A Theory of Encodings and Expressiveness

Rob van Glabbeek

Data61, CSIRO, Sydney, Australia

School of Computer Science and Engineering, University of New South Wales, Sydney, Australia

rvg@cs.stanford.edu

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 [39, 32], 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 [9]. 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 [30] 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 [30] only deals with the case that all (relevant) elements in the domain are denotable as the interpretations of closed terms. In [32] situations are described where such a restriction is undesirable. In addition, both [9] and [30] 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 [8, 54, 56, 15, 4, 5, 58, 65].

In [32] 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 [9] and [30]. 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 [32] by proposing a new notion of a valid translation that incorporates the correct and valid translations of [32] as special cases. It drops the congruence requirements as well as the restriction on denotability.

An extended abstract of this paper appeared in the proceedings of FoSSaCS 2018 [35].

As in [32], 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 Section 2 defines my new concept of a valid translation up to a semantic equivalence or preorder $\mathrel{\dot\sim}$. Roughly, a valid translation up to $\mathrel{\dot\sim}$ 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 the translation of an expression is semantically equivalent to the meaning of the expression being translated.

Section 3 shows that this concept generalises the notion of a correct translation from [32]: 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, Section 14 shows that my new concept of validity generalises the one of [32], and Section 15 establishes the coincidence of my new validity-based notion of expressiveness with the one from [32] 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 $\mathrel{\dot\sim}$ iff there exists a valid translation up to $\mathrel{\dot\sim}$ of the latter language into the former. Section 4 shows that the relation "being at least as expressive as" is a preorder on languages. This expressiveness preorder depends on the choice of $\mathrel{\dot\sim}$, and a coarser choice (making less semantic distinctions) yields a richer preorder of expressiveness inclusions.

Section 5 presents the widely used class of *closed-term* languages, in which the distinction between syntax and semantic is effectively dropped by taking the domain of values where the language is interpreted in to consist of the closed terms of the language. Section 6 illustrates my approach on a well-known case study: the encoding of the synchronous in the asynchronous π-calculus.

Section 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. Section 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, and vice versa.

Section 9 concludes that the framework established thus far is very suitable for comparing the expressiveness of languages, but falls short for the purpose of combining language features. This requires a congruence reflection theorem, provided in Section 13, for closed-term languages and preorders $\mathrel{\dot\sim}$ that satisfy some mild sanity requirements: the postulates formulated in Sections 10 and 11.

Section 12 defines when a translation is *compositional*, and shows that any valid translation up to $\mathrel{\dot\sim}$ can be modified into a compositional translation valid up to $\mathrel{\dot\sim}$, provided the languages and preorders $\mathrel{\dot\sim}$ satisfy the sanity requirements of Sections 10 and 11. Hence, for the purpose of comparing the expressive power of languages, valid translations between them may be presumed compositional.

Section 16 contemplates a more general, and arguably also simpler, concept of a valid translation, than the one of Section 2. However, it lacks appealing properties of the latter.

Section 17 speculates on suitable choices for $\mathrel{\dot\sim}$ when comparing the expressiveness of process calculi. Sections 18–19 compare my approach with *full abstraction*, and with the approach of Gorla [39].

2 Languages, correct and 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 [32], I represent a language $\mathscr{L}$ as a pair $(\mathbb{T}_{\mathscr{L}}, [\![\ \ ]\!]_{\mathscr{L}})$ of a set $\mathbb{T}_{\mathscr{L}}$ of valid expressions in $\mathscr{L}$ and a mapping $[\![\ \ ]\!]_{\mathscr{L}} : \mathbb{T}_{\mathscr{L}} \rightarrow \mathscr{D}_{\mathscr{L}}$ from $\mathbb{T}_{\mathscr{L}}$ in some set of meanings $\mathscr{D}_{\mathscr{L}}$.

Definition 1 ([32]) A *translation* from a language $\mathscr{L}$ into a language $\mathscr{L}'$ is a mapping $\mathscr{T} : \mathbb{T}_{\mathscr{L}} \rightarrow \mathbb{T}_{\mathscr{L}'}$.

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

Let $\mathscr{L}$ and $\mathscr{L}'$ be two languages of the type considered above, with semantic mappings

$$[\![\ \ ]\!]_{\mathscr{L}} : \mathbb{T}_{\mathscr{L}} \rightarrow ((\mathscr{X} \rightarrow \mathbf{V}) \rightarrow \mathbf{V}) \qquad \text{and} \qquad [\![\ \ ]\!]_{\mathscr{L}'} : \mathbb{T}_{\mathscr{L}'} \rightarrow ((\mathscr{X} \rightarrow \mathbf{V}') \rightarrow \mathbf{V}').$$

In order to compare these languages w.r.t. their expressive power I need a semantic equivalence or preorder $\sim$ that is defined on a unifying domain of interpretation $\mathbf{Z}$, with $\mathbf{V}, \mathbf{V}' \subseteq \mathbf{Z}$.[1] Intuitively, $v' \sim v$ 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 a translation of an expression with meaning v into an expression with meaning v'.

Correct and a *valid* translations up to a semantic equivalence or preorder $\sim$ were introduced in [32]. 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 $\mathscr{L}$ and $\mathscr{L}'$ 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 $\eta : \mathscr{X} \rightarrow \mathbf{V}'$, $\rho : \mathscr{X} \rightarrow \mathbf{V}$ I write $\eta\, \mathbf{R}\, \rho$ iff $\eta(X)\, \mathbf{R}\, \rho(X)$ for each $X \in \mathscr{X}$.

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

Thus $\mathscr{T}$ 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 $\mathscr{T}(E)$ is a counterpart of the value filled in for X in E.

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

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

[1] I will be chiefly interested in the case that $\sim$ is an equivalence—hence the choice of a symbol that looks like $\sim$. However, to establish Observation 2 and Theorem 2 below, it suffices to know that $\sim$ is reflexive and transitive. My convention is that the dotted end of $\sim$ points to a translation and the other end to an original—without offering an intuition for the possible asymmetry.

3 Correct = valid + congruence

In [32] the concept of a correct translation up to $\sim$ was defined, for $\sim$ a semantic equivalence on $\mathbf{Z}$. Here two valuations $\eta,\rho : \mathscr{X} \to \mathbf{Z}$ are called *$\sim$-equivalent*, $\eta \sim \rho$, if $\eta(X) \sim \rho(X)$ for each $X \in \mathscr{X}$. 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 $\mathscr{L}$ into $\mathscr{L}'$ up to $\sim$. Namely, the semantics of $\mathscr{L}$ describes, among others, how any $\mathscr{L}$-operator evaluates the argument value v, and this aspect of the language has no counterpart in $\mathscr{L}'$. Therefore, [32] requires

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

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

Definition 6 ([32]) A translation $\mathscr{T}$ from $\mathscr{L}$ into $\mathscr{L}'$ is *correct up to* $\sim$ iff (1) holds and $[\![\mathscr{T}(E)]\!]_{\mathscr{L}'}(\eta) \sim [\![E]\!]_{\mathscr{L}}(\rho)$ for all $E \in \mathbb{T}_{\mathscr{L}}$ and all valuations $\eta : \mathscr{X} \to \mathbf{V}'$ and $\rho : \mathscr{X} \to \mathbf{V}$ with $\eta \sim \rho$.

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

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

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

Proposition 1 ([32]) If $\mathscr{T}$ is a correct translation up to $\sim$ from $\mathscr{L}$ into $\mathscr{L}'$, then $\sim$ is a congruence for $\mathscr{L}$.

The existence of a correct translation up to $\sim$ from $\mathscr{L}$ into $\mathscr{L}'$ does not imply that $\sim$ is a congruence for $\mathscr{L}'$. However, $\sim$ has the properties of a congruence for those expressions of $\mathscr{L}'$ that arise as translations of expressions of $\mathscr{L}$, when restricting attention to valuations into $\mathbf{U} := \{v \in \mathbf{V}' \mid \exists v \in \mathbf{V}.\ v' \sim v\}$. In [32] this called a *congruence for* $\mathscr{T}(\mathscr{L})$.

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

Proposition 2 ([32]) If $\mathscr{T}$ is a correct translation up to $\sim$ from $\mathscr{L}$ into $\mathscr{L}'$, then $\sim$ is a congruence for $\mathscr{T}(\mathscr{L})$.

The following theorem tells that the notion of validity proposed in Section 2 can be seen as a generalisation of the notion of correctness from [32] that applies to equivalences (and preorders) $\mathrel{\backsim\!\!\!\!\!\bullet}$ that are not necessarily congruences for $\mathscr{L}$ or $\mathscr{T}(\mathscr{L})$.

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

Proof: By Definitions 4 and 5 any translation that is correct up to $\sim$ is surely valid up to $\sim$.

Suppose $\mathscr{T}$ is valid up to $\sim$ and $\sim$ is a congruence for $\mathscr{T}(\mathscr{L})$. Then there is a semantic translation $\mathbf{R} \subseteq \mathbf{V}' \times \mathbf{V}$ such that $\mathbf{R} \subseteq \sim$ and $\mathscr{T}$ is correct w.r.t. $\mathbf{R}$. To establish that $\mathscr{T}$ is correct up to $\sim$, let $E \in \mathbb{T}_{\mathscr{L}}$ and let $\eta : \mathscr{X} \to \mathbf{V}'$ and $\rho : \mathscr{X} \to \mathbf{V}$ be valuations with $\eta \sim \rho$. Let $\theta : \mathscr{X} \to \mathbf{V}'$ be a valuation with $\theta\ \mathbf{R}\ \rho$—it exists since $\mathbf{R}$ is a semantic translation. Now $\theta \sim \rho \sim \eta$, using that $\mathbf{R} \subseteq \sim$, so $\theta,\eta : \mathscr{X} \to \mathbf{U}$ and $\theta \sim \eta$. Hence $[\![\mathscr{T}(E)]\!]_{\mathscr{L}'}(\eta) \sim [\![\mathscr{T}(E)]\!]_{\mathscr{L}'}(\theta) \sim [\![E]\!]_{\mathscr{L}}(\rho)$, using that $\sim$ is a congruence for $\mathscr{T}(\mathscr{L})$ and that $\mathscr{T}$ is correct w.r.t. $\mathbf{R}$. $\square$

[2]This is called a *lean* congruence in [33]; in the presence of recursion, stricter congruence requirements are common. Those are not needed in this paper.

4 A hierarchy of expressiveness preorders

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

Observation 1 Let $\mathscr{T} : \mathbb{T}_{\mathscr{L}} \to \mathbb{T}_{\mathscr{L}'}$ be a translation from $\mathscr{L}$ into $\mathscr{L}'$, and let $\sim$ be finer than $\approx$. If $\mathscr{T}$ is valid up to $\sim$, 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 Observation 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 $\mathscr{L}$ and $\mathscr{L}'$ by showing that $\mathscr{L}$ *cannot* be encoded in $\mathscr{L}'$, a coarser equivalence or preorder yields a stronger claim.

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

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

Proof: For $i=1,2,3$ let $[\![\ ]\!]_{\mathscr{L}_i} : \mathbb{T}_{\mathscr{L}_i} \to ((\mathscr{X} \to \mathbf{V}_i) \to \mathbf{V}_i)$, and for $k=1,2$ let $\mathscr{T}_k : \mathbb{T}_{\mathscr{L}_k} \to \mathbb{T}_{\mathscr{L}_{k+1}}$ be valid translations up to $\sim$ from $\mathscr{L}_k$ to $\mathscr{L}_{k+1}$, based on semantic translations $\mathbf{R}_k \subseteq \mathbf{V}_{\mathscr{L}_{k+1}} \times \mathbf{V}_{\mathscr{L}_k}$. I will show that the translation $\mathscr{T}_2 \circ \mathscr{T}_1 : \mathbb{T}_{\mathscr{L}_1} \to \mathbb{T}_{\mathscr{L}_3}$ from $\mathscr{L}_1$ into $\mathscr{L}_3$, given by $\mathscr{T}_2 \circ \mathscr{T}_1(E) = \mathscr{T}_2(\mathscr{T}_1(E))$, is valid up to $\sim$. To this end, let $\mathbf{R}_2 \circ \mathbf{R}_1 \subseteq \mathbf{V}_3 \times \mathbf{V}_1$ be the relation given by $v_3\ \mathbf{R}_2 \circ \mathbf{R}_1\ v_1$ iff $\exists v_2 : v_3\ \mathbf{R}_2\ v_2 \wedge v_2\ \mathbf{R}_1\ v_1$—it is again a semantic translation, and satisfies $\mathbf{R}_2 \circ \mathbf{R}_1 \subseteq \sim$, using the transitivity of $\sim$. Now let $E \in \mathbb{T}_{\mathscr{L}_1}$, $\rho : \mathscr{X} \to \mathbf{V}_1$ and $\theta : \mathscr{X} \to \mathbf{V}_3$, with $\theta\ \mathbf{R}_2 \circ \mathbf{R}_1\ \rho$. Then there is a valuation $\eta : \mathscr{X} \to \mathbf{V}_2$ with $\theta\ \mathbf{R}_2\ \eta\ \mathbf{R}_1\ \rho$. Hence $[\![\mathscr{T}_2(\mathscr{T}_1(E))]\!]_{\mathscr{L}_3}(\theta) \sim [\![\mathscr{T}_1(E)]\!]_{\mathscr{L}_2}(\eta) \sim [\![E]\!]_{\mathscr{L}_1}(\rho)$, so $\mathscr{T}_2 \circ \mathscr{T}_1$ is correct w.r.t. $\mathbf{R}_2 \circ \mathbf{R}_1$. □

Theorem 2 and Observation 2 show that the relation "being at least as expressive as up to $\sim$" 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 $\mathbb{T}_{\mathscr{L}}$ of $\mathscr{L}$-expressions is inductively defined by:

- $X \in \mathbb{T}_{\mathscr{L}}$ for each variable $X \in \mathscr{X}$,
- $f(E_1, \dots, E_n) \in \mathbb{T}_{\mathscr{L}}$ for each n-ary operator f and expressions $E_i \in \mathbb{T}_{\mathscr{L}}$ for $i = 1, \dots, n$,
- 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 $\mathscr{L}$ as an n-ary operation $f^{\mathbf{V}} : \mathbf{V}^n \to \mathbf{V}$ on $\mathbf{V}$. Using the equations

$$[\![X]\!]_{\mathscr{L}}(\rho) = \rho(X) \qquad \text{and} \qquad [\![f(E_1, \dots, E_n)]\!]_{\mathscr{L}}(\rho) = f^{\mathbf{V}}([\![E_1]\!]_{\mathscr{L}}(\rho), \dots, [\![E_n]\!]_{\mathscr{L}}(\rho))$$

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

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

Definition 9 A *substitution* in $\mathscr{L}$ is a partial function $\sigma : \mathscr{X} \rightharpoonup \mathbb{T}_{\mathscr{L}}$ from the variables to the $\mathscr{L}$-expressions. For a given $\mathscr{L}$-expression $E \in \mathbb{T}_{\mathscr{L}}$, $E[\sigma] \in \mathbb{T}_{\mathscr{L}}$ denotes the $\mathscr{L}$-expression E in which each free occurrence of a variable $X \in \mathit{dom}(\sigma)$ 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 $E[\sigma]$.[3] A substitution is *closed* if it has the form $\sigma : \mathscr{X} \to \mathrm{T}_{\mathscr{L}}$.

An important class of languages used in concurrency theory are the ones where the distinction between syntax and semantic is effectively dropped by taking $\mathbf{V} = \mathrm{T}_{\mathscr{L}}$, 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 $[\![E]\!]_{\mathscr{L}}(\rho)$ for $E \in \mathbb{T}_{\mathscr{L}}$ and $\rho : \mathscr{X} \to \mathrm{T}_{\mathscr{L}}$ is defined to be $E[\rho] \in \mathrm{T}_{\mathscr{L}}$. I will call such languages *closed-term* languages.

6 Example: translating a synchronous into an asynchronous π-calculus

As an illustration of the concepts introduced above, consider the π-calculus as presented by Milner in [51], i.e., the one of Sangiorgi and Walker [76] without matching, τ-prefixing, and choice.

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

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

with x, y, z ranging over $\mathscr{N}$, and X over $\mathscr{X}$, the set of *process variables*. Process variables are not considered in [76], although they are common in languages like CCS [50] 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 Section 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

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

Here $P\{^w\!/\!_z\}$ denotes the process obtained by replacing each free occurrence of z in P by w.

[3] In languages with multiple kinds of binding, such as for process variables $X \in \mathscr{X}$ and for names $x \in \mathscr{N}$ in the π-calculus below, all types of bound variables may be renamed.

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

$$\frac{z \notin \mathsf{bn}(Q)}{\bar{x}z.P|x(y).Q \rightarrow P|Q\{z/y\}} \qquad \frac{P \rightarrow P'}{P|Q \rightarrow P'|Q} \qquad \frac{P \rightarrow P'}{(\nu z)P \rightarrow (\nu z)P'} \qquad \frac{Q \equiv P \quad P \rightarrow P' \quad P' \equiv Q'}{Q \rightarrow Q'}$$

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 $x \in \mathscr{N}$. 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, notation $P{\Downarrow_{\bar{x}}}$, 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* [52, 75, 76].

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 [76]. This is known to induce the same notion of weak barbed congruence [76]. 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, similar to the theory of testing of Hennessy and De Nicola [19]—see Section 17 for details. 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 $\zeta(X) = (\nu u)(\bar{x}u.\mathbf{0}|u(v).\bar{v}z.\mathbf{0})$ and $\rho(X) = \bar{x}z.\mathbf{0}$. Then $E[\zeta] \rightarrow (\nu u)\big(u(v).\bar{v}z.\mathbf{0}|\bar{u}v.\mathbf{0}\big) \rightarrow (\bar{v}z.\mathbf{0}){\downarrow_{\bar{v}}}$ but $(E[\rho]){\not\Downarrow_{\bar{v}}}$.

The asynchronous π-calculus, as introduced by Honda & Tokoro in [44] and by Boudol in [10], 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 $!x(z).\bar{x}z.\mathbf{0} \not\cong^c \mathbf{0}$ [76]. 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 [10] defined a translation $\mathscr{T}$ from π to aπ inductively as follows:

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

Example 1 shows that $\mathscr{T}$ is not valid up to $\cong^c$. In fact, it is not even valid up to $\cong^c_{\mathrm{a}}$. However, as shown in [47], 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 $\mathscr{L}$ interpreted in a semantic domain $\mathbf{V}$ if $[\![E]\!]_{\mathscr{L}}(\nu) \sim [\![E]\!]_{\mathscr{L}}(\rho)$ for any $\mathscr{L}$-expression E and any valuations $\nu,\rho : \mathscr{X} \rightarrow \mathbf{V}$ with $\nu \sim^1 \rho$. Here ν,ρ are $\sim^1$*-equivalent*, $\nu \sim^1 \rho$, if $\nu(X) \sim \rho(X)$ for some $X \in \mathscr{X}$ 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 $\mathscr{L}$ 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 $\mathscr{L}$. 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 $\mathscr{L}$ 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 [41] and easy to prove. It follows that for any equivalence relation $\sim$ there exists a largest 1-hole congruence for $\mathscr{L}$ contained in $\sim$. I will denote this 1-hole congruence by $\sim_{\mathscr{L}}^{1c}$, and call it the *congruence closure* of $\sim$ w.r.t. $\mathscr{L}$. One has $v_1 \sim_{\mathscr{L}}^{1c} v_2$ for $v_1, v_2 \in \mathbf{V}$ iff $[\![E]\!]_{\mathscr{L}}(\nu) \sim [\![E]\!]_{\mathscr{L}}(\rho)$ for any $\mathscr{L}$-expression E and any valuations $\nu,\rho : \mathscr{X} \rightarrow \mathbf{V}$ with $\nu(X) = v_1$ and $\rho(X) = v_2$ for some $X \in \mathscr{X}$ and $\nu(Y) = \rho(Y)$ for all variables $Y \neq X$. 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 $\mathscr{L}$, 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 $\mathscr{L}$ in which all operators and other constructs have a finite arity, so that each $E \in \mathbb{T}_{\mathscr{L}}$ contains only finitely many variables, there is no difference between a congruence and a 1-hole congruence, and thus $\sim_{\mathscr{L}}^{1c}$ 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 $\mathscr{L}$ and $\sim$

have the property that $\sim^{1c}_{\mathscr{L}}$ is in fact a congruence relation. As an example, consider weak bisimilarity [50]. 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 [31] is also a congruence for the infinitary sum, as well as for all other operators of CCS [50].

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

Proposition 3 Let $\mathscr{T}$ be a translation from $\mathscr{L}$ into $\mathscr{L}'$ 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 $\mathscr{T}(\mathscr{L})$.

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

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

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

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

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

Now suppose $\sim$ is a 1-hole congruence for $\mathscr{L}$. Let $E \in \mathbb{T}_{\mathscr{L}}$ and $\theta, \eta : \mathscr{X} \to \mathbf{R}(\mathbf{V})$ with $\theta \sim^1 \eta$. Then $\theta(X) \sim \eta(X)$ for some $X \in \mathscr{X}$ and $\theta(Y) = \eta(Y)$ for all $Y \neq X$. So there must be $\nu, \rho : \mathscr{X} \to \mathbf{V}$ 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. $\square$

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

Example 4 Let $\mathscr{L}$ be a language with two constants **Yes** and **No** and a unary operator $\neg$. Its semantics is given by $\mathbf{V} = \{0,1\}$, $\mathbf{Yes}^{\mathbf{V}} = 1$, $\mathbf{No}^{\mathbf{V}} = 0$ and $\neg^{\mathbf{V}}(b) = 1-b$. Let $\mathscr{L}'$ be the extension of $\mathscr{L}$ with the constant $\top$ and the semantic value $\top$—so $\mathbf{V}' = \{0,1,\top\}$—with $\top^{\mathbf{V}'} = \top$ and $\neg^{\mathbf{V}'}(\top) = \top$. Let $\sim$ be the semantic equivalence on $\mathbf{V}' \supseteq \mathbf{V}$ given by $0 \not\sim 1 \sim \top$. Then the identity function $\mathscr{I}$ is a translation from $\mathscr{L}$ into $\mathscr{L}'$ that is valid up to $\sim$. This is witnessed by the semantic translation $\mathbf{R} := \{(0,0),(1,1)\} \subseteq \sim$. The relation $\sim$ is a congruence for $\mathscr{L}$. In line with Proposition 4 it also is a congruence for $\mathscr{I}(\mathscr{L})$ on $\mathbf{R}(\mathbf{V}) = \{0,1\}$. However it fails to be a congruence for $\mathscr{I}(\mathscr{L})$ (on $\mathbf{U} = \{0,1,\top\}$). So by Proposition 2 the transition $\mathscr{I}$ is not correct up to $\sim$. Indeed, for $\rho : \mathscr{X} \to \mathbf{V}$ a valuation that sends $X \in \mathscr{X}$ to 1 and $\eta : \mathscr{X} \to \mathbf{V}'$ a valuation that sends X to $\top$, assuming $\rho(Y) = \eta(Y)$ for other variables Y, one has $\rho \sim \eta$, yet $[\![\mathscr{I}(\neg X)]\!]_{\mathscr{L}'}(\eta) = [\![\neg X]\!]_{\mathscr{L}'}(\eta) = \neg^{\mathbf{V}'}(\eta(X)) = \top \not\sim 0 = \neg^{\mathbf{V}}(\rho(X)) = [\![\neg X]\!]_{\mathscr{L}}(\rho)$.

8 A congruence closure property for valid translations

In many applications, semantic values in the domain of interpretation of a language $\mathscr{L}$ 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 $\mathscr{L}$.

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 Section 6, with $\stackrel{\bullet}{\approx}$ in the rôle of $\sim$ and $\cong^c$ in the rôle of $\sim^c$. Now Theorem 4, contributed in this section, says that if a translation from $\mathscr{L}$ into $\mathscr{L}'$ is valid up to $\sim$, then it is even valid up to an equivalence $\sim^{1c}_{\mathscr{L},\mathbf{R}}$ that extends $\sim^c$ from $\mathbf{V}$ to a subdomain $\mathbf{W}$ of $\mathbf{V}'$ that suffices for the interpretation of translated expressions from $\mathscr{L}$. This equivalence $\sim^{1c}_{\mathscr{L},\mathbf{R}}$ coincides with the congruence closure of $\sim$ on $\mathscr{L}$, as well as on $\mathscr{T}(\mathscr{L})$, and melts each equivalence class of $\mathbf{V}$ with exactly one of $\mathbf{W}$, and vice versa.

Let $\mathscr{L}$ and $\mathscr{L}'$ be languages with $[\![\ ]\!]_{\mathscr{L}} : \mathbb{T}_{\mathscr{L}} \to ((\mathscr{X} \to \mathbf{V}) \to \mathbf{V})$ and $[\![\ ]\!]_{\mathscr{L}'} : \mathbb{T}_{\mathscr{L}'} \to ((\mathscr{X} \to \mathbf{V}') \to \mathbf{V}')$. In this section I assume that $\mathbf{V} \cap \mathbf{V}' = \emptyset$. To apply the results to the general case, just adapt $\mathscr{L}'$ by using a copy of $\mathbf{V}'$—any preorder $\sim$ on $\mathbf{V} \cup \mathbf{V}'$ extends to this copy by considering each copied element $\sim$-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 $\mathscr{T}$ is correct w.r.t. the semantic translation $\mathbf{R}$, then $\equiv_{\mathbf{R}}$ is a 1-hole congruence for $\mathscr{L}$.

Proof: Let $E \in \mathbb{T}_{\mathscr{L}}$ and $\nu, \rho : \mathscr{X} \to \mathbf{V}$ with $\nu \equiv^1_{\mathbf{R}} \rho$. I have to show that $[\![E]\!]_{\mathscr{L}}(\nu) \equiv_{\mathbf{R}} [\![E]\!]_{\mathscr{L}}(\rho)$.

Let $X \in \mathscr{X}$ be such that $\nu(X) \equiv_{\mathbf{R}} \rho(X)$ and $\nu(Y) = \rho(Y)$ for all $Y \neq X$. Then, for some $n \geq 0$ there are $v_0, \ldots, v_n \in \mathbf{V}$ and $w_1, \ldots, w_n \in \mathbf{V}'$ with $\nu(X) = v_0\ \mathbf{R}^{-1}\ w_1\ \mathbf{R}\ v_1\ \mathbf{R}^{-1}\ w_2\ \mathbf{R}\ v_2\ \mathbf{R}^{-1} \ldots \mathbf{R}\ v_n = \rho(X)$. For $i = 0, \ldots, n$ let $\rho_i : \mathscr{X} \to \mathbf{V}$ be given by $\rho_i(X) = v_i$ and $\rho_i(Y) = \rho(Y)$ for $Y \neq X$, and for $i = 1, \ldots, n$ let $\eta_i : \mathscr{X} \to \mathbf{V}'$ be given by $\eta_i(X) = w_i$ and $\eta_i(Y) = \eta(Y)$ for $Y \neq X$, for some $\eta : \mathscr{X} \to \mathbf{V}'$ with $\eta\ \mathbf{R}\ \rho$. Then $\nu = \rho_0\ \mathbf{R}^{-1}\ \eta_1\ \mathbf{R}\ \rho_1\ \mathbf{R}^{-1}\ \eta_2\ \mathbf{R}\ \rho_2\ \mathbf{R}^{-1} \ldots \mathbf{R}\ \rho_n = \rho$. Hence $[\![E]\!]_{\mathscr{L}}(\rho_0)\ \mathbf{R}^{-1}\ [\![\mathscr{T}(E)]\!]_{\mathscr{L}'}(\eta_1)\ \mathbf{R}\ [\![E]\!]_{\mathscr{L}}(\rho_1)\ \mathbf{R}^{-1}\ [\![\mathscr{T}(E)]\!]_{\mathscr{L}'}(\eta_2)\ \mathbf{R}\ [\![E]\!]_{\mathscr{L}}(\rho_2)\ \mathbf{R}^{-1} \cdots \mathbf{R}\ [\![E]\!]_{\mathscr{L}}(\rho_n)$. Thus $[\![E]\!]_{\mathscr{L}}(\nu) \equiv_{\mathbf{R}} [\![E]\!]_{\mathscr{L}}(\rho)$. □

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

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

Proof: By assumption the translation $\mathscr{T}$ from $\mathscr{L}$ into $\mathscr{L}'$ is correct w.r.t. a semantic translation $\mathbf{R} \subseteq \sim$. Let $\sim^{1c}_{\mathscr{L},\mathbf{R}}$, the *congruence closure* of $\sim$ w.r.t. $\mathscr{L}$ and $\mathbf{R}$, be the binary relation on $\mathbf{V} \cup \mathbf{V}'$ defined by $w_1 \sim^{1c}_{\mathscr{L},\mathbf{R}} w_2$ iff $w_1 \equiv_{\mathbf{R}} v_1 \sim^{1c}_{\mathscr{L}} v_2 \equiv_{\mathbf{R}} w_2$ for some $v_1, v_2 \in \mathbf{V}$. Here $\sim^{1c}_{\mathscr{L}}$ is the largest 1-hole congruence for $\mathscr{L}$, defined on $\mathbf{V}$, that is contained in $\sim$.

Since $\equiv_{\mathbf{R}}$ is a 1-hole congruence for $\mathscr{L}$ contained in $\sim$ (by Theorem 3), and $\sim^{1c}_{\mathscr{L}}$ is the largest 1-hole congruence for $\mathscr{L}$ contained in $\sim$, one has $v \equiv_{\mathbf{R}} w \Rightarrow v \sim^{1c}_{\mathscr{L}} w$ for all $v, w \in \mathbf{V}$. From this it follows that $\sim^{1c}_{\mathscr{L},\mathbf{R}}$ is transitive. As $\equiv_{\mathbf{R}}$ and $\sim^{1c}_{\mathscr{L}}$ are reflexive and symmetric, so is $\sim^{1c}_{\mathscr{L},\mathbf{R}}$. Thus, $\sim^{1c}_{\mathscr{L},\mathbf{R}}$ is an equivalence relation. Since $\sim^{1c}_{\mathscr{L}}$ and $\mathbf{R}$, and hence also $\equiv_{\mathbf{R}}$, are contained in $\sim$, so is $\sim^{1c}_{\mathscr{L},\mathbf{R}}$. Moreover, $\mathbf{R} \subseteq \equiv_{\mathbf{R}} \subseteq \sim^{1c}_{\mathscr{L},\mathbf{R}}$, so $\mathscr{T}$ is valid up to $\sim^{1c}_{\mathscr{L},\mathbf{R}}$. It remains to check properties (1)–(3).

Let $E \in \mathbb{T}_{\mathscr{L}}$ be an $\mathscr{L}$-expression and $\nu, \rho : \mathscr{X} \to \mathbf{V}$ be valuations with $\nu \sim^1 \rho$. Then there are valuations $\nu', \rho' : \mathscr{X} \to \mathbf{V}$ with $\nu \equiv^1_{\mathbf{R}} \nu'\ (\sim^{1c}_{\mathscr{L}})^1\ \rho' \equiv^1_{\mathbf{R}} \rho$. Since $\equiv_{\mathbf{R}}$ and $\sim^{1c}_{\mathscr{L}}$ are 1-hole congruences, it follows that $[\![E]\!]_{\mathscr{L}}(\nu) \equiv_{\mathbf{R}} [\![E]\!]_{\mathscr{L}}(\nu') \sim^{1c}_{\mathscr{L}} [\![E]\!]_{\mathscr{L}}(\rho') \equiv_{\mathbf{R}} [\![E]\!]_{\mathscr{L}}(\rho)$, so $[\![E]\!]_{\mathscr{L}}(\nu) \sim^{1c}_{\mathscr{L},\mathbf{R}} [\![E]\!]_{\mathscr{L}}(\rho)$. Thus $\sim^{1c}_{\mathscr{L},\mathbf{R}}$ is a 1-hole congruence for $\mathscr{L}$ on $\mathbf{V}$. Since $\sim^{1c}_{\mathscr{L}}$ is the largest 1-hole congruence for $\mathscr{L}$ contained in $\sim$, it follows that, restricted to $\mathbf{V}$, $\sim^{1c}_{\mathscr{L},\mathbf{R}} \subseteq \sim^{1c}_{\mathscr{L}}$. By the reflexivity of $\equiv_{\mathbf{R}}$ one moreover has $\sim^{1c}_{\mathscr{L}} \subseteq \sim^{1c}_{\mathscr{L},\mathbf{R}}$, so $\sim^{1c}_{\mathscr{L},\mathbf{R}} = \sim^{1c}_{\mathscr{L}}$, i.e. (1) holds.

Let $\mathbf{W} := \{w \in \mathbf{V}' \mid \exists v \in \mathbf{V}.\ w \sim^{1c}_{\mathscr{L},\mathbf{R}} v\}$. By definition, $\mathbf{W} = \mathbf{R}(\mathbf{V})$. So by Proposition 3 $\mathbf{W}$ is closed under $\mathscr{T}(\mathscr{L})$, i.e. (2) holds.

By Proposition 4 $\sim^{1c}_{\mathscr{L},\mathbf{R}}$ is a 1-hole congruence for $\mathscr{T}(\mathscr{L})$ on $\mathbf{W} = \mathbf{R}(\mathbf{V})$, contained in $\sim$. Let $\approx$ be any other 1-hole congruence on $\mathbf{W}$ contained in $\sim$. Define the relation $\approx_\mathbf{R}$ on $\mathbf{V} \cup \mathbf{W}$ by $v_1 \approx_\mathbf{R} v_2$ iff $v_1 \equiv_\mathbf{R} w_1 \approx w_2 \equiv_\mathbf{R} v_2$ for some $w_1, w_2 \in \mathbf{W}$, and let $\approx^*_\mathbf{R}$ be its transitive closure. Then $\approx^*_\mathbf{R}$ is an equivalence relation on $\mathbf{V} \cup \mathbf{W}$. Since $\equiv_\mathbf{R}$ and $\approx$ are 1-hole congruences for $\mathscr{T}(\mathscr{L})$ on $\mathbf{W}$, by the same reasoning as above also $\approx^*_\mathbf{R}$ is a 1-hole congruence for $\mathscr{T}(\mathscr{L})$ on $\mathbf{W}$. As $\mathbf{R} \subseteq \equiv_\mathbf{R} \subseteq \approx^*_\mathbf{R}$, by Proposition 4, $\approx^*_\mathbf{R}$ is a 1-hole congruence for $\mathscr{L}$. Since $\sim^{1c}_{\mathscr{L}}$ is the largest 1-hole congruence for $\mathscr{L}$ contained in $\sim$, it follows that, restricted to $\mathbf{V}$, $\approx^*_\mathbf{R} \subseteq \sim^{1c}_{\mathscr{L}}$.

For each $w_1, w_2 \in \mathbf{W}$ with $w_1 \approx w_2$ there are $v_1, v_2 \in \mathbf{V}$ with $v_1 \equiv_\mathbf{R} w_1 \approx w_2 \equiv_\mathbf{R} v_2$. So $v_1 \approx_\mathbf{R} v_2$ and hence $v_1 \sim^{1c}_{\mathscr{L}} v_2$, and thus $w_1 \sim^{1c}_{\mathscr{L},\mathbf{R}} w_2$, by definition. So $\approx \subseteq \sim^{1c}_{\mathscr{L},\mathbf{R}}$ and (3) holds. □

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

Example 5 Let $\mathscr{L}$ be a language with two constants **Yes** and **No** and a binary operator **same**. Its semantics is given by $\mathbf{V} = \{0, 1, \top, \bot\}$, $\mathbf{Yes}^\mathbf{V} = 1$, $\mathbf{No}^\mathbf{V} = 0$ and $\mathbf{same}^\mathbf{V}(x,y) = 1$ if $x = y$, while $\mathbf{same}^\mathbf{V}(x,y) = 0$ if $x \neq y$. The semantic values $\top$ and $\bot$ are not denotable as the interpretation of closed terms. Let $\mathscr{L}'$ be an exact copy of $\mathscr{L}$, except that the semantic values are primed. Let $\sim$ be the semantic equivalence on $\mathbf{V} \cup \mathbf{V}'$ given by $0 \sim 0' \not\sim 1 \sim 1' \not\sim \top \sim \top' \sim \bot \sim \bot'$. Then the identity function $\mathscr{I}$ is a translation from $\mathscr{L}$ into $\mathscr{L}'$ that is valid up to $\sim$. This is witnessed by the semantic translation $\mathbf{R}^\dagger := \{(0',0), (1',1), (\top',\bot), (\bot',\top)\} \subseteq \sim$, or, alternatively, by the semantic translation $\mathbf{R} := \{(0',0), (1',1), (\top',\top), (\bot',\bot)\} \subseteq \sim$. Upon taking the congruence closure $\sim^1_{\mathscr{L}}$, the four semantic values of $\mathbf{V}$ become inequivalent. Yet, there are two candidates for $\sim^{1c}_{\mathscr{L},\mathbf{R}}$, namely $\mathbf{R}^\dagger$ and $\mathbf{R}$.

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

Proof: By the proof of Theorem 4 $\mathscr{T}$ is correct w.r.t. the semantic translation $\mathbf{R} \subseteq \sim^{1c}_{\mathscr{L},\mathbf{R}}$, the restriction of $\sim^{1c}_{\mathscr{L},\mathbf{R}}$ to $\mathbf{V}$ equals $\sim^1_{\mathscr{L}}$, and $\mathbf{R}(\mathbf{V}) = \mathbf{W}$. Using that $\sim^1_{\mathscr{L}}$ is a congruence for $\mathscr{L}$, by Proposition 4 $\sim^{1c}_{\mathscr{L},\mathbf{R}}$ is a congruence for $\mathscr{T}(\mathscr{L})$ on $\mathbf{R}(\mathbf{V})$. Since $\mathbf{R}(\mathbf{V}) = \mathbf{W}$, which for $\sim^{1c}_{\mathscr{L},\mathbf{R}}$ is the $\mathbf{U}$ used in Definition 8, $\sim^{1c}_{\mathscr{L},\mathbf{R}}$ is a congruence for $\mathscr{T}(\mathscr{L})$. So by Theorem 1, $\mathscr{T}$ is correct up to $\sim^{1c}_{\mathscr{L},\mathbf{R}}$. □

The languages π and aπ of Section 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 Corollary 1 Boudol's translation $\mathscr{T}$ is correct up to an equivalence $\stackrel{\bullet}{\approx}{}^c_{\pi,\mathbf{R}}$, 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 $\stackrel{\bullet}{\approx}$, and on the source domain T_π coincides with $\cong^c$. By Theorem 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 $\mathscr{T}(\pi)$ on $\mathbf{W}$ that is contained in $\stackrel{\bullet}{\approx}$. As $\cong^c_\mathrm{a}$ is a congruence for all of aπ on all of $\mathrm{T}_{\mathrm{a}\pi}$, and contained in $\stackrel{\bullet}{\approx}$, it is certainly a congruence for $\mathscr{T}(\pi)$ 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 [76] and thus strong full bisimilar by [76, Def. 2.2.2].) Consequently, their translations must be related by $\stackrel{\bullet}{\approx}{}^c_{\pi,\mathbf{R}}$. So, for distinct $u, v, y, w, x, z \in \mathscr{N}$,

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

Yet, these processes are not $\cong^c_\mathrm{a}$-equivalent, as can be seen by putting them in a context $x(y).x(y).\bar{r}(s)|X$. In this context, 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 $\mathscr{T}$ from $\mathscr{L}$ to $\mathscr{L}'$ up to $\sim$, and (as usual) $\sim^{1c}_{\mathscr{L}}$ is a congruence, then all truths that can be expressed in terms of $\mathscr{L}$ can be mimicked in $\mathscr{L}'$. For the congruence classes of $\sim^{1c}_{\mathscr{L}}$ translate bijectively to congruence classes of an induced equivalence relation on the domain of $\mathscr{T}(\mathscr{L})$ (within the domain of $\mathscr{L}'$), and all operations on those congruence classes that can be performed by contexts of $\mathscr{L}$ have a perfect counterpart in terms of contexts of $\mathscr{T}(\mathscr{L})$. 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 $\mathscr{L}_1$ and $\mathscr{L}_2$ have valid translations into a language $\mathscr{L}'$, then all that can be done with $\mathscr{L}_1$ can be mimicked in a fragment of $\mathscr{L}'$, and all that can be done with $\mathscr{L}_2$ can be mimicked in another fragment of $\mathscr{L}'$. In order for these two fragments to combine, one would like to employ a single congruence relation on $\mathscr{L}'$ that specialises to congruence relations for $\mathscr{T}_1(\mathscr{L}_1)$ and $\mathscr{T}_2(\mathscr{L}_2)$, which form the counterparts of relevant congruence relations for the source languages $\mathscr{L}_1$ and $\mathscr{L}_2$.

In terms of the translation $\mathscr{T}$ 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 Section 13 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 $\mathscr{L}$ is a closed-term language. The following example illustrated why that restriction needs to be imposed.

Example 6 Let $\mathscr{L}$ be a language with three constants $\top$, **Yes** and **No**. Its semantics is given by $\mathbf{V} = \{+,-\}$, $\top^{\mathbf{V}} = \mathbf{Yes}^{\mathbf{V}} = +$ and $\mathbf{No}^{\mathbf{V}} = -$. Let $\mathscr{L}'$ be a language with the same three constants and a unary operator **next**. Its semantics is given by $\mathbf{V}' = \{0,1,2\}$ with $\top^{\mathbf{V}'} = 2$, $\mathbf{Yes}^{\mathbf{V}'} = 1$ and $\mathbf{No}^{\mathbf{V}'} = 0$, while $\mathbf{next}^{\mathbf{V}'}(n) = (n+1) \bmod 3$. Let $\sim$ be the semantic equivalence on $\mathbf{V}' \cup \mathbf{V}$ given by $- \sim 0 \not\sim 1 \sim 2 \sim +$. Then the identity function $\mathscr{I}$ is a translation from $\mathscr{L}$ into $\mathscr{L}'$ that is valid up to $\sim$. This is witnessed by the semantic translation $\mathbf{R} := \{(0,-),(1,+),(2,+)\} \subseteq \sim$. The congruence closure of $\sim$ on $\mathbf{V}'$ is the identity relation. This relation cannot be extended to $\mathbf{V} \cup \mathbf{V}'$ in such a way that $\mathscr{T}$ remains valid. For 1 and 2 need to be different; yet to make $\mathscr{T}$ valid both need to be related to $+$.

10 A unique decomposition of terms

The results of Sections 12–16 apply only to languages that satisfy two postulates, and to preorders $\sim$ that "respect $\stackrel{\alpha}{=}$" (defined in Section 11). Below and in Section 11 I formulate these postulates.

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 $E \stackrel{\alpha}{=} F$ if expression E can be converted into F by multiple acts of α-conversion.

In languages where there are multiple types of bound variables, $\stackrel{\alpha}{=}$ allows conversion of all of them. In a π-calculus with recursion, for instance, there could be bound process variables $X \in \mathscr{X}$ as well as bound names $x \in \mathscr{N}$. The last two conversions in the right column of Definition 10 define α-conversion for π-calculus names.

The following notation and observation is used below and in Section 12.1.

Definition 18 Given two substitutions $\sigma, \xi : \mathscr{X} \rightharpoonup \mathbb{T}_{\mathscr{L}}$, their composition $\xi \bullet \sigma$ with $dom(\xi \bullet \sigma) = dom(\sigma)$ is given by $(\xi \bullet \sigma)(X) = \sigma(X)[\xi]$.

Observation 3 If $dom(\sigma) = fv(E)$ then $E[\sigma][\xi] \stackrel{\alpha}{=} E[\xi \bullet \sigma]$.

An expression E of the form $f(E_1,\dots,E_n)$ can be written as $H[\sigma]$, where H is an expression $f(X_1,\dots,X_n)$ and $\sigma : \{X_1,\dots,X_n\} \to \mathbb{T}_{\mathscr{L}}$ is the substitution with $\sigma(X_i) = E_i$ for $i = 1,\dots,n$. Here H and σ are completely determined by E, except for the choice of the variables $X_1,\dots,X_n$. The term H is called a *head* of E. In this paper, for the proofs of Section 12, I propose a unique decomposition of expressions E into H and σ, by making an arbitrary choice for the variables $X_1,\dots,X_n$. Moreover, I extend this decomposition to all terms E that are not variables. This requires a postulate that says, in essence, that such a decomposition is always possible, and restricting attention to languages satisfying this postulate.

Definition 19 ([32]) A term $E \in \mathbb{T}_{\mathscr{L}}$ is a *prefix* of a term F, written $E \leq F$, if $F \stackrel{\alpha}{=} E[\sigma]$ for some substitution σ with $dom(\sigma) = fv(E)$.

Note that $E[id_E] = E$, where $id_E : fv(E) \to \mathbb{T}_{\mathscr{L}}$ is the identity. Moreover, if $dom(\sigma) = fv(E)$ then $E[\sigma][\xi] \stackrel{\alpha}{=} E[\xi \bullet \sigma]$, by Observation 3. It follows that $\leq$ is reflexive and transitive, and hence a preorder. Write $\equiv$ for the kernel of $\leq$, i.e. $E \equiv F$ iff $E \leq F \wedge F \leq E$. Note that $E \stackrel{\alpha}{=} F$ implies $E \equiv F$. If $E \equiv F$ then E can be converted into F by means of an injective renaming of its variables.

Definition 20 ([32]) A term $H \in \mathbb{T}_{\mathscr{L}}$ is a *head* if H is not a single variable and $E \leq H$ implies that E is a single variable or $E \equiv H$. It is a *head of* another term F if it is a head, as well as a prefix of F.

Example 7 Let c be a constant, g a unary operator, f a binary operator, and $\mu X.E$ the recursion construct of CCS. Then $f(X,Y)$ is a head of the term $f(c,g(c))$, and $\mu X.f(Y,g(g(X)))$ is a head of $\mu X.f(g(c),g(g(X)))$. See [32] for further detail.

Postulate 1 ([32]) Each expression E, if not a variable, has a head, which is unique up to $\equiv$.

This 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. This means that in this section I consider only languages that satisfy this postulate.

Pick a representative from each $\equiv$-equivalence class of heads, and call the chosen representatives *standard heads*. Likewise, pick a representative from each $\stackrel{\alpha}{=}$-class of terms, and call a substitution σ where $\sigma(X)$ is such a representative for each $X \in dom(\sigma)$ a *standard substitution*. Now each expression $E \notin \mathscr{X}$ can uniquely be written as $E \stackrel{\alpha}{=} H[\sigma]$, with H a standard head and σ a standard substitution with $dom(\sigma) = fv(E)$. I will refer to the pair H, σ as the *standard decomposition* of E.

11 Invariance of meaning under α-conversion

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

In [32] I limited attention to languages satisfying

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

This postulate says that the meaning of an expression is invariant under α-conversion. It can be reformulated as the requirement that $\stackrel{\alpha}{=}_{\mathscr{L}}$ 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 $\stackrel{\alpha}{=}$-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 $\stackrel{\alpha}{=}_{\mathscr{L}}$ is an equivalence.

Postulate 2 $\stackrel{\alpha}{=}_{\mathscr{L}}$ is an equivalence relation.

This postulate is needed for the results of Sections 12–16. I also need to restrict attention to preorders $\sim$ with $\stackrel{\alpha}{=}_{\mathscr{L}} \subseteq \sim$. When that holds I say that the preorder $\sim$ *respects* $\stackrel{\alpha}{=}_{\mathscr{L}}$. If (2) holds—which strengthens of Postulate 2—then *any* preorder respects $\stackrel{\alpha}{=}_{\mathscr{L}}$.

12 Compositionality

An important property of translations, defined below, is *compositionality*. In this section show I that any valid translation up to a preorder $\sim$ 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 $\sim$ that respect $\stackrel{\alpha}{=}$.

Definition 21 A translation $\mathscr{T}$ from $\mathscr{L}$ into $\mathscr{L}'$ is *compositional* if
(1) $\mathscr{T}(E[\sigma]) \stackrel{\alpha}{=} \mathscr{T}(E)[\mathscr{T} \circ \sigma]$ for each $E \in \mathbb{T}_{\mathscr{L}}$ and $\sigma : fv(E) \to \mathbb{T}_{\mathscr{L}}$,
(2) $E \stackrel{\alpha}{=} F$ implies $\mathscr{T}(E) \stackrel{\alpha}{=} \mathscr{T}(F)$ for all $E, F \in \mathbb{T}_{\mathscr{L}}$,
(3) and moreover $\mathscr{T}(X) = X$ for each $X \in \mathscr{X}$.

In case $E = f(t_1,\dots,t_n)$ for certain $t_i \in \mathbb{T}_{\mathscr{L}}$ this amounts to $\mathscr{T}(f(t_1,\dots,t_n)) \stackrel{\alpha}{=} E_f(\mathscr{T}(t_1),\dots,\mathscr{T}(t_n))$, where $E_f := \mathscr{T}(f(X_1,\dots,X_n))$ and $E_f(u_1,\dots,u_n)$ denotes the result of the simultaneous substitution in this expression of the terms $u_i \in \mathbb{T}_{\mathscr{L}'}$ for the free variables X_i, for $i = 1,\dots,n$. The first requirement of Definition 21 is more general and covers language constructs other than functions, such as recursion. Requiring equality rather than $\stackrel{\alpha}{=}$ is too demanding, as the following example illustrates.

Example 8 Take a source language $\mathscr{L}$ that features an unary replication operator !, as in the π-calculus, and a target language $\mathscr{L}'$ that instead has a recursion construct $\mu X.E$, as in CCS; both languages have a constant 0. A suitable translation $\mathscr{T}$ satisfies $\mathscr{T}(!X_1) = \mu X.(X|X_1)$ and $\mathscr{T}(0) = 0$. Applying Definition 21 with $\sigma(X_1) = 0$ gives $\mathscr{T}(!0) \stackrel{\alpha}{=} \mu X.(X|0)$, whereas applying it with $\sigma(X_1) = X$ gives $\mathscr{T}(!X) \stackrel{\alpha}{=} \mu Y.(Y|X)$. Here the bound variable X needed to be renamed (into Y) to avoid capture of the free variable X that is substituted for X_1. Furthermore, applying Definition 21 on $\mathscr{T}(!X) \stackrel{\alpha}{=} \mu Y.(Y|X)$ with $\sigma(X) = 0$ gives $\mathscr{T}(!0) \stackrel{\alpha}{=} \mu Y.(Y|0)$. Since $X \neq Y$, this shows that $\stackrel{\alpha}{=}$ cannot consistently be replaced by $=$.

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

Proof: (1) $\mathscr{T}_2(\mathscr{T}_1(E[\sigma])) \stackrel{\alpha}{=} \mathscr{T}_2(\mathscr{T}_1(E)[\mathscr{T}_1 \circ \sigma]) \stackrel{\alpha}{=} \mathscr{T}_2(\mathscr{T}_1(E))[\mathscr{T}_2 \circ \mathscr{T}_1 \circ \sigma])$ for each $\sigma : \mathscr{X} \rightharpoonup \mathbb{T}_{\mathscr{L}_1}$ and $E \in \mathbb{T}_{\mathscr{L}_1}$. Here the derivation of the first $\stackrel{\alpha}{=}$ uses Property (2) of Definition 21—and this is the reason for requiring that property.

(2) $E \stackrel{\alpha}{=} F$ implies $\mathscr{T}_1(E) \stackrel{\alpha}{=} \mathscr{T}_1(F)$ and hence $\mathscr{T}_2(\mathscr{T}_1(E)) \stackrel{\alpha}{=} \mathscr{T}_2(\mathscr{T}_1(F))$ for all $E, F \in \mathbb{T}_{\mathscr{L}}$.

(3) $\mathscr{T}_2(\mathscr{T}_1(X)) = \mathscr{T}_2(X) = X$ for each $X \in \mathscr{X}$. $\square$

12.1 Translations that are compositional by construction

The following proposition shows that when verifying that a translation is compositional it suffices to check requirement (1) of Definition 21 for the case that the term E is a (standard) head only. In most applications (including the one of Section 6) a translation is defined inductively, in such a way that (1) for E a head, as well as (2) and (3), hold by definition; in such cases compositionality follows.

Proposition 5 Let $\mathscr{L}$ be a language satisfying Postulate 1. Any translation $\mathscr{T}$ from $\mathscr{L}$ into $\mathscr{L}'$ satisfying

$$\begin{array}{ll} \mathscr{T}(X) = X & \text{for } X \in \mathscr{X}\text{, and} \\ \mathscr{T}(E) \stackrel{\alpha}{=} \mathscr{T}(H)[\mathscr{T} \circ \sigma] & \text{when } E \stackrel{\alpha}{=} H[\sigma] \text{ with } H, \sigma \text{ the standard decomposition of } E. \end{array}$$

is compositional.

Proof: By assumption, this translation satisfies Properties (2) and (3) of Definition 21. Next, I show that $\mathscr{T}$ also satisfies Property (1), using induction on $E/_{\stackrel{\alpha}{=}}$. So let $E \in \mathbb{T}_{\mathscr{L}}$ and $\xi : fv(E) \rightarrow \mathbb{T}_{\mathscr{L}}$. I have to show that $\mathscr{T}(E[\xi]) \stackrel{\alpha}{=} \mathscr{T}(E)[\mathscr{T} \circ \xi]$. The case $E \in \mathscr{X}$ is trivial, so let $E \stackrel{\alpha}{=} H[\sigma]$, with H a standard head and σ a standard substitution with $dom(\sigma) = fv(H)$. For each free variable X of H, $\sigma(X)$ is a proper subterm of E up to $\stackrel{\alpha}{=}$, so by the induction hypothesis $\mathscr{T}(\sigma(X)[\xi]) \stackrel{\alpha}{=} \mathscr{T}(\sigma(X))[\mathscr{T} \circ \xi]$. Thus, for $X \in fv(H)$,

$$\begin{array}{rll} (\mathscr{T} \circ (\xi \bullet \sigma))(X) & = \mathscr{T}((\xi \bullet \sigma)(X)) & \text{by definition of functional composition } \circ \\ & = \mathscr{T}(\sigma(X)[\xi]) & \text{by definition of the operation } \bullet \text{ on substitutions} \\ & \stackrel{\alpha}{=} \mathscr{T}(\sigma(X))[\mathscr{T} \circ \xi] & \text{by induction, derived above} \\ & = ((\mathscr{T} \circ \xi) \bullet (\mathscr{T} \circ \sigma))(X) & \text{by definition of the operations } \circ \text{ and } \bullet. \end{array}$$

This shows that the substitutions $\mathscr{T} \circ (\xi \bullet \sigma)$ and $(\mathscr{T} \circ \xi) \bullet (\mathscr{T} \circ \sigma)$—both with domain $fv(H)$—are equal up to α-conversion, from which it follows that $F[\mathscr{T} \circ (\xi \bullet \sigma)] \stackrel{\alpha}{=} (F[\mathscr{T} \circ \sigma])[\mathscr{T} \circ \xi]$ for all terms $F \in \mathbb{T}_{\mathscr{L}'}$.

$$\begin{array}{rll} \text{Hence } \mathscr{T}(E[\xi]) & \stackrel{\alpha}{=} \mathscr{T}(H[\sigma][\xi]) & \text{since } E \stackrel{\alpha}{=} H[\sigma]\text{, and thus } E[\xi] \stackrel{\alpha}{=} H[\sigma][\xi]\text{, using (2)} \\ & \stackrel{\alpha}{=} \mathscr{T}(H[\xi \bullet \sigma]) & \text{by Observation 3} \\ & \stackrel{\alpha}{=} \mathscr{T}(H[\upsilon]) & \text{for a standard substitution } \upsilon \stackrel{\alpha}{=} \xi \bullet \sigma \\ & \stackrel{\alpha}{=} \mathscr{T}(H)[\mathscr{T} \circ \upsilon] & \text{by assumption} \\ & \stackrel{\alpha}{=} \mathscr{T}(H)[\mathscr{T} \circ (\xi \bullet \sigma)] & \text{replacing } \upsilon \text{ by } \xi \bullet \sigma\text{, using (2)} \\ & \stackrel{\alpha}{=} (\mathscr{T}(H)[\mathscr{T} \circ \sigma])[\mathscr{T} \circ \xi] & \text{derived above} \\ & \stackrel{\alpha}{=} \mathscr{T}(E)[\mathscr{T} \circ \xi] & \text{by assumption.} \end{array}$$

□

12.2 The denotation of a substitution

To obtain the compositionality of valid translations, I need the concept of the *denotation* of a substitution as a transformation of valuations. The semantic mapping $[\![\ ]\!]_{\mathscr{L}} : \mathbb{T}_{\mathscr{L}} \rightarrow ((\mathscr{X} \rightarrow \mathbf{V}) \rightarrow \mathbf{V})$ extends to substitutions σ by $[\![\sigma]\!]_{\mathscr{L}}(\rho)(X) := [\![\sigma(X)]\!]_{\mathscr{L}}(\rho)$ for all $X \in \mathscr{X}$ and $\rho : \mathscr{X} \rightarrow \mathbf{V}$—here σ is extended to a total function by $\sigma(Y) := Y$ for all $Y \notin dom(\sigma)$. Thus $[\![\sigma]\!]_{\mathscr{L}}$ is of type $(\mathscr{X} \rightarrow \mathbf{V}) \rightarrow (\mathscr{X} \rightarrow \mathbf{V})$, i.e. a map from valuations to valuations. The inductive nature of the semantic mapping $[\![\ ]\!]_{\mathscr{L}}$ ensures that for each expression $E \in \mathbb{T}_{\mathscr{L}}$, substitution $\sigma : \mathscr{X} \rightharpoonup \mathbb{T}_{\mathscr{L}}$ and valuation $\rho : \mathscr{X} \rightarrow \mathbf{V}$ there exists a term F with $E \stackrel{\alpha}{=} F$ such that $[\![E[\sigma]]\!]_{\mathscr{L}}(\rho) = [\![F[\sigma]]\!]_{\mathscr{L}}(\rho) = [\![F]\!]_{\mathscr{L}}([\![\sigma]\!]_{\mathscr{L}}(\rho))$, and hence

$$[\![E[\sigma]]\!]_{\mathscr{L}}(\rho) \stackrel{\alpha}{=}_{\mathscr{L}} [\![E]\!]_{\mathscr{L}}([\![\sigma]\!]_{\mathscr{L}}(\rho)). \tag{3}$$

In case E is $f(X_1, \ldots, X_n)$ this amounts to $[\![f(E_1, \ldots, E_n)]\!]_{\mathscr{L}}(\rho) = f^{\mathbf{V}}([\![E_1]\!]_{\mathscr{L}}(\rho), \ldots, [\![E_n]\!]_{\mathscr{L}}(\rho))$, but the above is more general and anticipates language constructs other than functions, such as recursion.

12.3 Closing a semantic translation under α-conversion

The following lemma (assuming Postulate 2) says that $\stackrel{\alpha}{=}_{\mathscr{L}}$ is a congruence.

Lemma 2 Let $E \in \mathbb{T}_{\mathscr{L}}$ and $\nu,\rho : \mathscr{X} \rightarrow \mathbf{V}$. If $\nu \stackrel{\alpha}{=}_{\mathscr{L}} \rho$ then $[\![E]\!]_{\mathscr{L}}(\nu) \stackrel{\alpha}{=}_{\mathscr{L}} [\![E]\!]_{\mathscr{L}}(\rho)$.

Proof: Suppose $\nu \stackrel{\alpha}{=}_{\mathscr{L}} \rho$. Then, for each $X \in \mathscr{X}$, $\nu(X) \stackrel{\alpha}{=}_{\mathscr{L}} \rho(X)$, so there are terms $E_X, F_X \in \mathbb{T}_{\mathscr{L}}$ with $E_X \stackrel{\alpha}{=} F_X$ and a valuation $\zeta_X : \mathscr{X} \rightarrow \mathbf{V}$ such that $[\![E_X]\!]_{\mathscr{L}}(\zeta_X) = \nu(X)$ and $[\![F_X]\!]_{\mathscr{L}}(\zeta_X) = \rho(X)$. By renaming of variables one can assure that $\mathit{fv}(E_X) \cap \mathit{fv}(E_Y) = \emptyset$ for any different $X,Y \in \mathit{fv}(E)$. Here I assume that the set $\mathscr{X}$ of variables is sufficiently large. Note that $\mathit{fv}(F_X) = \mathit{fv}(E_X)$ for all $X \in \mathit{fv}(E)$. Let $\zeta : \mathscr{X} \rightarrow \mathbf{V}$ be a valuation satisfying $\zeta(Z) = \zeta_X(Z)$ for any $Z \in \mathit{fv}(E_X)$ with $X \in \mathit{fv}(E)$. Define the substitutions $\sigma, \xi : \mathit{fv}(E) \rightarrow \mathbb{T}_{\mathscr{L}}$ by $\sigma(X) = E_X$ and $\xi(X) = F_X$ for all $X \in \mathit{fv}(E)$. Then $[\![\sigma(X)]\!]_{\mathscr{L}}(\zeta) = \nu(X)$ and $[\![\xi(X)]\!]_{\mathscr{L}}(\zeta) = \rho(X)$ for all $X \in \mathit{fv}(E)$. Hence, using (3), $[\![E]\!]_{\mathscr{L}}(\nu) = [\![E]\!]_{\mathscr{L}}([\![\sigma]\!]_{\mathscr{L}}(\zeta)) \stackrel{\alpha}{=}_{\mathscr{L}} [\![E[\sigma]]\!]_{\mathscr{L}}(\zeta)$ and $[\![E]\!]_{\mathscr{L}}(\rho) = [\![E]\!]_{\mathscr{L}}([\![\xi]\!]_{\mathscr{L}}(\zeta)) \stackrel{\alpha}{=}_{\mathscr{L}} [\![E[\xi]]\!]_{\mathscr{L}}(\zeta)$. As $\sigma(X) \stackrel{\alpha}{=} \xi(X)$ for all $X \in \mathscr{X}$ one has $E[\sigma] \stackrel{\alpha}{=} E[\xi]$, and thus $[\![E]\!]_{\mathscr{L}}(\nu) \stackrel{\alpha}{=}_{\mathscr{L}} [\![E[\sigma]]\!]_{\mathscr{L}}(\zeta) \stackrel{\alpha}{=}_{\mathscr{L}} [\![E[\xi]]\!]_{\mathscr{L}}(\zeta) \stackrel{\alpha}{=}_{\mathscr{L}} [\![E]\!]_{\mathscr{L}}(\rho)$. Hence $[\![E]\!]_{\mathscr{L}}(\nu) \stackrel{\alpha}{=}_{\mathscr{L}} [\![E]\!]_{\mathscr{L}}(\rho)$ with Postulate 2. □

Given a relation $\mathbf{R} \subseteq \mathbf{V}' \times \mathbf{V}$, define $\mathbf{R}^\alpha$ by $w \mathrel{\mathbf{R}} v$ iff $\exists w', v'.\ w \stackrel{\alpha}{=}_{\mathscr{L}'} w' \mathrel{\mathbf{R}} v' \stackrel{\alpha}{=}_{\mathscr{L}} v$.

Lemma 3 If a translation $\mathscr{T}$ between languages $\mathscr{L}$ and $\mathscr{L}'$ that satisfy Postulate 2 is correct w.r.t. a semantic translation $\mathbf{R}$, then it is also correct w.r.t. $\mathbf{R}^\alpha$.

Proof: Let $\mathbf{R} \subseteq \mathbf{V}' \times \mathbf{V}$ be a semantic translation, and $\mathscr{T}$ a translation that is correct w.r.t. $\mathbf{R}$. Let $E \in \mathbb{T}_{\mathscr{L}}$, $\eta : \mathscr{X} \rightarrow \mathbf{V}'$ and $\rho : \mathscr{X} \rightarrow \mathbf{V}$, with $\eta \mathrel{\mathbf{R}^\alpha} \rho$. Then there must be valuations $\theta : \mathscr{X} \rightarrow \mathbf{V}'$ and $\nu : \mathscr{X} \rightarrow \mathbf{V}$ with $\eta \stackrel{\alpha}{=}_{\mathscr{L}'} \theta \mathrel{\mathbf{R}} \nu \stackrel{\alpha}{=}_{\mathscr{L}} \rho$. Since $\mathscr{T}$ is correct w.r.t. $\mathbf{R}$ one has $[\![\mathscr{T}(E)]\!]_{\mathscr{L}'}(\theta) \mathrel{\mathbf{R}} [\![E]\!]_{\mathscr{L}}(\nu)$. By Lemma 2 $[\![\mathscr{T}(E)]\!]_{\mathscr{L}'}(\eta) \stackrel{\alpha}{=}_{\mathscr{L}'} [\![\mathscr{T}(E)]\!]_{\mathscr{L}'}(\theta) \mathrel{\mathbf{R}} [\![E]\!]_{\mathscr{L}}(\nu) \stackrel{\alpha}{=}_{\mathscr{L}} [\![E]\!]_{\mathscr{L}}(\rho)$ and thus $[\![\mathscr{T}(E)]\!]_{\mathscr{L}'}(\eta) \mathrel{\mathbf{R}^\alpha} [\![E]\!]_{\mathscr{L}}(\rho)$. □

12.4 Only the effect on standard heads matters

Proposition 6 Let $\mathscr{L}$ and $\mathscr{L}'$ be languages that satisfy Postulates 1 and 2, and $\mathbf{R} \subseteq \mathbf{V}' \times \mathbf{V}$ a semantic translation. A compositional translation $\mathscr{T} : \mathbb{T}_{\mathscr{L}} \rightarrow \mathbb{T}_{\mathscr{L}'}$ is correct w.r.t. $\mathbf{R}^\alpha$ iff $[\![\mathscr{T}(H)]\!]_{\mathscr{L}'}(\eta) \mathrel{\mathbf{R}^\alpha} [\![H]\!]_{\mathscr{L}}(\rho)$ for all standard heads $H \in \mathbb{T}_{\mathscr{L}}$ and all valuations $\eta : \mathscr{X} \rightarrow \mathbf{V}'$ and $\rho : \mathscr{X} \rightarrow \mathbf{V}$ with $\eta \mathrel{\mathbf{R}^\alpha} \rho$.

Proof: "Only if" follows immediately from Definition 3. So assume $[\![\mathscr{T}(H)]\!]_{\mathscr{L}'}(\eta) \mathrel{\mathbf{R}^\alpha} [\![H]\!]_{\mathscr{L}}(\rho)$ for all standard heads $H \in \mathbb{T}_{\mathscr{L}}$ and all $\eta : \mathscr{X} \rightarrow \mathbf{V}'$ and $\rho : \mathscr{X} \rightarrow \mathbf{V}$ with $\eta \mathrel{\mathbf{R}^\alpha} \rho$. I have to show that $\mathscr{T}$ is correct w.r.t. $\mathbf{R}^\alpha$, i.e. that $[\![\mathscr{T}(E)]\!]_{\mathscr{L}'}(\eta) \mathrel{\mathbf{R}^\alpha} [\![E]\!]_{\mathscr{L}}(\rho)$ for all terms $E \in \mathbb{T}_{\mathscr{L}}$ and all $\eta : \mathscr{X} \rightarrow \mathbf{V}'$ and $\rho : \mathscr{X} \rightarrow \mathbf{V}$ with $\eta \mathrel{\mathbf{R}^\alpha} \rho$. Let η and ρ be such valuations. I proceed with structural induction on E, up to $\stackrel{\alpha}{=}$. When handling a term $E \stackrel{\alpha}{=} H[\sigma]$, $\sigma(X)$ is, up to $\stackrel{\alpha}{=}$, a proper subterm of E for each free variable X of H. So by induction $[\![\mathscr{T}(\sigma(X))]\!]_{\mathscr{L}'}(\eta) \mathrel{\mathbf{R}^\alpha} [\![\sigma(X)]\!]_{\mathscr{L}}(\rho)$. The valuation $[\![\sigma]\!]_{\mathscr{L}}(\rho)$ is defined such that $[\![\sigma]\!]_{\mathscr{L}}(\rho)(X) = [\![\sigma(X)]\!]_{\mathscr{L}}(\rho)$ for each $X \in \mathscr{X}$. Likewise, $[\![\mathscr{T} \circ \sigma]\!]_{\mathscr{L}'}(\eta)(X) = [\![\mathscr{T}(\sigma(X))]\!]_{\mathscr{L}'}(\eta)$ for each $X \in \mathscr{X}$. Hence $[\![\mathscr{T} \circ \sigma]\!]_{\mathscr{L}'}(\eta) \mathrel{\mathbf{R}^\alpha} [\![\sigma]\!]_{\mathscr{L}}(\rho)$. (*)

- $$\begin{array}{rll} [\![\mathscr{T}(X)]\!]_{\mathscr{L}'}(\eta) = [\![X]\!]_{\mathscr{L}'}(\eta) = & \eta(X) & \text{by definitions of } \mathscr{T} \text{ and } [\![\ ]\!]_{\mathscr{L}'} \\ \mathbf{R}^\alpha & \rho(X) & \text{since } \eta \mathrel{\mathbf{R}^\alpha} \rho \\ = & [\![X]\!]_{\mathscr{L}}(\rho) & \text{by definition of } [\![\ ]\!]_{\mathscr{L}}. \end{array}$$
- $$\begin{array}{rll} [\![\mathscr{T}(E)]\!]_{\mathscr{L}'}(\eta) \stackrel{\alpha}{=}_{\mathscr{L}'} & [\![\mathscr{T}(H)[\mathscr{T} \circ \sigma]]\!]_{\mathscr{L}'}(\eta) & \text{by the compositionality of } \mathscr{T}\text{, since } E \stackrel{\alpha}{=} H[\sigma] \\ \stackrel{\alpha}{=}_{\mathscr{L}'} & [\![\mathscr{T}(H)]\!]_{\mathscr{L}'}([\![\mathscr{T} \circ \sigma]\!]_{\mathscr{L}'}(\eta)) & \text{by (3)} \\ \mathbf{R}^\alpha & [\![H]\!]_{\mathscr{L}}([\![\sigma]\!]_{\mathscr{L}}(\rho)) & \text{by assumption, using (*) above} \\ \stackrel{\alpha}{=}_{\mathscr{L}} & [\![H[\sigma]]\!]_{\mathscr{L}}(\rho) & \text{by (3)} \\ \stackrel{\alpha}{=}_{\mathscr{L}} & [\![E]\!]_{\mathscr{L}}(\rho) & \text{since } E \stackrel{\alpha}{=} H[\sigma]. \end{array}$$ □

12.5 Correct and valid transitions can be made compositional

Theorem 5 Let $\mathscr{L}$ and $\mathscr{L}'$ be languages satisfying Postulates 1 and 2. If any correct translation from $\mathscr{L}$ into $\mathscr{L}'$ w.r.t. $\mathbf{R}^\alpha$ exists, then there is a compositional translation from $\mathscr{L}$ into $\mathscr{L}'$ that is correct w.r.t. $\mathbf{R}^\alpha$.

Proof: Given a translation $\mathscr{T}_0$ that is correct w.r.t. $\mathbf{R}^\alpha$, define the translation $\mathscr{T}$ inductively by

$$\begin{array}{ll} \mathscr{T}(X) := X & \text{for } X \in \mathscr{X} \\ \mathscr{T}(E) := \mathscr{T}_0(H)[\mathscr{T} \circ \sigma] & \text{when } E \stackrel{\alpha}{=} H[\sigma] \text{ with } H, \sigma \text{ the standard decomposition of } E. \end{array}$$

By filling in $H[id_H]$ for E, with H a standard head and id_H the identity substitution on $fv(H)$, I obtain $\mathscr{T}(H) = \mathscr{T}_0(H)$ for each standard head H. Hence, by Proposition 5, $\mathscr{T}$ is compositional. Moreover, since $\mathscr{T}_0$ is correct w.r.t. $\mathbf{R}^\alpha$, for all standard heads $H \in \mathbb{T}_\mathscr{L}$ and all $\eta : \mathscr{X} \rightarrow \mathbf{V}'$ and $\rho : \mathscr{X} \rightarrow \mathbf{V}$ with $\eta \; \mathbf{R}^\alpha \; \rho$ one has $[\![\mathscr{T}(H)]\!]_{\mathscr{L}'}(\eta) = [\![\mathscr{T}_0(H)]\!]_{\mathscr{L}'}(\eta) \; \mathbf{R}^\alpha \; [\![H]\!]_\mathscr{L}(\rho)$. Thus, by Proposition 6, $\mathscr{T}$ is correct w.r.t. $\mathbf{R}^\alpha$. $\square$

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

Proof: Let $\mathscr{T} : \mathbb{T}_\mathscr{L} \rightarrow \mathbb{T}_{\mathscr{L}'}$ be valid up to $\sim$. Then $\mathscr{T}$ is correct w.r.t. some semantic translation $\mathbf{R} \subseteq \mathbf{V}' \times \mathbf{V}$ with $\mathbf{R} \subseteq \sim$. By Lemma 3 $\mathscr{T}$ is correct w.r.t. $\mathbf{R}^\alpha$. By Theorem 5 there exists a compositional translation $\mathscr{T}' : \mathbb{T}_\mathscr{L} \rightarrow \mathbb{T}_{\mathscr{L}'}$ that is correct w.r.t. $\mathbf{R}^\alpha$. Since $\sim$ respects $\stackrel{\alpha}{=}_\mathscr{L}$ and $\stackrel{\alpha}{=}_{\mathscr{L}'}$, one has $\stackrel{\alpha}{=}_\mathscr{L}, \stackrel{\alpha}{=}_{\mathscr{L}'}, \mathbf{R} \subseteq \sim$, and thus $\mathbf{R}^\alpha \subseteq \sim$. The statement about correct translations up to $\sim$ follows in the same way, or as a corollary by use of Theorem 1. $\square$

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 [32], but assuming (2), a stronger version of Postulate 2.

13 Translations from closed-term languages reflect target congruences

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

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

Proof: If $\mathbf{V} = \emptyset$, the statement is trivial. Otherwise, pick a valuation $\zeta : \mathscr{X} \rightarrow \mathbf{V}$. By assumption, $\mathscr{T}$ is correct w.r.t. a semantic translation $\mathbf{R} \subseteq \sim$. Now pick a valuation $\theta : \mathscr{X} \rightarrow \mathbf{V}'$ with $\theta \; \mathbf{R} \; \zeta$. Let $\equiv_\mathscr{T}$ be the smallest equivalence relation on $\mathbf{V} \cup \mathbf{V}'$ such that $[\![\mathscr{T}(P)]\!]_{\mathscr{L}'}(\theta) \equiv_\mathscr{T} [\![P]\!]_\mathscr{L}$ for all $P \in \mathrm{T}_\mathscr{L}$. Since $\mathscr{T}$ is correct w.r.t. $\mathbf{R}$, one has $[\![\mathscr{T}(P)]\!]_{\mathscr{L}'}(\theta) \; \mathbf{R} \; [\![P]\!]_\mathscr{L}(\zeta) = [\![P]\!]_\mathscr{L}$ for all $P \in \mathrm{T}_\mathscr{L}$. Hence $\equiv_\mathscr{T} \subseteq \sim$.

Define $\approx_\mathscr{T}$ on $\mathbf{V} \cup \mathbf{V}'$ by $v_1 \approx_\mathscr{T} v_2$ iff $v_1 \equiv_\mathscr{T} w_1 \approx w_2 \equiv_\mathscr{T} v_2$ for some $w_1, w_2 \in \mathbf{V}'$. Then $\approx_\mathscr{T} \subseteq \sim$. Since $\mathscr{L}$ is a closed-term language, for each $v \in \mathbf{V}$ there is exactly one $P \in \mathrm{T}_\mathscr{L}$ with $[\![P]\!]_\mathscr{L} = v$, namely $P = v$. Hence, each $\equiv_\mathscr{T}$-equivalence class on $\mathbf{V} \cup \mathbf{V}'$ contains exactly one element of $\mathbf{V}'$. It follows that $\approx_\mathscr{T}$ is transitive, and hence an equivalence relation. Moreover, on $\mathbf{V}'$ it coincides with $\approx$. It remains to show that $\mathscr{T}$ is valid up to $\approx_\mathscr{T}$. For then Theorem 1 implies that $\mathscr{T}$ is correct up to $\approx_\mathscr{T}$.

Let $\mathbf{R}_\mathscr{T} := \{(w, v) \mid v \in \mathrm{T}_\mathscr{L} \wedge w \stackrel{\alpha}{=}_{\mathscr{L}'} [\![\mathscr{T}(v)]\!]_{\mathscr{L}'}(\theta)\}$. Then $\mathbf{R}_\mathscr{T}$ is a semantic translation with $\mathbf{R}_\mathscr{T} \subseteq \approx_\mathscr{T}$, using that $\stackrel{\alpha}{=}_{\mathscr{L}'} \subseteq \approx$. To show validity of $\mathscr{T}$ up to $\approx_\mathscr{T}$ it suffices to establish that $\mathscr{T}$ is correct w.r.t. $\mathbf{R}_\mathscr{T}$. So let $E \in \mathbb{T}_\mathscr{L}$ and let $\eta : \mathscr{X} \rightarrow \mathbf{V}'$ and $\rho : \mathscr{X} \rightarrow \mathbf{V}$ be valuations with $\eta \; \mathbf{R}_\mathscr{T} \; \rho$. I have to show that $[\![\mathscr{T}(E)]\!]_{\mathscr{L}'}(\eta) \; \mathbf{R}_\mathscr{T} \; [\![E]\!]_\mathscr{L}(\rho)$. Since $\mathscr{L}$ is a closed-term language, ρ also is a closed substitution, and $[\![E]\!]_\mathscr{L}(\rho) := E[\rho] =: [\![E[\rho]]\!]_\mathscr{L}$. Let $\mathscr{T} \circ \rho : \mathscr{X} \rightarrow \mathrm{T}_{\mathscr{L}'}$ be the substitution with $(\mathscr{T} \circ \rho)(X) = \mathscr{T}(\rho(X))$ for all X. Then—filling in $(\eta(X),\rho(X))$ for (w,v) in the definition of $\mathbf{R}_{\mathscr{T}}$—$\eta(X) \stackrel{\alpha}{=}_{\mathscr{L}'} [\![(\mathscr{T}\circ\rho)(X)]\!]_{\mathscr{L}'}(\theta)$ for all $X\in\mathscr{X}$, i.e., $\eta \stackrel{\alpha}{=}_{\mathscr{L}'} [\![\mathscr{T}\circ\rho]\!]_{\mathscr{L}'}(\theta)$. So, using Lemma 2, (3) and compositionality,

$$[\![\mathscr{T}(E)]\!]_{\mathscr{L}'}(\eta) \stackrel{\alpha}{=}_{\mathscr{L}'} [\![\mathscr{T}(E)]\!]_{\mathscr{L}'}([\![\mathscr{T}\circ\rho]\!]_{\mathscr{L}'}(\theta)) \stackrel{\alpha}{=}_{\mathscr{L}'} [\![\mathscr{T}(E)[\mathscr{T}\circ\sigma]]\!]_{\mathscr{L}'}(\theta) \stackrel{\alpha}{=}_{\mathscr{L}'}$$
$$[\![\mathscr{T}(E[\sigma])]\!]_{\mathscr{L}'}(\theta)\ \mathbf{R}_{\mathscr{T}}\ [\![E[\sigma]]\!]_{\mathscr{L}} = [\![E]\!]_{\mathscr{L}}(\rho).$$

It follows that $[\![\mathscr{T}(E)]\!]_{\mathscr{L}'}(\eta)\ \mathbf{R}_{\mathscr{T}}\ [\![E]\!]_{\mathscr{L}}(\rho)$ using Postulate 2. □

Since by Theorem 6 $\mathscr{T}$ is correct up to $\approx_{\mathscr{T}}$, by Proposition 1 $\approx_{\mathscr{T}}$ is a congruence for $\mathscr{L}$, contained in $\sim$. Consequently, on $\mathbf{V}$, $\approx_{\mathscr{T}}$ is contained in $\sim^c_{\mathscr{L}}$, the coarsest 1-hole congruence for $\mathscr{L}$ contained in $\sim$. As remarked in Section 9, for the example of Section 6 this inclusion is strict.

14 Comparison with the definition of validity from [32]

An earlier definition of validity occurs in [32]. A shortcoming of that notion was that it only applied to languages for which all values in the domain of interpretation are denotable by closed terms. Here I show that the current notion of validity, which does not suffer from this limitation, generalises the one of [32]. Let $\mathscr{L}$ and $\mathscr{L}'$ be languages with $[\![\ ]\!]_{\mathscr{L}}:\mathbb{T}_{\mathscr{L}}\rightarrow((\mathscr{X}\rightarrow\mathbf{V})\rightarrow\mathbf{V})$ and $[\![\ ]\!]_{\mathscr{L}'}:\mathbb{T}_{\mathscr{L}'}\rightarrow((\mathscr{X}\rightarrow\mathbf{V}')\rightarrow\mathbf{V}')$. The property of a language that all values in the domain of interpretation are denotable by closed terms can be stated as

$$\forall v\in\mathbf{V}.\ \exists P\in\mathrm{T}_{\mathscr{L}}.\ [\![P]\!]_{\mathscr{L}}=v\ . \tag{4}$$

Definition 22 ([32]) A translation $\mathscr{T}$ from $\mathscr{L}$ into $\mathscr{L}'$ *respects* $\backsim$ if (1) holds and $[\![\mathscr{T}(P)]\!]_{\mathscr{L}'}(\eta)\backsim[\![P]\!]_{\mathscr{L}}$ for all closed $\mathscr{L}$-expressions $P\in\mathrm{T}_{\mathscr{L}}$ and all valuations $\eta:\mathscr{X}\rightarrow\mathbf{U}$, with $\mathbf{U}:=\{v\in\mathbf{V}'\mid\exists v\in\mathbf{V}.\ v'\backsim v\}$.

In case $\mathbf{V}=\emptyset$, then $\mathbf{U}=\emptyset$, so by lack of any $\eta:\mathscr{X}\rightarrow\mathbf{U}$ each translation $\mathscr{T}$ from $\mathscr{L}$ into $\mathscr{L}'$ respects $\backsim$. When $\mathbf{V}\neq\emptyset$ the valuation η is needed because $\mathscr{T}(P)$ need not be a closed term, even if P is.

Definition 23 In [32], a translation from $\mathscr{L}$ into $\mathscr{L}'$ is called *valid up to* $\backsim$ if it is compositional and respects $\backsim$, while $\mathscr{L}$ satisfies (4).

Whereas (4) is an unwanted limitation, Example 2 in [32] shows that simply skipping requirement (4) in Definition 23 would yield a criterion that is too weak. The solution proposed by the current paper is to skip (4), while simultaneously strengthening or replacing the requirement of respecting $\backsim$ by the one of being valid up to $\backsim$ (Section 2). The following result state that, for languages $\mathscr{L}$ and $\mathscr{L}'$ that satisfy Postulate 2, and preorders $\backsim$ that respects $\stackrel{\alpha}{=}_{\mathscr{L}}$ and $\stackrel{\alpha}{=}_{\mathscr{L}'}$, any translation that is valid up to $\backsim$ according to [32] is certainly valid according to Definition 5 of the current paper.

Theorem 7 Let $\mathscr{L}$ and $\mathscr{L}'$ satisfy Postulate 2, let $\mathscr{L}$ satisfy (4), and let $\mathscr{T}$ be a compositional translation from $\mathscr{L}$ into $\mathscr{L}'$ that respects $\backsim$, where $\backsim$ respects $\stackrel{\alpha}{=}_{\mathscr{L}}$ and $\stackrel{\alpha}{=}_{\mathscr{L}'}$. Then $\mathscr{T}$ is valid up to $\backsim$.

Proof: In case $\mathbf{V}=\emptyset$, there is a unique semantic transition $\mathbf{R}\subseteq\mathbf{V}'\times\mathbf{V}$, namely $\emptyset\subseteq\backsim$. Since there are no valuations $\rho:\mathscr{X}\rightarrow\mathbf{V}$, $\mathscr{T}$ is correct w.r.t. $\mathbf{R}$, and thus valid up to $\backsim$.

Otherwise, let $\theta:\mathscr{X}\rightarrow\mathbf{U}:=\{v\in\mathbf{V}'\mid\exists v\in\mathbf{V}.\ v'\backsim v\}$—it exists by (1)—and define the semantic translation $\mathbf{R}\subseteq\mathbf{V}'\times\mathbf{V}$ by $\mathbf{R}:=\{([\![\mathscr{T}(P)]\!]_{\mathscr{L}'}(\theta),[\![P]\!]_{\mathscr{L}})\mid P\in\mathrm{T}_{\mathscr{L}}\}$. Then $\mathbf{R}\subseteq\backsim$, since $\mathscr{T}$ respects $\backsim$. Since $\backsim$ respects $\stackrel{\alpha}{=}_{\mathscr{L}}$ and $\stackrel{\alpha}{=}_{\mathscr{L}'}$, also $\mathbf{R}^\alpha\subseteq\backsim$. Hence it suffices to show that $\mathscr{T}$ is correct w.r.t. $\mathbf{R}^\alpha$. So let $E\in\mathbb{T}_{\mathscr{L}}$, $\rho:\mathscr{X}\rightarrow\mathbf{V}$ and $\eta:\mathscr{X}\rightarrow\mathbf{V}'$ with $\eta\ \mathbf{R}^\alpha\ \rho$. I need to show that $[\![\mathscr{T}(E)]\!]_{\mathscr{L}'}(\eta)\ \mathbf{R}^\alpha\ [\![E]\!]_{\mathscr{L}}(\rho)$.

Pick $\bar\rho : \mathscr{X} \to \mathbf{V}$ and $\bar\eta : \mathscr{X} \to \mathbf{V}'$ such that $\eta \stackrel{\alpha}{=}_{\mathscr{L}'} \bar\eta \;\mathbf{R}\; \bar\rho \stackrel{\alpha}{=}_{\mathscr{L}} \rho$. Then there is a closed substitution $\sigma : \mathscr{X} \to \mathrm{T}_{\mathscr{L}}$ such that $\bar\rho(X) = \llbracket \sigma(X) \rrbracket_{\mathscr{L}}$ and $\bar\eta(X) = \llbracket \mathscr{T}(\sigma(X)) \rrbracket_{\mathscr{L}'}(\theta)$ for all $X \in X$, i.e., $\bar\rho = \llbracket \sigma \rrbracket_{\mathscr{L}}$ and $\bar\eta = \llbracket \mathscr{T} \circ \sigma \rrbracket_{\mathscr{L}'}(\theta)$. Hence,

$$
\begin{array}{rlll}
\llbracket \mathscr{T}(E) \rrbracket_{\mathscr{L}'}(\eta) & \stackrel{\alpha}{=}_{\mathscr{L}'} & \llbracket \mathscr{T}(E) \rrbracket_{\mathscr{L}'}(\bar\eta) & \text{by Lemma 2} \\
& = & \llbracket \mathscr{T}(E) \rrbracket_{\mathscr{L}'}(\llbracket \mathscr{T} \circ \sigma \rrbracket_{\mathscr{L}'}(\theta)) & \text{derived above} \\
& \stackrel{\alpha}{=}_{\mathscr{L}'} & \llbracket \mathscr{T}(E)[\mathscr{T} \circ \sigma] \rrbracket_{\mathscr{L}'}(\theta) & \text{by (3)} \\
& \stackrel{\alpha}{=}_{\mathscr{L}'} & \llbracket \mathscr{T}(E[\sigma]) \rrbracket_{\mathscr{L}'}(\theta) & \text{by compositionality of } \mathscr{T} \\
& \mathbf{R} & \llbracket E[\sigma] \rrbracket_{\mathscr{L}} & \text{by definition of } \mathbf{R} \\
& \stackrel{\alpha}{=}_{\mathscr{L}} & \llbracket E \rrbracket_{\mathscr{L}}(\llbracket \sigma \rrbracket_{\mathscr{L}}) & \text{by (3)} \\
& = & \llbracket E \rrbracket_{\mathscr{L}}(\bar\rho) & \text{derived above} \\
& \stackrel{\alpha}{=}_{\mathscr{L}} & \llbracket E \rrbracket_{\mathscr{L}}(\rho) & \text{by Lemma 2.} \quad \square
\end{array}
$$

15 The case where all semantic values are denotable by closed terms

Here I show that the notion of validity from the current paper agrees with the one of [32] when applied to languages for which all values are denotable by closed terms.

Usually one employs translations $\mathscr{T}$ with the property that for any $E \in \mathbb{T}_{\mathscr{L}}$ any free variable of $\mathscr{T}(E)$ is also a free variable of E—I call these *free-variable respecting translations*, or *fvr-translations* [32]. If $\mathscr{T}_0$ in the proof of Theorem 5 is an fvr-translation, then so is $\mathscr{T}$. Hence, replaying the proof of Corollary 2, one obtains:

Observation 4 Let $\mathscr{L}$ and $\mathscr{L}'$ be languages that satisfy Postulates 1 and 2, and $\sim$ a preorder that respects $\stackrel{\alpha}{=}_{\mathscr{L}}$ and $\stackrel{\alpha}{=}_{\mathscr{L}'}$. If any valid (or correct) fvr-translation from $\mathscr{L}$ into $\mathscr{L}'$ up to $\sim$ exists, then there exists a compositional fvr-translation that is valid (or correct) up to $\sim$.

An fvr-translation $\mathscr{T}$ from $\mathscr{L}$ into $\mathscr{L}'$ respects $\sim$ iff either $\mathbf{V} = \emptyset$, or (1) holds and $\llbracket \mathscr{T}(P) \rrbracket_{\mathscr{L}'} \sim \llbracket P \rrbracket_{\mathscr{L}}$ for all closed $\mathscr{L}$-expressions $P \in \mathrm{T}_{\mathscr{L}}$.

Observation 5 Any fvr-translation that is valid up to $\sim$ respects $\sim$.

Lemma 4 Let $\mathscr{L}'$ satisfy (4) and $\mathbf{V} \neq \emptyset$. If there is a translation from $\mathscr{L}$ into $\mathscr{L}'$ that is valid up to a preorder $\sim$, then there is a fvr-translation from $\mathscr{L}$ into $\mathscr{L}'$ that is valid up to $\sim$.

Proof: Let $\mathscr{T}$ be a translation from $\mathscr{L}$ into $\mathscr{L}'$ that is valid up to $\sim$. Let $\mathbf{R} \subseteq \sim$ be a semantic translation such that $\mathscr{T}$ is correct w.r.t. $\mathbf{R}$. Let $P \in \mathrm{T}_{\mathscr{L}'}$ be a closed $\mathscr{L}'$-term with $\llbracket P \rrbracket_{\mathscr{L}'} \;\mathbf{R}\; v$ for some $v \in \mathbf{V}$—such a P exists by (4). Let the translation $\mathscr{T}'$ be obtained from $\mathscr{T}$ by defining $\mathscr{T}'(E) := \mathscr{T}(E)[\sigma_E]$ for all $E \in \mathbb{T}_{\mathscr{L}}$—here σ_E is the substitution that (only) substitutes P for all variables X that occur in $\mathscr{T}(E)$ but not in E. Then $\mathscr{T}'$ is a fvr-translation. It remains to show that $\mathscr{T}'$ is correct w.r.t. $\mathbf{R}$.

So let $E \in \mathbb{T}_{\mathscr{L}}$, $\eta : \mathscr{X} \to \mathbf{V}'$ and $\rho : \mathscr{X} \to \mathbf{V}$ with $\eta \,\mathbf{R}\, \rho$. I aim to show that $\llbracket \mathscr{T}'(E) \rrbracket_{\mathscr{L}'}(\eta) \;\mathbf{R}\; \llbracket E \rrbracket_{\mathscr{L}}(\rho)$. Let $\bar\eta : \mathscr{X} \to \mathbf{V}'$ and $\bar\rho : \mathscr{X} \to \mathbf{V}$ be substitutions with $\bar\eta(X) = \eta(X)$ and $\bar\rho(X) = \rho(X)$ for all $X \in \mathit{fv}(E)$, such that $\bar\eta(Y) = \llbracket P \rrbracket_{\mathscr{L}'}$ and $\bar\rho(Y) = v$ for all $Y \in \mathit{fv}(\mathscr{T}(E)) \setminus \mathit{fv}(E)$. Then $\bar\eta \;\mathbf{R}\; \bar\rho$. Consequently,

$$\llbracket \mathscr{T}'(E) \rrbracket_{\mathscr{L}'}(\eta) = \llbracket \mathscr{T}(E)[\sigma_E] \rrbracket_{\mathscr{L}'}(\eta) = \llbracket \mathscr{T}(E) \rrbracket_{\mathscr{L}'}(\bar\eta) \;\mathbf{R}\; \llbracket E \rrbracket_{\mathscr{L}}(\bar\rho) = \llbracket E \rrbracket_{\mathscr{L}}(\rho). \qquad \square$$

The following result states that in case both $\mathscr{L}$ and $\mathscr{L}'$ satisfy (4), as well as Postulates 1 and 2, and $\sim$ respects $\stackrel{\alpha}{=}_{\mathscr{L}}$ and $\stackrel{\alpha}{=}_{\mathscr{L}'}$, the validity-based notion of expressiveness from [32] coincides with the one here.

Corollary 3 Let $\mathscr{L}$ and $\mathscr{L}'$ be languages satisfying Postulates 1 and 2, as well as (4), and $\sim$ a preorder respecting $\stackrel{\alpha}{=}_{\mathscr{L}}$ and $\stackrel{\alpha}{=}_{\mathscr{L}'}$. There exists a valid translation from $\mathscr{L}$ into $\mathscr{L}'$ up to $\sim$ iff there exists a compositional translation from $\mathscr{L}$ into $\mathscr{L}'$ that respects $\sim$.

Proof: Suppose a valid translation up to $\precsim$ exists. In case $\mathbf{V} = \emptyset$, by Corollary 2 there exists a compositional translation $\mathscr{T}$ that is valid up to $\sim$, and by the remark following Definition 22 $\mathscr{T}$ respects $\precsim$.

In case $\mathbf{V} \neq \emptyset$, by Lemma 4 there exists an fvr-translation that is valid up to $\precsim$. By Observation 4, there exists a compositional fvr-translation $\mathscr{T}$ that is valid up to $\sim$. By Observation 5, $\mathscr{T}$ respects $\precsim$.

The converse direction follows from Theorem 7. □

16 A potential generalisation of the concept of a valid translation

In this section I consider a potentially more liberal concept of a valid transition up to a preorder $\precsim$, namely a translation that is compositional and *preserves* $\precsim$, as in Definition 24 below. Like correctness w.r.t. a semantic translation (Definition 3) and consequently also correctness and validity up to $\precsim$ (Definitions 4, 5 and 6), the requirement of Definition 24 is that the translation preserves the meaning of expressions: the meaning of the translation of an expression E should be semantically equivalent to the meaning of E —see [32] for an elaboration. In fact, this should hold under any valuation of the variables occurring in E. The difference between Definitions 6 and 24 is that the former is based on universal quantification of all matching valuations in the target language, whereas the latter associates with any valuation in the source language a single matching valuation in the target language. This requires a translation $\mathscr{T} : \mathbb{T}_{\mathscr{L}} \to \mathbb{T}_{\mathscr{L}'}$ to have a *semantic counterpart* $\mathbf{T} : \mathbf{V} \to \mathbf{V}'$ that maps the possible meanings of $\mathscr{L}$-expressions into the possible meanings of $\mathscr{L}'$-expressions.

Definition 24 A translation $\mathscr{T}$ from $\mathscr{L}$ into $\mathscr{L}'$ *preserves* $\precsim$ iff there exists a mapping $\mathbf{T} : \mathbf{V} \to \mathbf{V}'$ such that $\mathbf{T}(v) \precsim v$ for all $v \in \mathbf{V}$ and $[\![\mathscr{T}(E)]\!]_{\mathscr{L}'}(\mathbf{T} \circ \rho) \precsim [\![E]\!]_{\mathscr{L}}(\rho)$ for all $E \in \mathbb{T}_{\mathscr{L}}$ and all $\rho : \mathscr{X} \to \mathbf{V}$.

Note that the existence of $\mathbf{T}$ implies (1) in Section 3.

Proposition 7 Any valid translation up to a preorder $\precsim$ preserves $\precsim$.

Proof: Let $\mathscr{T}$ be correct w.r.t. the semantic translation $\mathbf{R} \subseteq \precsim$. Take any $\mathbf{T} : \mathbf{V} \to \mathbf{V}'$ with $\mathbf{T} \subseteq \mathbf{R}$. Then $\mathbf{T} \circ \rho \ \mathbf{R} \ \rho$ for any valuation $\rho : \mathscr{X} \to \mathbf{V}$. Hence $\mathscr{T}$ preserves $\precsim$ using the semantic counterpart $\mathbf{T}$. □

Corollary 2 does not extend (from correct and valid translations up to $\precsim$) to translations that preserve $\precsim$.

Example 9 Let $\mathscr{L}$ be a language with unary operators S and 2, and $\mathscr{L}'$ a language with unary operators S and G_k for each $k > 0$. Their semantics is given by $\mathbf{V} = \mathbf{V}' = \mathbb{N}$, $S^{\mathbf{V}}(n) = S^{\mathbf{V}'}(n) = n+1$ for each $n \in \mathbb{N}$, $2^{\mathbf{V}}(n) = 2n$ for each $n \in \mathbb{N}$, and $G_k^{\mathbf{V}'}(n) = 2^k n + 1$ for each $n \in \mathbb{N}$ and $k > 0$.

Let $\sim$ be the equivalence relation on $\mathbb{N}$ defined by $n \sim m$ iff n and m differ only by a factor 2^k for some $k \in \mathbb{N}$, i.e., $n = m = 0$ or their prime factorisations, when leaving out the factors of 2, are the same.

Each term $E \in \mathbb{T}_{\mathscr{L}}$ is obtained from a single variable X by applications of the operators S and 2. Using the equation $2SF = SS2F$, each such term E can be rewritten into a unique *normal form* $S^i 2^j X$ for some $i, j \in \mathbb{N}$—notation $E \mapsto S^i 2^j X$. If $E \mapsto S^i 2^j X$ then $[\![S^i 2^j X]\!]_{\mathscr{L}}(\rho) = i + 2^j \rho(X) = [\![E]\!]_{\mathscr{L}}(\rho)$ for each valuation $\rho : \mathscr{X} \to \mathbb{N}$, and thus $[\![S^i 2^j X]\!]_{\mathscr{L}} \sim [\![E]\!]_{\mathscr{L}}$.

Let $\mathscr{T} : \mathbb{T}_{\mathscr{L}} \to \mathbb{T}_{\mathscr{L}'}$ be the translation given by
$$\begin{cases} \mathscr{T}(E) = \mathscr{T}(S^i 2^j X) & \text{if } E \mapsto S^i 2^j X \\ \mathscr{T}(2^j X) = X \\ \mathscr{T}(S^{i+1} X) = S^{i+1} X \\ \mathscr{T}(S^{i+1} 2^j X) = S^i G_j X. \end{cases}$$

Then $\mathscr{T}$ preserves $\sim$, taking $\mathbf{T}$ to be the identity function on $\mathbb{N}$. Moreover, the preconditions of Corollary 2 are trivially satisfied. Yet, there is no compositional translation that preserves $\sim$.

For suppose $\mathscr{T}'$ is such a compositional translation, with semantic counterpart $\mathbf{T}$. Then there is a term E_2 such that $\mathscr{T}'(2X) = E_2$. Using that $\mathscr{T}'$ preserves $\sim$, one has $[\![E_2]\!]_{\mathscr{L}'}(\mathbf{T} \circ \rho) \sim [\![2X]\!]_{\mathscr{L}}(\rho)$ for each $\rho : \mathscr{X} \to \mathbb{N}$. Take $\rho(X) = 0$ and $\rho(Y) = 7$ for each $Y \neq X$. Then $\mathbf{T}(\rho(Y)) \sim 7$, so $\mathbf{T}(\rho(Y)) \geq 7$, for each $Y \neq X$. Since $[\![2X]\!]_{\mathscr{L}}(\rho_0) = 0$, also $[\![E_2]\!]_{\mathscr{L}'}(\mathbf{T} \circ \rho_0) = 0$. As the operators S and G_k are strictly increasing, it follows that E_2 contains none of these operators, nor variables $Y \neq X$. So $E_2 = X$.

As $\mathscr{T}'$ is compositional, it follows that $\mathscr{T}'(S2X) = \mathscr{T}'(SX)$. Thus, taking $\rho(X) = 1$, one obtains $2 = [\![SX]\!]_{\mathscr{L}'}(\rho) \sim [\![\mathscr{T}'(SX)]\!]_{\mathscr{L}'}(\mathbf{T} \circ \rho) = [\![\mathscr{T}'(S2X)]\!]_{\mathscr{L}'}(\mathbf{T} \circ \rho) \sim [\![S2X]\!]_{\mathscr{L}}(\rho) = 3$, a contradiction.

For this reason, when using preservation of $\sim$ as a criterion like validity and correctness up to $\sim$, compositionality has to be required separately.

By Observation 4 and Proposition 7, assuming that $\mathscr{L}$ and $\mathscr{L}'$ satisfy Postulates 1 and 2 and $\sim$ respects $\stackrel{\alpha}{=}_{\mathscr{L}}$ and $\stackrel{\alpha}{=}_{\mathscr{L}'}$, if a valid fvr-translation from $\mathscr{L}$ into $\mathscr{L}'$ up to $\sim$ exists, then there exists a compositional fvr-translation that preserves $\sim$. Below I establish the reverse, thereby showing that my concept of a valid translation is quite general.

Theorem 8 Let $\mathscr{L}$ and $\mathscr{L}'$ be languages that satisfy Postulate 2, and $\sim$ a preorder that respects $\stackrel{\alpha}{=}_{\mathscr{L}}$ and $\stackrel{\alpha}{=}_{\mathscr{L}'}$. Any compositional fvr-translation $\mathscr{T} : \mathbb{T}_{\mathscr{L}} \to \mathbb{T}_{\mathscr{L}'}$ that preserves $\sim$ is valid up to $\sim$.

Proof: Let $\mathscr{T}$ be a compositional fvr-translation that preserves $\sim$, say by means of the semantic counterpart $\mathbf{T}\colon \mathbf{V} \to \mathbf{V}'$. Let $\mathbf{R} \subseteq \mathbf{V}' \times \mathbf{V}$ be the smallest relation containing $\mathbf{T}^{-1}$ such that $\mathscr{T}$ is correct w.r.t. $\mathbf{R}$. By Lemma 3 $\mathbf{T}$ is correct w.r.t. $\mathbf{R}^\alpha$. I need to show that $\mathbf{R}^\alpha \subseteq \sim$.

Claim: If $v' \,\mathbf{R}\, v$ then $v' \stackrel{\alpha}{=}_{\mathscr{L}'} [\![\mathscr{T}(E)]\!]_{\mathscr{L}'}(\mathbf{T} \circ \nu)$ and $[\![E]\!]_{\mathscr{L}}(\nu) \stackrel{\alpha}{=}_{\mathscr{L}} v$ for some $E \in \mathbb{T}_{\mathscr{L}}$ and $\nu : \mathscr{X} \to \mathbf{V}$.

Establishing the validity of this claim is sufficient, because by Definition 24 and the transitivity of $\sim$, using that $\stackrel{\alpha}{=}_{\mathscr{L}}, \stackrel{\alpha}{=}_{\mathscr{L}'} \subseteq \sim$, it immediately implies that $\mathbf{R}^\alpha \subseteq \sim$.

I prove the claim with induction on the construction of $\mathbf{R}$.

Induction base: If $v' \,\mathbf{R}\, v$ because $v' = \mathbf{T}(v)$ the claim holds by taking $E := X$ and $\nu(X) = v$.

Induction step: Let $v' = [\![\mathscr{T}(E)]\!]_{\mathscr{L}'}(\eta)$ and $v = [\![E]\!]_{\mathscr{L}}(\rho)$ for some $E \in \mathbb{T}_{\mathscr{L}}$, $\eta : \mathscr{X} \to \mathbf{V}'$ and $\rho : \mathscr{X} \to \mathbf{V}$ with $\eta \,\mathbf{R}\, \rho$. By induction one may assume that for each $X \in fv(E)$ there are $E_X \in \mathbb{T}_{\mathscr{L}}$ and $\nu_X : \mathscr{X} \to \mathbf{V}$ such that $\eta(X) \stackrel{\alpha}{=}_{\mathscr{L}'} [\![\mathscr{T}(E_X)]\!]_{\mathscr{L}'}(\mathbf{T} \circ \nu_X)$ and $[\![E_X]\!]_{\mathscr{L}}(\nu_X) \stackrel{\alpha}{=}_{\mathscr{L}} \rho(X)$. Let $(\sigma_X)_{X \in fv(E)}$ be a family of injective renamings of variables with $dom(\sigma_X) = fv(E_X)$, such that the sets $range(\sigma_X) := \sigma_X(fv(E_X))$ for $X \in fv(E)$ are pairwise disjoint. Here I assume the pool of variables to draw from is large enough. Note that $[\![\sigma_X]\!]_{\mathscr{L}}(\nu_X \circ \sigma_X^{-1}) = \nu_X$ and $[\![\sigma_X]\!]_{\mathscr{L}'}(\mathbf{T} \circ \nu_X \circ \sigma_X^{-1}) = \mathbf{T} \circ \nu_X$. Therefore, using (3), $[\![\mathscr{T}(E_X)]\!]_{\mathscr{L}'}(\mathbf{T} \circ \nu_X) \stackrel{\alpha}{=}_{\mathscr{L}'} [\![\mathscr{T}(E_X)[\sigma_X]]\!]_{\mathscr{L}'}(\mathbf{T} \circ \nu_X \circ \sigma_X^{-1})$ and $[\![E_X[\sigma_X]]\!]_{\mathscr{L}}(\nu_X \circ \sigma_X^{-1}) \stackrel{\alpha}{=}_{\mathscr{L}} [\![E_X]\!]_{\mathscr{L}}(\nu_X)$. By the compositionality of $\mathscr{T}$, $\mathscr{T}(E_X)[\sigma_X] \stackrel{\alpha}{=} \mathscr{T}(E_X[\sigma_X])$, so $\eta(X) \stackrel{\alpha}{=}_{\mathscr{L}'} [\![\mathscr{T}(E_X[\sigma_X])]\!]_{\mathscr{L}'}(\mathbf{T} \circ \nu_X \circ \sigma_X^{-1})$.

Now let $\nu\colon \mathscr{X} \to \mathbf{V}$ be a valuation such that $\nu(Y) = \nu_X \circ \sigma_X^{-1}(Y)$ if $Y \in fv(E_X[\sigma_X])$ for some $X \in fv(E)$. Furthermore, define the substitution $\sigma : fv(E) \to \mathbb{T}_{\mathscr{L}}$ by $\sigma(X) = E_X[\sigma_X]$. Then $\rho(X) \stackrel{\alpha}{=}_{\mathscr{L}} [\![\sigma(X)]\!]_{\mathscr{L}}(\nu)$ for all $X \in fv(E)$, so $v \stackrel{\alpha}{=}_{\mathscr{L}} [\![E]\!]_{\mathscr{L}}([\![\sigma]\!]_{\mathscr{L}}(\nu))$ by Lemma 2. Likewise, $\eta(X) \stackrel{\alpha}{=}_{\mathscr{L}'} [\![\mathscr{T}(\sigma(X))]\!]_{\mathscr{L}'}(\mathbf{T} \circ \nu)$ for all $X \in fv(E) \supseteq fv(\mathscr{T}(E))$, so $v' \stackrel{\alpha}{=}_{\mathscr{L}'} [\![\mathscr{T}(E)]\!]_{\mathscr{L}'}([\![\mathscr{T} \circ \sigma]\!]_{\mathscr{L}'}(\mathbf{T} \circ \nu))$. Now (3) yields $v \stackrel{\alpha}{=}_{\mathscr{L}} [\![E[\sigma]]\!]_{\mathscr{L}}(\nu)$ and $v' \stackrel{\alpha}{=}_{\mathscr{L}'} [\![\mathscr{T}(E)[\mathscr{T} \circ \sigma]]\!]_{\mathscr{L}'}(\mathbf{T} \circ \nu)$. By compositionality, $\mathscr{T}(E)[\mathscr{T} \circ \sigma] \stackrel{\alpha}{=} \mathscr{T}(E[\sigma])$, from which it follows that $v' \stackrel{\alpha}{=}_{\mathscr{L}'} [\![\mathscr{T}(E[\sigma])]\!]_{\mathscr{L}'}(\mathbf{T} \circ \nu)$. □

Corollary 4 Let $\mathscr{L}$ and $\mathscr{L}'$ be languages satisfying Postulates 1 and 2, and $\sim$ a preorder respecting $\stackrel{\alpha}{=}_{\mathscr{L}}$ and $\stackrel{\alpha}{=}_{\mathscr{L}'}$. There exists a valid fvr-translation from $\mathscr{L}$ into $\mathscr{L}'$ up to $\sim$ iff there exists a compositional fvr-translation from $\mathscr{L}$ into $\mathscr{L}'$ that preserves $\sim$.

There exist translations that are compositional and preserve $\sim$ but are not fvr-translations and not valid in the sense of Definition 5. Examples are hard to find and do not appear very natural. Moreover, major results I establish about valid translations (Theorems 4 and 6) do not generalise to such examples. For these reasons I prefer to exclude them from my definition of a valid translation.

Example 10 Let $\mathscr{L}$ be a language with constants $\mathbf{0}$ and $\mathbf{1}$, and a semantics given by $\mathbf{V}=\{a,b\}$, $\mathbf{0}^{\mathbf{V}}=a$ and $\mathbf{1}^{\mathbf{V}}=b$. Let $\mathscr{L}'$ be a language with a unary operator f, and a semantics given by $\mathbf{V}'=\{1,2,3,4\}$ and $f^{\mathbf{V}'}(n)=n{+}1(\text{mod } 4)$. Finally, let $\sim$ be the equivalence on $\mathbf{V}\cup\mathbf{V}'$ given by $a\sim 1\sim 2\not\sim 3\sim 4\sim b$.

The translation $\mathscr{T}:\mathbb{T}_{\mathscr{L}}\rightarrow\mathbb{T}_{\mathscr{L}'}$ given by $\mathscr{T}(\mathbf{0})=f(X_0)$ and $\mathscr{T}(\mathbf{1})=f(f(f(X_0)))$ for some $X_0\in\mathscr{X}$, and $\mathscr{T}(X)=X$ for all $X\in\mathscr{X}$, is compositional by construction, and preserves $\mathrel{\bullet\!\!\sim}$. This is witnessed by its semantic counterpart $\mathbf{T}:\mathbf{V}\rightarrow\mathbf{V}'$ given by $\mathbf{T}(a)=1$ and $\mathbf{T}(b)=4$.

Since $\mathrm{T}_{\mathscr{L}'}=\emptyset$, there do not exists fvr-translations from $\mathscr{L}$ to $\mathscr{L}'$. Up to symmetry, $\mathscr{T}$ is in fact the only translation from $\mathscr{L}$ to $\mathscr{L}'$ that preserves $\mathrel{\bullet\!\!\sim}$. For considering that $f^4(n)=n$ and the choice of X_0 is immaterial, any such translation $\mathscr{T}_{k,\ell}$ must satisfy $\mathscr{T}(\mathbf{0})=f^k(X_0)$ and $\mathscr{T}(\mathbf{1})=f^{\ell}(X_0)$, where $k,\ell\in\{0,1,2,3\}$. The only choices for $\mathbf{T}(a)$ are 1 or 2, and by symmetry one can pick $\mathbf{T}(a)=1$. Hence, to satisfy $[\![\mathscr{T}_{k,\ell}(a)]\!]_{\mathscr{L}'}(\mathbf{T}\circ\rho)\mathrel{\bullet\!\!\sim}[\![a]\!]_{\mathscr{L}}(\rho)$ when $\rho(X_0)=a$, one must take $k\in\{0,1\}$. To also satisfy this formula when $\rho(X_0)=b$, one must take $k=1$ and $\mathbf{T}(b)=4$. This forces $\ell\in\{2,3\}\cap\{3,0\}$, so $\mathscr{T}_{k,\ell}=\mathscr{T}_{1,3}$.

However, there does not exists a translation from $\mathscr{L}$ to $\mathscr{L}'$ that is valid up to $\mathrel{\bullet\!\!\sim}$. For by symmetry, using Proposition 7, $\mathscr{T}$ is the only candidate for such a translation. Suppose $\mathbf{R}$ is a semantic translation w.r.t. which $\mathscr{T}$ is correct. Then $1\ \mathbf{R}\ a$ or $2\ \mathbf{R}\ a$. Suppose $1\ \mathbf{R}\ a$. Take valuations ρ,η with $\eta\ \mathbf{R}\ \rho$, such that $\rho(X_0)=a$ and $\eta(X_0)=1$. As $\mathscr{T}$ is correct w.r.t. $\mathbf{R}$, one has

$$2=f^{\mathbf{V}'}(1)=[\![f(X_0)]\!]_{\mathscr{L}'}(\eta)=[\![\mathscr{T}(\mathbf{0})]\!]_{\mathscr{L}'}(\eta)\ \mathbf{R}\ [\![\mathbf{0}]\!]_{\mathscr{L}}(\rho)=a.$$

So one must have $2\ \mathbf{R}\ a$. Now take valuations ρ,η with $\eta\ \mathbf{R}\ \rho$, such that $\rho(X_0)=a$ and $\eta(X_0)=2$. As $\mathscr{T}$ is correct w.r.t. $\mathbf{R}$, one has

$$3=f^{\mathbf{V}'}(2)=[\![f(X_0)]\!]_{\mathscr{L}'}(\eta)=[\![\mathscr{T}(\mathbf{0})]\!]_{\mathscr{L}'}(\eta)\ \mathbf{R}\ [\![\mathbf{0}]\!]_{\mathscr{L}}(\rho)=a.$$

This contradicts the requirement that $\mathbf{R}\subseteq\sim$.

Theorem 6 does not extend to this translation. For $\mathscr{L}$ is a closed-term language and $\mathscr{L}'$ trivially satisfies Postulate 2 (for by lack of bound variables $\stackrel{\alpha}{=}_{\mathscr{L}'}$ is the identity). Any congruence $\approx$ for $\mathscr{L}'$ contained in $\sim$ must distinguish all four semantic values. Suppose $\mathscr{T}$ would preserve an equivalence $\approx_{\mathscr{T}}$ on $\mathbf{V}\cup\mathbf{V}'$, contained in $\approx$, that on $\mathbf{V}'$ coincides with $\approx$. Let $\mathbf{T}'$ be a semantic counterpart of $\mathscr{T}$. Then $\mathbf{T}'(a)=1$ or $\mathbf{T}'(a)=2$. Suppose $\mathbf{T}'(a)=1$. Take a valuation ρ with $\rho(X_0)=a$. Then

$$2=f^{\mathbf{V}'}(1)=[\![f(X_0)]\!]_{\mathscr{L}'}(\mathbf{T}'\circ\rho)=[\![\mathscr{T}(\mathbf{0})]\!]_{\mathscr{L}'}(\mathbf{T}'\circ\rho)\approx_{\mathscr{T}}[\![\mathbf{0}]\!]_{\mathscr{L}}(\rho)=a.$$

So one has $2\approx_{\mathscr{T}}a\approx_{\mathscr{T}}1$, a contradiction. The case $\mathbf{T}'(a)=2$ leads to a contradiction in the same way. Hence $\mathscr{T}$ does not preserve such an equivalence $\approx_{\mathscr{T}}$.

17 Validity up to barbed bisimilarity for process calculi

In this paper I defined a notion of a valid translation op to a semantic equivalence $\sim$ between general system description languages, dealing in Section 5 with the special case of closed-term languages. In this section I zoom in further on process calculi such as CCS [50] and the π-calculus [76], and ask which equivalences $\sim$ to use in studying their relative expressiveness. These languages have in common that their semantics can be given in terms of labelled transition systems, whose states, called *processes*, are the closed expressions in the language, and whose transitions are given by an operational semantics in the style of Plotkin [71].

For a wide class of process calculi without name-binding, the finest equivalence in regular employ is strong bisimilarity [50]. In proving that process calculus $\mathscr{L}'$ is at least as expressive as process calculus

$\mathscr{L}$ up to a semantic equivalence $\sim$, the choice of strong bisimilarity for $\sim$ thus yields the strongest result. Accordingly, Robert de Simone [79] showed that a wide class of process calculi, including CCS [50], CSP [11], ACP [6] and SCCS [49], are expressible up to strong bisimilarity in MEIJE [3].

However, strong bisimilarity is not suitable for comparing any two process calculi. As an example, consider the π-calculus with the early operational semantics versus the π-calculus with the late operational semantics [76]. Both operational semantics are meant to convey the same idea on how π-calculus processes interact. Consequently, one would hope and expect that the identity function between these versions of the π-calculus is a valid translation. However, it is not valid up to strong bisimilarity. The late semantics has transitions labelled $x(y)$, where the name y is a variable in which a value received on channel x will be stored, whereas the early semantics has transitions labelled xy, where the name y is a particular value received on channel x. Since these labels have a different shape, and strong bisimilarity requires matching labels, validity up to strong bisimilarity fails. Moreover, the problem cannot be resolved by simply renaming the labels, for the entire meaning of input transitions is different.

In the π-calculus, *strong early bisimilarity* [76] is a much more canonical semantic equivalence than strong bisimilarity, and the identity function *is* a valid transition between the early and late π-calculi up to strong early bisimilarity. However, the definition of early bisimilarity for the π-calculus with the late operational semantics [76] has a very different form than the definition of early bisimilarity for the early π-calculus [76]. The reason is that the same idea needs to be phrased in terms of rather different transition relations. Consequently, taking strong early bisimilarity as a unifying equivalence for comparing the late and early π-calculus appears to be somewhat ad hoc. Moreover, neither strong bisimilarity nor strong early bisimilarity would be suitable to compare, say, the π-calculus with CCS.

A canonical semantic equivalence that can be defined in a uniform way on CCS as well as on the late and early π-calculi is *strong barbed bisimilarity*, originally proposed by Milner and Sangiorgi in [52]. It is based on the idea that a τ-transition of a process P describes an actual reaction of the process P, whereas a translation labelled $a \neq \tau$ merely indicates a potential reaction of P when synchronising with a communication partner that is willing to engage in the complementary transition $\bar{a}$. (Likewise, in CSP [11] a transition $P \xrightarrow{a} P'$ can be regarded as a potential; one that is not realised when putting the process in a parallel composition involving synchronisation on the action a when the other component cannot partake in such a synchronisation. The only way to be sure that the CSP action a cannot be inhibited by any synchronisation context is to *conceal* it from the environment, using the CSP concealment operator $\_\backslash a$, renaming the action a into τ.) From this perspective, it appears natural to formulate semantic equivalences entirely in terms of the τ-transitions processes can do, and capture the communication potentials (manifested by the other transitions) merely by studying the behaviour of processes in contexts. Accordingly, *(strong) reduction bisimilarity* was defined in [52] as the version of strong bisimilarity that requires the transfer property for τ-transitions only, i.e., strong barbed bisimilarity as defined below, without the first clause, and reading $\xrightarrow{\tau}$ for $\rightarrow$. Naturally, reduction bisimilarity is not a congruence: the CCS processes $a.\mathbf{0}$ and $\mathbf{0}$ are reduction equivalent (for neither can do a τ-transition), yet $a.\mathbf{0}|\bar{a}.\mathbf{0}$ and $\mathbf{0}|\bar{a}.\mathbf{0}$ are not. The purpose of the relation is to define a reasonable semantic equivalence as its congruence closure. Indeed, for divergence-free CCS processes strong bisimilarity turned out to be the congruence closure of reduction bisimilarity [52]. Unfortunately, this failed for processes with divergences (infinite-sequences of τ-transitions) [52], and when lifted to the weak case, weak reduction congruence turned out to be the universal relation, and thus useless [52]. For this reason, reduction bisimilarity needed to be strengthened. The main idea is that processes like $a.\mathbf{0}$ and $\mathbf{0}$ can be told apart by placing them in a context $\_|\bar{a}.\omega$, where ω denotes a success state that might be reached by the environment in which a process is placed. The process $a.\mathbf{0}|\bar{a}.\omega$ can reach this success state by performing a τ-transition, whereas $\mathbf{0}|\bar{a}.\omega$ can not. Writing $P{\downarrow_\omega}$ to indicate that a process P has reached this success state, reduction bisimilarity can be strengthened by requiring that if $P{\downarrow_\omega}$ then any process Q equivalent to P should also satisfy $Q{\downarrow_\omega}$. This yields the notion of strong barbed bisimilarity. The predicate $\_{\downarrow_\omega}$ is called a *barb*. In general, one can use a collection of barbs.

Barbed bisimilarity is defined on closed-term languages $\mathscr{L}$ that are equipped with a binary *reduction relation* $\rightarrow\ \subseteq \mathrm{T}_\mathscr{L} \times \mathrm{T}_\mathscr{L}$ between processes and with a collection $\{{\downarrow_\omega} \subseteq \mathrm{T}_\mathscr{L} \mid \omega \in \Omega\}$ of unary predicates on processes, called *barbs*.

Definition 25 *Strong barbed bisimilarity* is the largest symmetric relation $\stackrel{\bullet}{\sim}\ \subseteq \mathrm{T}_\mathscr{L} \times \mathrm{T}_\mathscr{L}$ such that

- $P \stackrel{\bullet}{\sim} Q$ and $P{\downarrow_\omega}$ with $b \in \Omega$ implies $Q{\downarrow_\omega}$, and
- $P \stackrel{\bullet}{\sim} Q$ and $P \rightarrow P'$ implies $Q \rightarrow Q'$ for some Q' with $P' \stackrel{\bullet}{\sim} Q'$.

Strong barbed congruence, $\stackrel{\bullet}{\sim}{}^{1c}_{\mathscr{L}}$, is the congruence closure of $\stackrel{\bullet}{\sim}$ w.r.t. the language $\mathscr{L}$.

For process calculi equipped with a labelled transition system semantics, the reductions $P \rightarrow P'$ are simply the τ-transitions. However, for many languages it is possible to give a *reduction semantics* that generates the reductions directly, without first constricting a labelled transition system. Some languages, such as the λ-calculus, come with a natural reduction semantics, even when they have no labelled transition semantics at all.

The definition of the barbs on various process calculi is somewhat ad hoc. In [52] only one barb was used, and a process has this barb when it can perform an observable action, i.e., $P{\downarrow}$ iff $P \xrightarrow{a} P'$ for some process P' and action $a \neq \tau$. An alternative is to introduce a barb a for each action $a \neq \tau$, taking $P{\downarrow_a}$ iff $P \xrightarrow{a} P'$ for some P'. Both forms of barbs lead to the same notion of strong barbed congruence. In the π-calculus, x and $\bar{x}$ for $x \in \mathscr{N}$ are taken to be barbs (or just $\bar{x}$), with $P{\downarrow_\omega}$ iff $P \xrightarrow{\omega(y)} P'$ or $P \xrightarrow{\omega y} P'$ for some P'—cf. Section 6. Again, including barbs $x \in \mathscr{N}$ or not makes no difference in the resulting notion of strong barbed congruence. In fact, it does not matter at all how the barbs are defined, but only whether there are enough barbs, and how they propagate upwards through a context (that is, how the validity of $C[P]{\downarrow_\omega}$ is determined by the validity of $P{\downarrow_\omega}$). This is because in determining whether two processes P and Q are strongly barbed congruent, the barbs that help in this determination appear in contexts in which the processes P and Q are placed, rather than in P and Q themselves. For this reason, aiming for a fairly language-independent definition of barbed congruence, I propose the use of *external barbs* instead, defined as follows. Postulate a sufficiently large collection Ω of barbs and add each of them as a fresh constant to the language under consideration. Require them to propagate upwards through a context in the same way as τ-transitions or reduction steps. Thus, whenever the structural operational semantics of the language has a rule

$$\frac{P_i \rightarrow Q}{f(P_1,\ldots,P_n) \rightarrow R}$$

for some n-ary operator f, possibly with $\xrightarrow{\tau}$ in the rôle of $\rightarrow$, then postulate a rule

$$\frac{P_i{\downarrow_\omega}}{f(P_1,\ldots,P_n){\downarrow_\omega}}$$

for each barb $\omega \in \Omega$. Note that this agrees exactly with Definition 12 of (strong) output barbs in the π-calculus. It turns out that for CCS, the π-calculus, and in fact any language with a definition of strong barbed congruence that I am aware of, the above use of external barbs yields the same notion of strong barbed congruence as the original approach. It may be felt as a drawback that in defining strong barbed congruence on a language $\mathscr{L}$, $\mathscr{L}$ needs to be extended, namely by the added constants $\omega \in \Omega$. However, as a tool to define strong barbed congruence on $\mathscr{L}$ this is not a big problem, as the definition on the extended language naturally restricts to $\mathscr{L}$.

The treatment of barbs proposed above is strongly inspired by the success action ω in the treatment of *testing equivalences* by De Nicola & Hennessy [19], and by the criterion of *success sensitiveness* imposed by Gorla [39] on translations between process calculi. Both [19] and [39] use only a single success predicate (barb). In [39] $P{\downarrow_\omega}$ is defined to hold iff P has a "top-level unguarded occurrence" of ω. Gorla defines the latter concept only for languages that are equipped with a notion of *structural congruence* $\equiv$ as well as a parallel composition $|$. In that case P has a top-level unguarded occurrence of ω iff $P \equiv Q|\omega$, for some Q [39]. Specialised to the π-calculus, a *(top-level) unguarded* occurrence is one that not lays strictly within a subterm $\alpha.Q$, where α is τ, $\bar{x}y$ or $x(z)$. As far as I know, for all languages where Gorla's definition of $P{\downarrow_\omega}$ as well as mine apply, they yield the same result.

On CCS strong barbed congruence coincides with strong bisimilarity [52]. On the π-calculus strong barbed congruence coincides with strong early congruence, the congruence closure of strong early bisimilarity [76]. This testifies to the success of the barbed bisimilarity approach to defining canonical semantic equivalences for different process calculi in a uniform way. For many other calculi, strong barbed congruence is the primary *proposal* for a canonical semantic equivalence, and its explicit characterisation in a manner that does not involve quantification over all context is a task that merely follows.

This makes strong barbed bisimilarity an attractive candidate for the equivalence relation $\sim$ up to which translations between process calculi are proven valid. Theorem 4 says that if a translation between two process calculi is valid up to strong barbed bisimilarity, then it is also valid up an equivalence that on the source language coincides with strong barbed congruence, and on the image of the source language within the target language is also the congruence closure of strong barbed bisimilarity.

For "nonprompt encodings" [56], that "allow administrative (or book-keeping) steps to precede a committing step", strong (barbed) bisimilarity is too fine an equivalence to relate source processes and their translations. Here an appealing alternative is *weak barbed bisimilarity*, cf. Definition 13. On CCS weak barbed congruence coincides with weak bisimilarity as defined in [50]. On the π-calculus weak barbed congruence coincides with weak early congruence as used in [76], at least for image finite processes, or in general when allowing infinite sums in the π-calculus [75]. Section 6 mentioned Boudol's encoding of π into $a\pi$ as an example of a transition valid up to weak barbed bisimilarity.

In many situations one would reject translations between process calculi, that introduce (or eliminate) divergences. This can be captured in my framework by requiring validity op to *weak (barbed) bisimilarity with explicit divergence* [7]. It adds to Definition 13 the clause

- $P \stackrel{\bullet}{\approx} Q$ and $P{\Uparrow}$ implies $Q{\Uparrow}$.

Here $P{\Uparrow}$ denotes that there are P_i for $i \geq 0$ such that $P = P_0$ and $P_i \rightarrow P_{i+1}$ (or $P_i \xrightarrow{\tau} P_{i+1}$) for all $i \geq 0$. Using Remark 1, Theorem 2, Observation 3 and Corollary 3 in [34], the proof of Theorem 2 in [47] shows that Boudol's encoding of π into $a\pi$ is even valid up to weak barbed bisimilarity with explicit divergence.

In most industrial applications of process calculi the rôle once played by weak bisimilarity has largely been taken over by branching bisimilarity [37]—cf. [24, 18]—of which both a default version and two versions with explicit divergence are in use [36, 22]. The following proposal captures these through the barbed bisimilarity methodology:

Definition 26 *Divergence-preserving branching barbed bisimilarity* is the largest symmetric relation $\stackrel{\bullet}{\leftrightarrow}{}_b^\Delta \subseteq \mathrm{T}_\mathscr{L} \times \mathrm{T}_\mathscr{L}$ such that

- $P \stackrel{\bullet}{\leftrightarrow}{}_b^\Delta Q$ and $P{\downarrow_\omega}$ with $\omega \in \Omega$ implies $Q \Longrightarrow Q^\dagger$ for some $Q^\dagger$ with $P \stackrel{\bullet}{\leftrightarrow}{}_b^\Delta Q^\dagger$ and $Q^\dagger{\downarrow_\omega}$,
- $P \stackrel{\bullet}{\leftrightarrow}{}_b^\Delta Q$ and $P \rightarrow P'$ implies $Q \Longrightarrow Q^\dagger \ (\rightarrow)\ Q'$ for some $Q^\dagger, Q'$ with $P \stackrel{\bullet}{\leftrightarrow}{}_b^\Delta Q^\dagger$ and $P' \stackrel{\bullet}{\leftrightarrow}{}_b^\Delta Q'$, and
- $P \stackrel{\bullet}{\leftrightarrow}{}_b^\Delta Q$ and $P \rightarrow P_1 \rightarrow P_2 \rightarrow \ldots$ implies $Q \rightarrow Q'$ for some Q' with $P_k \stackrel{\bullet}{\leftrightarrow}{}_b^\Delta Q'$ for some $k \geq 0$.

Here $P\ (\rightarrow)\ Q$ abbreviates $(P = Q) \vee (P \rightarrow Q)$. See [36] for a number of equivalent versions of the last clause, i.e. with $Q \Longrightarrow\rightarrow Q'$. For *weakly divergence-preserving branching barbed bisimilarity* the last clause is weakened to "$P \overset{\bullet}{\leftrightarrow}{}_b^{\Delta} Q$ and $P{\Uparrow}$ implies $Q{\Uparrow}$", just as for weak barbed bisimilarity with explicit divergence above. Skipping the last clause altogether yields *branching barbed bisimilarity*. On CCS, the congruence closures of these equivalences yield *((weakly) divergence-preserving) branching bisimilarity*, as defined in the literature [22]; the proofs are not essentially different from the ones for strong and weak barbed bisimilarity. Using the remark below Observation 3 in [34], the proof of Theorem 2 in [47] can be adapted to show that Boudol's encoding of π into $a\pi$ is even valid up to divergence-preserving branching barbed bisimilarity.

18 Related work: full abstraction

The concept of *full abstraction* stems from Milner [48]. It indicates a particularly nice connection between a denotational and an operational semantics of a language $\mathscr{L}$. Here a denotational semantics is a function $[\![\ ]\!]_{\mathscr{L}}$ as introduced in Section 2, whereas an operational semantics is given by an *evaluation function* $\mathscr{E} : \mathrm{T}_{\mathscr{L}} \rightarrow \mathbb{O}$ from the closed terms, there called *programs*, to a set $\mathbb{O}$ of *observations*. Evaluation determines an equivalence relation on programs: $P \sim_{\mathscr{E}} Q$ iff $\mathscr{E}(P) = \mathscr{E}(P)$. Let $\sim_{\mathscr{E}}^{1c}$ be the congruence closure of $\sim_{\mathscr{E}}$ for the language $\mathscr{L}$, as defined in Section 7.

Definition 27 ([48]) The semantic function $[\![\ ]\!]_{\mathscr{L}}$ for $\mathscr{L}$ is *fully abstract* w.r.t. $\mathscr{E}$ iff for all $P,Q \in \mathrm{T}_{\mathscr{L}}$

$$[\![P]\!]_{\mathscr{L}} = [\![Q]\!]_{\mathscr{L}} \quad \Leftrightarrow \quad P \sim_{\mathscr{E}}^{1c} Q\,.$$

A semantic function $[\![\ ]\!]_{\mathscr{L}}$ always induces an equivalence relation on $\mathrm{T}_{\mathscr{L}}$ by $P \approx_{\mathscr{L}} Q$ iff $[\![P]\!]_{\mathscr{L}} = [\![Q]\!]_{\mathscr{L}}$. When this semantics is defined inductively as indicated in Section 5, $\approx_{\mathscr{L}}$ must be a congruence. Now $[\![\ ]\!]_{\mathscr{L}}$ is fully abstract w.r.t. $\mathscr{E}$ iff $\approx_{\mathscr{L}} = \sim_{\mathscr{E}}^{1c}$.

Note that any equivalence relation $\sim_{\mathscr{E}}$ can be extracted from an evaluation function $\mathscr{E} : \mathrm{T}_{\mathscr{L}} \rightarrow \mathbb{O}$, namely by taking $\mathbb{O}$ to be the set of $\sim_{\mathscr{E}}$-equivalence classes of closed terms, with $\mathscr{E}$ mapping each $p \in \mathrm{T}_{\mathscr{L}}$ to its own equivalence class. Likewise, each congruence relation $\approx$ on $\mathrm{T}_{\mathscr{L}}$ can be obtained from a semantics $[\![\ ]\!]_{\mathscr{L}}$: take $\mathbf{V}$ to be the set of $\approx$-equivalence classes of closed terms, and for $E \in \mathbb{T}_{\mathscr{L}}$ and $\rho : \mathscr{L} \rightarrow \mathbf{V}$ define $[\![E]\!]_{\mathscr{L}}(\rho)$ by to be the $\approx$-equivalence class of $E[\sigma]$, where $\sigma : \mathscr{X} \rightarrow \mathrm{T}_{\mathscr{L}}$ is a closed substitution that maps each variable X to a member of the $\approx$-equivalence class of closed terms $\rho(X)$. Since $\approx$ is a congruence, this definition is independent of the choice of σ.

Consequently, full abstraction can equally well be stated as a relation between two equivalence relations $\approx$ and $\sim$ on $\mathrm{T}_{\mathscr{L}}$:

$\approx$ is *fully abstract* w.r.t. $\mathscr{L}$ and $\sim$ iff $\approx = \sim_{\mathscr{L}}^{1c}$.

It is in this spirit that full abstraction has been employed in [29] and subsequent papers.

Riecke [73] and Shapiro [77] extend the notion of full abstraction to translations between languages. Riecke [73] compares languages $\mathscr{L}_{\mathrm{S}}$ and $\mathscr{L}_{\mathrm{T}}$ with a shared evaluation function $\mathscr{E} : \mathrm{T}_{\mathscr{L}_{\mathrm{S}}} \cup \mathrm{T}_{\mathscr{L}_{\mathrm{T}}} \rightarrow 2^{\mathbb{O}}$, associating with each closed expression a set of observations. Write $P \sqsubseteq^{\mathscr{E}} Q$, for $P,Q \in \mathrm{T}_{\mathscr{L}_{\mathrm{S}}} \cup \mathrm{T}_{\mathscr{L}_{\mathrm{T}}}$, if $\mathscr{E}(P) \subseteq \mathscr{E}(Q)$, and let $\sqsubseteq_{\mathscr{L}_i}^{\mathscr{E}}$ be the congruence closure of $\sqsubseteq^{\mathscr{E}}$ w.r.t. $\mathscr{L}_i$, for $i = \mathrm{S},\mathrm{T}$ (source and target). He calls a translation $\mathscr{T} : \mathbb{T}_{\mathscr{L}_{\mathrm{S}}} \rightarrow \mathbb{T}_{\mathscr{L}_{\mathrm{T}}}$ fully abstract iff, for all $P,Q \in \mathrm{T}_{\mathscr{L}_{\mathrm{S}}}$,

$$P \sqsubseteq_{\mathscr{L}_{\mathrm{S}}}^{\mathscr{E}} Q \quad \Leftrightarrow \quad \mathscr{T}(P) \sqsubseteq_{\mathscr{L}_{\mathrm{T}}}^{\mathscr{E}} \mathscr{T}(Q)\,.$$

The same notion occurs earlier in Mitchell [53], although not under the name "full abstraction", and using equivalence relations instead of preorders—taking $P \sim^{\mathscr{E}} Q$ iff $\mathscr{E}(P) = \mathscr{E}(Q)$. He compares the expressive power of programming languages in terms of *abstraction preserving reductions* between them. These are translations that are compositional as well as fully abstract in the above sense. Later work abstracts from the evaluation function $\mathscr{E}$, and casts full abstraction directly in terms of equivalences $\sim^{\mathrm{S}}$ and $\sim^{\mathrm{T}}$ on the source and target languages [74].

Felleisen [21] compares the expressive power of programming languages through a notion of *eliminability* of language constructs. With some effort, this approach can be seen to have significant similarities with the approach of Mitchell [53], although it allows certain degenerate reductions [53].

Whereas most work on expressiveness deals with closed-term languages, allowing a focus on syntax over semantics, Shapiro [77] works entirely on the semantic side, leaving the syntax largely implicit. His languages are triples $(\mathbf{V}, \mathscr{L}, \sim)$, consisting of a semantic domain $\mathbf{V}$, a collection $\mathscr{L}$ of partial operators on $\mathbf{V}$, and a semantic equivalence $\sim$ on $\mathbf{V}$. A *language embedding* of one such language $(\mathbf{V}_{\mathrm{S}}, \mathscr{L}_{\mathrm{S}}, \sim^{\mathrm{S}})$ into another $(\mathbf{V}_{\mathrm{T}}, \mathscr{L}_{\mathrm{T}}, \sim^{\mathrm{T}})$ is defined to be a homomorphism of the partial algebra $(\mathbf{V}_{\mathrm{S}}, \mathscr{L}_{\mathrm{S}})$ into the partial algebra $(\mathbf{V}_{\mathrm{T}}, \mathscr{L}_{\mathrm{T}})$. Recasting this definition in terms of my framework, this is a function $\mathbf{R}$: $\mathbf{V}_{\mathrm{S}} \rightarrow \mathbf{V}_{\mathrm{T}}$ such that there exists a compositional translation $\mathscr{T} : \mathscr{L}_1 \rightarrow \mathscr{L}_2$ correct w.r.t. $\mathbf{R}$ (as in Definition 3). For $i = \mathrm{S}, \mathrm{T}$ let $\sim^i_{\mathscr{L}_i}$ be the congruence closure of $\sim^i$ w.r.t. $\mathscr{L}_i$—Shapiro calls this the *fully-abstract* congruence of $\mathscr{L}_i$. Then a language embedding is deemed fully abstract iff, for all $v, w \in \mathbf{V}_{\mathrm{S}}$,

$$v \sim^{\mathrm{S}}_{\mathscr{L}_{\mathrm{S}}} w \quad \Leftrightarrow \quad \mathscr{T}(v) \sim^{\mathrm{T}}_{\mathscr{L}_{\mathrm{T}}} \mathscr{T}(w)\,.$$

In [78] these fully abstract language embeddings are used to classify a number of concurrent programming languages on expressive power.

In Nestmann & Pierce [56] the notion of full abstraction was generalised by dropping the requirement that the source and target equivalences being compared need to be congruence closures.

Definition 28 A translation $\mathscr{T} : \mathbb{T}_{\mathscr{L}_{\mathrm{S}}} \rightarrow \mathbb{T}_{\mathscr{L}_{\mathrm{T}}}$ is *fully abstract* w.r.t. the equivalences $\sim_{\mathrm{S}} \subseteq \mathrm{T}^2_{\mathscr{L}_{\mathrm{S}}}$ and $\sim_{\mathrm{T}} \subseteq \mathrm{T}^2_{\mathscr{L}_{\mathrm{T}}}$ if, for all $P, Q \in \mathrm{T}_{\mathscr{L}_{\mathrm{S}}}$,

$$P \sim_{\mathrm{S}} Q \quad \Leftrightarrow \quad \mathscr{T}(P) \sim_{\mathrm{T}} \mathscr{T}(Q)\,.$$

In this form, full abstraction has found widespread applications [54, 5]. Fu [23], for instance, bases a theory of expressiveness on full abstraction, with divergence-preserving branching barbed bisimilarity in the rôle of $\sim_{\mathrm{S}}$ and $\sim_{\mathrm{T}}$.

As stressed in [40, 62], the notion of full abstraction is meaningful only in relation to a well-chosen pair of source and target equivalences, and only in combination with a criterion like compositionality. In particular, for each encoding $\mathscr{T}$ and each target equivalence $\sim_{\mathrm{T}}$ there exits a source term equivalence $\sim_{\mathrm{S}}$, namely $\{(P,Q) \mid \mathscr{T}(P) \sim_{\mathrm{T}} \mathscr{T}(Q)\}$, such that $\mathscr{T}$ is fully abstract w.r.t. $\sim_{\mathrm{S}}$ and $\sim_{\mathrm{T}}$. For each injective encoding $\mathscr{T}$ and each source term relation $\sim_{\mathrm{S}}$, there exits $\sim_{\mathrm{T}} \subseteq \mathrm{T}^2_{\mathscr{L}_{\mathrm{T}}}$, namely $\{(\mathscr{T}(P), \mathscr{T}(Q)) \mid P \sim_{\mathrm{S}} Q\}$, such that $\mathscr{T}$ is fully abstract w.r.t. $\sim_{\mathrm{S}}$ and $\sim_{\mathrm{T}}$. Finally, for each pair $\sim_{\mathrm{S}}$ and $\sim_{\mathrm{T}}$ such that the cardinality of $\mathrm{T}_{\mathscr{L}_{\mathrm{T}}}/_{\sim_{\mathrm{T}}}$ greater than or equal to the cardinality of $\mathrm{T}_{\mathscr{L}_{\mathrm{S}}}/_{\sim_{\mathrm{S}}}$ there exists a translation from $\mathscr{L}_{\mathrm{S}}$ to $\mathscr{L}_{\mathrm{T}}$ that is fully abstract w.r.t. $\sim_{\mathrm{S}}$ and $\sim_{\mathrm{T}}$.

Naturally, any translation that is valid up to an equivalence $\sim$ as in Definition 5 of the current paper is also fully abstract, namely w.r.t. the same equivalence $\sim$ in the rôle of both $\sim_{\mathrm{S}}$ and $\sim_{\mathrm{T}}$. Furthermore, validity entails compositionality through Corollary 2. Both full abstraction and validity imply an injective translation from the $\sim_{\mathrm{S}}$-equivalence classes of closed source terms to the $\sim_{\mathrm{T}}$-equivalence classes of closed target terms. However, validity demands that this link between source and target terms is again governed by $\sim$, whereas full abstraction implies no counterpart to this crucial requirement.

A typical example of a valid transition is the encoding of the synchronous into the asynchronous π-calculus described in Section 6. This encoding is valid up to weak barbed bisimilarity $\dot{\approx}$. Consequently is it also fully abstract w.r.t. $\stackrel{\bullet}{\approx}$ and $\stackrel{\bullet}{\approx}$ according to Definition 28. However, it is not fully abstract w.r.t. $\stackrel{\bullet}{\approx}$ and $\stackrel{\bullet}{\approx}$ in the more original sense of Shapiro [77] and others, due to the fact that $\stackrel{\bullet}{\approx}$ is not a congruence for either the source or the target language. By Theorem 4 the same translation is also fully abstract w.r.t the congruence closure of $\stackrel{\bullet}{\approx}$ on the source language, and a somewhat artificial equivalence on the target language that is the congruence closure of $\stackrel{\bullet}{\approx}$ under translated source contexts. Although closer, this is still not a full-abstraction result according to Shapiro, as the latter equivalence fails to be a congruence for all of the target language. Finally, by Theorem 6, the same encoding is furthermore fully abstract w.r.t. the congruence closure of $\stackrel{\bullet}{\approx}$ on the target language, and an artificial congruence on the source language that is strictly finer than the congruence closure of $\stackrel{\bullet}{\approx}$ on the source language. This is a full abstraction result in the framework of [77]. However, in line with the observations of [40, 62], the latter two full abstraction results should not be regarded as lending additional credibility to this particular encoding of the synchronous into the asynchronous π-calculus. Since Theorems 4 and 6 hold in great generality, these full abstraction results are merely consequences of the validity of the encoding up to $\stackrel{\bullet}{\approx}$.

It is well known that Boudol's encoding—described in Section 6—of the synchronous into the asynchronous π-calculus fails to be fully abstract w.r.t. $\cong^c$ and $\cong^c_a$. Here $\cong^c$ is the congruence closure of $\stackrel{\bullet}{\approx}$ on the source language, and $\cong^c_a$ the congruence closure of $\stackrel{\bullet}{\approx}$ on the target language. A counterexample was given at the end of Section 8. The same analysis applies to a similar encoding proposed by Honda & Tokoro [44]. In [20] this problem is addressed by proposing a strict subcalculus $SA\pi$ of the target language that contains the image of the source language under of a version Honda & Tokoro's encoding, such that this encoding is fully abstract w.r.t. $\cong^c$ and the congruence closure of $\stackrel{\bullet}{\approx}$ w.r.t. $SA\pi$. In [72] a similar solution to the same problem was found earlier, but for a variant of Boudol's encoding from the *polyadic* π-calculus to the (monadic) asynchronous π-calculus. They defined a class of *well-typed* expressions in the asynchronous π-calculus, such that the well-typed expressions constitute a subcalculus of the target language that contains the image of the source language under the encoding. Again, the encoding is fully abstract w.r.t. $\cong^c$ and the congruence closure of $\stackrel{\bullet}{\approx}$ w.r.t. that sublanguage. By Theorem 4 such results can always be achieved, namely by taking as target language exactly the image of the source language under the encoding. What the results of [72, 20] add is that the sublanguage may be strictly larger than the image of the source language, and that its definition is not phrased in terms of the encoding.

19 Related work: validity of encodings according to Gorla

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) [74, 45, 8, 60, 56, 57, 54, 17, 16, 12, 15, 4, 5, 59, 58, 69, 14, 81, 13, 42, 70, 80, 1, 43, 66, 2]; see [38, 39] 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 [55], [61] and [63]. Gorla [39] 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. Since then, several authors have used Gorla's framework as a basis for establishing new valid encodings and separation results [38, 46, 68, 65, 67, 25, 26, 27, 28].

Like Boudol [9] and the present paper, Gorla requires a compositionality condition for encodings. However, his criterion differs on two counts from mine. It is stronger in that the expression E_f encoding an operator f may use each variable corresponding to an argument of f only once. A translation defined (in part) by $\mathscr{T}(f(P)) = g(\mathscr{T}(P)\|\mathscr{T}(P))$ for instance can be compositional according to Definition 21, but not according to Gorla's definition. It is weaker than mine 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 [32]. It is an interesting topic for future research to see if there are any valid encodability results à la [39] that suffer from my proposed strengthening of compositionality.

The second criterion of [39] 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 [32] I argue that this criterion as formalised in [39] is too restrictive.

The remaining three requirements of Gorla (the 'semantic' requirements) are very close to an instantiation of mine with a particular preorder $\sim\!\!\!\!\bullet$. If one takes $\sim\!\!\!\!\bullet$ to be weak barbed bisimilarity with explicit divergence (i.e. relating divergent states with divergent states only—see Section 17), using external barbs as defined in Section 17, then any valid translation in my sense satisfies Gorla's semantic criteria, provided that the equivalence $\asymp$ 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 [32] and [39] are further discussed in [64].

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] J. Åman Pohjola & J. Parrow (2014): *Priorities Without Priorities: Representing Preemption in Psi-Calculi*. In J. Borgström & S. Crafa, editors: Proceedings Combined 21st International Workshop on *Expressiveness in Concurrency* and 11th Workshop on *Structural Operational Semantics*, Rome, Italy, 1st September 2014, *Electronic Proceedings in Theoretical Computer Science* 160, Open Publishing Association, pp. 2–15, doi:10.4204/EPTCS.160.2.

[2] J. Åman Pohjola & J. Parrow (2016): *The Expressive Power of Monotonic Parallel Composition*. In P. Thiemann, editor: *Programming Languages and Systems:* Proceedings 25th *European Symposium on Programming*, ESOP'16, held as part of the European Joint Conferences on *Theory and Practice of Software*, ETAPS'16, Eindhoven, The Netherlands, April 2016, LNCS 9632, Springer, pp. 780–803, doi:10.1007/978-3-662-49498-1_30.

[3] D. Austry & G. Boudol (1984): *Algèbre de processus et synchronisations*. *Theoretical Computer Science* 30(1), pp. 91–131, doi:10.1016/0304-3975(84)90067-7.

[4] M. Baldamus, J. Parrow & B. Victor (2004): *Spi Calculus Translated to π-Calculus Preserving May-Tests*. In: Proceedings 19th IEEE Symposium on *Logic in Computer Science*, LICS'04, July 2004, Turku, Finland, IEEE Computer Society Press, pp. 22–31, doi:10.1109/LICS.2004.1319597.

[5] M. Baldamus, J. Parrow & B. Victor (2005): *A Fully Abstract Encoding of the pi-Calculus with Data Terms*. In L. Caires, G.F. Italiano, L. Monteiro, C. Palamidessi & M. Yung, editors: Proceedings 32nd *International Colloquium on Automata, Languages and Programming*, ICALP'05, Lisbon, Portugal, July 2005, LNCS 3580, Springer, pp. 1202–1213, doi:10.1007/11523468_97.

[6] J.A. Bergstra & J.W. Klop (1986): *Algebra of communicating processes*. In J.W. de Bakker, M. Hazewinkel & J.K. Lenstra, editors: *Mathematics and Computer Science*, CWI Monograph 1, North-Holland, Amsterdam, pp. 89–138.

[7] J.A. Bergstra, J.W. Klop & E.-R. Olderog (1987): *Failures without chaos: a new process semantics for fair abstraction*. In M. Wirsing, editor: *Formal Description of Programming Concepts – III, Proceedings of the* 3^{th} *IFIP WG 2.2 working conference*, Ebberup 1986, North-Holland, Amsterdam, pp. 77–103.

[8] M. Boreale (1998): *On the Expressiveness of Internal Mobility in Name-Passing Calculi*. Theoretical Computer Science 195(2), pp. 205–226, doi:10.1016/S0304-3975(97)00220-X.

[9] G. Boudol (1985): *Notes on algebraic calculi of processes*. In K. Apt, editor: Logics and Models of Concurrent Systems, Springer, pp. 261–303, doi:10.1007/978-3-642-82453-1_9. NATO ASI Series F13.

[10] G. Boudol (1992): *Asynchrony and the π-calculus (Note)*. Technical Report 1702, INRIA.

[11] S.D. Brookes, C.A.R. Hoare & A.W. Roscoe (1984): *A theory of communicating sequential processes*. Journal of the ACM 31(3), pp. 560–599, doi:10.1145/828.833.

[12] 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.

[13] D. Cacciagrano, F. Corradini, J. Aranda & F.D. Valencia (2008): *Linearity, Persistence and Testing Semantics in the Asynchronous Pi-Calculus*. Electronic Notes in Theoretical Computer Science 194(2), pp. 59–84, doi:10.1016/j.entcs.2007.11.006.

[14] D. Cacciagrano, F. Corradini & C. Palamidessi (2007): *Separation of synchronous and asynchronous communication via testing*. Theoretical Computer Science 386(3), pp. 218–235, doi:10.1016/j.tcs.2007.07.009.

[15] M. Carbone & S. Maffeis (2003): *On the Expressive Power of Polyadic Synchronisation in pi-calculus*. Nordic Journal of Computing 10(2), pp. 70–98.

[16] L. Cardelli, G. Ghelli & A.D. Gordon (2002): *Types for the Ambient Calculus*. Information and Computation 177(2), pp. 160–194, doi:10.1006/inco.2001.3121.

[17] L. Cardelli & A.D. Gordon (2000): *Mobile ambients*. Theoretical Computer Science 240(1), pp. 177–213, doi:10.1016/S0304-3975(99)00231-5.

[18] S. Cranen, J. F. Groote, J. J. A. Keiren, F. P. M. Stappers, E. P. de Vink, W. Wesselink & T. A. C. Willemse (2013): *An Overview of the mCRL2 Toolset and Its Recent Advances*. In Nir Piterman & Scott A. Smolka, editors: Proceedings Tools and Algorithms for the Construction and Analysis of Systems, TACAS '13, LNCS 7795, Springer, pp. 199–213, doi:10.1007/978-3-642-36742-7_15.

[19] R. De Nicola & M. Hennessy (1984): *Testing equivalences for processes*. Theoretical Computer Science 34, pp. 83–133, doi:10.1016/0304-3975(84)90113-0.

[20] W. Du, Z. Yang & H. Zhu (2018): *A Fully Abstract Encoding for Sub Asynchronous Pi Calculus*. In J. Pang & C. Zhang, editors: 12th International Symposium on Theoretical Aspects of Software Engineering, IEEE. To appear.

[21] M. Felleisen (1991): *On the Expressive Power of Programming Languages*. Science of Computer Programming 17(1-3), pp. 35–75, doi:10.1016/0167-6423(91)90036-W.

[22] W.J. Fokkink, R.J. van Glabbeek & B. Luttik (2017): *Divide and Congruence III: Stability & Divergence*. In R. Meyer & U. Nestmann, editors: Proceedings 28th International Conference on Concurrency Theory, CONCUR 2017, Leibniz International Proceedings in Informatics (LIPIcs) 85, Schloss Dagstuhl–Leibniz-Zentrum für Informatik, Dagstuhl, Germany, pp. 15:1–15:16, doi:10.4230/LIPIcs.CONCUR.2017.15.

[23] Y. Fu (2016): *Theory of interaction*. Theoretical Computer Science 611, pp. 1–49, doi:10.1016/j.tcs.2015.07.043.

[24] H. Garavel, F. Lang, R. Mateescu & W. Serwe (2011): *CADP 2010: A Toolbox for the Construction and Analysis of Distributed Processes*. In P.A. Abdulla & K.R.M. Leino, editors: Proceedings Tools and Algorithms for the Construction and Analysis of Systems, TACAS '11, LNCS 6605, Springer, pp. 372–387, doi:10.1007/978-3-642-19835-9_33.

[25] T. Given-Wilson (2014): *Expressiveness via Intensionality and Concurrency*. In G. Ciobanu & D. Méry, editors: Proceedings 11th International Colloquium on Theoretical Aspects of Computing, ICTAC'14, Bucharest, Romania, 2014, LNCS 8687, Springer, pp. 206–223, doi:10.1007/978-3-319-10882-7_13.

[26] T. Given-Wilson (2014): *On the Expressiveness of Intensional Communication*. In J. Borgström & S. Crafa, editors: Proceedings Combined 21st International Workshop on *Expressiveness in Concurrency* and 11th Workshop on *Structural Operational Semantics*, Rome, Italy, 1st September 2014, *Electronic Proceedings in Theoretical Computer Science* 160, Open Publishing Association, pp. 30–46, doi:10.4204/EPTCS.160.4.

[27] T. Given-Wilson & A. Legay (2015): *On the Expressiveness of Joining*. In S. Knight, I. Lanese, A. Lluch Lafuente & H. Torres Vieira, editors: Proceedings 8th *Interaction and Concurrency Experience*, Grenoble, France, 4-5th June 2015, *Electronic Proceedings in Theoretical Computer Science* 189, Open Publishing Association, pp. 99–113, doi:10.4204/EPTCS.189.9.

[28] T. Given-Wilson & A. Legay (2016): *On the Expressiveness of Symmetric Communication*. In A. Sampaio & F. Wang, editors: Proceedings 13th International Colloquium on *Theoretical Aspects of Computing*, ICTAC'16, Taipei, Taiwan, October 2016, LNCS 9965, Springer, pp. 139–157, doi:10.1007/978-3-319-46750-4_9.

[29] R.J. van Glabbeek (1993): *Full Abstraction in Structural Operational Semantics (extended abstract)*. In M. Nivat, C. Rattray, T. Rus & G. Scollo, editors: Proceedings of the 3^{rd} International Conference on *Algebraic Methodology and Software Technology*, AMAST'93, Twente, The Netherlands, June l993, Workshops in Computing, Springer, pp. 75–82. Available at http://theory.stanford.edu/~rvg/abstracts.html#28.

[30] R.J. van Glabbeek (1994): *On the expressiveness of ACP (extended abstract)*. In A. Ponse, C. Verhoef & S.F.M. van Vlijmen, editors: Proceedings First Workshop on the *Algebra of Communicating Processes*, ACP'94, Utrecht, The Netherlands, May 1994, Workshops in Computing, Springer, pp. 188–217, doi:10.1007/978-1-4471-2120-6_8.

[31] R.J. van Glabbeek (2005): *A Characterisation of Weak Bisimulation Congruence*. In A. Middeldorp, V. van Oostrom, F. van Raamsdonk & R. de Vrijer, editors: *Processes, Terms and Cycles: Steps on the Road to Infinity: Essays Dedicated to Jan Willem Klop on the Occasion of His 60th Birthday*, LNCS 3838, Springer, pp. 26–39, doi:10.1007/11601548_4.

[32] R.J. van Glabbeek (2012): *Musings on Encodings and Expressiveness*. In B. Luttik & M.A. Reniers, editors: Proceedings Combined 19th International Workshop on *Expressiveness in Concurrency* and 9th Workshop on *Structured Operational Semantics*, Newcastle upon Tyne, UK, September 3, 2012, *Electronic Proceedings in Theoretical Computer Science* 89, Open Publishing Association, pp. 81–98, doi:10.4204/EPTCS.89.7.

[33] R.J. van Glabbeek (2017): *Lean and Full Congruence Formats for Recursion*. In: Proceedings 32^{nd} Annual ACM/IEEE Symposium on *Logic in Computer Science*, LICS'17, Reykjavik, Iceland, June 2017, IEEE Computer Society Press, doi:10.1109/LICS.2017.8005142.

[34] R.J. van Glabbeek (2018): *On the Validity of Encodings of the Synchronous in the Asynchronous π-calculus*. *Information Processing Letters* 137, p. 1725, doi:10.1016/j.ipl.2018.04.015.

[35] R.J. van Glabbeek (2018): *A Theory of Encodings and Expressiveness (extended abstract)*. In C. Baier & U. Dal Lago, editors: Proceeding 21st International Conference on *Foundations of Software Science and Computational Structures*, FoSSaCS 2018; held as part of the *European Joint Conferences on Theory and Practice of Software*, ETAPS 2018, hessaloniki, Greece, April 2018, LNCS 10803, Springer, pp. 183–202, doi:10.1007/978-3-319-89366-2_10.

[36] R.J. van Glabbeek, B. Luttik & N. Trčka (2009): *Branching Bisimilarity with Explicit Divergence*. *Fundamenta Informaticae* 93(4), pp. 371–392, doi:10.3233/FI-2009-109.

[37] R.J. van Glabbeek & W.P. Weijland (1996): *Branching Time and Abstraction in Bisimulation Semantics*. *Journal of the ACM* 43(3), pp. 555–600, doi:10.1145/233551.233556.

[38] 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.

[39] D. Gorla (2010): *Towards a unified approach to encodability and separation results for process calculi*. *Information and Computation* 208(9), pp. 1031–1053, doi:10.1016/j.ic.2010.05.002.

[40] D. Gorla & U. Nestmann (2016): *Full abstraction for expressiveness: history, myths and facts*. Mathematical Structures in Computer Science 26(4), pp. 639–654, doi:10.1017/S0960129514000279.

[41] G. Grätzer (2010): *Lattice Theory: Foundation*. Springer, doi:10.1007/978-3-0348-0018-1.

[42] B. Haagensen, S. Maffeis & I. Phillips (2008): *Matching Systems for Concurrent Calculi*. Electronic Notes in Theoretical Computer Science 194(2), pp. 85–99, doi:10.1016/j.entcs.2007.11.004.

[43] M. Hatzel, C. Wagner, K. Peters & U. Nestmann (2015): *Encoding CSP into CCS*. In S. Crafa & D.E. Gebler, editors: Proceedings of the Combined 22th International Workshop on Expressiveness in Concurrency and 12th Workshop on Structural Operational Semantics, Madrid, Spain, August 2015, Electronic Proceedings in Theoretical Computer Science 190, Open Publishing Association, pp. 61–75, doi:10.4204/EPTCS.190.5.

[44] K. Honda & M. Tokoro (1991): *An Object Calculus for Asynchronous Communication*. In Pierre America, editor: Proc. European Conference on Object-Oriented Programming, ECOOP'91, LNCS 512, Springer, pp. 133–147, doi:10.1007/BFb0057019.

[45] C.T. Jensen (1994): *Interpreting Broadcast Communication in CCS with Priority Choice*. In: Proceedings of the 6th Nordic Workshop on Programming Theory, pp. 49–70.

[46] I. Lanese, J.A. Pérez, D. Sangiorgi & A. Schmitt (2010): *On the Expressiveness of Polyadic and Synchronous Communication in Higher-Order Process Calculi*. In S. Abramsky, C. Gavoille, C. Kirchner, F. Meyer auf der Heide & P.G. Spirakis, editors: Proceedings 37th International Colloquium on Automata, Languages and Programming, ICALP'10, Bordeaux, France, July 2010, Part II, LNCS 6199, Springer, pp. 442–453, doi:10.1007/978-3-642-14162-1_37.

[47] C. Lippert, S. Mennicke, R.J. van Glabbeek & U. Goltz (2018): *A Case Study on Evaluating Encodings Between Process Calculi*. Available at http://theory.stanford.edu/~rvg/abstracts.html#129.

[48] R. Milner (1975): *Processes: A Mathematical Model for Computing Agents*. In H.E. Rose & J.C. Shepherdson, editors: Logic Colloquium '73, Studies in Logic and the Foundations of Mathematics 50, Elsevier, pp. 157–173, doi:10.1016/S0049-237X(08)71948-7.

[49] R. Milner (1983): *Calculi for synchrony and asynchrony*. Theoretical Computer Science 25, pp. 267–310, doi:10.1016/0304-3975(83)90114-7.

[50] R. Milner (1990): *Operational and algebraic semantics of concurrent processes*. In J. van Leeuwen, editor: Handbook of Theoretical Computer Science, chapter 19, Elsevier Science Publishers B.V. (North-Holland), pp. 1201–1242. Alternatively see *Communication and Concurrency*, Prentice-Hall, Englewood Cliffs, 1989, of which an earlier version appeared as *A Calculus of Communicating Systems*, LNCS 92, Springer, 1980.

[51] R. Milner (1992): *Functions as Processes*. Mathematical Structures in Computer Science 2(2), pp. 119–141, doi:10.1017/S0960129500001407.

[52] R. Milner & D. Sangiorgi (1992): *Barbed Bisimulation*. In W. Kuich, editor: Proceedings 19th International Colloquium on Automata, Languages and Programming, ICALP'92, Vienna, Austria, July 1992, LNCS 623, Springer, pp. 685–695, doi:10.1007/3-540-55719-9_114.

[53] J.C. Mitchell (1993): *On Abstraction and the Expressive Power of Programming Languages*. Science of Computer Programming 21(2), pp. 141–163, doi:10.1016/0167-6423(93)90004-9. A earlier version was circulated in 1986 under the name *Lisp is not universal*.

[54] U. Nestmann (2000): *What is a "Good" Encoding of Guarded Choice?* Information and Computation 156(1-2), pp. 287–319, doi:10.1006/inco.1999.2822.

[55] U. Nestmann (2006): *Welcome to the Jungle: A Subjective Guide to Mobile Process Calculi*. In C. Baier & H. Hermanns, editors: Proceedings 17th International Conference on Concurrency Theory, CONCUR 2006, Bonn, Germany, August 2006, LNCS 4137, Springer, pp. 52–63, doi:10.1007/11817949_4.

[56] U. Nestmann & B.C. Pierce (2000): *Decoding Choice Encodings*. Information and Computation 163(1), pp. 1–59, doi:10.1006/inco.2000.2868. An earlier version appeared in Proc. CONCUR'96.

[57] 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.

[58] C. Palamidessi, V.A. Saraswat, F.D. Valencia & B Victor (2006): *On the Expressiveness of Linearity vs Persistence in the Asychronous Pi-Calculus*. In: Proceedings 21th IEEE Symposium on *Logic in Computer Science*, LICS'06, August 2006, Seattle, WA, USA, IEEE Computer Society Press, pp. 59–68, doi:10.1109/LICS.2006.39.

[59] C. Palamidessi & F.D. Valencia (2005): *Recursion vs Replication in Process Calculi: Expressiveness*. *Bulletin of the EATCS* 87, pp. 105–125.

[60] J. Parrow (2000): *Trios in concert*. In G.D. Plotkin, C. Stirling & M. Tofte, editors: *Proof, Language, and Interaction, Essays in Honour of Robin Milner*, The MIT Press, pp. 623–638.

[61] J. Parrow (2008): *Expressiveness of Process Algebras*. *Electronic Notes in Theoretical Computer Science* 209, pp. 173–186, doi:10.1016/j.entcs.2008.04.011.

[62] J. Parrow (2016): *General conditions for full abstraction*. *Mathematical Structures in Computer Science* 26(4), pp. 655–657, doi:10.1017/S0960129514000280.

[63] K. Peters (2012): *Translational Expressiveness. Comparing Process Calculi using Encodings*. Ph.D. thesis, TU Berlin, doi:10.14279/depositonce-3416.

[64] K. Peters & R.J. van Glabbeek (2015): *Analysing and Comparing Encodability Criteria*. In S. Crafa & D.E. Gebler, editors: Proceedings of the Combined 22th International Workshop on *Expressiveness in Concurrency* and 12th Workshop on *Structural Operational Semantics*, Madrid, Spain, 31st August 2015, *Electronic Proceedings in Theoretical Computer Science* 190, Open Publishing Association, pp. 46–60, doi:10.4204/EPTCS.190.4.

[65] K. Peters & U. Nestmann (2012): *Is It a "Good" Encoding of Mixed Choice?* In L. Birkedal, editor: Proceeding 15th International Conference on *Foundations of Software Science and Computational Structures*, FoSSaCS'12; held as part of the *European Joint Conferences on Theory and Practice of Software*, ETAPS'12, Tallinn, Estonia, March/April 2012, LNCS 7213, Springer, pp. 210–224, doi:10.1007/978-3-642-28729-9_14.

[66] K. Peters & U. Nestmann (2016): *Breaking symmetries*. *Mathematical Structures in Computer Science* 26(6), pp. 1054–1106, doi:10.1017/S0960129514000346.

[67] K. Peters, U. Nestmann & U. Goltz (2013): *On Distributability in Process Calculi*. In M. Felleisen & Ph. Gardner, editors: *Programming Languages and Systems:* Proceedings 22nd *European Symposium on Programming*, ESOP'13, held as part of the European Joint Conferences on *Theory and Practice of Software*, ETAPS'13, Rome, Italy, March 2013, LNCS 7792, Springer, pp. 310–329, doi:10.1007/978-3-642-37036-6_18.

[68] K. Peters, J.-W. Schicke & U. Nestmann (2011): *Synchrony vs Causality in the Asynchronous Pi-Calculus*. In B. Luttik & F. Valencia, editors: Proceedings 18th International Workshop on *Expressiveness in Concurrency*, *EPTCS* 64, pp. 89–103, doi:10.4204/EPTCS.64.7.

[69] I. Phillips & M.G. Vigliotti (2006): *Leader election in rings of ambient processes*. *Theoretical Computer Science* 356(3), pp. 468–494, doi:10.1016/j.tcs.2006.02.004.

[70] I. Phillips & M.G. Vigliotti (2008): *Symmetric electoral systems for ambient calculi*. *Information and Computation* 206(1), pp. 34–72, doi:10.1016/j.ic.2007.08.005.

[71] G.D. Plotkin (2004): *A Structural Approach to Operational Semantics*. *The Journal of Logic and Algebraic Programming* 60–61, pp. 17–139, doi:10.1016/j.jlap.2004.05.001. Originally appeared in 1981.

[72] P. Quaglia & D. Walker (2000): *On Synchronous and Asynchronous Mobile Processes*. In J. Tiuryn, editor: Proceedings Third International Conference on *Foundations of Software Science and Computation Structures*, FoSSaCS'00, held as part of the Joint European Conferences on *Theory and Practice of Software*, ETAPS'00, Berlin, Germany, March/April 2000, LNCS 1784, Springer, pp. 283–296, doi:10.1007/3-540-46432-8_19.

[73] J.G. Riecke (1991): *Fully Abstract Translations between Functional Languages*. In D.S. Wise, editor: Conference Record of the Eighteenth Annual ACM Symposium on *Principles of Programming Languages*, POPL'91, Orlando, Florida, USA, January 1991, ACM Press, pp. 245–254, doi:10.1145/99583.99617.

[74] D. Sangiorgi (1993): *From π-calculus to Higher-Order π-calculus — and back*. In M.-C. Gaudel & J.-P. Jouannaud, editors: Proceedings *Theory and Practice of Software Development*, TAPSOFT'93, International Joint Conference CAAP/FASE, Orsay, France, April 1993, LNCS 668, Springer, pp. 151–166, doi:10.1007/3-540-56610-4_62.

[75] D. Sangiorgi & D. Walker (2001): *On Barbed Equivalences in π-Calculus*. In K.G. Larsen & M. Nielsen, editors: Proceeding 12th International Conference on *Concurrency Theory*, CONCUR'01, Aalborg, Denmark, August 2001, LNCS 2154, Springer, pp. 292–304, doi:10.1007/3-540-44685-0_20.

[76] D. Sangiorgi & D. Walker (2001): *The π-calculus: A Theory of Mobile Processes*. Cambridge University Press.

[77] E.Y. Shapiro (1991): *Separating Concurrent Languages with Categories of Language Embeddings (Extended Abstract)*. In C. Koutsougeras & J.S. Vitter, editors: Proceedings 23rd Annual ACM Symposium on *Theory of Computing*, STOC'91, May 1991, New Orleans, Louisiana, USA, ACM, pp. 198–208, doi:10.1145/103418.103423.

[78] E.Y. Shapiro (1992): *Embeddings Among Concurrent Programming Languages (Preliminary Version)*. In R. Cleaveland, editor: Proceedings Third International Conference on *Concurrency Theory*, CONCUR '92, *Stony Brook, NY, USA, August 1992*, LNCS 630, Springer, pp. 486–503, doi:10.1007/BFb0084811.

[79] R. de Simone (1985): *Higher-level synchronising devices in* MEIJE-*SCCS*. *Theoretical Computer Science* 37, pp. 245–267, doi:10.1016/0304-3975(85)90093-3.

[80] C. Versari, N. Busi & R. Gorrieri (2009): *An expressiveness study of priority in process calculi*. *Mathematical Structures in Computer Science* 19(6), pp. 1161–1189, doi:10.1017/S0960129509990168.

[81] M.G. Vigliotti, I. Phillips & C. Palamidessi (2007): *Tutorial on separation results in process calculi via leader election problems*. *Theoretical Computer Science* 388(1-3), pp. 267–289, doi:10.1016/j.tcs.2007.09.001.