These in turn use notation given in Figure 8 for the various processes. The allowed forms for M are identical except for the use of the local types R^l_r, R^l_s in place of R^c_r, R^c_s.

Let the family of relations over configurations $\{\,\mathcal{R}^k \mid k \in \mathbb{N}\,\}$ be determined by the following constraints, where we assume in each clause that $\Gamma \sim \Delta$:

$$\begin{aligned}
\Gamma \rhd N_1 \;\mathcal{R}^k\; \Delta \rhd M_1 &\qquad \text{whenever } k \geq 2\\
\Gamma \rhd N_2(n) \;\mathcal{R}^k\; \Delta \rhd M_2(n) &\qquad \text{whenever } k \geq 6,\ n \in \mathsf{BN}\\
\Gamma \rhd N_i(n) \;\mathcal{R}^k\; \Delta \rhd M_i(n) &\qquad \text{whenever } k \geq 4,\ n \in \mathsf{BN}, i = 3, 41, 51, 42\\
\Gamma \rhd N_i(n) \;\mathcal{R}^k\; \Delta \rhd M_i(n) &\qquad \text{whenever } k \geq 4,\ n \in \mathsf{BN}, i = 3, 41, 42\\
\Gamma \rhd N_i(b) \;\mathcal{R}^k\; \Delta \rhd M_i(b) &\qquad \text{whenever } k \geq 0,\ b \in \mathsf{BK}, i = 51, 52, 53
\end{aligned}$$

It is fairly straightforward, although tedious, to prove that $\{\,\mathcal{R}^k \mid k \in \mathbb{N}\,\}$ satisfies the requirements of being a weak bisimulation in the wLTS of Section 3.3, up to structural equivalence and β-moves. This is facilitated by the fact that the code in each component of the pairs is identical.

Note that the configuration $\Gamma_{\mathsf{central}} \rhd \mathsf{Sys}_{\mathsf{central}}$ β-reduces to a configuration of the form $\Gamma \rhd N_1$ and $(\Gamma_{\mathsf{local}} \rhd \mathsf{Sys}_{\mathsf{local}})$ β-reduces to one of the form $\Delta \rhd M_1$, where $\Gamma \sim \Delta$, and thus (A.1) above follows.

A.2. **The publisher.** Here we revisit the publishing example developed in Example 2.8, Example 2.9 and Example 4.12; by exhibiting a witness bisimulation, again up to structural equivalence and β-moves, we show that

$$(\Gamma_{327} \rhd \mathrm{PA}_K) \sqsubseteq^0_{\mathsf{ewgt}} (\Gamma_{327} \rhd \mathrm{PA}) \tag{A.2}$$

subject to minor constraints on Γ; these constraints allow $\Gamma^o(\mathsf{p})$ to be finite. The systems PA and PA_K, in addition to cost-free communications,

- use resource news; in the definition of the cost environment from Example 2.8 this is recorded as a loss of 3, the cost of using news. In the abstract wLTS we are using this loss is paid for by the funds in $\Gamma^o_{327}(\mathsf{p})$, while it costs nothing to provide
- provide resource $\mathsf{publish}$; in the cost environment this is recorded as a gain of 6, namely the difference between providing it 7 and using it 1. Also this gain is added to the funds of $\Gamma^o_{327}(\mathsf{p})$.

There are also internal communications which have costs associated with them, namely the use and provision of adv; again this is recorded as a loss of 2 which must be funded by $\Gamma^o_{327}(\mathsf{p})$.

In order to describe the witness bisimulation we use the code abbreviations in Figure 10 and the system definitions in Figure 11. All environments we use have the form $\Gamma_{dyn}, \mathsf{news}{:}\mathsf{R}_n, \mathsf{publish}{:}\mathsf{R}_p$, and in order to fund the advertising we assume $\Gamma^o(\mathsf{a}) = \infty$. In the witness bisimulation $\{\,\mathcal{R}^k \mid k \in \mathbb{N}\,\}$ all $\mathcal{R}^k$ are identical and this unique relation $\mathcal{R}$ is characterised by the following constraints:

$$\begin{aligned}
\Gamma^{\mathsf{e}} \rhd PA_{K1} \;\mathcal{R}\; \Delta^{\mathsf{e}} \rhd PA_1 &\qquad 5 \leq \Gamma^o(\mathsf{p}),\ 5 \leq \Delta^o(\mathsf{p})\\
\Gamma^{\mathsf{e}} \rhd PA_{K2}(r) \;\mathcal{R}\; \Delta^{\mathsf{e}} \rhd PA_2(r) &\qquad 2 \leq \Gamma^o(\mathsf{p}),\ 2 \leq \Delta^o(\mathsf{p}),\ r \in \mathsf{Chan}\\
\Gamma^{\mathsf{e}} \rhd PA_{K3}(r) \;\mathcal{R}\; \Delta^{\mathsf{e}} \rhd PA_3(r) &\qquad r \in \mathsf{Chan}\\
\Gamma^{\mathsf{e}} \rhd PA_{Ki}(n) \;\mathcal{R}\; \Delta^{\mathsf{e}} \rhd PA_i(n) &\qquad 4 \leq i \leq 6,\ n \in \mathsf{News}\\
\Gamma^{\mathsf{e}} \rhd PA_{K7}(n) \;\mathcal{R}\; \Delta^{\mathsf{e}} \rhd PA_6(n) &\qquad n \in \mathsf{News}
\end{aligned}$$

$$
\begin{array}{lrl}
\text{Publisher:} & P_1(r_1) & \Leftarrow \mathsf{news}!\langle r_1\rangle.(\mathsf{new}r_2)P_2(r_1,r_2) \\
& P_2(r_1,r_2) & \Leftarrow \mathsf{adv}!\langle r_2\rangle.P_3(r_1,r_2) \\
& P_3(r_1,r_2) & \Leftarrow r_1?(n)\,.P_4(n,r_2) \\
& P_4(n,r_2) & \Leftarrow r_2?(d)\,.P_5(n,d) \\
& P_5(n,d) & \Leftarrow \mathsf{publish}?(z)\,.P_6(n,d,z) \\
& P_6(n,d,z) & \Leftarrow z!\langle n,d\rangle.(\mathsf{new}r_1)P_1(r_1) \\
\\
\text{Advertiser:} & A_1 & \Leftarrow \mathsf{adv}?(r)\,.(\mathsf{new}d)A_2(r,d) \\
& A_2 & \Leftarrow r!\langle d\rangle.A_1 \\
\\
\text{Publisher with kickback:} & P_{K1}(r_1) & \Leftarrow \mathsf{news}!\langle r_1\rangle(\mathsf{new}r_2,k)P_{K2}(r_1,r_2,k) \\
& P_{K2}(r_1,r_2,k) & \Leftarrow \mathsf{adv}!\langle k,r_2\rangle.P_{K3}(r_1,r_2,k) \\
& P_{K3}(r_1,r_2,k) & \Leftarrow r_1?(n)\,.P_{K4}(n,r_2,k) \\
& P_{K4}(n,r_2,k) & \Leftarrow r_2?(d)\,.P_{K5}(n,d,k) \\
& P_{K5}(n,d,k) & \Leftarrow \mathsf{publish}?(z)\,.P_{K6}(n,d,k,z) \\
& P_{K6}(n,d,k,z) & \Leftarrow k?.P_{K7}(n,d,z) \\
& P_{K7}(n,d,z) & \Leftarrow z!\langle n,d\rangle.(\mathsf{new}r_1)P_{K1}(r_1) \\
\\
\text{Advertiser with kickback:} & A_{K1} & \Leftarrow \mathsf{adv}?(k,r)\,.(\mathsf{new}d)A_{K2}(k,r,d) \\
& A_{K2}(k,r,d) & \Leftarrow r!\langle d\rangle.(A_{K1} \mid k!)
\end{array}
$$

Figure 10: Notation for publisher code

Here we use News to denote some set of news stories.

It is straightforward to show that this is indeed a weak amortised bisimulation in the abstract wLTS relative to the single external observer e. Since $\Gamma_{327} \rhd \mathrm{PA}_K$ β-reduces to $\Gamma_{327} \rhd \mathrm{PA}_{K1}$ and $\Gamma_{327} \rhd \mathrm{PA}$ β-reduces to $\Gamma_{327} \rhd \mathrm{PA}_1$, and $\Gamma_{327} \rhd \mathrm{PA}_{K1}\ \mathcal{R}\ \Gamma_{327} \rhd \mathrm{PA}_1$, the required (A.2) above follows.

Acknowledgments

The author would like to thank the referees for their very useful comments.

References

[BBDCS03] Franco Barbanera, Michele Bugliesi, Mariangiola Dezani-Ciancaglini, and Vladimiro Sassone. A calculus of bounded capacities. In Vijay A. Saraswat, editor, *ASIAN*, volume 2896 of *Lecture Notes in Computer Science*, pages 205–223. Springer, 2003.

[CG00] Luca Cardelli and Andrew D. Gordon. Mobile ambients. *Theor. Comput. Sci.*, 240(1):177–213, 2000.

Standard publisher:

$$\begin{aligned}
\mathrm{PA}_1 &\Leftarrow (\mathsf{newadv}, r_1)([P_1(r_1)]_\mathsf{p} \mid [A_1]_\mathsf{a}) \\
\mathrm{PA}_2(r_1) &\Leftarrow (\mathsf{newadv}, r_2)([P_2(r_1, r_2)]_\mathsf{p} \mid [A_1]_\mathsf{a}) \\
\mathrm{PA}_3(r_1) &\Leftarrow (\mathsf{newadv}, r_2, d)([P_3(r_1, r_2)]_\mathsf{p} \mid [A_2(r_2, d)]_\mathsf{a}) \\
\mathrm{PA}_4(n) &\Leftarrow (\mathsf{newadv}, r_2, d)([P_4(n, r_2)]_\mathsf{p} \mid [A_2(r_2, d)]_\mathsf{a}) \\
\mathrm{PA}_5(n) &\Leftarrow (\mathsf{newadv}, d)([P_5(n, d)]_\mathsf{p} \mid [A_1]_\mathsf{a}) \\
\mathrm{PA}_6(n) &\Leftarrow (\mathsf{newadv}, d)([P_6(n, d, r)]_\mathsf{p} \mid [A_{K1}]_\mathsf{a})
\end{aligned}$$

Publisher with kickback:

$$\begin{aligned}
\mathrm{PA}_{K1} &\Leftarrow (\mathsf{newadv}, r_1)([P_{K1}(r_1)]_\mathsf{p} \mid [A_{K1}]_\mathsf{a}) \\
\mathrm{PA}_{K2}(r_1) &\Leftarrow (\mathsf{newadv}, r_2, k)([P_{K2}(r_1, r_2, k)]_\mathsf{p} \mid [A_{K1}]_\mathsf{a}) \\
\mathrm{PA}_{K3}(r_1) &\Leftarrow (\mathsf{newadv}, k, r_2, d)([P_{K3}(r_1, r_2, k)]_\mathsf{p} \mid [A_{K2}(k, r_2, d)]_\mathsf{a}) \\
\mathrm{PA}_{K4}(n) &\Leftarrow (\mathsf{newadv}, k, r_2, d)([P_{K4}(n, r_2, k)]_\mathsf{p} \mid [A_{K2}(k, r_2, d)]_\mathsf{a}) \\
\mathrm{PA}_{K5}(n) &\Leftarrow (\mathsf{newadv}, k, r_2, d)([P_{K5}(n, d, k)]_\mathsf{p} \mid [A_{K1} \mid k!]_\mathsf{a}) \\
\mathrm{PA}_{K6}(n) &\Leftarrow (\mathsf{newadv}, k, r_2, d)([P_{K6}(n, d, r, k)]_\mathsf{p} \mid [A_{K1} \mid k!]_\mathsf{a}) \\
\mathrm{PA}_{K7}(n) &\Leftarrow (\mathsf{newadv}, d)([P_{K7}(n, d, r)]_\mathsf{p} \mid [A_{K1}]_\mathsf{a})
\end{aligned}$$

Figure 11: Publishing systems

[CGP08] Giuseppe Castagna, Nils Gesbert, and Luca Padovani. A theory of contracts for web services. In *POPL '08, 35th ACM Symposium on Principles of Programming Languages*, Jan 2008.

[CP07] Matthew Collinson and David Pym. Algebra and logic for resource-based systems modelling. Technical report, Hewlett-Packard Laboratories, 2007. Submitted for Publication.

[DKV09] Manfred Droste, Werner Kuich, and Heiko Vogler, editors. *Handbook of Weighted Automata*. EATCS Monographs in Theoretical Computer Science. Springer-Verlag, 2009.

[GS96] Jan Friso Groote and M. P. A. Sellink. Confluence for process verification. *Theor. Comput. Sci.*, 170(1-2):47–81, 1996.

[Hen07] Matthew Hennessy. *A distributed picalculus*. Cambridge University Press, 2007.

[HG08] Matthew Hennessy and Manish Gaur. Counting the cost in the picalculus (extended abstract). *Electr. Notes Theor. Comput. Sci.*, 2008. To appear. Preliminary version presented at *First Interaction and Concurrency Expierience (ICE'08)*, Reykjavik, July 2008.

[HR04] Matthew Hennessy and Julian Rathke. Typed behavioural equivalences for processes in the presence of subtyping. *Mathematical Structures in Computer Science*, 14:651–684, 2004.

[HT92] Kohei Honda and Mario Tokoro. On asynchronous communication semantics. In P. Wegner M. Tokoro, O. Nierstrasz, editor, *Proceedings of the ECOOP '91 Workshop on Object-Based Concurrent Computing*, volume 612 of *LNCS 612*. Springer-Verlag, 1992.

[KAK05] Astrid Kiehn and Sak Arun-Kumar. Amortised bisimulations. In Farn Wang, editor, *FORTE*, volume 3731 of *Lecture Notes in Computer Science*, pages 320–334. Springer, 2005.

[LV06] Gerald Lüttgen and Walter Vogler. Bisimulation on speed: a unified approach. *Theor. Comput. Sci.*, 360(1):209–227, 2006.

[Mil89] Robin Milner. *Communication and Concurrency*. Prentice-Hall, 1989.

[Mil99] Robin Milner. *Comunicating and mobile systems: the π-calculus*. Cambridge University Press, 1999.

[SW01] Davide Sangiorgi and David Walker. *The π-calculus: A Theory of Mobile Processes*. Cambridge University Press, 2001.

[Tel04] David Teller. Recollecting resources in the pi-calculus. In *Proceedings of IFIP TCS 2004*, pages 605–618. Kluwer Academic Publishing, 2004.

[Tof94] Chris M. N. Tofts. Processes with probablities, priority and time. *Formal Asp. Comput.*, 6(5):536–564, 1994.