A Theory of Encodings and Expressiveness
(extended abstract)

Rob van Glabbeek[1,2]

[1] Data61, CSIRO, Sydney, Australia
[2] Comput. Sci. and Engineering, University of New South Wales, Sydney, Australia

Abstract. This paper proposes a definition of what it means for one system description language to encode another one, thereby enabling an ordering of system description languages with respect to expressive power. I compare the proposed definition with other definitions of encoding and expressiveness found in the literature, and illustrate it on a well-known case study: the encoding of the synchronous in the asynchronous π-calculus.

1 Introduction

This paper, like [21,16], aims at answering the question what it means for one language to encode another one, and making the resulting definition applicable to order system description languages like CCS, CSP and the π-calculus with respect to their expressive power.

To this end it proposes a unifying concept of valid translation between two languages *up to* a semantic equivalence or preorder. It applies to languages whose semantics interprets the operators and recursion constructs as operations on a set of values, called a *domain*. Languages can be partially ordered by their expressiveness up to the chosen equivalence or preorder according to the existence of valid translations between them.

The concept of a [valid] translation between system description languages (or *process calculi*) was first formally defined by Boudol [3]. There, and in most other related work in this area, the domain in which a system description language is interpreted consists of the closed expressions from the language itself. In [14] I have reformulated Boudol's definition, while dropping the requirement that the domain of interpretation is the set of closed terms. This allows (but does not enforce) a clear separation of syntax and semantics, in the tradition of universal algebra. Nevertheless, the definition employed in [14] only deals with the case that all (relevant) elements in the domain are denotable as the interpretations of closed terms. In [16] situations are described where such a restriction is undesirable. In addition, both [3] and [14] require the semantic equivalence $\sim$ under which two languages are compared to be a congruence for both of them. This is too severe a restriction to capture many recent encodings [43,2,30,31,7,1,33,38].

In [16] I alleviated these two restrictions by proposing two notions of encoding: *correct* and *valid* translations up to $\sim$. Each of them generalises the proposals of [3] and [14]. The former drops the restriction on denotability as well as $\sim$ being a congruence for the whole target language, but it requires $\sim$

to be a congruence for the source language, as well as for the source's image within the target. The latter drops both congruence requirements (and allows $\sim$ to be a preorder rather than an equivalence), but at the expense of requiring denotability by closed terms. In situations where $\sim$ is a congruence for the source language's image within the target language *and* all semantic values are denotable, the two notions agree.

The current paper further generalises the work of [16] by proposing a new notion of a valid translation that incorporates the correct and valid translations of [16] as special cases. It drops the congruence requirements as well as the restriction on denotability.

As in [16], my aim is to generalise the concept of a valid translation as much as possible, so that it is uniformly applicable in many situations, and not just in the world of process calculi. Also, it needs to be equally applicable to encodability and separation results, the latter saying that an encoding of one language in another does not exists. At the same time, I try to derive this concept from a unifying principle, rather than collecting a set of criteria that justify a number of known encodability and separation results that are intuitively justified.

Overview of the paper Sect. 2 defines my new concept of a valid translation up to a semantic equivalence or preorder $\backsim$. Roughly, a valid translation of one language into another is a mapping from the expressions in the first language to those in the second that preserves their meaning, i.e. such that the meaning of a translated expression is semantically equivalent to the meaning of the original.

Sect. 3 shows that this concept generalises the notion of a correct translation from [16]: a translation is correct up to a semantic equivalence $\sim$ iff it is valid up to $\sim$ and $\sim$ is a congruence for the source language as well as for the image of the source language within the target language.

Likewise, [18]—the full version of this paper—establishes the coincidence of my validity-based notion of expressiveness with the one from [16] when applying both to languages for which all semantic values are denotable by closed terms.

One language is said to be at least as expressive as another up to $\backsim$ iff there exists a valid translation up to $\backsim$ of the latter language into the former. Sect. 4 shows that "being at least as expressive as" is a preorder on languages. This expressiveness preorder depends on the choice of $\backsim$, and a coarser choice (making less distinctions) yields a richer preorder of expressiveness inclusions.

Sect. 6 illustrates the framework on a well-known case study: the encoding of the synchronous in the asynchronous π-calculus.

Sect. 7 discusses the *congruence closure* of a semantic equivalence for a given language, and remarks that in the presence of operators with infinite arity it is not always a congruence. Sect. 8 states a useful congruence closure property for valid translations: if a translation between two languages exists that is valid up a semantic equivalence $\sim$, then it is even valid up to an equivalence that

- on the source language coincides with the congruence closure of $\sim$
- on the image of the source within the target language also coincides with the congruence closure of $\sim$
- melts each equivalence class of the source with exactly one of the target.

Sect. 9 concludes that the framework established thus far is great for comparing the expressiveness of languages, but falls short for the purpose of combining language features. This requires a congruence reflection theorem, provided in Sect. 12, for languages satisfying postulates formulated in Sects. 5, 10 and 11.

Sect. 12 defines when a translation is *compositional*, and shows that any valid translation up to $\precsim$ can be modified into a compositional translation valid up to $\precsim$. This requires restricting attention to languages and preorders $\precsim$ that satisfy some mild sanity requirements—the postulates of Sects. 10 and 11. Hence, for the purpose of comparing the expressive power of languages, valid translations between them may be presumed compositional.

Sect. 13 compares my approach with the one of Gorla [21], and concludes. Omitted proofs and counterexamples (marked by ¶) can be found in [18].

2 Languages, valid translations, and expressiveness

A language consists of *syntax* and *semantics*. The syntax determines the valid expressions in the language. The semantics is given by a mapping $[\![\ \ ]\!]$ that associates with each valid expression its meaning, which can for instance be an object, concept or statement.

Following [16], I represent a language $\mathcal{L}$ as a pair $(\mathbb{T}_{\mathcal{L}}, [\![\ \ ]\!]_{\mathcal{L}})$ of a set $\mathbb{T}_{\mathcal{L}}$ of valid expressions in $\mathcal{L}$ and a mapping $[\![\ \ ]\!]_{\mathcal{L}} : \mathbb{T}_{\mathcal{L}} \to \mathcal{D}_{\mathcal{L}}$ from $\mathbb{T}_{\mathcal{L}}$ in some set of meanings $\mathcal{D}_{\mathcal{L}}$.

Definition 1 ([16]) A *translation* from a language $\mathcal{L}$ into a language $\mathcal{L}'$ is a mapping $\mathcal{T} : \mathbb{T}_{\mathcal{L}} \to \mathbb{T}_{\mathcal{L}'}$.

In this paper, I consider single-sorted languages $\mathcal{L}$ in which *expressions* or *terms* are built from variables (taken from a set $\mathcal{X}$) by means of operators (including constants) and possibly recursion constructs. For such languages the meaning $[\![E]\!]_{\mathcal{L}}$ of an $\mathcal{L}$-expression E is a function of type $(\mathcal{X} \to \mathbf{V}) \to \mathbf{V}$ for a given sets of *values* $\mathbf{V}$. It associates a value $[\![E]\!]_{\mathcal{L}}(\rho) \in \mathbf{V}$ to E that depends on the choice of a *valuation* $\rho\colon \mathcal{X} \to \mathbf{V}$. The valuation associates a value from $\mathbf{V}$ with each variable.

Since normally the names of variables are irrelevant and the cardinality of the set of variables satisfies only the requirement that it is "sufficiently large", no generality is lost by insisting that two (system description) languages whose expressiveness is being compared employ the same set of (process) variables. On the other hand, two languages $\mathcal{L}$ and $\mathcal{L}'$ may be interpreted in different domains of values $\mathbf{V}$ and $\mathbf{V}'$.

Let $\mathcal{L}$ and $\mathcal{L}'$ be languages as considered above, with semantic mappings

$$[\![\ \ ]\!]_{\mathcal{L}} : \mathbb{T}_{\mathcal{L}} \to ((\mathcal{X} \to \mathbf{V}) \to \mathbf{V}) \quad \text{and} \quad [\![\ \ ]\!]_{\mathcal{L}'} : \mathbb{T}_{\mathcal{L}'} \to ((\mathcal{X} \to \mathbf{V}') \to \mathbf{V}').$$

In order to compare these languages w.r.t. their expressive power I need a semantic equivalence or preorder $\precsim$ that is defined on a unifying domain of interpretation $\mathbf{Z}$, with $\mathbf{V}, \mathbf{V}' \subseteq \mathbf{Z}$.[1] Intuitively, $v' \precsim v$ with $v \in \mathbf{V}$ and $v' \in \mathbf{V}'$ means that values v and v' are sufficiently alike for our purposes, so that one can accept

[1] I will be chiefly interested in the case that $\precsim$ is an equivalence—hence the choice of a symbol that looks like $\sim$. However, to establish Obs. 2 and Thm. 2 below, it

a translation of an expression with meaning v into an expression with meaning v'. Below, target values of a translation (in $\mathbf{V}'$) are written on the left.

Correct and a *valid* translations up to a semantic equivalence or preorder $\backsim$ were introduced in [16]. Here I redefine these concepts in terms of a new concept of *correctness w.r.t. a semantic translation*.

Definition 2 Let $\mathbf{V}$ and $\mathbf{V}'$ be domains of values in which two languages $\mathcal{L}$ and $\mathcal{L}'$ are interpreted. A *semantic translation* from $\mathbf{V}$ into $\mathbf{V}'$ is a relation $\mathbf{R} \subseteq \mathbf{V}' \times \mathbf{V}$ such that $\forall v \in \mathbf{V}. \exists v' \in \mathbf{V}'. v' \mathbf{R} v$.

Thus every semantic value in $\mathbf{V}$ needs to have a counterpart in $\mathbf{V}'$—possibly multiple ones. For valuations $\eta : \mathcal{X} \to \mathbf{V}'$, $\rho : \mathcal{X} \to \mathbf{V}$ I write $\eta \mathbf{R} \rho$ iff $\eta(X) \mathbf{R} \rho(X)$ for each $X \in \mathcal{X}$.

Definition 3 A translation $\mathcal{T} : \mathbb{T}_{\mathcal{L}} \to \mathbb{T}_{\mathcal{L}'}$ is *correct* w.r.t. a semantic translation $\mathbf{R}$ if $[\![\mathcal{T}(E)]\!]_{\mathcal{L}'}(\eta) \mathbf{R} [\![E]\!]_{\mathcal{L}}(\rho)$ for all expressions $E \in \mathbb{T}_{\mathcal{L}}$ and all valuations $\eta : \mathcal{X} \to \mathbf{V}'$ and $\rho : \mathcal{X} \to \mathbf{V}$ with $\eta \mathbf{R} \rho$.

Thus $\mathcal{T}$ is correct iff the meaning of the translation of an expression E is a counterpart of the meaning of E, no matter what values are filled in for the variables, provided that the value filled in for a given variable X occurring in the translation $\mathcal{T}(E)$ is a counterpart of the value filled in for X in E.

Definition 4 A translation $\mathcal{T} : \mathbb{T}_{\mathcal{L}} \to \mathbb{T}_{\mathcal{L}'}$ is *correct* up to $\backsim$ iff $\backsim$ is an equivalence, the restriction $\mathbf{R}$ of $\backsim$ to $\mathbf{V}' \times \mathbf{V}$ is a semantic translation, and $\mathcal{T}$ is correct w.r.t. $\mathbf{R}$.

Definition 5 A translation $\mathcal{T}$ is *valid* up to $\backsim$ iff it is correct w.r.t. some semantic translation $\mathbf{R} \subseteq \backsim$. Language $\mathcal{L}'$ is at least as *expressive* as $\mathcal{L}$ up to $\backsim$ if a translation valid up to $\backsim$ from $\mathcal{L}$ into $\mathcal{L}'$ exists.

Example 4 in [18] illustrates both notions and shows their difference.

3 Correct = valid + congruence

In [16] the concept of a correct translation up to $\sim$ was defined, for $\sim$ a semantic equivalence on $\mathbf{Z}$. Here two valuations $\eta, \rho : \mathcal{X} \to \mathbf{Z}$ are called $\sim$*-equivalent*, $\eta \sim \rho$, if $\eta(X) \sim \rho(X)$ for each $X \in \mathcal{X}$. In case there exists a $v \in \mathbf{V}$ for which there is no $\sim$-equivalent $v' \in \mathbf{V}'$, there is no correct translation from $\mathcal{L}$ into $\mathcal{L}'$ up to $\sim$. Namely, the semantics of $\mathcal{L}$ describes, among others, how any $\mathcal{L}$-operator evaluates the argument value v, and this aspect of the language has no counterpart in $\mathcal{L}'$. Therefore, [16] requires

$$\forall v \in \mathbf{V}. \exists v' \in \mathbf{V}'. v' \sim v. \tag{1}$$

This implies that for any valuation $\rho : \mathcal{X} \to \mathbf{V}$ there is an $\eta : \mathcal{X} \to \mathbf{V}'$ with $\eta \sim \rho$.

Definition 6 ([16]) A translation $\mathcal{T}$ from $\mathcal{L}$ into $\mathcal{L}'$ is *correct up to* $\sim$ iff (1) holds and $[\![\mathcal{T}(E)]\!]_{\mathcal{L}'}(\eta) \sim [\![E]\!]_{\mathcal{L}}(\rho)$ for all $E \in \mathbb{T}_{\mathcal{L}}$ and all valuations $\eta : \mathcal{X} \to \mathbf{V}'$ and $\rho : \mathcal{X} \to \mathbf{V}$ with $\eta \sim \rho$.

suffices to know that $\backsim$ is reflexive and transitive. My convention is that the dotted end of $\backsim$ points to a translation and the other end to an original—without offering an intuition for the possible asymmetry.

Note that this definition agrees completely with Def. 4. Requirement (1) above corresponds to $\mathbf{R}$ being a semantic translation in Def. 4.

If a correct translation up to $\sim$ from $\mathcal{L}$ into $\mathcal{L}'$ exists, then $\sim$ must be a congruence for $\mathcal{L}$.

Definition 7 An equivalence relation $\sim$ is a *congruence* for a language $\mathcal{L}$ interpreted in a semantic domain $\mathbf{V}$ if $[\![E]\!]_{\mathcal{L}}(\nu) \sim [\![E]\!]_{\mathcal{L}}(\rho)$ for any $\mathcal{L}$-expression E and any valuations $\nu, \rho \colon \mathcal{X} \to \mathbf{V}$ with $\nu \sim \rho$.[2]

Proposition 1 ([16]) If $\mathcal{T}$is a correct translation up to $\sim$ from $\mathcal{L}$ into $\mathcal{L}'$, then $\sim$ is a congruence for $\mathcal{L}$

The existence of a correct translation up to $\sim$ from $\mathcal{L}$ into $\mathcal{L}'$ does not imply that $\sim$ is a congruence for $\mathcal{L}'$. However, $\sim$ has the properties of a congruence for those expressions of $\mathcal{L}'$ that arise as translations of expressions of $\mathcal{L}$, when restricting attention to valuations into $\mathbf{U} := \{v \in \mathbf{V}' \mid \exists v \in \mathbf{V}.\ v' \sim v\}$. In [16] this called a *congruence for* $\mathcal{T}(\mathcal{L})$.

Definition 8 Let $\mathcal{T} : \mathbb{T}_{\mathcal{L}} \to \mathbb{T}_{\mathcal{L}'}$ be a translation from $\mathcal{L}$ into $\mathcal{L}'$. An equivalence $\sim$ on $\mathbf{V}'$ is a *congruence for* $\mathcal{T}(\mathcal{L})$ if $[\![\mathcal{T}(E)]\!]_{\mathcal{L}'}(\theta) \sim [\![\mathcal{T}(E)]\!]_{\mathcal{L}'}(\eta)$ for any $E \in \mathbb{T}_{\mathcal{L}}$ and $\theta, \eta \colon \mathcal{X} \to \mathbf{U}$ with $\theta \sim \eta$.

Proposition 2 ([16]) If $\mathcal{T}$ is a correct translation up to $\sim$ from $\mathcal{L}$ into $\mathcal{L}'$, then $\sim$ is a congruence for $\mathcal{T}(\mathcal{L})$.

The following theorem tells that the notion of validity proposed in Sect. 2 can be seen as a generalisation of the notion of correctness from [16] that applies to equivalences (and preorders) $\precsim$ that need not be congruences for $\mathcal{L}$ or $\mathcal{T}(\mathcal{L})$.

Theorem 1 A translation $\mathcal{T}$ from $\mathcal{L}$ into $\mathcal{L}'$ is correct up to a semantic equivalence $\sim$ iff it is valid up to $\sim$ and $\sim$ is a congruence for $\mathcal{T}(\mathcal{L})$. ¶

4 A hierarchy of expressiveness preorders

An equivalence or preorder $\precsim$ on a class $\mathbf{Z}$ is said to be *finer*, *stronger*, or *more discriminating* than another equivalence or preorder $\approx$ on $\mathbf{Z}$ if $v \precsim w \Rightarrow v \approx w$ for all $v, w \in \mathbf{Z}$.

Observation 1 Let $\mathcal{T} : \mathbb{T}_{\mathcal{L}} \to \mathbb{T}_{\mathcal{L}'}$ be a translation from $\mathcal{L}$ into $\mathcal{L}'$, and let $\precsim$ be finer than $\approx$. If $\mathcal{T}$ is valid up to $\precsim$, then it is also valid up to $\approx$.

The quality of a translation depends on the choice of the equivalence or preorder up to which it is valid. Any two languages are equally expressive up to the universal equivalence, relating any two processes. Hence, the equivalence or preorder needs to be chosen carefully to match the intended applications of the languages under comparison. In general, as shown by Obs. 1, using a finer equivalence or preorder yields a stronger claim that one language can be encoded in another. On the other hand, when separating two languages $\mathcal{L}$ and $\mathcal{L}'$ by showing that $\mathcal{L}$ *cannot* be encoded in $\mathcal{L}'$, a coarser equivalence yields a stronger claim.

[2] This is called a *lean* congruence in [17]; in the presence of recursion, stricter congruence requirements are common. Those are not needed in this paper.

Observation 2 The identity is a valid translation up to any preorder from any language into itself.

Theorem 2 If valid translations up to $\backsim$ exists from $\mathcal{L}_1$ into $\mathcal{L}_2$ and from $\mathcal{L}_2$ into $\mathcal{L}_3$, then there is a valid translation up to $\backsim$ from $\mathcal{L}_1$ into $\mathcal{L}_3$. ¶

Thm. 2 and Obs. 2 show that the relation "being at least as expressive as up to $\backsim$" is a preorder on languages.

5 Closed-term languages

The languages considered in this paper feature *variables*, *operators* of *arity* $n \in \mathbb{N}$, and/or other constructs. The set $\mathbb{T}_{\mathcal{L}}$ of $\mathcal{L}$-expressions is inductively defined by:

- $X \in \mathbb{T}_{\mathcal{L}}$ for each variable $X \in \mathcal{X}$,
- $f(E_1, \ldots, E_n) \in \mathbb{T}_{\mathcal{L}}$ for each n-ary operator f and expressions $E_i \in \mathbb{T}_{\mathcal{L}}$,
- and clauses for the other constructs, if any.

Examples of other constructs are the infinite summation operator $\sum_{i \in I} E_i$ of CCS, which takes arbitrary many arguments, or the recursion construct $\mu X.E$, that has one argument, but *binds* all occurrences of X in that argument.

In general a construct has a number (possibly infinite) of argument expressions and it may bind certain variables within some of its arguments—the *scope* of the binding. An occurrence of a variable X in an expression is *bound* if it occurs within the scope of a construct that binds X, and *free* otherwise.

The semantics of such a language is given, in part, by a domain of values $\mathbf{V}$, and an interpretation of each n-ary operator f of $\mathcal{L}$ as an n-ary operation $f^{\mathbf{V}} : \mathbf{V}^n \to \mathbf{V}$ on $\mathbf{V}$. Using the equations

$$[\![X]\!]_{\mathcal{L}}(\rho) = \rho(X) \quad \text{and} \quad [\![f(E_1, \ldots, E_n)]\!]_{\mathcal{L}}(\rho) = f^{\mathbf{V}}([\![E_1]\!]_{\mathcal{L}}(\rho), \ldots, [\![E_n]\!]_{\mathcal{L}}(\rho))$$

this allows an inductive definition of the meaning $[\![E]\!]_{\mathcal{L}}$ of an $\mathcal{L}$-expression E. Moreover, $[\![E]\!]_{\mathcal{L}}(\rho)$ only depends on the restriction of ρ to the set $fv(E)$ of variables occurring free in E.

The set $\mathrm{T}_{\mathcal{L}} \subseteq \mathbb{T}_{\mathcal{L}}$ of *closed terms* of $\mathcal{L}$ consists of those $\mathcal{L}$-expressions $E \in \mathbb{T}_{\mathcal{L}}$ with $fv(E) = \emptyset$. If $P \in \mathrm{T}_{\mathcal{L}}$ and $\mathbf{V} \neq \emptyset$ then $[\![P]\!]_{\mathcal{L}}(\rho)$ is independent of the choice of $\rho : \mathcal{X} \to \mathbf{V}$, and therefore denoted $[\![P]\!]_{\mathcal{L}}$.

Definition 9 A *substitution* in $\mathcal{L}$ is a partial function $\sigma : \mathcal{X} \rightharpoonup \mathbb{T}_{\mathcal{L}}$ from the variables to the $\mathcal{L}$-expressions. For a given $\mathcal{L}$-expression $E \in \mathbb{T}_{\mathcal{L}}$, $E[\sigma] \in \mathbb{T}_{\mathcal{L}}$ denotes the $\mathcal{L}$-expression E in which each free occurrence of a variable $X \in dom(\sigma)$ is replaced by $\sigma(X)$, while renaming bound variables in E so as to avoid a free variable Y occurring in an expression $\sigma(X)$ ending up being bound in $E[\sigma]$. A substitution is *closed* if it has the form $\sigma : \mathcal{X} \to \mathrm{T}_{\mathcal{L}}$.

An important class of languages used in concurrency theory are the ones where the distinction between syntax and semantic is effectively dropped by taking $\mathbf{V} = \mathrm{T}_{\mathcal{L}}$, i.e. where the domain of values where the language is interpreted in consists of the closed terms of the language. Here a valuation is the same as a closed substitution, and $[\![E]\!]_{\mathcal{L}}(\rho)$ for $E \in \mathbb{T}_{\mathcal{L}}$ and $\rho : \mathcal{X} \to \mathrm{T}_{\mathcal{L}}$ is defined to be $E[\rho] \in \mathrm{T}_{\mathcal{L}}$. I will call such languages *closed-term* languages.

6 Translating a synchronous into an asynchronous π

As an illustration of the concepts introduced above, consider the π-calculus as presented in [28], i.e., the one of [44] without matching, τ-prefixing, and choice.

Given a set of *names* $\mathcal{N}$, the set $\mathbb{T}_\pi$ of *process expressions* or *terms* E of the calculus is given by

$$E ::= X \quad | \quad \mathbf{0} \quad | \quad \bar{x}y.E \quad | \quad x(z).E \quad | \quad E|E' \quad | \quad (\nu z)E \quad | \quad !E$$

with x, y, z ranging over $\mathcal{N}$, and X over $\mathcal{X}$, the set of *process variables*. Process variables are not considered in [44], although they are common in languages like CCS [27] that feature a recursion construct. Since process variables form a central part of my notion of a valid or correct translation, here they have simply been added. This works generally. In Sect. 12 I show that for the purpose of accessing whether one language is as expressive as another, translations between them can be assumed to be compositional. This important result would be lost if process variables were dropped from the language. In that case compositionality would need to be stated as a separate requirement for valid translations.

Closed process expressions are called *processes*. The π-calculus is usually presented as a closed-term language, in that the semantic value associated with a closed term is simply itself. Yet, the real semantics is given by a reduction relation between processes, defined below.

Definition 10 An occurrence of a name z in π-calculus process $P \in \mathrm{T}_\pi$ is *bound* if it occurs within a subexpression $x(z).P'$ or $(\nu z)P'$ of P; otherwise it is *free*. Let $\mathsf{n}(P)$ (resp. $\mathsf{bn}(P)$) be the set of names occurring (bound) in $P \in \mathrm{T}_\pi$. *Structural congruence*, $\equiv$, is the smallest congruence relation on processes satisfying

$$\begin{array}{rclrclrcl} P_1|(P_2|P_3) & \equiv & (P_1|P_2)|P_3 & !P & \equiv & P|!P & (\nu w)(P|Q) & \equiv & P|(\nu w)Q \\ P_1|P_2 & \equiv & P_2|P_1 & (\nu z)\mathbf{0} & \equiv & \mathbf{0} & x(z).P & \equiv & x(w).P\{^w/_z\} \\ P|\mathbf{0} & \equiv & P & (\nu z)(\nu w)P & \equiv & (\nu w)(\nu z)P & (\nu z)P & \equiv & (\nu w)P\{^w/_z\} \ . \end{array}$$

Here the rightmost column only holds when $w \notin \mathsf{n}(P)$, and $P\{^w/_z\}$ denotes the process obtained by replacing each free occurrence of z in P by w.

Definition 11 The *reduction relation*, $\rightarrow \subseteq \mathrm{T}_\pi \times \mathrm{T}_\pi$, is generated by the following rules.

$$\frac{}{\bar{x}z.P|x(y).Q \rightarrow P|Q\{^z/_y\}} \ (z \notin \mathsf{bn}(Q))$$

$$\frac{P \rightarrow P'}{P|Q \rightarrow P'|Q} \qquad \frac{P \rightarrow P'}{(\nu z)P \rightarrow (\nu z)P'} \qquad \frac{Q \equiv P \quad P \rightarrow P' \quad P' \equiv Q'}{Q \rightarrow Q'}$$

Let $\Longrightarrow$ be the reflexive and transitive closure of $\rightarrow$. The observable behaviour of π-calculus processes is often stated in terms of the outputs they can produce (abstracting from the value communicated on an output channel).

Definition 12 Let $x \in \mathcal{N}$. A process P has a *strong output barb* on x, notation $P{\downarrow_{\bar{x}}}$, if P can perform an output action $\bar{x}z$. This is defined inductively:

$$(\bar{x}z.(P)){\downarrow_{\bar{x}}} \qquad \frac{P{\downarrow_{\bar{x}}}}{(P|Q){\downarrow_{\bar{x}}}} \qquad \frac{Q{\downarrow_{\bar{x}}}}{(P|Q){\downarrow_{\bar{x}}}} \qquad \frac{P{\downarrow_{\bar{x}}} \quad x \neq z}{((\nu z)P){\downarrow_{\bar{x}}}} \qquad \frac{P{\downarrow_{\bar{x}}}}{(!P){\downarrow_{\bar{x}}}}$$

A process P has a *weak output barb* on x, $P{\Downarrow_{\bar{x}}}$, if there is a P' with $P \Longrightarrow P'{\downarrow_{\bar{x}}}$.

A common semantic equivalence applied in the π-calculus is *weak barbed congruence* [29,44].

Definition 13 *Weak (output) barbed bisimilarity* is the largest symmetric relation $\dot{\approx} \subseteq \mathrm{T}_\pi \times \mathrm{T}_\pi$ such that
- $P \dot{\approx} Q$ and $P{\downarrow_{\bar{x}}}$ implies $Q{\Downarrow_{\bar{x}}}$, and
- $P \dot{\approx} Q$ and $P \Longrightarrow P'$ implies $Q \Longrightarrow Q'$ for some Q' with $P' \dot{\approx} Q'$.

Weak barbed congruence, $\cong^c$, is the largest congruence included in $\dot{\approx}$.

Often *input barbs*, defined similarly, are included in the definition of weak barbed bisimilarity [44]. This is known to induce the same notion of weak barbed congruence [44]. Another technique for defining weak barbed congruence is to use a barb, or set of barbs, external to the language under investigation, that are added to the language as constants [21], similar to the theory of testing of [9]. This method is useful for languages with a reduction semantics that do not feature a clear notion of barb, or where there is ambiguity in which barbs should be counted and which not, or for comparing languages with different kinds of barb.

Example 1 $\bar{x}z.\mathbf{0} \not\cong^c (\nu u)(\bar{x}u.\mathbf{0}|u(v).\bar{v}z.\mathbf{0})$.
For let $E := X|x(u).\bar{u}v.\mathbf{0}$ with $\rho(X) = \bar{x}z.\mathbf{0}$ and $\zeta(X) = (\nu u)(\bar{x}u.\mathbf{0}|u(v).\bar{v}z.\mathbf{0})$.
Then $E[\zeta] \to (\nu u)\big(u(v).\bar{v}z.\mathbf{0}|\bar{u}v.\mathbf{0}\big) \to (\bar{v}z.\mathbf{0}){\downarrow_{\bar{v}}}$ but $(E[\rho]){\not\Downarrow_{\bar{v}}}$.

The asynchronous π-calculus, as introduced by Honda & Tokoro in [24] and by Boudol in [4], is the sublanguage aπ of the fragment π of the π-calculus presented above where all subexpressions $\bar{x}y.E$ have the form $\bar{x}y.\mathbf{0}$. *Asynchronous barbed congruence*, $\cong^c_{\mathrm{a}}$, is the largest congruence *for the asynchronous π-calculus* included in $\dot{\approx}$. Since $a\pi$ is a sublanguage of π, $\cong^c_{\mathrm{a}}$ is at least as coarse an equivalence as $\cong^c$, i.e. $\cong^c \subseteq \cong^c_{\mathrm{a}}$. The inclusion is strict, since $!x(z).\bar{x}z.\mathbf{0} \cong^c_{\mathrm{a}} \mathbf{0}$, yet $!x(z).\bar{x}z.\mathbf{0} \not\cong^c \mathbf{0}$ [44]. Since all expressions used in Example 1 belong to aπ, one even has $\bar{x}z.\mathbf{0} \not\cong^c_{\mathrm{a}} (\nu u)(\bar{x}u.\mathbf{0}|u(v).\bar{v}z.\mathbf{0})$.

Boudol [4] defined a translation $\mathcal{T}$ from π to aπ inductively as follows:

$$\begin{aligned}
\mathcal{T}(X) &= X && \text{for } X \in \mathcal{X}\\
\mathcal{T}(\mathbf{0}) &= \mathbf{0}\\
\mathcal{T}(\bar{x}z.P) &= (u)(\bar{x}u|u(v).(\bar{v}z|\mathcal{T}(P))) && \text{choosing } u,v \notin \mathsf{n}(P),\ u \neq v\\
\mathcal{T}(x(y).P) &= x(u).(v)(\bar{u}v|v(y).\mathcal{T}(P)) && \text{choosing } u,v \notin \mathsf{n}(P),\ u \neq v\\
\mathcal{T}(P|Q) &= (\mathcal{T}(P)|\mathcal{T}(Q))\\
\mathcal{T}(!P) &= \ !\mathcal{T}(P)\\
\mathcal{T}((\nu x)P) &= (\nu x)\mathcal{T}(P)
\end{aligned}$$

Example 1 shows that $\mathcal{T}$ is not valid up to $\cong^c$. In fact, it is not even valid up to $\cong^c_{\mathrm{a}}$. However, as shown in [25], it is valid up to $\dot{\approx}$. Since $\dot{\approx}$ is not a congruence (for π or aπ) it is not correct up to $\dot{\approx}$.

7 Congruence closure

Definition 14 An equivalence relation $\sim$ is a *1-hole congruence* for a language $\mathcal{L}$ interpreted in a semantic domain $\mathbf{V}$ if $[\![E]\!]_{\mathcal{L}}(\nu) \sim [\![E]\!]_{\mathcal{L}}(\rho)$ for any $\mathcal{L}$-expression E and any valuations $\nu, \rho : \mathcal{X} \to \mathbf{V}$ with $\nu \sim^1 \rho$. Here ν, ρ are $\sim^1$*-equivalent*, $\nu \sim^1 \rho$, if $\nu(X) \sim \rho(X)$ for some $X \in \mathcal{X}$ and $\nu(Y) = \rho(Y)$ for all variables $Y \neq X$.

An *n-hole congruence* for any finite $n \in \mathbb{N}$ can be defined in the same vain, and it is well known and easy to check that a 1-hole congruence $\sim$ is also an n-hole congruence, for any $n \in \mathbb{N}$. However, in the presence of operators with infinitely many arguments, a 1-hole congruence need not be a congruence.

Example 2 Let $\mathbf{V}$ be $(\mathbb{N} \times \mathbb{N}) \cup \{\infty\}$, with the well-order $\leq$ on $\mathbf{V}$ inherited lexicographically from the default order on $\mathbb{N}$ and ∞ the largest element. So $(n, m) \leq (n', m')$ iff $n \leq n' \vee (n = m \wedge m \leq m')$. Consider the language $\mathcal{L}$ with constants 0, 1 and (1), interpreted in $\mathbf{V}$ as $(0, 0)$, $(1, 0)$ and $(0, 1)$, respectively, the binary operator $+$, interpreted by $(n_1, m_1) +^{\mathbf{V}} (n_2, m_2) = (n_1{+}n_2, m_1{+}m_2)$ and $\infty + E = E + \infty = \infty$, and the construct $\sup(E_i)_{i \in I}$ that takes any number of arguments (dependent on the set of the index sets I). The interpretation of sup in $\mathbf{V}$ is to take the supremum of its arguments w.r.t. the well-order $\leq$. In case sup is given finitely many arguments, it simply returns the largest. However $\sup((n, i))_{i \in \mathbb{N}} = (n{+}1, 0)$.

Now let the equivalence relation $\sim$ on $\mathbf{V}$ be defined by $(n, m) \sim (n', m')$ iff $n = n'$, leaving ∞ in an equivalence class of its own. This relation is a 1-hole congruence on $\mathcal{L}$. Hence, it is also a 2-hole congruence, so one has

$$\big((n_1, m_1){\sim}(n_1', m_1') \wedge (n_2, m_2){\sim}(n_2', m_2')\big) \Rightarrow (n_1, m_1){+}(n_2, m_2) \sim (n_1', m_1'){+}(n_2', m_2').$$

Yet it fails to be a congruence: $(n, i) \sim (n, 0)$ for all $i \in \mathbb{N}$, but

$$(n{+}1, 0) = \sup((n, i))_{i \in \mathbb{N}} \not\sim \sup((n, 0))_{i \in \mathbb{N}} = (n, 0).$$

It is well known and easy to check that the collection of equivalence relations on any domain $\mathbf{V}$, ordered by inclusion, forms a complete lattice—namely the intersection of arbitrary many equivalence relations is again an equivalence relation. Likewise, the collection of 1-hole congruences for $\mathcal{L}$ is also a complete lattice, and moreover a complete sublattice of the complete lattice of equivalence relations on $\mathbf{V}$. The latter implies that for any collection C of 1-hole congruence relations, the least equivalence relation that contains all elements of C (exists and) happens to be a 1-hole congruence relation. Again, this is a property that is well known [22] and easy to prove. It follows that for any equivalence relation $\sim$ there exists a largest 1-hole congruence for $\mathcal{L}$ contained in $\sim$. I will denote this 1-hole congruence by $\sim_{\mathcal{L}}^{1c}$, and call it the *congruence closure* of $\sim$ w.r.t. $\mathcal{L}$. One has $v_1 \sim_{\mathcal{L}}^{1c} v_2$ for $v_1, v_2 \in \mathbf{V}$ iff $[\![E]\!]_{\mathcal{L}}(\nu) \sim [\![E]\!]_{\mathcal{L}}(\rho)$ for any $\mathcal{L}$-expression E and any valuations $\nu, \rho : \mathcal{X} \to \mathbf{V}$ with $\nu(X) = v_1$ and $\rho(X) = v_2$ for some $X \in \mathcal{X}$ and $\nu(Y) = \rho(Y)$ for all $Y \neq X$. Such results do not generally hold for congruences.

Example 3 Continue Example 2, but skipping the operator $+$. Let $\sim_k$ be the equivalence on $\mathbf{V}$ defined by $(n, m) \sim_k (n', m')$ iff $n = n' \wedge (m = m' \vee m, m' \leq k)$. It is easy to check that all $\sim_k$ for $k \in \mathbb{N}$ are congruences on the reduced $\mathcal{L}$, and contained in $\sim$. Yet their least upper bound (in the lattice of equivalence relations on $\mathbf{V}$) is $\sim$, which is not a congruence itself. In particular, there is no largest congruence contained in $\sim$.

When dealing with languages $\mathcal{L}$ in which all operators and other constructs have a finite arity, so that each $E \in \mathbb{T}_{\mathcal{L}}$ contains only finitely many variables, there is no difference between a congruence and a 1-hole congruence, and thus

$\sim_{\mathcal{L}}^{1c}$ is a congruence relation for any equivalence $\sim$. I will apply the theory of expressiveness presented in this paper also to languages like CCS that have operators (such as $\sum_{i\in I} E_i$) of infinite arity. However, in all such cases I'm currently aware of, the relevant choices of $\mathcal{L}$ and $\sim$ have the property that $\sim_{\mathcal{L}}^{1c}$ is in fact a congruence relation. As an example, consider weak bisimilarity [27]. This equivalence relation fails to be a congruence for $\sum$. However, the coarsest 1-hole congruence contained in this relation, often called *rooted* weak bisimilarity, happens to be a congruence. In fact, when congruence-closing weak bisimilarity w.r.t. the binary sum, the result [15] is also a congruence for the infinitary sum, as well as for all other operators of CCS [27].

Definition 15 Let $\mathcal{T}$ be a translation from $\mathcal{L}$ into $\mathcal{L}'$. A subset $\mathbf{W}$ of $\mathbf{V}'$ is *closed* under $\mathcal{T}(\mathcal{L})$ if $[\![\mathcal{T}(E)]\!](\eta) \in \mathbf{W}$ for any expression $E \in \mathbb{T}_{\mathcal{L}}$ and valuation $\eta : \mathcal{X} \to \mathbf{W}$. An equivalence $\sim$ on $\mathbf{W}$ is a *congruence* (respectively *1-hole congruence*) for $\mathcal{T}(\mathcal{L})$ on $\mathbf{W}$ if for any $E \in \mathbb{T}_{\mathcal{L}}$ and $\theta, \eta : \mathcal{X} \to \mathbf{W}$ with $\theta \sim \eta$ (respectively $\theta \sim^1 \eta$) one has $[\![\mathcal{T}(E)]\!]_{\mathcal{L}'}(\theta) \sim [\![\mathcal{T}(E)]\!]_{\mathcal{L}'}(\eta)$.

Proposition 3 Let $\mathcal{T}$ be a translation from $\mathcal{L}$ into $\mathcal{L}'$ that is correct w.r.t. a semantic translation $\mathbf{R} \subseteq \mathbf{V}' \times \mathbf{V}$. Let $\mathbf{R}(\mathbf{V}) := \{v' \in \mathbf{V}' \mid \exists v \in \mathbf{V}.\ v'\ \mathbf{R}\ v\}$. Then $\mathbf{R}(\mathbf{V})$ is closed under $\mathcal{T}(\mathcal{L})$.

Proof: Let $E \in \mathbb{T}_{\mathcal{L}}$ and $\eta : \mathcal{X} \to \mathbf{R}(\mathbf{V})$. Take $\rho : \mathcal{X} \to \mathbf{V}$ with $\rho\ \mathbf{R}\ \eta$. Then $[\![\mathcal{T}(E)]\!]_{\mathcal{L}'}(\eta)\ \mathbf{R}\ [\![E]\!]_{\mathcal{L}}(\rho)$. Since $[\![E]\!]_{\mathcal{L}}(\rho) \in \mathbf{V}$ one has $[\![\mathcal{T}(E)]\!]_{\mathcal{L}'}(\eta) \in \mathbf{R}(\mathbf{V})$. □

Proposition 4 Let the translation $\mathcal{T}$ from $\mathcal{L}$ into $\mathcal{L}'$ be correct w.r.t. the semantic translation $\mathbf{R} \subseteq \sim$. Then $\sim$ is a (1-hole) congruence for $\mathcal{L}$ iff it is a (1-hole) congruence for $\mathcal{T}(\mathcal{L})$ on $\mathbf{R}(\mathbf{V})$.

Proof: First suppose $\sim$ is a congruence for $\mathcal{L}$. Let $E \in \mathbb{T}_{\mathcal{L}}$ and $\theta, \eta : \mathcal{X} \to \mathbf{R}(\mathbf{V})$ with $\theta \sim \eta$. By the definition of $\mathbf{R}(\mathbf{V})$ there are valuations $\nu, \rho : \mathcal{X} \to \mathbf{V}$ with $\theta\ \mathbf{R}\ \nu$ and $\eta\ \mathbf{R}\ \rho$. Now $\nu \sim \theta \sim \eta \sim \rho$, so

$$[\![\mathcal{T}(E)]\!]_{\mathcal{L}'}(\theta)\ \mathbf{R}\ [\![E]\!]_{\mathcal{L}}(\nu) \sim [\![E]\!]_{\mathcal{L}}(\rho)\ \mathbf{R}^{-1}\ [\![\mathcal{T}(E)]\!]_{\mathcal{L}'}(\eta)$$

and hence $[\![\mathcal{T}(E)]\!]_{\mathcal{L}'}(\theta) \sim [\![\mathcal{T}(E)]\!]_{\mathcal{L}'}(\eta)$. The other direction proceeds in the same way.

Now suppose $\sim$ is a 1-hole congruence for $\mathcal{L}$. Let $E \in \mathbb{T}_{\mathcal{L}}$ and $\theta, \eta : \mathcal{X} \to \mathbf{R}(\mathbf{V})$ with $\theta \sim^1 \eta$. Then $\theta(X) \sim \eta(X)$ for some $X \in \mathcal{X}$ and $\theta(Y) = \eta(Y)$ for all $Y \neq X$. So there must be $\nu, \rho : \mathcal{X} \to \mathbf{V}$ with $\theta\ \mathbf{R}\ \nu$, $\eta\ \mathbf{R}\ \rho$ and $\nu(Y) = \rho(Y)$ for all $Y \neq X$. Since $\nu(X) \sim \theta(X) \sim \eta(X) \sim \rho(X)$ it follows that $\nu \sim^1 \rho$. The conclusion proceeds as above, and the other direction goes likewise. □

The requirement of being a congruence for $\mathcal{T}(\mathcal{L})$ on $\mathbf{R}(\mathbf{V})$ is slightly weaker than that of being a congruence for $\mathcal{T}(\mathcal{L})$—cf. Def. 8—for it proceeds by restricting attention to valuations into $\mathbf{R}(\mathbf{V}) \subseteq \mathbf{U}$. ¶

8 A congruence closure property for valid translations

In many applications, semantic values in the domain of interpretation of a language $\mathcal{L}$ are only meaningful up to a semantic equivalence $\sim^c$, and the intended

semantic domain could just as well be seen as the set of $\sim^c$-equivalence classes of values. For this purpose it is essential that $\sim^c$ is a congruence for $\mathcal{L}$. Often $\sim^c$ is the congruence closure of a coarser semantic equivalence $\sim$, so that two values end up being identified iff they are $\sim$-equivalent in every context. An example of this occurred in Sect. 6, with $\dot{\approx}$ in the rôle of $\sim$ and $\cong^c$ in the rôle of $\sim^c$. Now Thm. 4, contributed in this section, says that if a translation from $\mathcal{L}$ into $\mathcal{L}'$ is valid up to $\sim$, then it is even valid up to an equivalence $\sim^{1c}_{\mathcal{L},\mathbf{R}}$ that extends $\sim^c$ from $\mathbf{V}$ to a subdomain $\mathbf{W}$ of $\mathbf{V}'$ that suffices for the interpretation of translated expressions from $\mathcal{L}$. This equivalence $\sim^{1c}_{\mathcal{L},\mathbf{R}}$ coincides with the congruence closure of $\sim$ on $\mathcal{L}$, as well as on $\mathcal{T}(\mathcal{L})$, and melts each equivalence class of $\mathbf{V}$ with exactly one of $\mathbf{W}$, and vice versa.

Let $\mathcal{L}$ and $\mathcal{L}'$ be languages with $[\![\ ]\!]_{\mathcal{L}} : \mathbb{T}_{\mathcal{L}} \to ((\mathcal{X} \to \mathbf{V}) \to \mathbf{V})$ and $[\![\ ]\!]_{\mathcal{L}'} : \mathbb{T}_{\mathcal{L}'} \to ((\mathcal{X} \to \mathbf{V}') \to \mathbf{V}')$. In this section I assume that $\mathbf{V} \cap \mathbf{V}' = \emptyset$. To apply the results to the general case, just adapt $\mathcal{L}'$ by using a copy of $\mathbf{V}'$—any preorder $\precsim$ on $\mathbf{V} \cup \mathbf{V}'$ extends to this copy by considering each copied element $\precsim$-equivalent to the original.

Definition 16 Given any semantic translation $\mathbf{R}$, let $\equiv_{\mathbf{R}} \subseteq (\mathbf{V} \cup \mathbf{V}')^2$ be the smallest equivalence relation on $\mathbf{V} \cup \mathbf{V}'$ containing $\mathbf{R}$.

Theorem 3 If a translation $\mathcal{T}$ is correct w.r.t. the semantic translation $\mathbf{R}$, then $\equiv_{\mathbf{R}}$ is a 1-hole congruence for $\mathcal{L}$. ¶

By Prop. 4 $\equiv_{\mathbf{R}}$ also is a 1-hole congruence for $\mathcal{T}(\mathcal{L})$ on $\mathbf{R}(\mathbf{V})$. Only the subset $\mathbf{R}(\mathbf{V})$ of $\mathbf{V}'$ matters for the purpose of translating $\mathcal{L}$ into $\mathcal{L}'$. On $\mathbf{V}' \setminus \mathbf{R}(\mathbf{V})$ the equivalence $\equiv_{\mathbf{R}}$ is the identity.

Theorem 4 Let $\mathcal{T}$ be a translation from a language $\mathcal{L}$, with semantic domain $\mathbf{V}$, into a language $\mathcal{L}'$, with domain $\mathbf{V}'$, that is valid up to a semantic equivalence $\sim$. Then $\mathcal{T}$ is even valid up to a semantic equivalence $\sim^{1c}_{\mathcal{L},\mathbf{R}}$, contained in $\sim$, such that (1) the restriction of $\sim^{1c}_{\mathcal{L},\mathbf{R}}$ to $\mathbf{V}$ is the largest 1-hole congruence for $\mathcal{L}$ contained in $\sim$, (2) the set $\mathbf{W} := \{v \in \mathbf{V}' \mid \exists v \in \mathbf{V}.\ v' \sim^{1c}_{\mathcal{L},\mathbf{R}} v\}$ is closed under $\mathcal{T}(\mathcal{L})$, and (3) the restriction of $\sim^{1c}_{\mathcal{L},\mathbf{R}}$ to $\mathbf{W}$ is the largest 1-hole congruence for $\mathcal{T}(\mathcal{L})$ on $\mathbf{W}$ that is contained in $\sim$. ¶

Note that each equivalence class of $\sim^{1c}_{\mathcal{L},\mathbf{R}}$ on $\mathbf{V} \cup \mathbf{W}$ melts an equivalence class of $\sim^{1c}_{\mathcal{L},\mathbf{R}}$ on $\mathbf{V}$ with one of $\sim^{1c}_{\mathcal{L},\mathbf{R}}$ on $\mathbf{W}$. Moreover, on $\mathbf{V}$ the relation is completely determined by $\mathcal{L}$ and $\sim$. However, in general the whole relation $\sim^{1c}_{\mathcal{L},\mathbf{R}}$ is not completely determined by $\mathcal{L}$ and $\sim$. ¶

Corollary 1 Let $\mathcal{T}$ be a translation from a language $\mathcal{L}$, with semantic domain $\mathbf{V}$, into a language $\mathcal{L}'$, with domain $\mathbf{V}'$, valid up to a semantic equivalence $\sim$, and suppose the congruence closure $\sim^1_{\mathcal{L}}$ of $\sim$ w.r.t. $\mathcal{L}$ is in fact a congruence. Then $\mathcal{T}$ is correct up to the equivalence $\sim^{1c}_{\mathcal{L},\mathbf{R}}$ described in Thm. 4. ¶

The languages π and aπ of Sect. 6 do not feature operators (or other constructs) of infinite arity. Hence the congruence closure $\sim^{1c}_{\pi}$ or $\sim^{1c}_{\mathrm{a}\pi}$ of an equivalence $\sim$ on π or aπ is always a congruence. So by Cor. 1 Boudol's translation $\mathcal{T}$ is correct up to an equivalence $\dot{\approx}^{c}_{\pi,\mathbf{R}}$, defined on the disjoint union of the domains T_π and $\mathrm{T}_{\mathrm{a}\pi}$ on which the two languages are interpreted. This equivalence is contained in $\dot{\approx}$, and on the source domain T_π coincides with $\cong^c$. By Thm. 4, the

restriction of $\stackrel{\bullet}{\approx}^{c}_{\pi,\mathbf{R}}$ to a subdomain $\mathbf{W} \subseteq \mathrm{T}_{\mathrm{a}\pi}$ is the largest congruence for $\mathcal{T}(\pi)$ on $\mathbf{W}$ that is contained in $\sim$. As $\cong^{c}_{\mathrm{a}}$ is a congruence for all of $\mathrm{a}\pi$ on all of $\mathrm{T}_{\mathrm{a}\pi}$, and contained in $\stackrel{\bullet}{\approx}$, it is certainly a congruence for $\mathcal{T}(\pi)$ on $\mathbf{W}$, and thus contained in $\stackrel{\bullet}{\approx}^{c}_{\pi,\mathbf{R}}$. This inclusion turns out to be strict. As an illustration of that, note that $\bar{x}z.\mathbf{0}|\bar{x}z.\mathbf{0} \cong^{c} \bar{x}z.\bar{x}z\mathbf{0}$. (This follows since these processes are strong (early) bisimilar [44] and thus strong full bisimilar by [44, Def. 2.2.2].) Consequently, their translations must be related by $\stackrel{\bullet}{\approx}^{c}_{\pi,\mathbf{R}}$. So, for distinct $u, v, y, w, x, z \in \mathcal{N}$,

$$(u)(\bar{x}u|u(v).(\bar{v}z|\mathbf{0}))\big|(u)(\bar{x}u|u(v).(\bar{v}z|\mathbf{0})) \stackrel{\bullet}{\approx}^{c}_{\pi,\mathbf{R}} (y)(\bar{x}y|u(w).(\bar{w}z|(u)(\bar{x}u|u(v).(\bar{v}z|\mathbf{0})))).$$

Yet, these processes are not $\cong^{c}_{\mathrm{a}}$-equivalent, as can be seen by putting them in a context $x(y).x(y).\bar{r}(s)|X$. There, only the left-hand side has a weak barb $\Downarrow_{\bar{r}}$.

9 Integrating language features through translations

The results of the previous section show how valid translations are satisfactory for comparing the expressiveness of languages. If there is a valid translation $\mathcal{T}$ from $\mathcal{L}$ to $\mathcal{L}'$ up to $\sim$, and (as usual) $\sim^{1c}_{\mathcal{L}}$ is a congruence, then all truths that can be expressed in terms of $\mathcal{L}$ can be mimicked in $\mathcal{L}'$. For the congruence classes of $\sim^{1c}_{\mathcal{L}}$ translate bijectively to congruence classes of an induced equivalence relation on the domain of $\mathcal{T}(\mathcal{L})$ (within the domain of $\mathcal{L}'$), and all operations on those congruence classes that can be performed by contexts of $\mathcal{L}$ have a perfect counterpart in terms of contexts of $\mathcal{T}(\mathcal{L})$. This state of affairs was illustrated on Boudol's translation from a synchronous to an asynchronous π-calculus.

There is however one desirable property of translations between languages that has not yet been achieved, namely to combine the powers of two languages into one unified language. If both languages $\mathcal{L}_1$ and $\mathcal{L}_2$ have valid translations into a language $\mathcal{L}'$, then all that can be done with $\mathcal{L}_1$ can be mimicked in a fragment of $\mathcal{L}'$, and all that can be done with $\mathcal{L}_2$ can be mimicked in another fragment of $\mathcal{L}'$. In order for these two fragments to combine, one would like to employ a single congruence relation on $\mathcal{L}'$ that specialises to congruence relations for $\mathcal{T}_1(\mathcal{L}_1)$ and $\mathcal{T}_2(\mathcal{L}_2)$, which form the counterparts of relevant congruence relations for the source languages $\mathcal{L}_1$ and $\mathcal{L}_2$.

In terms of the translation $\mathcal{T}$ from π to $\mathrm{a}\pi$, the equivalence $\cong^{c}_{\mathrm{a}}$ on $\mathrm{T}_{\mathrm{a}\pi}$ would be the right congruence relation to consider for $\mathrm{a}\pi$. Ideally, this congruence would extend to an equivalence $\cong^{c}_{\pi,\mathrm{a}\pi}$ on the disjoint union $\mathrm{T}_{\pi} \uplus \mathrm{T}_{a\pi}$, such that the restriction of $\cong^{c}_{\pi,\mathrm{a}\pi}$ to T_{π} is a congruence for π. Necessarily, this congruence on T_{π} would have to distinguish the terms $\bar{x}z.\mathbf{0}|\bar{x}z.\mathbf{0}$ and $\bar{x}z.\bar{x}z\mathbf{0}$, since their translations are distinguished by $\cong^{c}_{\mathrm{a}}$. One therefore expects $\cong^{c}_{\pi,\mathrm{a}\pi}$ on T_{π} to be strictly finer than $\cong^{c}$. Here it is important that the union of T_{π} and $\mathrm{T}_{a\pi}$ on which this congruence is defined is required to be disjoint. For if one considers $\mathrm{T}_{a\pi}$ as a subset of T_{π}, then we obtain that the restriction of $\cong^{c}_{\pi,\mathrm{a}\pi}$ to that subset (1) coincides with $\cong^{c}_{\mathrm{a}}$ and (2) is strictly finer than $\cong^{c}$. This contradicts the fact that $\cong^{c}$ is strictly finer than $\cong^{c}_{\mathrm{a}}$.

In Sect. 12 I will show that such a congruence $\cong^{c}_{\pi,\mathrm{a}\pi}$ indeed exists. In fact, under a few very mild conditions this result holds generally, provided that the source language $\mathcal{L}$ is a closed-term language. ¶

10 A unique decomposition of terms

The results of Sect. 12 apply only to languages satisfying two postulates, formulated below, and to preorders $\precsim$ that "respect $\overset{\alpha}{=}$", defined in Sect. 11.

Definition 17 *α-conversion* is the act of renaming all occurrences of a bound variable X within the scope of its binding into another variable, say Y, while avoiding capture of free variables. Here one speaks of *capture* when a free occurrence of Y turns into a bound one.

Write $E \overset{\alpha}{=} F$ if expression E can be converted into F by acts of α-conversion. In languages where there are multiple types of bound variables, $\overset{\alpha}{=}$ allows conversion of all of them. In a π-calculus with recursion, for instance, there could be bound process variables $X \in \mathcal{X}$ as well as bound names $x \in \mathcal{N}$. The last two conversions in the right column of Def. 10 define α-conversion for names.

Postulate 1 ([16], paraphrased) There exists a class of expressions called *standard heads*, and a class of substitutions called *standard substitutions*, such that for each expression E, if not a variable, there are unique standard heads H and substitutions σ such that $E \overset{\alpha}{=} H[\sigma]$.

A term $f(c, g(c))$, for instance, can be written as $H[\sigma]$ where $H = f(X_1, X_2)$ is a head, and $\sigma : \{X_1, X_2\} \to \mathbb{T}_{\mathcal{L}}$ is given by $\sigma(X_1) = c$ and $\sigma(X_2) = g(c)$. The head H is standardised by means of a particular (arbitrary) choice for its argument variables X_1 and X_2. σ is standardised through a particular choice of the bound variables that may occur in the expressions $\sigma(X)$. A head for a recursive expression $\mu X.f(g(c), g(g(X)))$ is $\mu X.f(Y, g(g(X)))$. See [16] for further detail.

This postulate is easy to show for each common type of system description language, and I am not aware of any counterexamples. However, while striving for maximal generality, I consider languages with (recursion-like) constructs that are yet to be invented, and in view of those, this principle has to be postulated rather than derived.

11 Invariance of meaning under α-conversion

Write $v \overset{\alpha}{=}_{\mathcal{L}} w$, with $v, w \in \mathbf{V}$, iff there are terms $E, F \in \mathbb{T}_{\mathcal{L}}$ with $E \overset{\alpha}{=} F$, and a valuation $\zeta : \mathcal{X} \to \mathbf{V}$ such that $[\![E]\!]_{\mathcal{L}}(\zeta) = v$ and $[\![F]\!]_{\mathcal{L}}(\zeta) = w$. This relation is reflexive and symmetric.

In [16] I limited attention to languages satisfying

$$\textit{if } E \overset{\alpha}{=} F \textit{ then } [\![E]\!]_{\mathcal{L}} = [\![F]\!]_{\mathcal{L}}. \tag{2}$$

This postulate says that the meaning of an expression is invariant under α-conversion. It can be reformulated as the requirement that $\overset{\alpha}{=}_{\mathcal{L}}$ is the identity relation. This postulate is satisfied by all my intended applications, except for the important class of closed-term languages. Languages like CCS and the π-calculus can be regarded as falling in this class (although it is also possible to declare the meaning of a term under a valuation to be an $\overset{\alpha}{=}$-equivalence class of closed terms). To bring this type of application within the scope of my theory, here I weaken this postulate by requiring merely that $\overset{\alpha}{=}_{\mathcal{L}}$ is an equivalence.

Postulate 2 $\overset{\alpha}{=}_{\mathcal{L}}$ is an equivalence relation.

This postulate is needed in Sect. 12. I also need to restrict attention to preorders $\lesssim$ with $\stackrel{\alpha}{=}_{\mathcal{L}} \subseteq \lesssim$. When that holds I say that the preorder $\lesssim$ *respects* $\stackrel{\alpha}{=}_{\mathcal{L}}$. If (2) holds—which strengthens of Postulate 2—then *any* preorder respects $\stackrel{\alpha}{=}_{\mathcal{L}}$.

12 Compositionality

An important property of translations, defined below, is *compositionality*. In this section show I that any valid translation up to a preorder $\lesssim$ can be modified into such a translation that moreover is compositional, provided one restricts attention to languages that satisfy Postulates 1 and 2, and preorders $\lesssim$ that respect $\stackrel{\alpha}{=}$.

Definition 18 A translation $\mathcal{T}$ from $\mathcal{L}$ into $\mathcal{L}'$ is *compositional* if
(1) $\mathcal{T}(E[\sigma]) \stackrel{\alpha}{=} \mathcal{T}(E)[\mathcal{T} \circ \sigma]$ for each $E \in \mathbb{T}_{\mathcal{L}}$ and $\sigma : fv(E) \to \mathbb{T}_{\mathcal{L}}$,
(2) $E \stackrel{\alpha}{=} F$ implies $\mathcal{T}(E) \stackrel{\alpha}{=} \mathcal{T}(F)$ for all $E, F \in \mathbb{T}_{\mathcal{L}}$,
(3) and moreover $\mathcal{T}(X) = X$ for each $X \in \mathcal{X}$.
In case $E = f(t_1, \ldots, t_n)$ for certain $t_i \in \mathbb{T}_{\mathcal{L}}$ this amounts to $\mathcal{T}(f(t_1, \ldots, t_n)) \stackrel{\alpha}{=} E_f(\mathcal{T}(t_1), \ldots, \mathcal{T}(t_n))$, where $E_f := \mathcal{T}(f(X_1, \ldots, X_n))$ and $E_f(u_1, \ldots, u_n)$ denotes the result of the simultaneous substitution in this expression of the terms $u_i \in \mathbb{T}_{\mathcal{L}'}$ for the free variables X_i, for $i = 1, \ldots, n$. The first requirement of Def. 18 is more general and covers language constructs other than functions, such as recursion. Requiring equality rather than $\stackrel{\alpha}{=}$ is too demanding. ¶

Lemma 1 If $\mathcal{T}_1 : \mathbb{T}_{\mathcal{L}_1} \to \mathbb{T}_{\mathcal{L}_2}$ and $\mathcal{T}_2 : \mathbb{T}_{\mathcal{L}_2} \to \mathbb{T}_{\mathcal{L}_3}$ are compositional translations, then so is their composition $\mathcal{T}_2 \circ \mathcal{T}_1 : \mathbb{T}_{\mathcal{L}_1} \to \mathbb{T}_{\mathcal{L}_3}$, defined by $\mathcal{T}_2 \circ \mathcal{T}_1(E) := \mathcal{T}_2(\mathcal{T}_1(E))$ for all $E \in \mathcal{L}_1$.

Proof: (1) $\mathcal{T}_2(\mathcal{T}_1(E[\sigma])) \stackrel{\alpha}{=} \mathcal{T}_2(\mathcal{T}_1(E)[\mathcal{T}_1 \circ \sigma]) \stackrel{\alpha}{=} \mathcal{T}_2(\mathcal{T}_1(E))[\mathcal{T}_2 \circ \mathcal{T}_1 \circ \sigma])$ for each $\sigma : \mathcal{X} \rightharpoonup \mathbb{T}_{\mathcal{L}_1}$ and $E \in \mathbb{T}_{\mathcal{L}_1}$. Here the derivation of the first $\stackrel{\alpha}{=}$ uses Property (2) of Def. 18—and this is the reason for requiring that property.
(2) $E \stackrel{\alpha}{=} F$ implies $\mathcal{T}_1(E) \stackrel{\alpha}{=} \mathcal{T}_1(F)$ and $\mathcal{T}_2(\mathcal{T}_1(E)) \stackrel{\alpha}{=} \mathcal{T}_2(\mathcal{T}_1(F))$ for all $E, F \in \mathbb{T}_{\mathcal{L}}$.
(3) $\mathcal{T}_2(\mathcal{T}_1(X)) = \mathcal{T}_2(X) = X$ for each $X \in \mathcal{X}$. □

Theorem 5 Let $\mathcal{L}$ and $\mathcal{L}'$ be languages that satisfy Postulates 1 and 2, and $\lesssim$ a preorder that respects $\stackrel{\alpha}{=}_{\mathcal{L}}$ and $\stackrel{\alpha}{=}_{\mathcal{L}'}$. If any valid (or correct) translation from $\mathcal{L}$ into $\mathcal{L}'$ up to $\lesssim$ exists, then there exists a compositional translation that is valid (or correct) up to $\lesssim$. ¶

Hence, for the purpose of comparing the expressive power of languages, valid translations between them can be assumed to be compositional. For correct translations this was already established in [16], but assuming (2), a stronger version of Postulate 2.

I can now establish the theorem promised in Sect. 9. In view of Thm. 5, no great sacrifices are made by assuming that the translation $\mathcal{T}$ is compositional. Other "mild conditions" needed are Postulate 2 for $\mathcal{L}'$ and $\approx$ respecting $\stackrel{\alpha}{=}_{\mathcal{L}'}$.

Theorem 6 Let $\mathcal{L}$ be a closed-term language and $\mathcal{L}'$ a language that satisfies Postulate 2. Let $\mathcal{T}$ be a compositional translation from $\mathcal{L}$ into $\mathcal{L}'$ that is valid up to $\sim$. Let $\approx$ be any congruence for $\mathcal{L}'$ containing $\stackrel{\alpha}{=}_{\mathcal{L}'}$ and contained in $\sim$. Then $\mathcal{T}$ is correct up to an equivalence $\approx_{\mathcal{T}}$ on $\mathbf{V} \cup \mathbf{V}'$, contained in $\sim$, that on $\mathbf{V}'$ coincides with $\approx$. ¶

13 Related work

The concept of *full abstraction* stems from Milner [26]. It indicates a satisfactory connection between a denotational and an operational semantics of a language. Riecke [42] and Shapiro [45] adapt this notion to translations between languages.

Definition 19 A translation $\mathcal{T}\colon \mathbb{T}_{\mathcal{L}_\mathrm{S}} \to \mathbb{T}_{\mathcal{L}_\mathrm{T}}$ is *fully abstract* w.r.t. the equivalences $\sim_\mathrm{S} \subseteq \mathrm{T}^2_{\mathcal{L}_\mathrm{S}}$ and $\sim_\mathrm{T} \subseteq \mathrm{T}^2_{\mathcal{L}_\mathrm{T}}$ if, for all $P, Q \in \mathrm{T}_{\mathcal{L}_\mathrm{S}}$, $P \sim_\mathrm{S} Q \Leftrightarrow \mathcal{T}(P) \sim_\mathrm{T} \mathcal{T}(Q)$.

In [42,45], $\sim_\mathrm{S}$ and $\sim_\mathrm{T}$ are required to be congruence closures—see [18] for more detail. The simplified definition above was used in [31,30,1]. Fu [10] bases a theory of expressiveness on full abstraction, with a divergence-preserving form of barbed branching bisimilarity [19] in the rôle of $\sim_\mathrm{S}$ and $\sim_\mathrm{T}$. A comparison of full abstraction with the approach of the present paper appears in [18].

In the last twenty years, a great number of encodability and separation results have appeared, comparing CCS, Mobile Ambients, and several versions of the π-calculus (with and without recursion; with mixed choice, separated choice or asynchronous) [43,2,34,31,32,30,8,5,7,1,33,41,46,6,38,40,23,11,12,13,39]; see [20], [21] for an overview. Many of these results employ different and somewhat ad-hoc criteria on what constitutes a valid encoding, and thus are hard to compare with each other. Several of these criteria are discussed and compared in [35] and [36]. Gorla [21] collected some essential features of these approaches and integrated them in a proposal for a valid encoding that justifies most encodings and some separation results from the literature.

Like Boudol [3] and the present paper, Gorla requires a compositionality condition for encodings. However, his criterion is weaker than mine (cf. Def. 18) in that the expression E_f encoding an operator f may be dependent on the set of names occurring freely in the expressions given as arguments of f. This issue is further discussed in [16]. It is an interesting topic for future research to see if there are any valid encodability results à la [21] that suffer from my proposed strengthening of compositionality.

The second criterion of [21] is a form of invariance under name-substitution. It serves to partially undo the effect of making the compositionality requirement name-dependent. In my setting I have not yet found the need for such a condition. In [16] I argue that this criterion as formalised in [21] is too restrictive.

The remaining three requirements of Gorla (the 'semantic' requirements) are very close to an instantiation of mine with a particular preorder $\mathrel{\wasylozenge}$. If one takes $\mathrel{\wasylozenge}$ to be weak barbed bisimilarity with explicit divergence (i.e. relating divergent states with divergent states only), using barbs external to the language, as discussed in Sect. 6, then an valid translation in my sense satisfies Gorla's semantic criteria, provided that the equivalence $\equiv$ on the target language that acts as a parameter in Gorla's third criterion is also taken to be weak barbed bisimilarity with explicit divergence. The precise relationships between the proposals of [16] and [21] are further discussed in [37].

Further work is needed to sort out to what extent the two approaches have relevant differences when evaluating encoding and separation results from the literature. Another topic for future work is to sort out how dependent known encoding and separation results are on the chosen equivalence or preorder.

References

1. M. Baldamus, J. Parrow & B. Victor (2005): *A Fully Abstract Encoding of the pi-Calculus with Data Terms.* In: Proc. ICALP'05, LNCS 3580, Springer, pp. 1202–1213, doi:10.1007/11523468_97.
2. M. Boreale (1998): *On the Expressiveness of Internal Mobility in Name-Passing Calculi.* TCS 195(2), pp. 205–226, doi:10.1016/S0304-3975(97)00220-X.
3. G. Boudol (1985): *Notes on algebraic calculi of processes.* In K. Apt, editor: *Logics and Models of Concurrent Systems*, Springer, pp. 261–303. NATO ASI Series F13.
4. G. Boudol (1992): *Asynchrony and the π-calculus (Note).* Tech. Rep. 1702, INRIA.
5. N. Busi, M. Gabbrielli & G. Zavattaro (2009): *On the expressive power of recursion, replication and iteration in process calculi. Mathematical Structures in Computer Science* 19(6), pp. 1191–1222, doi:10.1017/S096012950999017X.
6. D. Cacciagrano, F. Corradini, J. Aranda & F.D. Valencia (2008): *Linearity, Persistence and Testing Semantics in the Asynchronous Pi-Calculus.* ENTCS 194(2), pp. 59–84, doi:10.1016/j.entcs.2007.11.006.
7. M. Carbone & S. Maffeis (2003): *On the Expressive Power of Polyadic Synchronisation in pi-calculus. Nordic Journal of Computing* 10(2), pp. 70–98.
8. L. Cardelli & A.D. Gordon (2000): *Mobile ambients.* TCS 240(1), pp. 177–213, doi:10.1016/S0304-3975(99)00231-5.
9. R. De Nicola & M. Hennessy (1984): *Testing equivalences for processes.* TCS 34, pp. 83–133, doi:10.1016/0304-3975(84)90113-0.
10. Y. Fu (2016): *Theory of interaction. Theoretical Computer Science* 611, pp. 1–49, doi:10.1016/j.tcs.2015.07.043.
11. T. Given-Wilson (2014): *Expressiveness via Intensionality and Concurrency.* In: Proc. ICTAC'14, LNCS 8687, pp. 206–223, doi:10.1007/978-3-319-10882-7_13.
12. T. Given-Wilson (2014): *On the Expressiveness of Intensional Communication.* In: Proc. EXPRESS/SOS'14, EPTCS 160, pp. 30–46, doi:10.4204/EPTCS.160.4.
13. T. Given-Wilson & A. Legay (2015): *On the Expressiveness of Joining.* In: Proc. ICE'15, EPTCS 189, pp. 99–113, doi:10.4204/EPTCS.189.9.
14. R.J. van Glabbeek (1994): *On the expressiveness of ACP (extended abstract).* In: Proc. ACP'94, Workshops in Computing, Springer, pp. 188–217. Available at http://theory.stanford.edu/~rvg/abstracts.html#31.
15. R.J. van Glabbeek (2005): *A Characterisation of Weak Bisimulation Congruence.* In: *Processes, Terms and Cycles*, LNCS 3838, pp. 26–39, doi:10.1007/11601548_4.
16. R.J. van Glabbeek (2012): *Musings on Encodings and Expressiveness.* In: Proc. EXPRESS/SOS'12, EPTCS 89, pp. 81–98, doi:10.4204/EPTCS.89.7.
17. R.J. van Glabbeek (2017): *Lean and Full Congruence Formats for Recursion.* In: Proc. LICS'17, doi:10.1109/LICS.2017.8005142.
18. R.J. van Glabbeek (2018): *A Theory of Encodings and Expressiveness.* Available at http://theory.stanford.edu/~rvg/abstracts.html#tra. Full version of current paper.
19. R.J. van Glabbeek, B. Luttik & N. Trčka (2009): *Branching Bisimilarity with Explicit Divergence. Fund. Inform.* 93(4), pp. 371–392, doi:10.3233/FI-2009-109.
20. D. Gorla (2010): *A taxonomy of process calculi for distribution and mobility. Distributed Computing* 23(4), pp. 273–299, doi:10.1007/s00446-010-0120-6.
21. D. Gorla (2010): *Towards a unified approach to encodability and separation results for process calculi.* I&C 208(9), pp. 1031–1053, doi:10.1016/j.ic.2010.05.002.
22. G. Grätzer (2010): *Lattice Theory: Foundation.* doi:10.1007/978-3-0348-0018-1.
23. M. Hatzel, C. Wagner, K. Peters & U. Nestmann (2015): *Encoding CSP into CCS.* In: Proc. EXPRESS/SOS'15, EPTCS 190, pp. 61–75, doi:10.4204/EPTCS.190.5.

24. K. Honda & M. Tokoro (1991): *An Object Calculus for Asynchronous Communication*. In: *Proc.* ECOOP'91, LNCS 512, pp. 133–147, doi:10.1007/BFb0057019.
25. C. Lippert, S. Mennicke, R.J. van Glabbeek & U. Goltz (2018): *A Case Study on Evaluating Encodings Between Process Calculi*. Technical report, available at http://theory.stanford.edu/~rvg/abstracts.html#129.
26. R. Milner (1975): *Processes: A Mathematical Model for Computing Agents*. In: *Logic Colloquium '73*, *Studies in Logic and the Foundations of Mathematics* 50, Elsevier, pp. 157–173, doi:10.1016/S0049-237X(08)71948-7.
27. R. Milner (1990): *Operational and algebraic semantics of concurrent processes*. In: *Handbook of Theoretical Computer Science*, chapter 19, pp. 1201–1242.
28. R. Milner (1992): *Functions as Processes*. *Mathematical Structures in Computer Science* 2(2), pp. 119–141, doi:10.1017/S0960129500001407.
29. R. Milner & D. Sangiorgi (1992): *Barbed Bisimulation*. In: Proc. ICALP'92, LNCS 623, Springer, pp. 685–695, doi:10.1007/3-540-55719-9_114.
30. U. Nestmann (2000): *What is a "Good" Encoding of Guarded Choice?* I&C 156(1-2), pp. 287–319, doi:10.1006/inco.1999.2822.
31. U. Nestmann & B.C. Pierce (2000): *Decoding Choice Encodings*. I&C 163(1), pp. 1–59, doi:10.1006/inco.2000.2868. An earlier version appeared in Proc. CONCUR'96.
32. C. Palamidessi (2003): *Comparing The Expressive Power Of The Synchronous And Asynchronous Pi-Calculi*. *Mathematical Structures in Computer Science* 13(5), pp. 685–719, doi:10.1017/S0960129503004043.
33. C. Palamidessi, V.A. Saraswat, F.D. Valencia & B Victor (2006): *On the Expressiveness of Linearity vs Persistence in the Asychronous Pi-Calculus*. In: Proc. LICS'06, IEEE Computer Society Press, pp. 59–68, doi:10.1109/LICS.2006.39.
34. J. Parrow (2000): *Trios in concert*. In: *Proof, Language, and Interaction, Essays in Honour of Robin Milner*, The MIT Press, pp. 623–638.
35. J. Parrow (2008): *Expressiveness of Process Algebras*. ENTCS 209, pp. 173–186, doi:10.1016/j.entcs.2008.04.011.
36. K. Peters (2012): *Translational Expressiveness. Comparing Process Calculi using Encodings*. Ph.D. thesis, TU Berlin, doi:10.14279/depositonce-3416.
37. K. Peters & R.J. van Glabbeek (2015): *Analysing and Comparing Encodability Criteria*. In: EXPRESS/SOS'15, EPTCS 190, pp. 46–60, doi:10.4204/EPTCS.190.4.
38. K. Peters & U. Nestmann (2012): *Is It a "Good" Encoding of Mixed Choice?* In: Proc. FoSSaCS'12, LNCS 7213, pp. 210–224, doi:10.1007/978-3-642-28729-9_14.
39. K. Peters & U. Nestmann (2016): *Breaking symmetries*. *Mathematical Structures in Computer Science* 26(6), pp. 1054–1106, doi:10.1017/S0960129514000346.
40. K. Peters, U. Nestmann & U. Goltz (2013): *On Distributability in Process Calculi*. In: Proc. ESOP'13, LNCS 7792, pp. 310–329, doi:10.1007/978-3-642-37036-6_18.
41. I. Phillips & M.G. Vigliotti (2006): *Leader election in rings of ambient processes*. TCS 356(3), pp. 468–494, doi:10.1016/j.tcs.2006.02.004.
42. J.G. Riecke (1991): *Fully Abstract Translations between Functional Languages*. In: Proc. POPL'91, ACM Press, pp. 245–254, doi:10.1145/99583.99617.
43. D. Sangiorgi (1993): *From π-calculus to Higher-Order π-calculus — and back*. In: Proc. TAPSOFT'93, LNCS 668, pp. 151–166, doi:10.1007/3-540-56610-4_62.
44. D. Sangiorgi & D. Walker (2001): *The π-calculus: A Theory of Mobile Processes*. Cambridge University Press.
45. E.Y. Shapiro (1991): *Separating Concurrent Languages with Categories of Language Embeddings*. In: STOC'91, ACM, pp. 198–208, doi:10.1145/103418.103423.
46. M.G. Vigliotti, I. Phillips & C. Palamidessi (2007): *Tutorial on separation results in process calculi via leader election problems*. TCS 388(1-3), pp. 267–289, doi:10.1016/j.tcs.2007.09.001.