Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in Theoretical Computer Science 229 (2009) 117–129

Electronic Notes in Theoretical Computer Science

www.elsevier.com/locate/entcs

Counting the Cost in the Picalculus (Extended Abstract)

Matthew Hennessy [1],[2]

Department of Computer Science
Trinity College Dublin
Dublin, Ireland

Manish Gaur[3]

Department of Informatics
University of Sussex
Brighton, UK

Abstract

We design a new variation on the picalculus, π_{cost}, in which the use of channels or resources must be paid for. Processes operate relative to a *cost environment*, and communications can only happen if principals have provided sufficient funds for the channels associated with the communications.
We define a bisimulation-based behavioural preorder in which two processes are related if, intuitively, they exhibit the same behaviour but one may be more efficient than the other. We justify our choice of preorder by proving that it is characterised by three intuitive properties which behavioural preorders should satisfy in a framework in which the use of resources must be funded.

Keywords: picalculus, resources, cost bisimulations

1 Introduction

The *picalculus* [20] is a basic abstract formal language for describing communicating processes and has a very developed behavioural theory [28], expressed as an equivalence relation between process descriptions; $P \approx Q$ signifies that, although P and Q may be intentionally very different they offer essentially the same behaviour to users.

[1] This work is a part of the doctoral studies of the second author and is supported by Commonwealth Scholarship Commission UK (Ref: INCS-2005-145). The support of SFI, Ireland is also acknowledged.

[2] Email: `Matthew.Hennessy@cs.tcd.ie`

[3] Email: `M.H.C.Gaur@sussex.ac.uk`

doi:10.1016/j.entcs.2009.06.042

The basic language and its related theory has been extended in myriad ways in order to incorporate many different aspects of concurrent behaviour [1,26,8]. In one family of extensions the judgements of the behavioural theory take the form

$$\Gamma \models P \approx Q \tag{1}$$

where Γ represents some aspect of the infrastructure in which the processes P, Q operate. The primary example, initiated in [25], is when Γ is a type environment describing the type of the communicating channels used by P, Q. But in [9,23] it represents the state of the underlying network, recording for example the current connectivity between the sites at which processes execute, or the failures which have occurred.

In this short paper we show how this framework, in particular the version from [13,11], can also be adapted to develop a theory in which there is a cost associated with the resources used in a computation. Here Γ will represent a *cost environment*, which could record for example the cost of using particular channels or resources, the current funds available to the various principals involved, and could also keep a tally of the total funds which have been expended so far. Indeed if the latter is included in the notion of a *cost environment* then (1) could be adapted to judgements of the form

$$\Gamma \models P \preccurlyeq Q \tag{2}$$

meaning informally that, relative to the cost environment Γ, processes P and Q offer essentially the same behaviour to users, but that Q is as least as efficient as P, and possibly more efficient.

We envisage two immediate applications for these ideas. The first is web services, [2]. In [17,6] a basic theory of contracts for web services is introduced, based on a variation of CCS, [19]. Our use of *cost environments* could immediately be applied here, and indeed we intend to pursue this line of work in future publications. The second is in the development of a more realistic theory of networked processes. Communication across a network is not instantaneous; by introducing some representation of routers into the process description language, we can associate as the cost of a communication the number of routers through which the message has to travel. This is pursued in [10].

The current paper seeks to lay the foundations for a theory of *costed process behaviour*. In Section 2 we describe a very simple variation on the (asynchronous) *picalculus*, which we call π_{cost}, in which channels are viewed as resources, as in [6], but which can only be used if sufficient funds are available. The reduction semantics is relative to a *cost environment*, so that the judgements are of the form

$$\Gamma_1 \triangleright P_1 \longrightarrow \Gamma_2 \triangleright P_2$$

We refer to the pairs $(\Gamma_i \triangleright P_i)$ as *systems*. The rules governing the judgements are minor variations on those used in the standard reduction semantics for the

$P, Q ::=$		**Process terms**
	$u?(x)\,.P$	Provide resource u
	$u!\langle v\rangle.P$	Use resource u
	$\mathsf{subscribe}(o, u, c).P$	Subscribe to resource u
	$\mathsf{if}\ u = v\ \mathsf{then}\ P\ \mathsf{else}\ Q$	Matching
	$(\mathsf{new}\, a\!:\!\mathsf{R})\ P$	Resource creation
	$P \mid Q$	Concurrency
	$*P$	Repetition
	stop	Termination
	$\mathsf{del}(a, v)$	Asynchronous message delivery

Fig. 1. Syntax of π_{cost}

(asynchronous) *picalculus*; it turns out that the rules only depend on three high-level operations on *cost environments*. However we also give a concrete instantiation of *cost environment* which supports these operations.

In Section 4 we define a labelled transition system for π_{cost}, and use the resulting actions to define the relation referred to above, (2), using a (minor) variation on the standard definition of (asynchronous) bisimulation equivalence, [14,4,28]. We claim that this does indeed form the basis for an adequate theory of *costed process behaviour*. To support this claim we offer one theorem, Theorem 4.4, which says that this relation is completely determined by three natural properties of behavioural relations between systems. These properties are outlined in Section 3, and the main ingredient is the manner in which processes are observed, in particular who pays the cost of performing observations. The paper ends with some remarks on related and future work.

2 The language π_{cost}

We assume a set of *channel* or *resource* names Chan, ranged over by $a, b, c, \ldots$ whose use requires some cost. As already stated we have two examples in mind. The first is where these names actually represent web services, as in [6], and the second is where they represent the transmission of data through routers in a distributed network. We also assume a set of *principals* or *owners* Own, ranged over by o, who register for these resources and pay for their use. The syntax of π_{cost} is then given in Figure 1, and is essentially a very minor extension to the *picalculus*; the meta-variables u, v range over *identifiers*, which are either resource names $a \in \mathsf{Chan}$, or variables x from a distinct set Var. We employ the standard abbreviations associated with the *picalculus*, and associated terminology.

Since resource usage incurs a cost, the execution of processes is now relative to a *cost environment* Γ; this records which owners are registered for which resources, and both the costs required to use resources, and the effect of actually using them. Thus judgements of the reduction semantics take the form

$$\Gamma_1 \triangleright P_1 \longrightarrow \Gamma_2 \triangleright P_2$$

where P_i are processes, that is closed terms from π_{cost}, and Γ_i represent *cost environments.*

There are many possibilities for *cost environments*, and we will provide a particular instance shortly. But no matter how they are defined, we need to be able to define at least three operations on them:

- *resource charging*: $\Gamma_1 \xrightarrow{a} \Gamma_2$ means that relative to Γ_1 sufficient funds are available for the use of resource a, and if it is used, the consumption of appropriate funds is recorded in the transformation from Γ_1 to Γ_2.
- *resource subscription*: $\Gamma_1 \xrightarrow{\mathsf{sub}(o,a,c)} \Gamma_2$ records the effect of allowing owner o to subscribe, with the funds c, to the resource a.
- *resource registration*: $\Gamma, a\!:\!\mathsf{R}$ records the result of extending Γ with a *new* resource named a, with the information contained in the *type* R. In this paper these types will take the form $\langle \mathsf{R}_c, \mathsf{R}_s \rangle$, where R_c is a usage cost, and and R_s records the amount of funds which owners have allocated to the resource.

Relative to these operations, the reduction semantics for π_{cost} is then defined as the least relation which satisfies the rules in Figure 2. This uses the standard *structural equivalence* between process terms of the *picalculus* which is recalled in Figure 3.

The idea behind this semantics is that $a!\langle v \rangle.P$ is a request to use the service a with parameter v; so with rule (R-OUT) it spawns an *atom* $\mathsf{del}(a, v)$, which is implicitly delivered through the network to the site of the resource a. In (R-COMM) this request is satisfied, at least if the *cost environment* allows it, that is $\Gamma_1 \xrightarrow{a} \Gamma_2$. We have not yet actually specified this relation, but one would also expect it to record the cost of this request. Most of the remaining rules are standard from the *picalculus*, but the novel (R-SUBSCRIBE) allows an owner to subscribe to a channel, that is allocate funds for the use of the channel.

But the final rule (R-NEW) is non-standard. In $\Gamma_1 \triangleright (\mathsf{new}\, b\!:\!\mathsf{R})\, P$, the process P may evolve by using the internal resource b. In general this requires the expenditure of funds, and therefore will effect funds available for any subsequent use of b. This is reflected in the change of the type, from R to R'; the possible values for R' are deduced by examining the possible evolution of P relative to the extended cost environment $\Gamma_1, b\!:\!\mathsf{R}$.

For the remainder of the paper we take a *cost environment* Γ to consist of a 4-tuple $\langle \Gamma^c, \Gamma^o, \Gamma^s, \Gamma^r \rangle$ where

- $\Gamma^c : \mathsf{Chan} \rightharpoonup N^\infty$
  $\Gamma^c(a)$ records the cost of using the resource a; since Γ^c is a partial function it also implicitly records the valid resources known to Γ, namely $\mathrm{dom}(\Gamma^c)$.

(r-out)
$$\Gamma \triangleright a!\langle b\rangle.P \longrightarrow \Gamma \triangleright \mathsf{del}(a,b)\,|P$$

(r-comm)
$$\frac{\Gamma_1 \xrightarrow{a} \Gamma_2}{\Gamma_1 \triangleright \mathsf{del}(a,b)\,|a?(x)\,.P \longrightarrow \Gamma_2 \triangleright P\{\!\!|{}^b\!/\!{}_x|\!\!\}}$$

(r-subscribe)
$$\frac{\Gamma_1 \xrightarrow{\mathsf{sub}(o,a,c)} \Gamma_2}{\Gamma_1 \triangleright \mathsf{subscribe}(o,a,c).P \longrightarrow \Gamma_2 \triangleright P}$$

(r-match)
$$\Gamma \triangleright \mathsf{if}\ a = a\ \mathsf{then}\ P\ \mathsf{else}\ Q \longrightarrow \Gamma \triangleright P$$

(r-mismatch)
$$\Gamma \triangleright \mathsf{if}\ a = b\ \mathsf{then}\ P\ \mathsf{else}\ Q \longrightarrow \Gamma \triangleright Q\ \ a \neq b$$

(r-struct)
$$\frac{P \equiv P',\ \Gamma_1 \triangleright P \longrightarrow \Gamma_2 \triangleright Q,\ Q \equiv Q'}{\Gamma_1 \triangleright P' \longrightarrow \Gamma_2 \triangleright Q'}$$

(r-cntx)
$$\Gamma_1 \triangleright P \longrightarrow \Gamma_2 \triangleright Q$$
$$\Gamma_1 \triangleright P\,|\,R \longrightarrow \Gamma_2 \triangleright Q\,|\,R$$
$$\Gamma_1 \triangleright R\,|\,P \longrightarrow \Gamma_2 \triangleright R\,|\,Q$$

(r-new)
$$\frac{\Gamma_1, b{:}\mathsf{R} \triangleright P \longrightarrow \Gamma_2, b{:}\mathsf{R}' \triangleright Q}{\Gamma_1 \triangleright (\mathsf{new}\,b{:}\mathsf{R})\,P \longrightarrow \Gamma_2 \triangleright (\mathsf{new}\,b{:}\mathsf{R}')\,Q}$$

Fig. 2. Reduction semantics

(s-scope – extrusion)	$(\mathsf{new}\,a{:}\mathsf{R})(P\,|\,Q) \equiv P\ \,|\ \,(\mathsf{new}\,a{:}\mathsf{R})\,Q$	if $a \notin \mathsf{fn}(P)$
(s-monoid – com)	$P\,|\,Q \equiv Q\,|\,P$	
(s-monoid – assoc)	$(P\,|\,Q)\,|\,R \equiv P\,|\,(Q\,|\,R)$	
(s-monoid – id)	$P\,|\,\mathsf{stop} \equiv P$	
(s-new – flip)	$(\mathsf{new}\,a{:}\mathsf{R})\,(\mathsf{new}\,b{:}\mathsf{S})\,P \equiv (\mathsf{new}\,b{:}\mathsf{S})\,(\mathsf{new}\,a{:}\mathsf{R})\,P$	if $a \neq b$
(s-new)	$(\mathsf{new}\,a{:}\mathsf{R})\,P \equiv P$	if $a \notin \mathsf{fn}(P)$
(s-rec)	$*P \equiv P\,|\,*P$	

Fig. 3. Structural equivalence of π-Cost

- $\Gamma^o : \mathsf{Own} \longrightarrow N^\infty$
  For an owner $o \in \mathsf{Own}$, $\Gamma^o(o)$ records the (unsubscribed) funds which o has in the system. These are available for o to allocate to particular resources, via the $\mathsf{subscribe}(o,a,c)$ command.
- $\Gamma^s : \mathsf{Chan} \rightharpoonup (\mathsf{Own} \rightharpoonup N^\infty)$
  $\Gamma^s(a)$ records the subscriptions that owners have on resource a; since $\Gamma^s(a)$ is a partial function it also implicitly records the owners registered to use a, namely

dom($\Gamma^s(a)$). We also use $|\Gamma^s(a)|$ to denote $\sum\{\,\Gamma^s(a)(o) \mid o \in \mathrm{dom}(\Gamma^s(a))\,\}$, the entire funds available for the use of the resource a.

- $\Gamma^r : N^\infty$
  This is a record of the cost which has already been expended by the system.

The required operations are defined as follows:

- *resource charging*: informally $\Gamma_1 \xrightarrow{a} \Gamma_2$ if there are sufficient funds subscribed to a in Γ_1 to cover the costs of using it, and Γ_2 records their consumption. Formally it holds when
  · $|\Gamma^s_2(a)| = |\Gamma^s_1(a)| - \Gamma^c_1(a)$
  · $\Gamma^r_2 = \Gamma^r_1 + \Gamma^c_1(a)$
  · $\Gamma^c_2 = \Gamma^c_1$, $\Gamma^o_2 = \Gamma^o_1$, and $\Gamma^s_2(b) = \Gamma^s_1(b)$ whenever $b \neq a$.
  Note that here no record is kept of which owners actually contributed to this particular use of the resource a.

- *resource subscription*: Intuitively $\Gamma_1 \xrightarrow{\mathsf{sub}(o,a,c)} \Gamma_2$ if Γ_2 can be constructed from Γ_1, by decreasing $\Gamma^o_1(o)$ by c, and increasing $\Gamma^s_1(a)(o)$ by the same amount. Formally it holds when
  · $o \in \mathrm{dom}(\Gamma^s_1(a))$; that is o is actually registered to use resource a
  · $\Gamma^o_2(o) = \Gamma^o_1(o) - c$
  · $\Gamma^s_2(a)(o) = \Gamma^s_1(a)(o) + c$
  · $\Gamma^c_2 = \Gamma^c_1$, $\Gamma^r_2 = \Gamma^r_1$, and $\Gamma^o_2(o') = \Gamma^o_1(o')$ whenever $o' \neq o$, and $\Gamma^s_2(b) = \Gamma^s_1(b)$ for every other b different from a.

- *resource registration*: The cost environment $\Gamma, a{:}\mathsf{R}$, is only defined if a is *fresh* to Γ, that is, if a is not in dom(Γ^c). In this case it gives the new cost environment Φ defined by
  · $\Phi^c(b) = \begin{cases} \Gamma^c(b) & \text{if } b \in \mathrm{dom}(\Gamma) \\ \mathsf{R}_c & \text{a=b} \end{cases}$
  · $\Phi^s(b) = \begin{cases} \Gamma^s(b) & \text{if } b \in \mathrm{dom}(\Gamma) \\ \mathsf{R}_s & \text{a=b} \end{cases}$
  So we require R_s to be a partial function in $(\mathsf{Own} \rightharpoonup N^\infty)$. Note that this also implicitly defines the set of owners registered to use the new channel a, namely dom(R_s).
  · Φ^o and Φ^r are taken to be Γ^o and Γ^r respectively.

The pair $(\Gamma \triangleright P)$ is called a *system* if $\mathsf{fn}(P) \subseteq \mathrm{dom}(\Gamma^c)$, that is every free resource name in P is known to the *cost environment* Γ. We use $\mathcal{S}$ to denote the set of all systems.

Proposition 2.1 *If* $(\Gamma_1 \triangleright P_1)$ *is a system and* $(\Gamma_1 \triangleright P_1) \longrightarrow (\Gamma_2 \triangleright P_2)$ *then* $(\Gamma_2 \triangleright P_2)$ *is also a system.* □

Reductions in a system affects it's cost environment, and as a sanity check we can describe precisely the kinds of changes which are possible:

Proposition 2.2 *Suppose* $(\Gamma_1 \triangleright P_1) \longrightarrow (\Gamma_2 \triangleright P_2)$. *Then*

- $\Gamma_1 = \Gamma_2$
- *or* $\Gamma_1 \xrightarrow{a} \Gamma_2$, *for some resource* a
- *or* $\Gamma_1 \xrightarrow{\mathsf{sub}(o,a,c)} \Gamma_2$, *for some resource* a, *owner* o *and cost* c. □

3 Observing systems

Here we adapt the standard theory of *reduction barbed congruence*, [14,28,13,11], to π_{cost}. The theory enables one to say that relative to an *environment* Γ the processes P_1 and P_2 are *observationally equivalent*. We modify this in two ways. In the first we will actually relate systems, $\Gamma_1 \triangleright P_1$ and $\Gamma_2 \triangleright P_2$, thereby enabling us to compare, for example, the same process running under different *cost environments*. Secondly, because our *cost environments* accumulate expenditure we will be able to define what it means for one system to be more efficient than another, while offering similar observational behaviour to observers: $(\Gamma_1 \triangleright P_1) \preccurlyeq_{cbp} (\Gamma_2 \triangleright P_2)$.

Observations:

There is lots of scope for defining what it means to observe processes in scenarios where communication, and therefore observation, must be paid for. In this preliminary paper we take a simple approach, in which the observations of a system $(\Gamma \triangleright P)$ are paid for by the funds available within the *cost environment* Γ; in other words observers are allowed access to the funds available in Γ.

Because π_{cost} is based on the *asynchronous picalculus* it turns out that only one kind of observable is required. Intuitively $(\Gamma \triangleright P) \Downarrow^c \mathsf{del}(a)$ means that it will cost the system *at most* c for an observer to be assured that some value can be delivered to the resource a.

First let us define *strong observations*. We write $(\Gamma \triangleright P) \downarrow^c \mathsf{del}(a)$ whenever

- $P \equiv (\mathsf{new}\,\tilde{b}{:})(\mathsf{del}(a,v)\,|Q)$, where a does not occur in $(\tilde{b})$
- $\Gamma \xrightarrow{a} \Gamma'$ for some Γ'
- $\Gamma^c(a) \leq c$

So this means that an observer can immediately obtain some value on resource a, and the cost of obtaining it is at most c. Then *weak observations* are defined by letting

$$(\Gamma \triangleright P) \Downarrow^c \mathsf{del}(a)$$

whenever $(\Gamma \triangleright P) \longrightarrow^* (\Phi \triangleright Q)$ where $(\Phi \triangleright Q) \downarrow^d \mathsf{del}(a)$, for some d such that $d + (\Phi^r - \Gamma^r) \leq c$. Here the total cost to the system is still at most c, taking into account the cost required to get to the state where the actual (strong) observation can be made.

We say that a relation $\mathcal{R} \subseteq \mathcal{S} \times \mathcal{S}$ is *observation improving* if, whenever $S_1 \;\mathcal{R}\; S_2$, $S_1 \Downarrow^c \mathsf{del}(a)$ implies $S_2 \Downarrow^c \mathsf{del}(a)$.
Intuitively this means that any observation made on the system S_1 can be made on S_2 for a possibly smaller cost.

Contextual:

A relation $\mathcal{R} \subseteq \mathcal{S} \times \mathcal{S}$ is called *contextual* if

(i) $(\Gamma_1 \triangleright P_1) \mathcal{R} (\Gamma_2 \triangleright P_2)$ implies $(\Gamma_1 \triangleright P_1 \,|\, O) \mathcal{R} (\Gamma_2 \triangleright P_2 \,|\, O)$, whenever $(\Gamma_1 \triangleright P_1 \,|\, O)$ and $(\Gamma_2 \triangleright P_2 \mid O)$ are both systems

(ii) $(\Gamma_1 \triangleright P_1) \mathcal{R} (\Gamma_2 \triangleright P_2)$ implies $(\Gamma_1, a{:}\mathsf{R} \triangleright P_1) \mathcal{R} (\Gamma_2, a{:}\mathsf{R} \triangleright P_2)$, whenever a is fresh to Γ_i.

Reduction cost improving:

A relation $\mathcal{R} \subseteq \mathcal{S} \times \mathcal{S}$ is called *reduction cost improving* if, whenever $(\Gamma_1 \triangleright P_1) \mathcal{R}$ $(\Gamma_2 \triangleright P_2)$

(i) $(\Gamma_1 \triangleright P_1) \longrightarrow (\Gamma'_1 \triangleright P'_1)$ implies $(\Gamma_2 \triangleright P_2) \longrightarrow^* (\Gamma'_2 \triangleright P'_2)$ for some system $(\Gamma'_2 \triangleright P'_2)$ such that $(\Gamma'^r_2 - \Gamma^r_2) \leq (\Gamma'^r_1 - \Gamma^r_1)$ and $(\Gamma'_1 \triangleright P'_1) \mathcal{R} (\Gamma'_2 \triangleright P'_2)$.

(ii) conversely $(\Gamma_2 \triangleright P_2) \longrightarrow (\Gamma'_2 \triangleright P'_2)$ implies $(\Gamma_1 \triangleright P_1) \longrightarrow^* (\Gamma'_1 \triangleright P'_1)$ for some system $(\Gamma'_1 \triangleright P'_1)$ such that $(\Gamma'^r_2 - \Gamma^r_2) \leq (\Gamma'^r_1 - \Gamma^r_1)$ and $(\Gamma'_1 \triangleright P'_1) \mathcal{R} (\Gamma'_2 \triangleright P'_2)$.

Here $(\Gamma'^r_i - \Gamma^r_i)$ represents the cost of doing the reduction $(\Gamma_i \triangleright P_i) \longrightarrow (\Gamma'_i \triangleright P'_i)$. So $(\Gamma_1 \triangleright P_1) \mathcal{R} (\Gamma_2 \triangleright P_2)$ means that the systems can mimic each other's reductions, but the reductions from $(\Gamma_2 \triangleright P_2)$ are no more expensive, and possibly cheaper, than those from $(\Gamma_1 \triangleright P_1)$.

Definition 3.1 Cost barbed precongruence:

Let $\preccurlyeq_{cbp} \subseteq \mathcal{S} \times \mathcal{S}$ be the largest relation which is

(i) observation improving

(ii) contextual

(iii) reduction cost improving.

The main result of the paper is a non-contextual purely coinductive characterisation of this observational preorder between systems.

4 Bisimulation equivalence for π_{cost}

In Figure 4 we give a set of rules for deriving judgements of the form $(\Gamma_1 \triangleright P_1) \xrightarrow{\mu} (\Gamma_2 \triangleright P_2)$, where μ can take one of the forms

(i) internal action, τ:

(ii) input, $a?b$, $(b{:}\mathsf{R})a?b$: input by resource a of a known or fresh name, respectively

(iii) output: $\mathsf{del}(a,b)$, $(b{:}\mathsf{R})\,\mathsf{del}(a,b)$: delivery of known or fresh name, respectively, to resource a

(iv) external subscription, $\mathsf{sub}(o,a,c)$: subscription by owner o to resource a

(v) external consumption, τ_a: use by some external entity of resource a.

We use α to range over the free actions $a?b$ or $\mathsf{del}(a,b)$, and in the rules we employ the standard *complementary* notation for them, α denoting the complement of α. For

(L-IN)
$$\frac{\Gamma \xrightarrow{a} \Gamma'}{\Gamma \triangleright a?(x)\,P \xrightarrow{a?b} \Gamma' \triangleright P\{\!|^{b}\!/\!_{x}|\!\}} \quad b \in \mathrm{dom}(\Gamma^c)$$

(L-IN)
$$\frac{\Gamma \xrightarrow{a} \Gamma'}{\Gamma \triangleright a?(x)\,P \xrightarrow{(b:\mathsf{R})a?b} \Gamma', b:\mathsf{R} \triangleright P\{\!|^{b}\!/\!_{x}|\!\}} \quad b \notin \mathrm{dom}(\Gamma^c)$$

(L-ASY)
$$\Gamma \triangleright P \xrightarrow{a?b} \Gamma \triangleright P \,|\, \mathsf{del}(a,b) \quad b \in \mathrm{dom}(\Gamma^c)$$

(L-ASY)
$$\Gamma \triangleright P \xrightarrow{(b:\mathsf{R})a?b} \Gamma, b:\mathsf{R} \triangleright P \,|\, \mathsf{del}(a,b) \quad b \notin \mathrm{dom}(\Gamma^c)$$

(L-OUT)
$$\Gamma \triangleright a!\langle b\rangle.P \xrightarrow{\tau} \Gamma \triangleright \mathsf{del}(a,b)\,|P$$

(L-DEL)
$$\frac{\Gamma \xrightarrow{a} \Gamma'}{\Gamma \triangleright \mathsf{del}(a,b) \xrightarrow{\mathsf{del}(a,b)} \Gamma' \triangleright \mathsf{stop}}$$

(L-COMM)
$$\frac{\Gamma \triangleright Q \xrightarrow{\alpha} \Gamma' \triangleright Q', \quad \Gamma \triangleright P \xrightarrow{\overline{\alpha}} \Gamma' \triangleright P'}{\Gamma \triangleright P \,|\, Q \xrightarrow{\tau} \Gamma' \triangleright (P' \,|\, Q')}$$

(L-COMM)
$$\frac{\Gamma \triangleright Q \xrightarrow{(b:\mathsf{R})\alpha} \Gamma', b{:}\mathsf{R} \triangleright Q', \quad \Gamma \triangleright P \xrightarrow{(b:\mathsf{R})\overline{\alpha}} \Gamma', b{:}\mathsf{R} \triangleright P'}{\Gamma \triangleright P \,|\, Q \xrightarrow{\tau} \Gamma' \triangleright (\mathsf{new}\, b{:}\mathsf{R})(P' \,|\, Q')}$$

(L-OPEN)
$$\frac{\Gamma, b{:}\mathsf{R} \triangleright P \xrightarrow{\mathsf{del}(a,b)} \Gamma' \triangleright P'}{\Gamma \triangleright (\mathsf{new}\, b{:}\mathsf{R})\, P \xrightarrow{(b:\mathsf{R})\, \mathsf{del}(a,b)} \Gamma' \triangleright P'} \quad a \neq b$$

(L-SUBSCRIBE)
$$\frac{\Gamma \xrightarrow{\mathsf{sub}(o,a,c)} \Gamma'}{\Gamma \triangleright \mathsf{subscribe}(o,a,c).P \xrightarrow{\tau} \Gamma' \triangleright P}$$

(L-MATCH)
$$\Gamma \triangleright \mathsf{if}\ a = a\ \mathsf{then}\ P\ \mathsf{else}\ Q \xrightarrow{\tau} \Gamma \triangleright P$$

(L-MISMATCH)
$$\Gamma \triangleright \mathsf{if}\ a = b\ \mathsf{then}\ P\ \mathsf{else}\ Q \xrightarrow{\tau} \Gamma \triangleright Q \quad a \neq b$$

(L-CNTX)
$$\frac{\Gamma, b{:}\mathsf{R} \triangleright P \xrightarrow{\mu} \Gamma', b{:}\mathsf{R}' \triangleright P'}{\Gamma \triangleright (\mathsf{new}\, b{:}\mathsf{R})\, P \xrightarrow{\mu} \Gamma' \triangleright (\mathsf{new}\, b{:}\mathsf{R}')\, P'} \quad b \notin \mathsf{n}(\mu)$$

(L-CNTX)
$$\frac{\Gamma \triangleright P \xrightarrow{\mu} \Gamma' \triangleright P'}{\Gamma \triangleright P \,|\, Q \xrightarrow{\mu} \Gamma' \triangleright P' \,|\, Q}$$

(L-EXT.SUBSCRIBE)
$$\frac{\Gamma \xrightarrow{\mathsf{sub}(o,a,c)} \Gamma'}{\Gamma \triangleright P \xrightarrow{\mathsf{sub}(o,a,c)} \Gamma' \triangleright P}$$

(L-EXT.COM)
$$\frac{\Gamma \xrightarrow{a} \Gamma'}{\Gamma \triangleright P \xrightarrow{\tau_a} \Gamma' \triangleright P}$$

Fig. 4. An action semantics for π_{cost}

convenience, we sometimes use $(\tilde{b}{:}\tilde{\mathsf{R}})\alpha$ to denote an arbitrary action; α is considered to be a degenerate instance of $(\tilde{b}{:}\tilde{\mathsf{R}})\alpha$, where the sequence $(\tilde{b}{:}\tilde{\mathsf{R}})$ is empty. We will also assume, as usual, that all bound names are *fresh* in the context in which they are used.

Many of the rules are a very simple modification of those used in the standard action semantics for the asynchronous *picalculus*, to take into account the presence of *cost environments*. Resource charging $\Gamma \xrightarrow{a} \Gamma'$ is required for both input (L-IN) and delivery (L-DEL). Resource registration is required in (L-OPEN), as is usual for the *picalculus*, but also in (L-CNTX) because of the effect that internal moves may have on resource types. The rule (L-ASY) is required because our language is asynchronous. Intuitively it represents an attempt by a user to observe a process P performing the input action $a?v$, by sending it the package $\mathsf{del}(a,v)$. This is ignored by P, and the resulting system is $P \,|\, \mathsf{del}(a,v)$. Note it does not require any intervention of the *cost environment*; intuitively a request has been made to the resource a, but is has not yet been serviced. The use of (L-ASY) has been discussed at length in [11], and was originally suggested in [14].

There are two novel actions which take into account the indirect effect that observers may have on the *cost environment* of a system. The first, (L-EXT.SUBSCRIBE), models some observer adding some funds to the resource a, while (L-EXT.COMM) is required to take into account the use of a resource a by some external party.

We can perform a number of sanity checks on these rules. For example one can show that if $(\Gamma_1 \triangleright P_1) \xrightarrow{(b:\mathsf{R})\alpha} (\Gamma_2 \triangleright P_2)$ Then $\Gamma_2 = \Phi, b{:}\mathsf{R}$ for some Γ_2 such that $\Gamma_1 \xrightarrow{a} \Phi$, where a is the channel used in α. In fact the cost to the system of performing this action is precisely the cost of using this channel: $\Phi^r - \Gamma_1^r = \Gamma_1^c(a)$.

The actions also preserve systems:

Proposition 4.1 *If* $(\Gamma_1 \triangleright P_1)$ *is a system and* $(\Gamma_1 \triangleright P_1) \xrightarrow{\mu} (\Gamma_2 \triangleright P_2)$ *then* $(\Gamma_2 \triangleright P_2)$ *is also a system.* □

As another sanity check we can relate the internal actions of the action semantics of Figure 4 with the reduction semantics of Figure 2. We lift the structural equivalence from processes to systems by writing $\Gamma_1 \triangleright P_1 \equiv \Gamma_2 \triangleright P_2$ to mean $P_1 \equiv P_2$ and $\Gamma_1 = \Gamma_2$.

Lemma 4.2

(i) $S_1 \longrightarrow S_2$ *implies* $S_1 \xrightarrow{\tau} S_2'$ *for some* $S_2' \equiv S_2$

(ii) $S_1 \xrightarrow{\tau} S_2$ *implies either* $S_2 \equiv S_1$ *or* $S_1 \longrightarrow S_2$. □

The bisimulation equivalence is defined using a slight abstraction from these judgements. As usual we ignore the type information on the freshly exported resource names [13], but more importantly we explicitly record the cost of actions:

(i) $(\Gamma_1 \triangleright P_1) \xrightarrow{(b)\,\mathsf{del}(a,b)}{}^{c} (\Gamma_2 \triangleright P_2)$ whenever $(\Gamma_1 \triangleright P_1) \xrightarrow{(b:\mathsf{R})\,\mathsf{del}(a,b)} (\Gamma_2 \triangleright P_2)$ can be deduced from the rules for some R, where $(\Gamma_2^r - \Gamma_1^r) \leq c$.

(ii) For all other μ, we write $(\Gamma_1 \triangleright P_1) \xrightarrow{\mu}{}^{c} (\Gamma_2 \triangleright P_2)$ whenever $(\Gamma_1 \triangleright P_1) \xrightarrow{\mu} (\Gamma_2 \triangleright P_2)$ can be deduced from the rules in Figure 4, where again $(\Gamma_2^r - \Gamma_1^r) \leq c$.

This means intuitively that the system can perform the μ action with *at most* cost c. These are extended to weak actions $(\Gamma_1 \triangleright P_1) \overset{\mu}{\Longrightarrow}{}^{c} (\Gamma_2 \triangleright P_2)$ in the standard manner, where the cost c is the accumulation of the cost bound associated with the action μ together with the cost bounds of all the pre and post internal τ actions.

Definition 4.3 Cost bisimulation

A relation $\mathcal{R} \subseteq \mathcal{S} \times \mathcal{S}$ is a *cost bisimulation* if whenever $S_1 \,\mathcal{R}\, S_2$,

(i) $S_1 \xrightarrow{\mu}{}^{c} S_1'$ implies $S_2 \overset{\hat{\mu}}{\Longrightarrow}{}^{c} S_2'$ for some S_2' such that $S_1' \,\mathcal{R}\, S_2'$

(ii) conversely, $S_2 \xrightarrow{\mu}{}^{c} S_2'$ implies $S_1 \overset{\hat{\mu}}{\Longrightarrow}{}^{c} S_1'$ for some S_1' such that $S_1' \,\mathcal{R}\, S_2'$.

Here we are using the notation $\overset{\hat{\mu}}{\Longrightarrow}{}^{c}$ to mean:

- $\overset{\tau}{\Longrightarrow}{}^{c} \cup \mathrm{Id}$, where Id is the identity relation over configurations, when μ is τ
- $\overset{\mu}{\Longrightarrow}{}^{c}$, otherwise.

Let $\sqsubseteq_{\text{cost}}$ be the largest cost bisimulation, that is $S_1 \sqsubseteq_{\text{cost}} S_2$ whenever there is some cost bisimulation $\mathcal{R}$ such that $S_1 \mathcal{R} S_2$. Intuitively this means that the systems S_1 and S_2 are bisimulation equivalent in the traditional sense, but that the latter is more cost efficient than the former. Note that in general, when resources have non-trivial costs, this relation will not be symmetric.

Theorem 4.4 (Full abstraction) *Suppose $dom(\Gamma_1^c) = dom(\Gamma_2^c)$. Then $(\Gamma_1 \triangleright P_1) \sqsubseteq_{cost} (\Gamma_2 \triangleright P_2)$ if and only if $(\Gamma_1 \triangleright P_1) \preccurlyeq_{cbp} (\Gamma_2 \triangleright P_2)$.*

Proof. (Outline) The structure of the proof is very similar to the corresponding one in [13], Corollary 6.9. In one direction it is sufficient to prove that cost bisimulation $\sqsubseteq_{\mathsf{cost}}$ satisfies the three defining properties in Definition 3.1. The only difficulty here is **Contextuality**; the proof is long, but not as complicated as the corresponding proof in [13], Proposition 6.4. However some care is required because actions can change the types of bound resources. This phenomenon already occured in [9].

To prove the converse it is sufficient to show that the relation $\preccurlyeq_{cbp}$ is a *cost bisimulation* between systems. The essence of this result is to show that the derived actions can be simulated by suitable contexts. A version of the *Definability result*, Proposition 4.4 of [13], must be established for *cost environments*, where the simulating contexts should not expend more funds than the original action. □

5 Conclusion

In this short paper we have shown how the well-established theory of typed bisimulation equivalence for the *picalculus* can be easily adapted to provide an adequate theory of *costed process behaviour*, in which actions can only be performed if there are sufficient funds available to pay for them. Moreover the theory is relatively independent of the precise details of the *cost environment* relative to which computations takes place.

We intend to pursue this line of work in two directions. In the first we wish to apply it to the various calculi being developed for web services, such as those in [17,6], and to see to what extent practical examples can be treated. In the second, more theoretical, we intend to revisit the idea of observing *costed processes*, as discussed in Section 3. There we assumed that observers of a system had access to the funds of the system; a more realistic point of view would be that observers were required to provide themselves the funds necessary to perform observations. This change should have some implications for the required labelled transition system, but at the moment their extent is unclear.

There is already a considerable literature on topics related to this line of research. For example in [16] an *efficiency preorder* is defined between CCS processes; here the cost, or speed of a (weak) action simply depends on the number of internal moves it contains. Interesting properties of this preorder were further studied in [21]. In [15] a cost is associated with a subset of actions (which can not be synchronised) and a theory of *amortised bisimulations* is developed in this framework. Here *amortised* refers to the fact that the cost of each individual action is not compared; instead it is the overall cost which counts, where the high cost of one action may be compensated for by another at a low cost. It should be possible to develop *amortised bisimulations* for π_{cost}; but an interesting theoretical question is how the resulting equivalence can be justified in terms of observations. *Faster than* preorders between processes have also been developed in work on timed process algebras; see [18] for an attempt at unifying different approaches.

In [3] there is a slightly different notion of the cost of a computation. The setting is *mobile ambients* [5] and the cost of computation is in terms of *space consumption*; essentially mobile agents can only migrate if the target location has sufficient capacity to accommodate it. Finally in [7] (and related publications such as [27]) a quite general theory of *resource-based* computation is being developed. The setting is SCCS [22], but the operational semantics is with respect to a *resource*. The generality is obtained by only requiring certain operations on the *resource*; in effect their use of *resource* is very similar to our use of *cost environments*, although the required operations are quite different. However they also have resource based modal logic for expressing properties of processes. The interesting point about the logic, a variation on Hennessy-Milner logic [12], is that satisfiabilty is resource dependent, being based on the bunched logic of [24]. It would be interesting to see if a similar logic could be developed for π_{cost}.

References

[1] Martín Abadi and Andrew D. Gordon. A calculus for cryptographic protocols: The spi calculus. In *Fourth ACM Conference on Computer and Communications Security*, pages 36–47. ACM Press, 1997.

[2] Gustavo Alonso, Fabio Casati, Harumi A. Kuno, and Vijay Machiraju. *Web Services - Concepts, Architectures and Applications*. Data-Centric Systems and Applications. Springer, 2004.

[3] 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.

[4] G. Boudol. Asynchrony and the π-calculus. Technical Report 1702, INRIA-Sophia Antipolis, 1992.

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

[6] G. Castagna, N. Gesbert, and L. Padovani. A theory of contracts for web services. In *POPL '08, 35th ACM Symposium on Principles of Programming Languages*, jan 2008.

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

[8] Vincent Danos and Jean Krivine. Formal molecular biology done in ccs-r. *Electr. Notes Theor. Comput. Sci.*, 180(3):31–49, 2007.

[9] Adrian Francalanza and Matthew Hennessy. A theory for observational fault tolerance. *Information and Computation*, 206(6): 711-759, 2008.

[10] Manish Gaur. PhD thesis, University of Sussex, 2008. to appear.

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

[12] Matthew Hennessy and Robin Milner. Algebraic laws for nondeterminism and concurrency. *Journal of the ACM*, 32(1):137–161, January 1985.

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

[14] 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.

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

[16] S. Arun-Kumar and M. Hennessy. An efficiency preorder for processes. *Acta Informatica*, 29(8):737–760, 1992.

[17] Cosimo Laneve and Luca Padovani. The *must* preorder revisited. In Luís Caires and Vasco Thudichum Vasconcelos, editors, *CONCUR*, volume 4703 of *Lecture Notes in Computer Science*, pages 212–225. Springer, 2007.

[18] Gerald Lüttgen and Walter Vogler. Bisimulation on speed: A unified approach. In Vladimiro Sassone, editor, *FoSSaCS*, volume 3441 of *Lecture Notes in Computer Science*, pages 79–94. Springer, 2005.

[19] R. Milner. *Communication and Concurrency*. Prentice-Hall, 1989.

[20] R. Milner. *Communicating and mobile systems: the π-calculus*. Cambridge University Press, 1999.

[21] R. Milner and D. Sangiorgi. Techniques of weak bisimulation up-to, 1992.

[22] Robin Milner. Calculi for synchrony and asynchrony. *Theor. Comput. Sci.*, 25:267–310, 1983.

[23] Rocco De Nicola, Daniele Gorla, and Rosario Pugliese. Basic observables for a calculus for global computing. *Inf. Comput.*, 205(10):1491–1525, 2007.

[24] Peter W. O'Hearn and David J. Pym. The logic of bunched implications. *Bulletin of Symbolic Logic*, 5(2):215–244, 1999.

[25] B. Pierce and D. Sangiorgi. Typing and subtyping for mobile processes. *Journal of Mathematical Structures in Computer Science*, 6(5):409–454, 1996. An extended abstract in *Proc. LICS 93*, IEEE Computer Society Press.

[26] Benjamin C. Pierce and David N. Turner. Pict: A programming language based on the pi-calculus. In G. Plotkin, C. Stirling, and M. Tofte, editors, *Proof, Language and Interaction: Essays in Honour of Robin Milner*. MIT Press, 2000.

[27] David J. Pym and Chris M. N. Tofts. Systems modelling via resources and processes: Philosophy, calculus, semantics, and logic. *Electr. Notes Theor. Comput. Sci.*, 172:545–587, 2007.

[28] D. Sangiorgi and D. Walker. *The π-calculus: a Theory of Mobile Processes*. Cambridge University Press, 2001.