Formal Methods Laboratory

Romanian Academy

FML Technical Report

TR no. FML-10-02

Printed:
August 2013

Continuation Semantics for Asynchronous Concurrency

G.Ciobanu and E.N.Todoran

Prepared by
Formal Methods Laboratory.

Series of
Technical Reports (Institute of Computer Science Iaşi)

ISSN 1842 - 1490

Series of Technical Reports (Institute of Computer Science Iaşi)
ISSN 1842 - 1490

Continuation Semantics for Asynchronous Concurrency

by

Gabriel Ciobanu [1,2]
Institute of Computer Science,
Romanian Academy, Iaşi
`gabriel@iit.tuiasi.ro`

Eneia Nicolae Todoran
Technical University of Cluj-Napoca
Department of Computer Science
400027 Cluj-Napoca, Romania
`Eneia.Todoran@cs.utcluj.ro`

ABSTRACT

The paper presents a method of reasoning about the behavior of asynchronous programs in denotational models designed with metric spaces and continuation semantics for concurrency.

[1]Head of Formal Methods Laboratory (FML)
[2]Contact person

Contents

1 Introduction

Metric spaces provide a convenient framework for designing denotational models of concurrency [4]. The main mathematical tool used in this approach to semantics is Banach's theorem, which states that a contracting function on a complete metric space has a *unique* fixed point. Contractions play a central role in designing and relating concurrency semantics [21] and there is a general method of solving reflexive domain equations in a category of complete metric spaces [1]. By using the methodology of metric semantics in [31] we introduced a "continuation semantics for concurrency" (CSC). The CSC technique can be used to model both sequential and parallel composition in interleaving semantics while providing the general advantages of the classic technique of continuations [12].

In continuation semantics a program is conceptually divided into a *current* statement and the *rest* of the program. A continuation is a representation of the behavior of the rest of the program, an *evaluation context* for the denotation of the current statement [16]. The distinctive characteristic of the CSC technique is the modeling of continuations as application-specific structures of computations (partially evaluated denotations) rather than the functions to some answer type that are used in the classic technique of continuations [30]. Intuitively it is a semantic formalization of a (process) scheduler, a *denotational scheduler*. Computations are grouped in a continuation. There is one *active* computation (the denotation of the current statement). Each computation remains active only until it performs and atomic action. Next, another computation taken from the continuation is planned for execution. In this way it can be obtained the desired interleaving behavior for parallel composition.

We do not know whether the domain of CSC is fully abstract.[1] However, as continuations are semantic evaluation contexts [16], it is easy to prove that a denotational semantics designed with CSC is *correct* with respect to a corresponding operational model (see section 5 of [31]). Also, the domain of CSC is very general. Only the structure of continuations needs to be adapted to the (concurrent) language under study [34]. In this sense the CSC technique provides *flexibility* in the denotational design of concurrent languages. For example, in the case of a sequential language a continuation can be a stack of computations. It is also natural to use the concept of a multiset to represent parallel computations. For a general combination of sequential and parallel composition a continuation is a tree of computations.

In this paper we use CSC to investigate the semantics of a simple concurrent language $\mathcal{L}$ embodying a mechanism of asynchronous communication. We work within the mathematical framework of 1-bounded complete metric spaces, by following the approach advocated in [3]. In theory syncronization is used because it is simple to express. Based on the results expressed in process algebra, asynchronous interaction is primitive, and syncronous interaction could be expressed in terms of asyncronous interaction; see, e.g. [19]. Moreover, asynchronous interaction is easier to implement. The relation between synchronous and

[1]In fact, we are not aware of any full abstractness result for a concurrent language designed with continuations, although various papers employ continuations in the denotational description of concurrent languages [3, 13, 2, 28].

asynchronous communication is a topic of recent and current research [25, 35]. Although synchronous interaction cannot always be expressed in terms of asynchronous primitives, asynchronous interaction is a basic mechanism in many distributed computing systems, including Internet and Web aplications.

The language $\mathcal{L}$ that we study in this paper embodies the paradigm of asynchronous communication introduced in [10]. The paradigm consists of a language based on a set of atomic actions and operators for sequential composition, nondeterministic choice and parallel composition. The semantics of atomic actions is defined with respect to an abstract set of states; the model includes special states indicating suspension and deadlock. The atomic actions are interpreted as state transformations. As explained in [10], various asynchronous computing models can be obtained as instances of this paradigm, including concurrent constraint programming [29], and also in other languages like dataflow and asynchronous CSP. $\mathcal{L}$ extends the paradigmatic language studied in [10] with recursion.

For the language under consideration, we show that the semantic operators designed with CSC satisfy some concurrency laws, such as the associativity and commutativity of parallel composition. Continuation-based models rely heavily on manipulations of higher-order functions. It may be difficult to reason directly in terms of higher-order functions. Therefore we introduce a *left merge* operator and we obtain a finite axiomatization of the parallel composition (or *merge*) operator. Any nonrecursive asynchronous concurrent program is thus provably equivalent to a corresponding nondeterministic sequential program. Obviously, the approach is inspired by classic process algebra theories [23, 8] this approach being adapted by us to a continuation-based framework.

The main contribution is given by the proofs that continuation semantics satisfies some basic laws such as the associativity of the parallel and sequential composition operators. Each semantic property, also called a *law* here, can be proved by identifying a corresponding invariant of the computation. Such an invariant is expressed as a relation between continuation structures. The identification of semantic properties from the invariants of the computation is common in classic bisimulation semantics [22]. The idea is adapted to a denotational framework based on continuations, by using arguments of the kind '$\frac{1}{2} \cdot \varepsilon \leq \varepsilon \Rightarrow \varepsilon = 0$', which are standard in metric semantics [3]. In our case ε is the distance between two behavioraly equivalent continuations, before and after a computation step, respectively. The effect of each computation step is given by the $\frac{1}{2}$ contracting factor. Therefore $\varepsilon = 0$ and the desired property follows.

The rest of the paper is organized as follows. Section 2 presents some theoretical preliminaries. Section 3 defines formally the language $\mathcal{L}$ and presents a mathematical structure that we use to define the domain of continuations. In section 4 we present a denotational semantics for $\mathcal{L}$ designed with CSC. In section 5 we prove the various laws that are satisfied by this semantics; we obtain a finite axiomatization of parallel composition. Section 6 provides some concluding remarks and objectives of future research.

2 Preliminaries

The notation $(x \in)X$ introduces the set X with typical element x ranging over X. Let $f \in X \to Y$ be a function. The function $[\, f \mid x \mapsto y\,] : X \to Y$, is defined (for $x, x' \in X, y \in Y$) by: $[\, f \mid x \mapsto y\,](x') = \textsf{if } x' = x \textsf{ then } y \textsf{ else } f(x')$. We use the notation $[\, f \mid x_1 \mapsto y_1 \mid x_2 \mapsto y_2]$ as an abbreviation for $[\,[\, f \mid x_1 \mapsto y_1\,] \mid x_2 \mapsto y_2\,]$. If $f : X \to X$ and $f(x) = x$ we call x a *fixed point* of f. When this fixed point is unique (see Theorem 2.1) we write $x = fix(f)$.

The denotational semantics given in this paper is built within the mathematical framework of *1-bounded complete metric spaces*, following the approach advocated in [3]. We work with the following notions which we assume known: *metric* and *ultrametric* space, *isometry* (distance preserving bijection between metric spaces, denoted by '$\cong$'), *complete* metric space, and *compact* set.

2.0.1 Examples.

1. Let X be an arbitrary set. The *discrete metric* on X ($d : X \times X \to [0, 1]$) is defined (for $x, y \in X$) as follows:

$$\begin{aligned} d(x, y) &= 0 \quad \textsf{if } x = y \\ &= 1 \quad \textsf{if } x \neq y \end{aligned}$$

   (X, d) is a complete ultrametric space.

2. Let $(a \in)A$ be a nonempty set, and let $(x, y \in)A^\infty = A^* \cup A^\omega$, where $A^*(A^\omega)$ is the set of all finite (infinite) sequences over A. One can define a metric over A^∞ as follows:

$$d(x, y) = 2^{-\sup\{\, n \mid x(n) = y(n)\,\}}$$

   where $x(n)$ denotes the prefix of x of length n, in case $length(x) \geq n$, and x otherwise (by convention, $2^{-\infty} = 0$). d is a Baire-like metric. (A^∞, d) is a complete ultrametric space.

We recall that if (X, d_X), (Y, d_Y) are metric spaces, a function $f : X \to Y$ is a *contraction* if $\exists c \in \mathbb{R},\ 0 \leq c < 1,\ \forall x_1, x_2 \in X : d_Y(f(x_1), f(x_2)) \leq c \cdot d_X(x_1, x_2)$. In metric semantics it is customary to attach a $\frac{1}{2}$-contracting factor to each computation step. When $c = 1$ the function f is called *nonexpansive*. In what follows we denote the set of all nonexpansive functions from X to Y by $X \overset{1}{\to} Y$. Banach's fixed point theorem [7] is at the core of metric semantics.

Theorem 2.1 (Banach) *Let (X, d_X) be a complete metric space. Each contraction $f : X \to X$ has a* unique *fixed point.*

Definition 2.2 *Let $(X, d_X), (Y, d_Y)$ be (ultra) metric spaces. On $(x \in)X$, $(f{\in})X \to Y$ (the function space), $(x, y){\in}X{\times}Y$ (the Cartesian product), $u, v{\in}X + Y$ (the disjoint union of X and Y, which can be defined by: $X + Y = (\{1\} \times X) \cup (\{2\} \times Y)$), and $U, V{\in}\mathcal{P}(X)$ (the power set of X) one can define the following metrics:*

(a) $d_{\frac{1}{2}\cdot X} : X \times X \to [0, 1] \quad d_{\frac{1}{2}\cdot X}(x_1, x_2) = \frac{1}{2} \cdot d_X(x_1, x_2)$

(b) $d_{X \to Y} : (X \to Y) \times (X \to Y) \to [0, 1]$

$$d_{X \to Y}(f_1, f_2) = \sup_{x \in X} d_Y(f_1(x), f_2(x))$$

(c) $d_{X \times Y} : (X \times Y) \times (X \times Y) \to [0, 1]$

$$d_{X \times Y}((x_1, y_1), (x_2, y_2)) =$$

$$max\{d_X(x_1, x_2), d_Y(y_1, y_2)\}$$

(d) $d_{X+Y} : (X + Y) \times (X + Y) \to [0, 1]$

$$d_{X+Y}(u, v) = \text{ if } (u, v \in X) \text{ then } d_X(u, v)$$

$$\text{else if } (u, v{\in}Y) \text{ then } d_Y(u, v) \text{ else } 1$$

(e) $d_H : \mathcal{P}(X) \times \mathcal{P}(X) \to [0, 1]$*:*

$$d_H(U, V) = max\{\sup_{u \in U} d(u, V), \sup_{v \in V} d(v, U)\}$$

where $d(u,W) = \inf_{w \in W} d(u, w)$ *and by convention* $\sup \emptyset = 0$, $\inf \emptyset = 1$ *(d_H is the* Hausdorff *metric).*

We use the abbreviation $\mathcal{P}_{nco}(\cdot)$ to denote the power set of *non-empty and compact* subsets of '$\cdot$'. Also, we often suppress the metrics part in domain definitions, and write, e.g., $\frac{1}{2} \cdot X$ instead of $(X, d_{\frac{1}{2}\cdot X})$.

Remark 2.3 *Let $(X, d_X), (Y, d_Y), d_{\frac{1}{2}\cdot X}$, $d_{X \to Y}, d_{X \times Y}$, d_{X+Y} and d_H be as in Definition 2.2. In case d_X, d_Y are ultrametrics, so are $d_{\frac{1}{2}\cdot X}, d_{X \to Y}, d_{X \times Y}, d_{X+Y}$ and d_H. Moreover, if $(X, d_X), (Y, d_Y)$ are complete then $\frac{1}{2} \cdot X$, $X \to Y$, $X \xrightarrow{1} Y$, $X \times Y$,$X + Y$, and $\mathcal{P}_{\text{nco}}(X)$ (with the metrics defined above) are also complete metric spaces [3].*

We also use the abbreviation $\mathcal{P}_{finite}(\cdot)$ to denote the power sets of *finite* subsets of '$\cdot$'. In general, the construct $\mathcal{P}_{finite}(\cdot)$ does not give rise to a complete space. In our study, we use it to create a structure that we endow with the discrete metric. Any set endowed with the discrete metric is a complete ultrametric space.

3 Syntax and continuation structure for $\mathcal{L}$

The syntax of $\mathcal{L}$ is given in BNF in Definition 3.1. The basic components are a set $(a \in)Act$ of *atomic actions* and a set $(x \in)RVar$ of *recursion variables*. There is a special symbol $\delta \in Act$, whose behavior is explained below. ;, $+$ and $\|$ are operators for sequential, nondeterministic and parallel composition, respectively. $\|$ is also called a *merge* operator, and $⌊⌊$ is the *left merge* operator.

Definition 3.1 *(Syntax of $\mathcal{L}$)*

(a) *(Statements)* $s(\in Stat) ::= a \mid x \mid s;s \mid s+s \mid s ⌊⌊\, s \mid s \parallel s$

(b) *(Guarded statements)* $g(\in GStat) ::= a \mid g;s \mid g+g \mid g ⌊⌊\, s \mid g\|g$

(d) *(Declarations)* $(D \in)Decl = RVar \to GStat$

(e) *(Programs)* $(\rho \in)\mathcal{L} = Decl \times Stat$

The meaning of atomic actions is defined by an interpretation function $I : Act \to \Sigma \to (\{\uparrow\} \cup \Sigma)$, where $(\sigma \in)\Sigma$ is a set of *states*. If $I(a)(\sigma) = \uparrow$ the action a cannot proceed in state σ; its execution is *suspended*. When all processes are suspended *deadlock* occurs. Notice that $I(\delta)(\sigma) = \uparrow, \forall \sigma \in \Sigma$, i.e. the action δ suspends in all states. $\mathcal{L}$ incorporates the mechanism of *asynchronous communication* studied in [10]. As explained in [10], this form of asynchronous communication can be encountered in concurrent constraint programming, and also in other languages like dataflow or asynchronous CSP.

We employ an approach to recursion based on *declarations* and *guarded statements* [3]. In a guarded statement each recursive call is preceded by at least one elementary action, which guarantees the fact that the semantic operators are contracting functions in the present metric setting. For the sake of brevity (and without loss of generality) in what follows we assume a fixed declaration $D \in Decl$, and all considerations in any given argument refer to this fixed D.

For inductive proofs we introduce a complexity measure ς that decreases upon recursive calls. ς is well-defined due to our restriction to guarded recursion.

Definition 3.2 *(Complexity measure) The function $\varsigma : Stat \to \mathbb{N}$ is given by*

$$\begin{array}{rcl} \varsigma(a) & = & 1 \\ \varsigma(x) & = & 1 + \varsigma(D(x)) \\ \varsigma(s_1 \;\mathsf{op}\; s_2) & = & 1 + \varsigma(s_1) \quad \mathsf{op} \in \{;\, , ⌊⌊\,\} \\ \varsigma(s_1 \;\mathsf{op}\; s_2) & = & 1 + max\{\varsigma(s_1), \varsigma(s_2)\} \quad \mathsf{op} \in \{+, \|\} \end{array}$$

In the CSC approach a continuation is a structured configuration of computations. For example, in the case of a sequential language a continuation is a *stack* of computations. It is also natural to use the concept of a *multiset* to represent parallel computations. For a general combination of parallel and sequential composition a continuation is a tree of computations with active computations at the leaves. For example, when the denotation of a program fragment $(s_1 \parallel s_2); s_3$ is computed, the denotations of s_1 and s_2 become leaves in such a tree and the denotation of s_3 becomes an inner node. This behavior is inspired by the concept of a *cactus stack* [9], a stack with multiple tops that can be active concurrently. In order to define such domains of trees of computations we employ a (partially ordered) set of *identifiers* Id. $(\alpha \in)Id$ is the set of all finite, possibly empty,sequences over $\{1,2\}$, and $\alpha \leq \alpha'$ iff α is a prefix of α'.

In this paper we use the symbol '$\cdot$' as a concatenation operator over sequences, hence we can represent any nonempty identifier $\alpha \in Id$ by a finite sequence $\alpha = i_1 \cdot \ldots \cdot i_n$, where $i_1, \ldots, i_n \in \{1,2\}$. We use the symbol λ to represent the empty sequence over $\{1,2\}$ $(\lambda \in Id)$.

Definition 3.3

(a) *Let* $(\alpha \in)Id = \{1,2\}^*$ *be a set of* identifiers*, equipped with the following partial ordering:* $\alpha \leq \alpha$*' iff* $\alpha' = \alpha \cdot i_1 \cdot \ldots \cdot i_n$ *for* $i_1, \cdots, i_n \in \{1,2\}, n \geq 0$. *We define* $\alpha < \alpha'$ *iff* $\alpha \leq \alpha'$ *and* $\alpha \neq \alpha'$. *If* $A \in \mathcal{P}(Id)$, *we denote by* $\leq_A$ *the restriction of* $\leq$ *to* A.

(b) *We define a function max* : $\mathcal{P}(Id) \to \mathcal{P}(Id)$ *by:*

$$max(A) = \{\alpha \mid \alpha \text{ is a maximal element of} (A, \leq_A)\}$$

Remark 3.4 $\lambda \leq \alpha$, *for any* $\alpha \in Id$, *which means that* λ *is the* least *element of* Id. *Also, when* $A \in \mathcal{P}(Id)$, α *is a* maximal *element of* $(A, \leq_A)$ *if* $\alpha \in A$ *and* $\forall \alpha' \in A : \neg(\alpha < \alpha')$. *The concept of a* tree *that we use in this paper is taken from set theory, where a tree is a partially ordered set in which the predecessors of each element are well-ordered. A set is well-ordered if it is linearly ordered and every nonempty subset has a least element. A set is linearly ordered if any two elements are comparable. There are several books on set theory that provide formal definitions of these concepts; see, e.g., [20]. Here we only explain the concept of a tree by means of an example.*

$(Id, \leq)$ *is a partially ordered set, i.e.* $\leq$ *is a binary relation over* Id *which is reflexive, transitive and antisymmetric. In this paper we only work with finite trees. If* $A \subseteq Id$ *is a finite subset of* Id *then* $(A, \leq_A)$ *is a finite tree. For example, let* $A = \{\alpha \cdot 1, \alpha \cdot 2, \alpha \cdot 1 \cdot 1, \alpha \cdot 1 \cdot 2, \alpha \cdot 2 \cdot 1, \alpha \cdot 2 \cdot 2, \alpha \cdot 1 \cdot 1 \cdot 1\}$, *for some* $\alpha \in Id$. $(A, \leq_A)$ *is a finite tree. The* maximal elements *of* $(A, \leq_A)$ *are exactly the* leaves *of the tree:* $max(A) = \{\alpha \cdot 1 \cdot 1 \cdot 1, \alpha \cdot 1 \cdot 2,\ \alpha \cdot 2 \cdot 1, \alpha \cdot 2 \cdot 2\}$. *The predecessors of each element in* A *are well-ordered. For example,* $\alpha \cdot 1 \cdot 1 \cdot 1 > \alpha \cdot 1 \cdot 1$ *and* $\alpha \cdot 1 \cdot 1 > \alpha \cdot 1$. *The set of predecessors of* $\alpha \cdot 1 \cdot 1 \cdot 1$ *is* $\{\alpha \cdot 1 \cdot 1, \alpha \cdot 1\}$, *which is linearly ordered, i.e. any two elements in* $\{\alpha \cdot 1 \cdot 1, \alpha \cdot 1\}$ *are comparable. In general,* α_1 *and* α_2 *are comparable iff* $\alpha_1 \leq \alpha_2$ *or* $\alpha_2 \leq \alpha_1$. *Obviously, every nonempty subset of* $\{\alpha \cdot 1 \cdot 1, \alpha \cdot 1\}$ *has a least element. In fact, every finite linearly ordered set is well-ordered.*

Let $(x \in)\mathbf{X}$ be a metric domain, i.e. a complete metric space. We use the following notation:

$$\{\!|\mathbf{X}|\!\} \overset{not.}{=} \mathcal{P}_{finite}(Id) \times (Id \to \mathbf{X})$$

Let $\alpha \in Id$, $(\pi, \theta) \in \{\!|\mathbf{X}|\!\}$ with $\pi \in \mathcal{P}_{finite}(Id)$, $\theta \in Id \to \mathbf{X}$. We define $id : \{\!|\mathbf{X}|\!\} \to \mathcal{P}_{finite}(Id)$, $id(\pi, \theta) = \pi$. We also use the following abbreviations:

$$\begin{array}{rcll}
(\pi,\theta)(\alpha) & \overset{not.}{=} & \theta(\alpha) & (\in \mathbf{X}) \\
(\pi,\theta) \setminus \alpha & \overset{not.}{=} & (\pi \setminus \{\alpha\}, \theta) & (\in \{\!|\mathbf{X}|\!\}) \\
[\,(\pi,\theta) \mid \alpha \mapsto x\,] & \overset{not.}{=} & (\pi \cup \{\alpha\}, [\,\theta \mid \alpha \mapsto x\,]) & (\in \{\!|\mathbf{X}|\!\})
\end{array}$$

The basic idea is that we treat (π, θ) as a 'function' with finite graph $\{(\alpha, \theta(\alpha)) \mid \alpha \in \pi\}$, thus ignoring the behaviour of θ for any $\alpha \notin \pi$ (π is the 'domain' of (π, θ)). We use this mathematical structure to represent finite partially ordered *bags* (or multisets)[2] of computations. The set Id is used to distinguish between multiple occurrences of a computation in such a bag. We endow both sets Id and $\mathcal{P}_{finite}(Id)$ with discrete metrics; every set with a discrete metric is a complete ultrametric space. By using the composite metrics given in Definition 2.2 $\{\!|\mathbf{X}|\!\}$ becomes also a metric domain. The operators behave as follows. $id(\pi, \theta)$ returns the collection of identifiers for the valid computations contained in the bag (π, θ), $(\pi, \theta)(\alpha)$ returns the computation with identifier α, $(\pi, \theta) \setminus \alpha$ removes the computation with identifier α, and $[\,(\pi, \theta) \mid \alpha \mapsto x\,]$ replaces the computation with identifier α.

By a slight abuse, we use the same notations (including the operator id and the abbreviations $(\cdot)(\alpha), (\cdot) \setminus \alpha, [\,\cdot \mid \alpha \mapsto x\,]$) when $(x \in)X$ is an ordinary set: $\{\!|X|\!\} = \mathcal{P}_{finite}(Id) \times (Id \to X)$; in this case we do not endow $\{\!|X|\!\}$ with a metric.

4 Continuation semantics for $\mathcal{L}$

We design a continuation-based denotational semantics for $\mathcal{L}$. As a semantic universe for the final yield of our denotational model we employ a standard linear-time domain $(p \in)\mathbf{P} = \mathcal{P}_{nco}(\Sigma^* \cup \Sigma^* \cdot \{\delta\} \cup \Sigma^\omega)$. Here $\Sigma^*(\Sigma^\omega)$ denotes the collection of all finite (infinite) sequences over Σ. An element of $\Sigma^* \cdot \{\delta\}$ is a finite sequence terminated with the symbol δ, which denotes *deadlock*. Also, we use the symbol λ to represent the empty sequence. This is a slight abuse of notation since we also use the symbol λ to represent the empty sequence over $\{1, 2\}$; however it is always clear from the context which is the type of λ (either $\lambda \in Id$ $(= \{1, 2\}^*)$, or $\lambda \in \Sigma^*$). We view $(q \in)\Sigma^* \cup \Sigma^* \cdot \{\delta\} \cup \Sigma^\omega$ as a complete ultrametric space by endowing it with the Baire metric (see section 2). We use the notation $\sigma \cdot p = \{\sigma \cdot q \mid q \in p\}$, for any $\sigma \in \Sigma$ and $p \in \mathbf{P}$. The type of the denotational semantics $[\![\cdot]\!]$ for $\mathcal{L}$ is $Stat \to \mathbf{D}$, where:

$$\begin{array}{rcl}
\mathbf{D} & \cong & \mathbf{Cont} \overset{1}{\to} \Sigma \to \mathbf{P} \\
(\gamma \in)\mathbf{Cont} & = & Id \times \mathbf{Kont} \\
(\kappa \in)\mathbf{Kont} & = & \{\!|\tfrac{1}{2} \cdot \mathbf{D}|\!\}
\end{array}$$

[2]We avoid using the notion of a *partially ordered multiset* which is a more refined structure – see [6], or chapter 16 of [3].

In the equations given above the sets Σ, Id (and $\mathcal{P}_{finite}(Id)$) are endowed with the discrete (ultra)metric. The composed metric spaces are built up using the metrics of Definition 2.2. To conclude that such a system of equations has a solution, which is unique up to isometry, we rely on the general method developed in [1]. The solution for $\mathbf{D}$ is obtained as a complete ultrametric space. In [1], the family of complete (ultra)metric spaces is made into a category $\mathcal{C}$. It is proved that a generalized form of Banach's fixed point theorem holds, stating that a functor $\mathcal{F} : \mathcal{C} \to \mathcal{C}$ that is *contracting* (in a sense) has a unique fixed point (up to isometry). Intuitively, in the equation above the relevant functor is contracting as a consequence of the fact that the recursive occurence of $\mathbf{D}$ is preceded by a $\frac{1}{2}$ factor.

The construction $\{\!|\frac{1}{2}\cdot \mathbf{D}|\!\} = \mathcal{P}_{finite}(Id) \times (Id \to \frac{1}{2}\cdot \mathbf{D})$ was introduced in section 3. In the sequel ϑ ranges over $Id \to \frac{1}{2}\cdot \mathbf{D}$. We call an element of **Kont** a *closed continuation* and an element of **Cont** an *open continuation*. A (closed or open) continuation is a representation of what remains to be computed from the program [30]. A closed continuation $\kappa \in \mathbf{Kont}$ is a self-contained structure of computations. An open continuation $(\alpha, \kappa) \in \mathbf{Cont}$ behaves like an evaluation context [16] for the denotational mapping $[\![\cdot]\!]$. In an expression $[\![s]\!](\alpha, \kappa)$, $[\![s]\!]$ is the *active computation* which is evaluated with respect to (α, κ). Intuitively, an open continuation (α, κ) is a structured configuration of computations which contains a *hole*, indicating the conceptual position of the active computation. The position of the 'hole' is given in this representation by the identifier α, which is not an element of $id(\kappa)$ ($[\![\cdot]\!]$ is designed to preserve this invariant property: $\alpha \notin id(\kappa)$ and $\alpha \in max(\{\alpha\} \cup id(\kappa))$).

The denotational function $[\![\cdot]\!]$ is defined in 4.1 with the aid of a mapping kc, which is called a *scheduler*. The denotational function maps an open continuation to a program behavior. After producing an elementary step the denotational function transmits the control to the scheduler. The scheduler receives as parameter a closed continuation κ which it maps to a corresponding program behavior. If the continuation κ is empty ($id(\kappa) = \emptyset$) the scheduler terminates the computation. Otherwise the scheduler activates a computatation (denotation) selected in a nondeterministic manner from the continuation; it decomposes a closed continuation into a computation and a corresponding open continuation and then it executes the former with the latter as continuation.

The semantics of nondeterministic choice in $\mathcal{L}$ is given by the operator $+ : (\mathbf{P} \times \mathbf{P}) \to \mathbf{P}$. This definition reflects that $p_1 + p_2$ blocks only if both p_1 and p_2 block. It is easy to check that $+$ is well-defined, nonexpansive, associative, commutative and idempotent. Also, $p + \{\delta\} = p$, for any $p \in \mathbf{P}$.

$$p_1 + p_2 = \{q \mid q \in p_1 \cup p_2, q \neq \delta\} \cup \{\delta \mid \delta \in p_1 \cap p_2\}$$

Definition 4.1 *(Denotational semantics for $\mathcal{L}$)*

(a) Let $kc : \mathbf{Kont} \to \Sigma \to \mathbf{P}$ *be given by:*

$$kc(\kappa)(\sigma) \;=\; \textsf{if } (id(\kappa) = \emptyset) \textsf{ then } \{\lambda\} \textsf{ else } +_{\alpha \in max(id(\kappa))} \; \kappa(\alpha)(\alpha, \kappa \setminus \alpha)(\sigma)$$

(b) *We define* $\Phi : (Stat \to \mathbf{D}) \to (Stat \to \mathbf{D})$ *(for* $\phi \in (Stat \to \mathbf{D})$*) by:*

$$\begin{array}{rcl}
\Phi(\phi)(a)(\alpha,\kappa)(\sigma) & = & \mathsf{if}\ (I(a)(\sigma) = \uparrow)\ \mathsf{then}\ \{\delta\} \\
& & \mathsf{else}\ I(a)(\sigma) \cdot kc(\kappa)(I(a)(\sigma)) \\
\Phi(\phi)(x)(\alpha,\kappa)(\sigma) & = & \Phi(\phi)(D(x))(\alpha,\kappa)(\sigma) \\
\Phi(\phi)(s_1;s_2)(\alpha,\kappa)(\sigma) & = & \Phi(\phi)(s_1)(\alpha\cdot 1,[\,\kappa \mid \alpha \mapsto \phi(s_2)\,])(\sigma) \\
\Phi(\phi)(s_1+s_2)(\alpha,\kappa)(\sigma) & = & \Phi(\phi)(s_1)(\alpha,\kappa)(\sigma)\ +\ \Phi(\phi)(s_2)(\alpha,\kappa)(\sigma) \\
\Phi(\phi)(s_1 \mathbin{\lfloor\!\lfloor} s_2)(\alpha,\kappa)(\sigma) & = & \Phi(\phi)(s_1)(\alpha\cdot 1,[\,\kappa \mid \alpha\cdot 2 \mapsto \phi(s_2)\,])(\sigma) \\
\Phi(\phi)(s_1 \parallel s_2)(\alpha,\kappa)(\sigma) & = & \Phi(\phi)(s_1)(\alpha\cdot 1,[\,\kappa \mid \alpha\cdot 2 \mapsto \phi(s_2)\,])(\sigma)\ + \\
& & \Phi(\phi)(s_2)(\alpha\cdot 1,[\,\kappa \mid \alpha\cdot 2 \mapsto \phi(s_1)\,])(\sigma)
\end{array}$$

(c) *We put* $[\![\cdot]\!] = fix(\Phi)$. *Let* $\alpha_0 = \lambda$, *and* $\kappa_0 = (\emptyset, \vartheta_0)$, *where* $\vartheta_0(\alpha) = [\![\delta]\!]$, $\forall \alpha \in Id$. $(\alpha_0,\kappa_0) \in \mathbf{Cont}$ *is the* empty continuation. *Notice that* $kc(\kappa_0)(\sigma) = \{\lambda\}$, *for any* $\sigma \in \Sigma$. *We also define* $\mathcal{D}[\![\cdot]\!] : Stat \to \Sigma \to \mathbf{P}$ *by:*

$$\mathcal{D}[\![s]\!] = [\![s]\!](\alpha_0,\kappa_0)$$

The semantics of atomic actions is defined with the aid of the interpretation function I introduced in the paragraph that follows after Definition 3.1.

A continuation is a tree of computations with active elements at the leaves (the maximal elements with respect to '$\leq$'). In the case of a sequential composition $(s_1;s_2)$ the computations $[\![s_1]\!]$ and $[\![s_2]\!]$ are given the identifiers $\alpha\cdot 1$ and α, respectively $(\alpha\cdot 1 > \alpha)$. The scheduler function kc gives priority to the computations at the leaves of the tree that represents the continuation. Therefore $[\![s_2]\!]$ will only be evaluated after the completion of the evaluation of $[\![s_1]\!]$. In the case of a parallel composition $(s_1 \parallel s_2)$ the computations $[\![s_1]\!]$ and $[\![s_2]\!]$ are given identifiers $(\alpha \cdot 1$ and $\alpha \cdot 2)$ that are incomparable (with respect to $\leq$) therefore the computations $[\![s_1]\!]$ and $[\![s_2]\!]$ are evaluated in an interleaved manner.

The denotational semantics $[\![\cdot]\!]$ is defined as the (unique) fixed point of the higher-order mapping Φ. It may not be obvious why on the right-hand sides of the equations given in Definition 4.1(b), in some places we use $\Phi(\phi)$ while in others we use ϕ. The definition of $\Phi(\phi)$ is organized by induction on $\varsigma(s)$ (see Definition 3.2). The computations $\phi(s)$ only occur in the continuation and are always executed after an elementary step performed by the active computation. This step ensures the contractiveness of Φ and is reflected by the $\frac{1}{2}\cdot$ factor in the definition of the domain of computations. Definition 4.1 is justified by Lemmas 4.2 and 4.3, whose proofs are omitted. Similar Lemmas are given in [31, 32]. See, e.g., the proofs of Lemmas 3.13 and 3.14 in [31].[3]

[3]In Lemma 4.3(b), $\Phi(\phi)(s)$ is (only) nonexpansive (rather than contractive) in the continuation. Still, this implies that Φ is $\frac{1}{2}\cdot$ -contractive in ϕ. Intuitively, the distance between denotations halves while they are stored into a continuation. This is a consequence of the $\frac{1}{2}\cdot$ -contracting factor in our domain equation. This also explains the occurence of the multiplication factor '$2\cdot$' (rather than '$\frac{1}{2}\cdot$') in Lemma 4.2(b).

Lemma 4.2

(a) The mapping kc (see Definition 4.1) is well-defined.

(b) $\forall \kappa_1, \kappa_2 \in \mathbf{Kont} : d(kc(\kappa_1), kc(\kappa_2)) \leq 2 \cdot d(\kappa_1, \kappa_2)$

Lemma 4.3 *For all* $\phi \in (Stat \rightarrow \mathbf{D}), s \in Stat, \alpha \in Id, \kappa \in \mathbf{Kont}, \sigma \in \Sigma$*:*

(a) $\Phi(\phi)(s)(\alpha, \kappa)(\sigma) \in \mathbf{P}$ *(it is well defined),*

(b) $\Phi(\phi)(s)$ *is nonexpansive (in* (α, κ)*), and*

(c) Φ *is* $\frac{1}{2}$ *- contractive (in* ϕ*).*

5 Concurrency laws in continuation semantics

We present a method of describing the behavior of concurrent systems in a denotational model designed with CSC, using a representation of continuations as structured configurations of computations. For the language $\mathcal{L}$ we show that the semantic operators satisfy laws that are usually included in concurrency theories, such as the associativity and commutativity of parallel composition.

Various properties can be proved for all continuations by simple manipulations of the equations that define the denotational mapping $[\![\cdot]\!]$.

Lemma 5.1 *For all* $s, s_1, s_2, s_3 \in Stat$:

(a) $[\![s_1 + s_2]\!] = [\![s_2 + s_1]\!]$ (*commutativity of* $+$)
(b) $[\![(s_1 + s_2) + s_3]\!] = [\![s_1 + (s_2 + s_3)]\!]$ (*associativity of* $+$)
(c) $[\![s + s]\!] = [\![s]\!]$ (*idempotency of* $+$)
(d) $[\![(s_1 + s_2); s_3]\!] = [\![s_1; s_3 + s_2; s_3]\!]$ (*right distributivity of* ; over $+$)
(e) $[\![s + \delta]\!] = [\![s]\!]$
(f) $[\![\delta; s]\!] = [\![\delta]\!]$
(g) $[\![s_1 \parallel s_2]\!] = [\![s_1 \lfloor\!\lfloor s_2 + s_2 \lfloor\!\lfloor s_1]\!]$
(h) $[\![(s_1 + s_2) \lfloor\!\lfloor s_3]\!] = [\![s_1 \lfloor\!\lfloor s_3 + s_2 \lfloor\!\lfloor s_3]\!]$ (*right distributivity of* $\lfloor\!\lfloor$ over $+$)
(i) $[\![s_1 \parallel s_2]\!] = [\![s_2 \parallel s_1]\!]$ (*commutativity of* $\parallel$)

Proof: In order to prove that $[\![s]\!] = [\![\overline{s}]\!]$, for $s, \overline{s} \in Stat$, it is enough to show that $[\![s]\!](\alpha, \kappa)(\sigma) = [\![\overline{s}]\!](\alpha, \kappa)(\sigma)$, for arbitrary $(\alpha, \kappa) \in \mathbf{Cont}$, $\sigma \in \Sigma$.

(a) $[\![s_1 + s_2]\!](\alpha, \kappa)(\sigma)$

$= [\![s_1]\!](\alpha, \kappa)(\sigma) + [\![s_2]\!](\alpha, \kappa)(\sigma)$ [+ is commutative]

$= [\![s_2]\!](\alpha, \kappa)(\sigma) + [\![s_1]\!](\alpha, \kappa)(\sigma)$

$= [\![s_2 + s_1]\!](\alpha, \kappa)(\sigma)$

(b) $[\![(s_1 + s_2) + s_3]\!](\alpha, \kappa)(\sigma)$

$= [\![s_1 + s_2]\!](\alpha, \kappa)(\sigma) + [\![s_3]\!](\alpha, \kappa)(\sigma)$

$= ([\![s_1]\!](\alpha, \kappa)(\sigma) + [\![s_2]\!](\alpha, \kappa)(\sigma)) + [\![s_3]\!](\alpha, \kappa)(\sigma)$ [+ is associative]

$= [\![s_1]\!](\alpha, \kappa)(\sigma) + ([\![s_2]\!](\alpha, \kappa)(\sigma) + [\![s_3]\!](\alpha, \kappa)(\sigma))$

$= [\![s_1]\!](\alpha, \kappa)(\sigma) + [\![s_2 + s_3]\!](\alpha, \kappa)(\sigma)$

$= [\![s_1 + (s_2 + s_3)]\!](\alpha, \kappa)(\sigma)$

(a) $[\![s + s]\!](\alpha, \kappa)(\sigma)$

$= [\![s]\!](\alpha, \kappa)(\sigma) + [\![s]\!](\alpha, \kappa)(\sigma)$ [+ is idempotent]

$= [\![s]\!](\alpha, \kappa)(\sigma)$

(d) $[\![(s_1 + s_2); s_3]\!](\alpha, \kappa)(\sigma)$

$= [\![s_1 + s_2]\!](\alpha \cdot 1, [\,\kappa \mid \alpha \mapsto [\![s_3]\!]\,])(\sigma)$

$= [\![s_1]\!](\alpha \cdot 1, [\,\kappa \mid \alpha \mapsto [\![s_3]\!]\,])(\sigma) + [\![s_2]\!](\alpha \cdot 1, [\,\kappa \mid \alpha \mapsto [\![s_3]\!]\,])(\sigma)$

$= [\![s_1; s_3]\!](\alpha, \kappa)(\sigma) + [\![s_2; s_3]\!](\alpha, \kappa)(\sigma)$

$= [\![(s_1; s_3) + (s_2; s_3)]\!](\alpha, \kappa)(\sigma)$

(e) $[\![s + \delta]\!](\alpha, \kappa)(\sigma)$

$= [\![s]\!](\alpha, \kappa)(\sigma) + [\![\delta]\!](\alpha, \kappa)(\sigma)$

$= [\![s]\!](\alpha, \kappa)(\sigma) + \{\delta\}$

$= [\![s]\!](\alpha, \kappa)(\sigma)$

(f) $[\![\delta; s]\!](\alpha, \kappa)(\sigma)$

$= [\![\delta]\!](\alpha \cdot 1, [\,\kappa \mid \alpha \mapsto [\![s]\!]\,])(\sigma)$

$= \{\delta\}$

$= [\![\delta]\!](\alpha, \kappa)(\sigma)$

(g) $[\![s_1 \parallel s_2]\!](\alpha, \kappa)(\sigma)$

$= [\![s_1]\!](\alpha \cdot 1, [\,\kappa \mid \alpha \cdot 2 \mapsto [\![s_2]\!]\,])(\sigma) + [\![s_2]\!](\alpha \cdot 1, [\,\kappa \mid \alpha \cdot 2 \mapsto [\![s_1]\!]\,])(\sigma)$

$= [\![s_1 \lfloor\!\lfloor s_2]\!](\alpha, \kappa)(\sigma) + [\![s_2 \lfloor\!\lfloor s_1]\!](\alpha, \kappa)(\sigma)$

$= [\![s_1 \lfloor\!\lfloor s_2 + s_2 \lfloor\!\lfloor s_1]\!](\alpha, \kappa)(\sigma)$

(h) $[\![(s_1 + s_2) \lfloor\!\lfloor s_3]\!](\alpha, \kappa)(\sigma)$

$= [\![s_1 + s_2]\!](\alpha \cdot 1, [\, \kappa \mid \alpha \cdot 2 \mapsto [\![s_3]\!] \,])(\sigma)$

$= [\![s_1]\!](\alpha \cdot 1, [\, \kappa \mid \alpha \cdot 2 \mapsto [\![s_3]\!] \,])(\sigma) + [\![s_2]\!](\alpha \cdot 1, [\, \kappa \mid \alpha \cdot 2 \mapsto [\![s_3]\!] \,])(\sigma)$

$= [\![s_1 \lfloor\!\lfloor s_3]\!](\alpha, \kappa)(\sigma) + [\![s_2 \lfloor\!\lfloor s_3]\!](\alpha, \kappa)(\sigma)$

$= [\![(s_1 \lfloor\!\lfloor s_3) + (s_2 \lfloor\!\lfloor s_3)]\!](\alpha, \kappa)(\sigma)$

(i) $[\![s_1 \parallel s_2]\!](\alpha, \kappa)(\sigma)$

$= [\![s_1]\!](\alpha \cdot 1, [\, \kappa \mid \alpha \cdot 2 \mapsto [\![s_2]\!] \,])(\sigma) + [\![s_2]\!](\alpha \cdot 1, [\, \kappa \mid \alpha \cdot 2 \mapsto [\![s_1]\!] \,])(\sigma)$

[+ is commutative]

$= [\![s_2]\!](\alpha \cdot 1, [\, \kappa \mid \alpha \cdot 2 \mapsto [\![s_1]\!] \,])(\sigma) + [\![s_1]\!](\alpha \cdot 1, [\, \kappa \mid \alpha \cdot 2 \mapsto [\![s_2]\!] \,])(\sigma)$

$= [\![s_2 \parallel s_1]\!](\alpha, \kappa)(\sigma)$ ■

5.1 Continuations and configurations

All the above proofs are straightforward. However, the flexibility provided by continuations comes at a price. Some properties may require more complex arguments and can be obtained for continuations that contain *only* denotations of program statements. We introduce the auxiliary notion of a *configuration* and a notion of *isomorphism* over configurations. A configuration is a structure of $\mathcal{L}$ statements. A continuation may contain arbitrary values of the type $\mathbf{D}$. We prove the desired properties for continuations that can be obtained as semantified versions of configurations, i.e. for continuations that contain only denotations of statements. This represents an invariant of the denotational semantics, and ensures its consistency just because the initial continuation is empty and the denotational semantics adds to the continuation only denotations of statements. The function K defined in 5.2(b) maps a configuration to a corresponding continuation.

Definition 5.2

(a) *We define the set of* closed configurations $(k \in) Konf = \{\!| Stat |\!\}$.[4] *A closed configuration is a finite (partially ordered) bag (multiset) of statements (∈Stat). Also, we define the set Conf of* open configurations *by:*

[4]In this case the construct $\{\!| \cdot |\!\}$ is used to define an ordinary set; see the explanation given in the final part of section 3.

$$Conf = \{(\alpha, k) \mid (\alpha, k) \in (Id \times Konf), \alpha \notin id(k), \alpha \in max(\{\alpha\} \cup id(k))\}$$

(b) We define $K : Konf \to$**Kont** *as follows:* $K(k) = (id(k), \vartheta)$*, where* $\vartheta(\alpha) = [\![k(\alpha)]\!], \forall \alpha \in Id$.

Definition 5.3

(a) We say that two closed configurations $k_1, k_2 \in Konf$ *are* isomorphic*, and we write* $k_1 \cong k_2$*, iff there exists a bijection* $\mu : id(k_1) \to id(k_2)$ *such that:*

(i) $\mu(\alpha') \leq \mu(\alpha'') \Leftrightarrow \alpha' \leq \alpha'', \quad \forall \alpha', \alpha'' \in id(k_1)$

(ii) $k_2(\mu(\alpha)) = k_1(\alpha), \quad \forall \alpha \in id(k_1)$

(b) We say that two open configurations $(\alpha_1, k_1), (\alpha_2, k_2) \in Conf$ *are* isomorphic*, and write* $(\alpha_1, k_1) \cong (\alpha_2, k_2)$ *iff there exists a bijection* $\mu : (\{\alpha_1\} \cup id(k_1)) \to (\{\alpha_2\} \cup id(k_2))$ *such that:*

(i) $\mu(\alpha_1) = \alpha_2$

(ii) $\mu(\alpha') \leq \mu(\alpha'') \Leftrightarrow \alpha' \leq \alpha'', \forall \alpha', \alpha'' \in \{\alpha_1\} \cup id(k_1)$

(iii) $k_2(\mu(\alpha)) = k_1(\alpha), \quad \forall \alpha \in id(k_1)$

Obviously, $\forall (\alpha, k) \in Conf : (\alpha, k) \cong (\alpha, k)$ and if $(\alpha_1, k_1), (\alpha_2, k_2) \in Conf$ then $(\alpha_1, k_1) \cong (\alpha_2, k_2) \Rightarrow k_1 \cong k_2$. Also, the following Lemma is given without proof (which is simple enough and can be used by the reader as proof exercise).

Lemma 5.4 *For all* $k, k_1, k_2 \in Konf$

(a) For any $\alpha \in Id, s \in Stat : \quad K[\, k \mid \alpha \mapsto s \,] = [\, K(k) \mid \alpha \mapsto [\![s]\!] \,]$

(b) $id(k) = id(K(k))$

(c) If $k_1 \cong k_2$ *then* $id(k_1) = \emptyset \Leftrightarrow id(k_2) = \emptyset$

(d) $K(k) \setminus \alpha = K(k \setminus \alpha)$*, for any* $\alpha \in Id$.

(e) If $\alpha \notin id(k)$ *then* $([\, k \mid \alpha \mapsto s \,] \setminus \alpha) \cong k$.

(f) For any $s \in Stat$ *and* $\alpha, \alpha' \in Id, \alpha' \neq \alpha$: $[\, k \mid \alpha \mapsto s \,] \setminus \alpha' = [\, k \setminus \alpha' \mid \alpha \mapsto s \,]$.

In Corollary 5.6 we show that any two continuations that correspond to isomorphic configurations behave the same. This result is obtained by combining Lemma 5.5 with an argument '$\varepsilon \leq \frac{1}{2} \cdot \varepsilon \Rightarrow \varepsilon = 0$'. Lemma 5.5 identifies the property - in this case the isomorphism between configurations - that is preserved by each computation step. The effect of each computation step is given in the present metric setting by the $\frac{1}{2}$-contracting factor.

Lemma 5.5

(a) For all $k_1, k_2 \in Konf$ *with* $k_1 \cong k_2$ *and* $\sigma \in \Sigma$*, there exists* $\overline{s} \in Stat$*,* $(\overline{\alpha}_1, \overline{k}_1), (\overline{\alpha}_2, \overline{k}_2) \in Conf$ *with* $(\overline{\alpha}_1, \overline{k}_1) \cong (\overline{\alpha}_2, \overline{k}_2)$ *such that:*

$$\begin{aligned} & d(kc(K(k_1))(\sigma), kc(K(k_2))(\sigma)) \leq \\ & \quad d([\![\overline{s}]\!](\overline{\alpha}_1, K(\overline{k}_1))(\sigma), [\![\overline{s}]\!](\overline{\alpha}_2, K(\overline{k}_2))(\sigma)) \end{aligned}$$

(b) For all $s \in Stat, (\alpha_1, k_1), (\alpha_2, k_2) \in Conf$ *with* $(\alpha_1, k_1) \cong (\alpha_2, k_2)$ *and* $\sigma \in \Sigma$*, there exists* $\overline{s} \in Stat, (\overline{\alpha}_1, \overline{k}_1), (\overline{\alpha}_2, \overline{k}_2) \in Conf$ *with* $(\overline{\alpha}_1, \overline{k}_1) \cong (\overline{\alpha}_2, \overline{k}_2)$ *and* $\overline{\sigma} \in \Sigma$ *such that:*

$$\begin{aligned} & d([\![s]\!](\alpha_1, K(k_1))(\sigma), [\![s]\!](\alpha_2, K(k_2))(\sigma)) \leq \\ & \quad \tfrac{1}{2} \cdot d([\![\overline{s}]\!](\overline{\alpha}_1, K(\overline{k}_1))(\overline{\sigma}), [\![\overline{s}]\!](\overline{\alpha}_2, K(\overline{k}_2))(\overline{\sigma})) \end{aligned}$$

Proof: For 5.5(a) we distinguish two subcases. If (by Lemma 5.4(b) and (c)) $id(K(k_1)) = \emptyset = id(K(k_1))$ then

$$d(kc(K(k_1))(\sigma), kc(K(k_2))(\sigma)) = d(\{\lambda\}, \{\lambda\}) = 0.$$

Otherwise,

$$\begin{aligned} & d(kc(K(k_1))(\sigma), kc(K(k_2))(\sigma)) \\ & = d(+_{\alpha \in max(id(K(k_1)))} K(k_1)(\alpha)(\alpha, K(k_1) \setminus \alpha)(\sigma), \\ & \qquad +_{\alpha \in max(id(K(k_2)))} K(k_2)(\alpha)(\alpha, K(k_2) \setminus \alpha)(\sigma)) \qquad \text{[Lemma 5.4(b)]} \\ & = d(+_{\alpha \in max(id(k_1))} K(k_1)(\alpha)(\alpha, K(k_1) \setminus \alpha)(\sigma), \\ & \qquad +_{\alpha \in max(id(k_2))} K(k_2)(\mu(\alpha))(\mu(\alpha), K(k_2) \setminus \mu(\alpha))(\sigma)) \\ & \qquad \text{[+ is nonexpansive]} \\ & = max\{d(K(k_1)(\alpha)(\alpha, K(k_1) \setminus \alpha)(\sigma), K(k_2)(\mu(\alpha))(\mu(\alpha), K(k_2) \setminus \mu(\alpha))(\sigma)) \\ & \qquad \mid \alpha \in max(id(k_1))\} \qquad \text{[Lemma 5.4(d)]} \\ & = max\{d([\![s]\!](\alpha, K(k_1 \setminus \alpha))(\sigma), [\![s]\!](\mu(\alpha), K(k_2 \setminus \mu(\alpha)))(\sigma)) \\ & \qquad \mid \alpha \in max(id(k_1)), s = k_1(\alpha) = k_2(\mu(\alpha))\} \end{aligned}$$

where $\mu : id(k_1) \to id(k_2)$ is a bijection that satisfies the properties given in Definition 5.3(a). As $k_1 \cong k_2$, $(\alpha, k_1 \setminus \alpha) \cong (\mu(\alpha), k_2 \setminus \mu(\alpha)) \in Conf$, for any $\alpha \in max(id(k_1))$. Therefore $\exists \overline{s} \in Stat, \overline{\alpha} \in Id$,[5] such that $\overline{s} = k_1(\overline{\alpha}) = k_2(\mu(\overline{\alpha}))$, $(\overline{\alpha}, k_1 \setminus \overline{\alpha}) \cong (\mu(\overline{\alpha}), k_2 \setminus \mu(\overline{\alpha}))$ and

$$d(kc(K(k_1))(\sigma), kc(K(k_2))(\sigma))$$

[5]$\overline{\alpha} \in max(id(k_1))$.

$\leq d([\![\overline{s}]\!](\overline{\alpha}, K(k_1 \setminus \overline{\alpha}))(\sigma), [\![\overline{s}]\!](\mu(\overline{\alpha}), K(k_2 \setminus \mu(\overline{\alpha})))(\sigma))$

which concludes the proof of 5.5(a).

Next, we treat 5.5(b) by induction on $\varsigma(s)$. Four subcases.

Case $[s = a]$ when $I(a)(\sigma) = \uparrow$.

$d([\![a]\!](\alpha_1, K(k_1))(\sigma), [\![a]\!](\alpha_2, K(k_2))(\sigma))$
$= d(\{\delta\}, \{\delta\}) = 0$

Case $[s = a]$ when $I(a)(\sigma) = \overline{\sigma} \in \Sigma$.

$d([\![a]\!](\alpha_1, K(k_1))(\sigma), [\![a]\!](\alpha_2, K(k_2))(\sigma))$
$= d(\overline{\sigma} \cdot kc(K(k_1))(\overline{\sigma}), \overline{\sigma} \cdot kc(K(k_2))(\overline{\sigma}))$
$= \frac{1}{2} \cdot\ d(kc(K(k_1))(\overline{\sigma}), kc(K(k_2))(\overline{\sigma}))\ ^{(5.5.b.1)}$

By Lemma 5.5(a), $\exists \overline{s} \in Stat, (\overline{\alpha}_1, \overline{k}_1) \cong (\overline{\alpha}_2, \overline{k}_2) \in Conf$ such that:
$^{(5.5.b.1)} \leq \frac{1}{2} \cdot\ d([\![\overline{s}]\!](\overline{\alpha}_1, K(\overline{k}_1))(\overline{\sigma}), [\![\overline{s}]\!](\overline{\alpha}_2, K(\overline{k}_2))(\overline{\sigma}))$

Case $[s = x]$

$d([\![x]\!](\alpha_1, K(k_1))(\sigma), [\![x]\!](\alpha_2, K(k_2))(\sigma))$
$= d([\![D(x)]\!](\alpha_1, K(k_1))(\sigma), [\![D(x)]\!](\alpha_2, K(k_2))(\sigma))\ ^{(5.5.b.2)}$

One can use the induction hypothesis $(\varsigma(D(x)) < \varsigma(x))$ and infer that $\exists \overline{s} \in Stat$, $(\overline{\alpha}_1, \overline{k}_1) \cong (\overline{\alpha}_2, \overline{k}_2) \in Conf$, such that:
$^{(5.5.b.2)} \leq \frac{1}{2} \cdot\ d([\![\overline{s}]\!](\overline{\alpha}_1, K(\overline{k}_1))(\overline{\sigma}), [\![\overline{s}]\!](\overline{\alpha}_2, K(\overline{k}_2))(\overline{\sigma}))$

Case $[s = s_1 \lfloor\!\lfloor s_2]$

$d([\![s_1 \lfloor\!\lfloor s_2]\!](\alpha_1, K(k_1))(\sigma), [\![s_1 \lfloor\!\lfloor s_2]\!](\alpha_2, K(k_2))(\sigma))$
$= d([\![s_1]\!](\alpha_1 \cdot 1, [\,K(k_1) \mid \alpha_1 \cdot 2 \mapsto [\![s_2]\!]\,])(\sigma),$
$\quad [\![s_1]\!](\alpha_2 \cdot 1, [\,K(k_2) \mid \alpha_2 \cdot 2 \mapsto [\![s_2]\!]\,])(\sigma)) \qquad$ [Lemma 5.4(a)]
$= d([\![s_1]\!](\alpha_1 \cdot 1, K[\,k_1 \mid \alpha_1 \cdot 2 \mapsto s_2\,])(\sigma),$
$\quad [\![s_1]\!](\alpha_2 \cdot 1, K[\,k_2 \mid \alpha_2 \cdot 2 \mapsto s_2\,])(\sigma))\ ^{(5.5.b.3)}$

It is easy to check that $(\alpha_1, k_1) \cong (\alpha_2, k_2)$ implies

$(\alpha_1 \cdot 1, [\,k_1 \mid \alpha_1 \cdot 2 \mapsto s_2\,]) \cong (\alpha_2 \cdot 1, [\,k_2 \mid \alpha_2 \cdot 2 \mapsto s_2\,])$

For example, if $\mu : (\{\alpha_1\} \cup id(k_1)) \to (\{\alpha_2\} \cup id(k_2))$ is a bijection that satisfies the properties given in Definition 5.3(b), we can prove the above isomorhism by defining a bijection μ' as follows:[6]

[6]Notice that:

$id([\,k_1 \mid \alpha_1 \cdot 2 \mapsto s_2\,]) = \{\alpha_1 \cdot 2\} \cup id(k_1)$
$id([\,k_2 \mid \alpha_2 \cdot 2 \mapsto s_2\,]) = \{\alpha_2 \cdot 2\} \cup id(k_2)$

$$\mu' : (\{\alpha_1 \cdot 1, \alpha_1 \cdot 2\} \cup id(k_1)) \to (\{\alpha_2 \cdot 1, \alpha_2 \cdot 2\} \cup id(k_2))$$
$$\mu'(\alpha_1 \cdot 1) = \alpha_2 \cdot 1$$
$$\mu'(\alpha_1 \cdot 2) = \alpha_2 \cdot 2$$
$$\mu'(\alpha) = \mu(\alpha), \text{ for any } \alpha \in id(k_1)$$

Therefore, as $\varsigma(s_1) < \varsigma(s_1 \lfloor\!\lfloor s_2)$, we can use the induction hypothesis and we infer that $\exists \overline{s} \in Stat, \overline{\sigma} \in \Sigma, (\overline{\alpha}_1, \overline{k}_1) \cong (\overline{\alpha}_2, \overline{k}_2) \in Conf$, such that

$$^{(5.5.b.3)} \leq \tfrac{1}{2} \cdot d([\![\overline{s}]\!](\overline{\alpha}_1, K(\overline{k}_1))(\overline{\sigma}), [\![\overline{s}]\!](\overline{\alpha}_2, K(\overline{k}_2))(\overline{\sigma}))$$

Case $[s = s_1 \parallel s_2]$

$$d([\![s_1 \parallel s_2]\!](\alpha_1, K(k_1))(\sigma), [\![s_1 \parallel s_2]\!](\alpha_2, K(k_2))(\sigma))$$
$$= d([\![s_1]\!](\alpha_1 \cdot 1, [\, K(k_1) \mid \alpha_1 \cdot 2 \mapsto [\![s_2]\!]\,])(\sigma) +$$
$$[\![s_2]\!](\alpha_1 \cdot 1, [\, K(k_1) \mid \alpha_1 \cdot 2 \mapsto [\![s_1]\!]\,])(\sigma),$$
$$[\![s_1]\!](\alpha_2 \cdot 1, [\, K(k_2) \mid \alpha_2 \cdot 2 \mapsto [\![s_2]\!]\,])(\sigma) +$$
$$[\![s_2]\!](\alpha_2 \cdot 1, [\, K(k_2) \mid \alpha_2 \cdot 2 \mapsto [\![s_1]\!]\,])(\sigma))$$

[Lemma 5.4(a), + is nonexpansive]

$$\leq max\{d([\![s_1]\!](\alpha_1 \cdot 1, K[\, k_1 \mid \alpha_1 \cdot 2 \mapsto s_2\,])(\sigma),$$
$$[\![s_1]\!](\alpha_2 \cdot 1, K[\, k_2 \mid \alpha_2 \cdot 2 \mapsto s_2\,])(\sigma))^{\ (5.5.b.4)}\ ,$$
$$d([\![s_2]\!](\alpha_1 \cdot 1, K[\, k_1 \mid \alpha_1 \cdot 2 \mapsto s_1\,])(\sigma),$$
$$[\![s_2]\!](\alpha_2 \cdot 1, K[\, k_2 \mid \alpha_2 \cdot 2 \mapsto s_1\,])(\sigma))^{\ (5.5.b.5)}\ \}$$

It is easy to check that $(\alpha_1, k_1) \cong (\alpha_2, k_2)$ implies

$$(\alpha_1 \cdot 1, [\, k_1 \mid \alpha_1 \cdot 2 \mapsto s_2\,]) \cong (\alpha_2 \cdot 1, [\, k_2 \mid \alpha_2 \cdot 2 \mapsto s_2\,])$$
$$(\alpha_1 \cdot 1, [\, k_1 \mid \alpha_1 \cdot 2 \mapsto s_1\,]) \cong (\alpha_2 \cdot 1, [\, k_2 \mid \alpha_2 \cdot 2 \mapsto s_1\,])$$

For the first isomorphism we can define a bijection exactly in the same way as we did for the subcase $[s = s_1 \lfloor\!\lfloor s_2]$. As $\varsigma(s_1) < \varsigma(s_1 \parallel s_2)$, we can use the induction hypothesis for $^{(5.5.b.4)}$ and we infer that $\exists \overline{s} \in Stat, (\overline{\alpha}_1, \overline{k}_1) \cong (\overline{\alpha}_2, \overline{k}_2) \in Conf$ and $\overline{\sigma} \in \Sigma$ such that:

$$^{(5.5.b.4)} \leq \tfrac{1}{2} \cdot d([\![\overline{s}]\!](\overline{\alpha}_1, K(\overline{k}_1))(\overline{\sigma}), [\![\overline{s}]\!](\overline{\alpha}_2, K(\overline{k}_2))(\overline{\sigma}))$$

$^{(5.5.b.5)}$ can be handled in a similar manner and the desired result follows immediately. ■

Corollary 5.6

(a) For all $s \in Stat, (\alpha_1, k_1) \cong (\alpha_2, k_2)\ (\in Conf)$: $[\![s]\!](\alpha_1, K(k_1)) = [\![s]\!](\alpha_2, K(k_2))$.

(b) For all $k_1 \cong k_2\ (\in Konf)$: $kc(K(k_1)) = kc(K(k_2))$.

Proof: Let

$$(w \in)W = \{\ (s, (\alpha_1, k_1), (\alpha_2, k_2), \sigma)$$
$$\mid s \in Stat, (\alpha_1, k_1), (\alpha_2, k_2) \in Conf : (\alpha_1, k_1) \cong (\alpha_2, k_2), \sigma \in \Sigma\}$$

For $(s, (\alpha_1, k_1), (\alpha_2, k_2), \sigma) \in W$ we use the notation:

$$\varepsilon_I(s, (\alpha_1, k_1), (\alpha_2, k_2), \sigma) \overset{not.}{=} d([\![s]\!](\alpha_1, K(k_1))(\sigma), [\![s]\!](\alpha_2, K(k_2))(\sigma))$$

Let $(s, (\alpha_1, k_1), (\alpha_2, k_2), \sigma) \in W$. By 5.5(b) there exists $(\overline{s}, (\overline{\alpha}_1, \overline{k}_1), (\overline{\alpha}_2, \overline{k}_2), \overline{\sigma}) \in W$ such that:

$$\varepsilon_I(s, (\alpha_1, k_1), (\alpha_2, k_2), \sigma) \leq \tfrac{1}{2} \cdot \varepsilon_I(\overline{s}, (\overline{\alpha}_1, \overline{k}_1), (\overline{\alpha}_2, \overline{k}_2), \overline{\sigma})$$

and thus $\sup_{w \in W} \varepsilon_I(w) \leq \frac{1}{2} \cdot \sup_{\overline{w} \in W} \varepsilon_I(\overline{w})$, where $w = (s, (\alpha_1, k_1), (\alpha_2, k_2), \sigma)$ and $\overline{w} = (\overline{s}, (\overline{\alpha}_1, \overline{k}_1), (\overline{\alpha}_2, \overline{k}_2), \overline{\sigma})$. This means that we have $\sup_{w \in W} \varepsilon_I(w) = 0$, i.e. $d([\![s]\!](\alpha_1, K(k_1))(\sigma), [\![s]\!](\alpha_2, K(k_2))(\sigma)) = 0$ and thus $[\![s]\!](\alpha_1, K(k_1))(\sigma) = [\![s]\!](\alpha_2, K(k_2))(\sigma)$ for any $\sigma \in \Sigma$, which implies Corollary 5.6(a). Corollary 5.6(b) follows immediately from Lemma 5.5(a) and Corollary 5.6(a). ■

5.2 Continuations and syntactic contexts

We show that in the CSC approach continuations can be used to reason in a compositional manner upon the behavior of concurrent programs. For this purpose we introduce a notion of syntactic context for the class of $\mathcal{L}$ statements.

Definition 5.7 *(Contexts for $\mathcal{L}$)*

$$C ::= (\cdot) \mid a \mid x \mid C;C \mid C + C \mid C \lfloor\!\lfloor C \mid C \| C$$

We denote by $C(s)$ the result of substituting s for all occurences of $(\cdot)$ in C. Formally, this substitution can be defined inductively: $(\cdot)(s) = s, a(s) = a, x(s) = x$ and $(C_1 \ \mathsf{op}\ C_2)(s) = C_1(s) \ \mathsf{op}\ C_2(s)$, where $\mathsf{op} \in \{;, +, \lfloor\!\lfloor, \|\}$.

Lemma 5.8 shows that program properties are preserved in any syntactic context by all CSC continuations containing only denotations of statements. The proof relies on an auxiliarry Lemma 5.9 which involves again the identification of an appropriate computing invariant and the use of contraction $\varepsilon \leq \frac{1}{2} \cdot \varepsilon \Rightarrow \varepsilon = 0$.

Lemma 5.8 *If $s_1, s_2 \in Stat$ are such that for all $(\alpha, k) \in Conf$:*

$$[\![s_1]\!](\alpha, K(k)) = [\![s_2]\!](\alpha, K(k))$$

then for all $(\alpha, k) \in Conf$ and for all contexts C:

$$[\![C(s_1)]\!](\alpha, K(k)) = [\![C(s_2)]\!](\alpha, K(k))$$

Proof: By structural induction on C. Cases $[C = a]$ and $[C = x]$ are trivial. The case $[C = (\cdot)]$ follows by the assumption. A non-trivial case is the context for parallel composition $[C = C_1 \parallel C_2]$. Let $(\alpha, k) \in \mathit{Conf}$ and $\sigma \in \Sigma$. We have to prove that $[\![(C_1 \parallel C_2)(s_1)]\!](\alpha, K(k))(\sigma) = [\![(C_1 \parallel C_2)(s_2)]\!](\alpha, K(k))(\sigma)$. We compute as follows:

$$[\![(C_1 \parallel C_2)(s_1)]\!](\alpha, K(k))(\sigma) = [\![C_1(s_1) \parallel C_2(s_1)]\!](\alpha, K(k))(\sigma)$$

$$= [\![C_1(s_1)]\!](\alpha \cdot 1, [\, K(k) \mid \alpha \cdot 2 \mapsto [\![C_2(s_1)]\!]\,])(\sigma)\ ^{(5.8.1)} \quad +$$

$$[\![C_2(s_1)]\!](\alpha \cdot 1, [\, K(k) \mid \alpha \cdot 2 \mapsto [\![C_1(s_1)]\!]\,])(\sigma)\ ^{(5.8.2)}$$

We handle $^{(5.8.1)}$ first.

$$^{(5.8.1)} = [\![C_1(s_1)]\!](\alpha \cdot 1, [\, K(k) \mid \alpha \cdot 2 \mapsto [\![C_2(s_1)]\!]\,])(\sigma) \quad \text{[Lemma 5.4(a)]}$$

$$= [\![C_1(s_1)]\!](\alpha \cdot 1, K[\, k \mid \alpha \cdot 2 \mapsto (C_2(s_1))\,])(\sigma) \quad \text{[Ind. hyp.]}$$

$$= [\![C_1(s_2)]\!](\alpha \cdot 1, K[\, k \mid \alpha \cdot 2 \mapsto (C_2(s_1))\,])(\sigma) \quad \text{[Lemma 5.4(a)]}$$

$$= [\![C_1(s_2)]\!](\alpha \cdot 1, [\, K(k) \mid \alpha \cdot 2 \mapsto [\![C_2(s_1)]\!]\,])(\sigma)$$

$(\alpha, k) \in \mathit{Conf}$ implies $(\alpha \cdot 1, k) \in \mathit{Conf}$. By the induction hypothesis:

$$[\![C_2(s_1)]\!](\alpha \cdot 1, K(k))(\sigma) = [\![C_2(s_2)]\!](\alpha \cdot 1, K(k))(\sigma)$$

Hence, by Lemma 5.9(b) we have:

$$[\![C_1(s_2)]\!](\alpha \cdot 1, [\, K(k) \mid \alpha \cdot 2 \mapsto [\![C_2(s_1)]\!]\,])(\sigma)$$

$$= [\![C_1(s_2)]\!](\alpha \cdot 1, [\, K(k) \mid \alpha \cdot 2 \mapsto [\![C_2(s_2)]\!]\,])(\sigma)$$

Similarly:

$$^{(5.8.2)} = [\![C_2(s_1)]\!](\alpha \cdot 1, [\, K(k) \mid \alpha \cdot 2 \mapsto [\![C_1(s_1)]\!]\,])(\sigma)$$

$$= [\![C_2(s_2)]\!](\alpha \cdot 1, [\, K(k) \mid \alpha \cdot 2 \mapsto [\![C_1(s_2)]\!]\,])(\sigma)$$

Therefore:

$$[\![(C_1 \parallel C_2)(s_1)]\!](\alpha, K(k))(\sigma) = {}^{(5.8.1)} + {}^{(5.8.2)}$$

$$= [\![C_1(s_2)]\!](\alpha \cdot 1, [\, K(k) \mid \alpha \cdot 2 \mapsto [\![C_2(s_2)]\!]\,])(\sigma) +$$

$$[\![C_2(s_2)]\!](\alpha \cdot 1, [\, K(k) \mid \alpha \cdot 2 \mapsto [\![C_1(s_2)]\!]\,])(\sigma)$$

$$= [\![C_1(s_2) \parallel C_2(s_2)]\!](\alpha, K(k))(\sigma) = [\![(C_1 \parallel C_2)(s_2)]\!](\alpha, K(k))(\sigma)$$

■

Lemma 5.9 *If $s_1, s_2 \in \mathit{Stat}$ are such that $[\![s_1]\!](\alpha, K(k))(\sigma) = [\![s_2]\!](\alpha, K(k))(\sigma)$ for all $(\alpha, k) \in \mathit{Conf}, \sigma \in \Sigma$ then* *(a) for all* $k \in Konf, \overline{\alpha} \in Id, \sigma \in \Sigma$

$$kc[\,K(k) \mid \overline{\alpha} \mapsto [\![s_1]\!]\,](\sigma) = kc[\,K(k) \mid \overline{\alpha} \mapsto [\![s_2]\!]\,](\sigma)$$

(b) for all $s \in Stat, (\alpha, k) \in Conf, \sigma \in \Sigma, \overline{\alpha} \in Id, \neg(\overline{\alpha} \geq \alpha)$*:*

$$[\![s]\!](\alpha, [\,K(k) \mid \overline{\alpha} \mapsto [\![s_1]\!]\,])(\sigma) = [\![s]\!](\alpha, [\,K(k) \mid \overline{\alpha} \mapsto [\![s_2]\!]\,])(\sigma)$$

Proof: For (a) we show that $\forall k \in Konf, \sigma \in \Sigma, \overline{\alpha} \in Id, \quad \exists s' \in Stat, (\alpha', k') \in Conf$ such that $\neg(\overline{\alpha} \geq \alpha')$ and:

$$\begin{aligned}
&d(kc[\,K(k) \mid \overline{\alpha} \mapsto [\![s_1]\!]\,](\sigma), kc[\,K(k) \mid \overline{\alpha} \mapsto [\![s_2]\!]\,](\sigma))\\
&\leq d([\![s']\!](\alpha', [\,K(k') \mid \overline{\alpha} \mapsto [\![s_1]\!]\,])(\sigma), [\![s']\!](\alpha', [\,K(k') \mid \overline{\alpha} \mapsto [\![s_2]\!]\,])(\sigma))\\
&\overset{not.}{=} \ \varepsilon_c(s', (\alpha', k'), \overline{\alpha}, s_1, s_2, \sigma) \ ^{(5.9.1)}
\end{aligned}$$

For (b) we show that $\forall s \in Stat, (\alpha, k) \in Conf, \overline{\alpha} \in Id, \sigma \in \Sigma$ with $\neg(\overline{\alpha} \geq \alpha), \quad \exists s' \in Stat, (\alpha', k') \in Conf, \sigma' \in \Sigma$ such that $\neg(\overline{\alpha} \geq \alpha')$ and:[7]

$$\begin{aligned}
&\varepsilon_c(s, (\alpha, k), \overline{\alpha}, s_1, s_2, \sigma)\\
&= d([\![s]\!](\alpha, [\,K(k) \mid \overline{\alpha} \mapsto [\![s_1]\!]\,])(\sigma), [\![s]\!](\alpha, [\,K(k) \mid \overline{\alpha} \mapsto [\![s_2]\!]\,])(\sigma))\\
&\leq \tfrac{1}{2} \cdot \ d([\![s']\!](\alpha', [\,K(k') \mid \overline{\alpha} \mapsto [\![s_1]\!]\,])(\sigma'), [\![s']\!](\alpha', [\,K(k') \mid \overline{\alpha} \mapsto [\![s_2]\!]\,])(\sigma'))\\
&= \tfrac{1}{2} \cdot \ \ \varepsilon_c(s', (\alpha', k'), \overline{\alpha}, s_1, s_2, \sigma') \ ^{(5.9.2)}
\end{aligned}$$

The invariant property that is preserved by the computation steps is given here by the condition that $(\alpha, k) \in Conf$ and $\neg(\overline{\alpha} \geq \alpha)$.

Let

$$\begin{aligned}
(w \in)W = \{&(s, (\alpha, k), \overline{\alpha}, s_1, s_2, \sigma)\\
&\mid s, s_1, s_2 \in Stat, (\alpha, k) \in Conf, \overline{\alpha} \in Id, \neg(\overline{\alpha} \geq \alpha), \sigma \in \Sigma\}
\end{aligned}$$

We infer that

$$sup_{w \in W} \ \varepsilon_c(w) \leq \tfrac{1}{2} \cdot \ sup_{w' \in W} \ \varepsilon_c(w')$$

where $w = (s, (\alpha, k), \overline{\alpha}, s_1, s_2, \sigma), w' = (s', (\alpha', k'), \overline{\alpha}, s_1, s_2, \sigma')$. But

$$sup_{w \in W} \ \varepsilon_c(w) = sup_{w' \in W} \ \varepsilon_c(w')$$

Therefore we infer $sup_{w \in W} \ \varepsilon_c(w) = 0$, which implies Lemma 5.9(b). Next, by using this result and $^{(5.9.1)}$ we obtain immediately Lemma 5.9(a). In the sequel we prove $^{(5.9.1)}$ and $^{(5.9.2)}$.

First, we handle $^{(5.9.1)}$. Let $\kappa_1, \kappa_2 \in \mathbf{Kont}$, $k_1, k_2 \in Kont$

[7] $\neg(\overline{\alpha} \geq \alpha)$ is the invariant property that is preserved by the computation steps.

$$\kappa_1 = [\, K(k) \mid \overline{\alpha} \mapsto [\![s_1]\!] \,]$$

$$\kappa_2 = [\, K(k) \mid \overline{\alpha} \mapsto [\![s_2]\!] \,]$$

$$k_1 = [\, k \mid \overline{\alpha} \mapsto s_1 \,]$$

$$k_2 = [\, k \mid \overline{\alpha} \mapsto s_2 \,]$$

By Lemma 5.4(a), $\kappa_1 = K(k_1), \kappa_2 = K(k_2)$. Obviously, $id(k_1) = id(k_2) = id(k) \cup \{\overline{\alpha}\}$. Also, by Lemma 5.4(b), $id(\kappa_1) = id(k_1) = id(k_2) = id(\kappa_2) = id(k) \cup \{\overline{\alpha}\}$. We distinguish two subcases. If $\overline{\alpha} \notin max(id(k) \cup \{\overline{\alpha}\})$ we compute as follows:

$$\begin{aligned}
& d(kc(\kappa_1)(\sigma), kc(\kappa_2)(\sigma)) \\
& = d(+_{\alpha \in max(id(k))} \kappa_1(\alpha)(\alpha, \kappa_1 \setminus \alpha)(\sigma), +_{\alpha \in max(id(k))} \kappa_2(\alpha)(\alpha, \kappa_2 \setminus \alpha)(\sigma)) \\
& \qquad [+ \text{ is nonexpansive}] \\
& = max\{d(\kappa_1(\alpha)(\alpha, \kappa_1 \setminus \alpha)(\sigma), \kappa_2(\alpha)(\alpha, \kappa_2 \setminus \alpha)(\sigma)) \\
& \qquad \mid \alpha \in max(id(k))\} \;^{(5.9.3)}
\end{aligned}$$

By Lemma 5.4(a) and 5.4(d)

$$\kappa_1 \setminus \alpha = K(k_1) \setminus \alpha = K(k_1 \setminus \alpha) = K[\, k \setminus \alpha \mid \overline{\alpha} \mapsto s_1 \,] = [\, K(k \setminus \alpha) \mid \overline{\alpha} \mapsto [\![s_1]\!] \,]$$

$$\kappa_2 \setminus \alpha = K(k_2) \setminus \alpha = K(k_2 \setminus \alpha) = K[\, k \setminus \alpha \mid \overline{\alpha} \mapsto s_2 \,] = [\, K(k \setminus \alpha) \mid \overline{\alpha} \mapsto [\![s_2]\!] \,]$$

Also, notice that for any $\alpha \in max(id(k))$: $\kappa_1(\alpha) = \kappa_2(\alpha) = [\![k(\alpha)]\!]$. Therefore we have:

$$\begin{aligned}
^{(5.9.3)} = max\{ & d([\![k(\alpha)]\!](\alpha, [\, K(k \setminus \alpha) \mid \overline{\alpha} \mapsto [\![s_1]\!] \,])(\sigma), \\
& [\![k(\alpha)]\!](\alpha, [\, K(k \setminus \alpha) \mid \overline{\alpha} \mapsto [\![s_2]\!] \,])(\sigma)) \\
& \mid \alpha \in max(id(k))\}
\end{aligned}$$

Clearly, this means that $\exists s' \in Stat, (\alpha', k') \in Conf$ with $\neg(\overline{\alpha} \geq \alpha')$ such that:[8]

$$\begin{aligned}
& d(kc(\kappa_1)(\sigma), kc(\kappa_2)(\sigma)) \\
& \leq d([\![s']\!](\alpha', [\, K(k') \mid \overline{\alpha} \mapsto [\![s_1]\!] \,])(\sigma), [\![s']\!](\alpha', [\, K(k') \mid \overline{\alpha} \mapsto [\![s_2]\!] \,])(\sigma))
\end{aligned}$$

Next, we treat the subcase $\overline{\alpha} \in (max(id(k)) \cup \{\overline{\alpha}\})$. In this subcase:

$$\begin{aligned}
& d(kc(\kappa_1)(\sigma), kc(\kappa_2)(\sigma)) \\
& = d((+_{\alpha \in max(id(k)), \neg(\overline{\alpha} \geq \alpha)} \kappa_1(\alpha)(\alpha, \kappa_1 \setminus \alpha)(\sigma)) + [\![s_1]\!](\overline{\alpha}, \kappa_1 \setminus \overline{\alpha})(\sigma),
\end{aligned}$$

[8] More precisely, for some $\alpha \in max(id(k))$, $s' = k(\alpha), \alpha' = \alpha, k' = k \setminus \alpha$.

$$(+_{\alpha\in max(id(k)),\neg(\overline{\alpha}\geq\alpha)}\kappa_2(\alpha)(\alpha,\kappa_2\setminus\alpha)(\sigma)) + [\![s_2]\!](\overline{\alpha},\kappa_2\setminus\overline{\alpha})(\sigma))$$

$[+$ is nonexpansive, $\kappa_1 = K(k_1), \kappa_2 = K(k_2), k_1\setminus\overline{\alpha}\cong k\cong k_2\setminus\overline{\alpha}$,

Corollary 5.6(a)$]$

$$max\{d([\![s_1]\!](\overline{\alpha},K(k))(\sigma),[\![s_2]\!](\overline{\alpha},K(k))(\sigma)),$$
$$max\{d(\kappa_1(\alpha)(\alpha,\kappa_1\setminus\alpha)(\sigma),\kappa_2(\alpha)(\alpha,\kappa_2\setminus\alpha)(\sigma))$$
$$\mid \alpha\in max(id(k)),\neg(\overline{\alpha}\geq\alpha)\}\}\ ^{(5.9.4)}$$

By assumption $[\![s_1]\!](\overline{\alpha},K(k))(\sigma) = [\![s_2]\!](\overline{\alpha},K(k))(\sigma)$. Also, for any $\alpha\in max(id(k))$, $\neg(\overline{\alpha}\geq\alpha) : \kappa_1(\alpha) = \kappa_2(\alpha) = k(\alpha)$, and $\kappa_1\backslash\alpha = [\,K(k\backslash\alpha)\mid\overline{\alpha}\mapsto[\![s_1]\!]\,]$, $\kappa_2\backslash\alpha = [\,K(k\backslash\alpha)\mid\overline{\alpha}\mapsto[\![s_2]\!]\,]$. Therefore:

$$^{(5.9.4)} = max\{d([\![k(\alpha)]\!](\alpha,[\,K(k\setminus\alpha)\mid\overline{\alpha}\mapsto[\![s_1]\!]\,])(\sigma),$$
$$[\![k(\alpha)]\!](\alpha,[\,K(k\setminus\alpha)\mid\overline{\alpha}\mapsto[\![s_2]\!]\,])(\sigma))$$
$$\mid \alpha\in max(id(k)),\neg(\overline{\alpha}\geq\alpha)\}$$

Clearly, this means that $\exists s'\in Stat,(\alpha',k')\in Conf$ with $\neg(\overline{\alpha}\geq\alpha')$ such that:

$$d(kc(\kappa_1)(\sigma),kc(\kappa_2)(\sigma))$$
$$\leq d([\![s']\!](\alpha',[\,K(k')\mid\overline{\alpha}\mapsto[\![s_1]\!]\,])(\sigma),[\![s']\!](\alpha',[\,K(k')\mid\overline{\alpha}\mapsto[\![s_2]\!]\,])(\sigma))$$

This concludes the proof of 5.9(1).

In the sequel we prove $^{(5.9.2)}$. More precisely, we show that $\forall s\in Stat,(\alpha,k)\in Conf,\overline{\alpha}\in Id$ with $\neg(\overline{\alpha}\geq\alpha)$ and $\sigma\in\Sigma$, $\exists s'\in Stat,(\alpha',k')\in Conf$ with $\neg(\overline{\alpha}\geq\alpha')$ and $\sigma'\in\Sigma$ such that:

$$\varepsilon_c(s,(\alpha,k),\overline{\alpha},s_1,s_2,\sigma)\leq\tfrac{1}{2}\cdot\ \varepsilon_c(s',(\alpha',k'),\overline{\alpha},s_1,s_2,\sigma')$$

We proceed by induction on $\varsigma(s)$. Two subcases.

Case $[s=a]$ when $I(a)(\sigma)=\sigma'\in\Sigma$.

$$\varepsilon_c(a,(\alpha,k),\overline{\alpha},s_1,s_2,\sigma)$$
$$= d([\![a]\!](\alpha,[\,K(k)\mid\overline{\alpha}\mapsto[\![s_1]\!]\,])(\sigma),[\![a]\!](\alpha,[\,K(k)\mid\overline{\alpha}\mapsto[\![s_2]\!]\,])(\sigma))$$
$$= d(\sigma'\cdot kc([\,K(k)\mid\overline{\alpha}\mapsto[\![s_1]\!]\,])(\sigma'),\sigma'\cdot kc([\,K(k)\mid\overline{\alpha}\mapsto[\![s_2]\!]\,])(\sigma'))\ ^{(5.9.5)}$$

By $^{(5.9.1)}$ $\exists s'\in Stat,(\alpha',k')\in Conf$ with $\neg(\overline{\alpha}\geq\alpha')$ such that:

$$^{(5.9.5)}\ \leq\tfrac{1}{2}\cdot\ d([\![s']\!](\alpha',[\,K(k')\mid\overline{\alpha}\mapsto[\![s_1]\!]\,])(\sigma'),$$
$$[\![s']\!](\alpha',[\,K(k')\mid\overline{\alpha}\mapsto[\![s_2]\!]\,])(\sigma'))$$

$= \frac{1}{2} \cdot \ \varepsilon_c(s', (\alpha', k'), \overline{\alpha}, s_1, s_2, \sigma')$

Case $[s = s^1 \parallel s^2]$

$$\begin{aligned}
&\varepsilon_c(s^1 \parallel s^2, (\alpha, k), \overline{\alpha}, s_1, s_2, \sigma)\\
&= d([\![s^1 \parallel s^2]\!](\alpha, [\,K(k) \mid \overline{\alpha} \mapsto [\![s_1]\!]\,])(\sigma),\\
&\quad [\![s^1 \parallel s^2]\!](\alpha, [\,K(k) \mid \overline{\alpha} \mapsto [\![s_2]\!]\,])(\sigma))\\
&= d([\![s^1]\!](\alpha \cdot 1, [\,K(k) \mid \overline{\alpha} \mapsto [\![s_1]\!] \mid \alpha \cdot 2 \mapsto [\![s^2]\!]\,])(\sigma) +\\
&\quad [\![s^2]\!](\alpha \cdot 1, [\,K(k) \mid \overline{\alpha} \mapsto [\![s_1]\!] \mid \alpha \cdot 2 \mapsto [\![s^1]\!]\,])(\sigma),\\
&\quad [\![s^1]\!](\alpha \cdot 1, [\,K(k) \mid \overline{\alpha} \mapsto [\![s_2]\!] \mid \alpha \cdot 2 \mapsto [\![s^2]\!]\,])(\sigma) +\\
&\quad [\![s^2]\!](\alpha \cdot 1, [\,K(k) \mid \overline{\alpha} \mapsto [\![s_2]\!] \mid \alpha \cdot 2 \mapsto [\![s^1]\!]\,])(\sigma))\\
&\quad [\text{'+' is nonexpansive; } \neg(\overline{\alpha} \geq \alpha) \Rightarrow \neg(\overline{\alpha} \geq \alpha \cdot 1), \neg(\overline{\alpha} \geq \alpha \cdot 2);\\
&\quad\ \text{Lemma 5.4(a)}]\\
&\leq \mathit{max}\{d([\![s^1]\!](\alpha \cdot 1, [\,K[\,k \mid \alpha \cdot 2 \mapsto s^2\,] \mid \overline{\alpha} \mapsto [\![s_1]\!]\,])(\sigma),\\
&\qquad [\![s^1]\!](\alpha \cdot 1, [\,K[\,k \mid \alpha \cdot 2 \mapsto s^2\,] \mid \overline{\alpha} \mapsto [\![s_2]\!]\,])(\sigma))\ ^{(5.9.6)}\\
&\qquad d([\![s^2]\!](\alpha \cdot 1, [\,K[\,k \mid \alpha \cdot 2 \mapsto s^1\,] \mid \overline{\alpha} \mapsto [\![s_1]\!]\,])(\sigma),\\
&\qquad [\![s^2]\!](\alpha \cdot 1, [\,K[\,k \mid \alpha \cdot 2 \mapsto s^1\,] \mid \overline{\alpha} \mapsto [\![s_2]\!]\,])(\sigma))\ ^{(5.9.7)}\ \}
\end{aligned}$$

As $\neg(\overline{\alpha} \geq \alpha \cdot 1)$ and $\neg(\overline{\alpha} \geq \alpha \cdot 2)$, we can apply the induction hypothesis and we infer that $\exists s'_1 \in \mathit{Stat}, (\alpha'_1, k'_1) \in \mathit{Conf}$ with $\neg(\overline{\alpha} \geq \alpha'_1)$ and $\sigma'_1 \in \Sigma$, and $\exists s'_2 \in \mathit{Stat}, (\alpha'_2, k'_2) \in \mathit{Conf}$ with $\neg(\overline{\alpha} \geq \alpha'_2)$ and $\sigma'_2 \in \Sigma$, such that:

$$\begin{aligned}
^{(5.9.6)}\ &\leq \tfrac{1}{2} \cdot d([\![s'_1]\!](\alpha'_1, [\,K(k'_1) \mid \overline{\alpha} \mapsto [\![s_1]\!]\,])(\sigma'_1),\\
&\qquad [\![s'_1]\!](\alpha'_1, [\,K(k'_1) \mid \overline{\alpha} \mapsto [\![s_2]\!]\,])(\sigma'_1))\ ^{(5.9.6')}
\end{aligned}$$

$$\begin{aligned}
^{(5.9.7)}\ &\leq \tfrac{1}{2} \cdot d([\![s'_2]\!](\alpha'_2, [\,K(k'_2) \mid \overline{\alpha} \mapsto [\![s_1]\!]\,])(\sigma'_2),\\
&\qquad [\![s'_2]\!](\alpha'_2, [\,K(k'_2) \mid \overline{\alpha} \mapsto [\![s_2]\!]\,])(\sigma'_2))\ ^{(5.9.7')}
\end{aligned}$$

Finally, we obtain:

$$\begin{aligned}
&\varepsilon_c(s^1 \parallel s^2, (\alpha, k), \overline{\alpha}, s_1, s_2, \sigma)\\
&\leq \mathit{max}\{\ ^{(5.9.6')}\ ,\ ^{(5.9.7')}\ \}\\
&= \mathit{max}\{\tfrac{1}{2} \cdot \ \varepsilon_c(s'_1, (\alpha'_1, k'_1), \overline{\alpha}, s_1, s_2, \sigma'_1), \tfrac{1}{2} \cdot \ \varepsilon_c(s'_2, (\alpha'_2, k'_2), \overline{\alpha}, s_1, s_2, \sigma'_2)\}
\end{aligned}$$

This implies immediately the desired result. ■

5.3 Concurrency laws

This section concludes with Theorem 5.12, which presents the main results of the paper. Theorem 5.12 allows us to reason in a compositional manner upon the behavior of $\mathcal{L}$ asynchronous programs. The denotational semantics $[\![\cdot]\!]$ preserves the following invariant property: continuations contain only computations denotable by program statements. The initial continuation (α_0, κ_0) (see Definition 4.1(c)) is empty (contains no computations) and each equation in the definition of $[\![\cdot]\!]$ adds only denotations of statements to the continuation. The properties given in Theorem 5.12 hold for continuations containing *only* computations denotable by program statements, which is sufficient in practice. The proof of Theorem 5.12 uses some auxiliary results given as Lemma 5.10 and Lemma 5.11. Essentially, Lemma 5.10 and Lemma 5.11 identify (non-isomorphic) continuation structures - specific of sequential composition and parallel composition, respectively - that behave the same.

Lemma 5.10 *For all $\tilde{s}, s_1, s_2 \in Stat, \sigma \in \Sigma, \tilde{\alpha}, \alpha \in Id, k \in Konf$ such that $(\tilde{\alpha}, k) \in Conf, (\alpha, k) \in Conf$, $(\tilde{\alpha} \neq \alpha), \neg(\tilde{\alpha} \leq \alpha)$ and $\neg(\alpha \leq \tilde{\alpha})$ we have:*

(a) $kc[\, K(k) \mid \alpha \mapsto [\![s_1 \parallel s_2]\!]\,](\sigma) = kc[\, K(k) \mid \alpha \cdot 1 \mapsto [\![s_1]\!] \mid \alpha \cdot 2 \mapsto [\![s_2]\!]\,](\sigma)$

(b) $[\![\tilde{s}]\!](\tilde{\alpha}, [\, K(k) \mid \alpha \mapsto [\![s_1 \parallel s_2]\!]\,])(\sigma) =$

$[\![\tilde{s}]\!](\tilde{\alpha}, [\, K(k) \mid \alpha \cdot 1 \mapsto [\![s_1]\!] \mid \alpha \cdot 2 \mapsto [\![s_2]\!]\,])(\sigma)$

Proof: We use the notation $Q(\tilde{\alpha}, \alpha)$, for $\tilde{\alpha}, \alpha \in Id$, to express the fact that $\tilde{\alpha}$ and α are different[9] and incomparable with respect to '$\leq$'.

$$Q(\tilde{\alpha}, \alpha) \overset{not.}{=} (\neg(\tilde{\alpha} \leq \alpha)) \wedge (\neg(\alpha \leq \tilde{\alpha}))$$

Also, we use the notation

$$P_{\parallel}(\tilde{\alpha}, \alpha, k) \overset{not.}{=} Q(\tilde{\alpha}, \alpha) \wedge ((\tilde{\alpha}, k) \in Conf) \wedge ((\alpha, k) \in Conf)$$

$P_{\parallel}$ is the invariant property which is preserved by the computation steps.

We proceed as follows. For 5.10(a) we show that $\forall s_1, s_2 \in Stat, \sigma \in \Sigma, \alpha \in Id, k \in Konf$ with $(\alpha, k) \in Conf$, $\exists s' \in Stat, \alpha' \in Id, k' \in Konf$ with $P_{\parallel}(\alpha', \alpha, k')$ such that:

$$\begin{aligned}
& d(kc[\, K(k) \mid \alpha \mapsto [\![s_1 \parallel s_2]\!]\,](\sigma), kc[\, K(k) \mid \alpha \cdot 1 \mapsto [\![s_1]\!] \mid \alpha \cdot 2 \mapsto [\![s_2]\!]\,](\sigma)) \\
& \leq d([\![s']\!](\alpha', [\, K(k') \mid \alpha \mapsto [\![s_1 \parallel s_2]\!]\,])(\sigma), \\
& \qquad [\![s']\!](\alpha', [\, K(k') \mid \alpha \cdot 1 \mapsto [\![s_1]\!] \mid \alpha \cdot 2 \mapsto [\![s_2]\!]\,])(\sigma)) \quad (5.10.1) \\
& \overset{not.}{=} \; \varepsilon_{\parallel}(s', \alpha', k', \alpha, s_1, s_2, \sigma)
\end{aligned}$$

Also, for 5.10(b) we show that $\forall \tilde{s}, s_1, s_2 \in Stat, \sigma \in \Sigma, \tilde{\alpha}, \alpha \in Id, k \in Konf$ such that $P_{\parallel}(\tilde{\alpha}, \alpha, k)$, $\exists s' \in Stat, \sigma' \in \Sigma, \alpha' \in Id, k' \in Konf$ with $P_{\parallel}(\alpha', \alpha, k')$ such that:

$$\begin{aligned}
& \varepsilon_{\parallel}(\tilde{s}, \tilde{\alpha}, k, \alpha, s_1, s_2, \sigma) \\
& = d([\![\tilde{s}]\!](\tilde{\alpha}, [\, K(k) \mid \alpha \mapsto [\![s_1 \parallel s_2]\!]\,])(\sigma), \\
& \qquad [\![\tilde{s}]\!](\tilde{\alpha}, [\, K(k) \mid \alpha \cdot 1 \mapsto [\![s_1]\!] \mid \alpha \cdot 2 \mapsto [\![s_2]\!]\,])(\sigma))
\end{aligned}$$

[9]Notice that $Q(\tilde{\alpha}, \alpha) \Rightarrow \tilde{\alpha} \neq \alpha$.

$$\leq \tfrac{1}{2} \cdot d([\![s']\!](\alpha', [\, K(k') \mid \alpha \mapsto [\![s_1 \parallel s_2]\!] \,])(\sigma'),$$
$$[\![s']\!](\alpha', [\, K(k') \mid \alpha \cdot 1 \mapsto [\![s_1]\!] \mid \alpha \cdot 2 \mapsto [\![s_2]\!] \,])(\sigma')) \quad {}^{(5.10.2)}$$
$$\overset{not.}{=} \varepsilon_{\parallel}(s', \alpha', k', \alpha, s_1, s_2, \sigma')$$

If we put $(w \in)W = Stat \times Id \times Conf \times Id \times Stat \times Stat \times \Sigma$ we infer that:

$$sup_{\ w \in W \,:\, P_{\parallel}(\tilde{\alpha}, \alpha, k)} \ \ \varepsilon_{\parallel}(w) \leq \tfrac{1}{2} \cdot sup_{\ w' \in W \,:\, P_{\parallel}(\alpha', \alpha, k')} \ \ \varepsilon_{\parallel}(w')$$

where $w = (\tilde{s}, \tilde{\alpha}, k, \alpha, s_1, s_2, \sigma)$ and $w' = (s', \alpha', k', \alpha, s_1, s_2, \sigma')$.
But obviously, $sup_{\ w \in W \,:\, P_{\parallel}(\tilde{\alpha}, \alpha, k)} \ \varepsilon_{\parallel}(w) = sup_{\ w' \in W \,:\, P_{\parallel}(\alpha', \alpha, k')} \ \varepsilon_{\parallel}(w')$, so

$$sup_{\ w \in W \,:\, P_{\parallel}(\tilde{\alpha}, \alpha, k)} \ \ \varepsilon_{\parallel}(w) = 0$$

which implies immediately 5.10(b). Next, by using this result and ${}^{(5.10.1)}$ we obtain immediately 5.10(a). In the sequel we prove ${}^{(5.10.1)}$ and ${}^{(5.10.2)}$.

First, we prove ${}^{(5.10.1)}$. Let $s_1, s_2 \in Stat, \sigma \in \Sigma, \alpha \in Id, k \in Konf$ with $(\alpha, k) \in Conf$. Let also $\kappa_1 = [\, K(k) \mid \alpha \mapsto [\![s_1 \parallel s_2]\!] \,]$, $\kappa_2 = [\, K(k) \mid \alpha \cdot 1 \mapsto [\![s_1]\!] \mid \alpha \cdot 2 \mapsto [\![s_2]\!] \,]$, $k_1 = [\, k \mid \alpha \mapsto (s_1 \parallel s_2) \,]$ and $k_2 = [\, k \mid \alpha \cdot 1 \mapsto s_1 \mid \alpha \cdot 2 \mapsto s_2 \,]$. By Lemma 5.4(a) $\kappa_1 = K(k_1), \kappa_2 = K(k_2)$. We compute as follows:

$$= d(kc(\kappa_1)(\sigma), kc(\kappa_2)(\sigma)) \qquad \text{[Lemma 5.4(b)]}$$
$$= d([\![s_1 \parallel s_2]\!](\alpha, [\, K(k) \mid \alpha \mapsto [\![s_1 \parallel s_2]\!] \,] \setminus \alpha)(\sigma) +$$
$$(+_{\alpha' \in max(id(k_1)), \alpha' \neq \alpha} \ \kappa_1(\alpha')(\alpha', \kappa_1 \setminus \alpha')(\sigma)),$$
$$[\![s_1]\!](\alpha \cdot 1, [\, K(k) \mid \alpha \cdot 1 \mapsto [\![s_1]\!] \mid \alpha \cdot 2 \mapsto [\![s_2]\!] \,] \setminus \alpha \cdot 1)(\sigma) +$$
$$[\![s_2]\!](\alpha \cdot 2, [\, K(k) \mid \alpha \cdot 1 \mapsto [\![s_1]\!] \mid \alpha \cdot 2 \mapsto [\![s_2]\!] \,] \setminus \alpha \cdot 2)(\sigma) +$$
$$(+_{\alpha' \in max(id(k_1)), \alpha' \neq \alpha \cdot 1, \alpha' \neq \alpha \cdot 2} \ \kappa_2(\alpha')(\alpha', \kappa_2 \setminus \alpha')(\sigma))) \quad {}^{(5.10.3)}$$

By Lemma 5.4(a) and 5.4(d)

$$[\, K(k) \mid \alpha \mapsto [\![s_1 \parallel s_2]\!] \,] \setminus \alpha = (K[\, k \mid \alpha \mapsto (s_1 \parallel s_2) \,]) \setminus \alpha$$
$$= K([\, k \mid \alpha \mapsto (s_1 \parallel s_2) \,] \setminus \alpha)$$

By Lemma 5.4(e), $[\, k \mid \alpha \mapsto (s_1 \parallel s_2) \,] \setminus \alpha \cong k$. Similarly,

$$[\, K(k) \mid \alpha \cdot 1 \mapsto [\![s_1]\!] \mid \alpha \cdot 2 \mapsto [\![s_2]\!] \,] \setminus \alpha \cdot 1$$
$$= K([\, k \mid \alpha \cdot 1 \mapsto s_1 \mid \alpha \cdot 2 \mapsto s_2 \,] \setminus \alpha \cdot 1)$$

and $[\, k \mid \alpha \cdot 1 \mapsto s_1 \mid \alpha \cdot 2 \mapsto s_2 \,] \setminus \alpha \cdot 1 \cong [\, k \mid \alpha \cdot 2 \mapsto s_2 \,]$. Also

$$[\, K(k) \mid \alpha \cdot 1 \mapsto [\![s_1]\!] \mid \alpha \cdot 2 \mapsto [\![s_2]\!] \,] \setminus \alpha \cdot 2$$

$$= K([\,k \mid \alpha\cdot 1 \mapsto s_1 \mid \alpha\cdot 2 \mapsto s_2\,] \setminus \alpha\cdot 2)$$

and $[\,k \mid \alpha\cdot 1 \mapsto s_1 \mid \alpha\cdot 2 \mapsto s_2\,] \setminus \alpha\cdot 2 \cong [\,k \mid \alpha\cdot 1 \mapsto s_1\,]$. In addition, it is easy to check that $(\alpha\cdot 2, [\,k \mid \alpha\cdot 1 \mapsto s_1\,]) \cong (\alpha\cdot 1, [\,k \mid \alpha\cdot 2 \mapsto s_1\,])$. Therefore, by Corollary 5.6(a) and taking into account that the semantic operator $+$ is nonexpansive we can compute as follows:

$$\begin{aligned}
{}^{(5.10.3)} \le max\{ & d([\![s_1 \parallel s_2]\!](\alpha, K(k))(\sigma),\\
& [\![s_1]\!](\alpha\cdot 1, [\,K(k) \mid \alpha\cdot 2 \mapsto [\![s_2]\!]\,])(\sigma) +\\
& [\![s_2]\!](\alpha\cdot 1, [\,K(k) \mid \alpha\cdot 2 \mapsto [\![s_1]\!]\,])(\sigma))\ {}^{(5.10.4)},\\
& d(+_{\alpha'\in max(id(k_1)),\alpha'\neq\alpha}\ \kappa_1(\alpha')(\alpha', \kappa_1 \setminus \alpha')(\sigma),\\
& +_{\alpha'\in max(id(k_1)),\alpha'\neq\alpha\cdot 1,\alpha'\neq\alpha\cdot 2}\ \kappa_2(\alpha')(\alpha', \kappa_2 \setminus \alpha')(\sigma))\ {}^{(5.10.5)}\ \}
\end{aligned}$$

Obviously, ${}^{(5.10.4)} = 0$ from the definition of the denotational mapping $[\![\cdot]\!]$. Next, we treat ${}^{(5.10.5)}$. As $(\alpha, k) \in Conf$ and $\alpha\cdot 1 \ge \alpha, \alpha\cdot 2 \ge \alpha$, it is easy to see that:

$$\begin{aligned}
&max(id(k_1)) = \{\alpha\} \cup \{\alpha' \mid \alpha' \in max(id(k)), Q(\alpha', \alpha)\}\\
&max(id(k_2)) = \{\alpha\cdot 1, \alpha\cdot 2\} \cup \{\alpha' \mid \alpha' \in max(id(k)), Q(\alpha', \alpha)\}
\end{aligned}$$

Therefore

$$\begin{aligned}
{}^{(5.10.5)} = \ & d(+_{\alpha'\in max(id(k)),Q(\alpha',\alpha)}\ \kappa_1(\alpha')(\alpha', \kappa_1 \setminus \alpha')(\sigma)\\
& +_{\alpha'\in max(id(k)),Q(\alpha',\alpha)}\ \kappa_2(\alpha')(\alpha', \kappa_2 \setminus \alpha')(\sigma)\\
& [+ \text{ is nonexpansive }]\\
\le\ & max\{d(\kappa_1(\alpha')(\alpha', \kappa_1 \setminus \alpha')(\sigma), \kappa_2(\alpha')(\alpha', \kappa_2 \setminus \alpha')(\sigma))\\
& \quad \mid \alpha' \in max(id(k)), Q(\alpha', \alpha)\}
\end{aligned}$$

This means that we have:

$$\begin{aligned}
& d(kc(\kappa_1)(\sigma), kc(\kappa_2)(\sigma)) \le max\{\ {}^{(5.10.4)}\ ,\ {}^{(5.10.5)}\ \} = max\{0,\ {}^{(5.10.5)}\ \}\\
& \le max\{d(\kappa_1(\alpha')(\alpha', \kappa_1 \setminus \alpha')(\sigma), \kappa_2(\alpha')(\alpha', \kappa_2 \setminus \alpha')(\sigma))\\
& \quad \mid \alpha' \in max(id(k)), Q(\alpha', \alpha)\}\ {}^{(5.10.6)}
\end{aligned}$$

We only treat the case when the above set (the argument of max in ${}^{(5.10.6)}$) is nonempty. Recall that $\kappa_1 = [\,K(k) \mid \alpha \mapsto [\![s_1 \parallel s_2]\!]\,]$ and $\kappa_2 = [\,K(k) \mid \alpha\cdot 1 \mapsto [\![s_1]\!] \mid \alpha\cdot 2 \mapsto [\![s_2]\!]\,]$, so

$${}^{(5.10.6)} = max\{d(K(k)(\alpha')(\alpha', [\,K(k) \mid \alpha \mapsto [\![s_1 \parallel s_2]\!]\,] \setminus \alpha')(\sigma)$$

$$K(k)(\alpha')(\alpha', [\, K(k) \mid \alpha \cdot 1 \mapsto [\![s_1]\!] \mid \alpha \cdot 2 \mapsto [\![s_2]\!] \,] \setminus \alpha')(\sigma))$$

$$\mid \alpha' \in max(id(k)), Q(\alpha', \alpha)\}$$

$$[\alpha' \in max(id(k)), (\alpha, k) \in Conf, Q(\alpha', \alpha) \Rightarrow$$

$$\alpha' \neq \alpha, \alpha' \neq \alpha \cdot 1, \alpha' \neq \alpha \cdot 2, \text{ Lemma 5.4(a), 5.4(d) and 5.4(f)}]$$

$$= max\{d([\![k(\alpha')]\!](\alpha', [\, K(k \setminus \alpha') \mid \alpha \mapsto [\![s_1 \parallel s_2]\!] \,])(\sigma),$$

$$[\![k(\alpha')]\!](\alpha', [\, K(k \setminus \alpha') \mid \alpha \cdot 1 \mapsto [\![s_1]\!] \mid \alpha \cdot 2 \mapsto [\![s_2]\!] \,])(\sigma))$$

$$\mid \alpha' \in max(id(k)), Q(\alpha', \alpha)\} \ ^{(5.10.7)}$$

As $(\alpha, k) \in Conf$, it is easy to check that $\forall \alpha' \in max(id(k))$ with $Q(\alpha', \alpha) : P_{\|}(\alpha', \alpha, k \setminus \alpha')$. Therefore, by taking the maximal element of $^{(5.10.7)}$ we obtain immediately the desired result $^{(5.10.1)}$, i.e. we infer that $\exists \overline{s}' \in Stat, \overline{\sigma}' \in \Sigma, \overline{\alpha}' \in Id, \overline{k}' \in Konf(\overline{k}' = k \setminus \overline{\alpha}', \overline{s}' = k(\overline{\alpha}'))$ satisfying the invariant property $P_{\|}(\overline{\alpha}', \alpha, \overline{k}')$ and such that

$$d(kc[\, K(k) \mid \alpha \mapsto [\![s_1 \parallel s_2]\!] \,](\sigma), kc[\, K(k) \mid \alpha \cdot 1 \mapsto [\![s_1]\!] \mid \alpha \cdot 2 \mapsto [\![s_2]\!] \,](\sigma))$$

$$\leq d([\![\overline{s}']\!](\overline{\alpha}', [\, K(\overline{k}') \mid \alpha \mapsto [\![s_1 \parallel s_2]\!] \,])(\sigma),$$

$$[\![\overline{s}']\!](\overline{\alpha}', [\, K(\overline{k}') \mid \alpha \cdot 1 \mapsto [\![s_1]\!] \mid \alpha \cdot 2 \mapsto [\![s_2]\!] \,])(\sigma))$$

which concludes the proof of $^{(5.10.1)}$.

Next, we prove $^{(5.10.2)}$. We proceed by induction on $\varsigma(\tilde{s})$ using $^{(5.10.1)}$. In the computations given below, by assumption $P_{\|}(\tilde{\alpha}, \alpha, k)$. Three subcases.

Case $[\tilde{s} = a]$ when $I(a)(\sigma) = \sigma' \in \Sigma$.

$$d([\![a]\!](\tilde{\alpha}, [\, K(k) \mid \alpha \mapsto [\![s_1 \parallel s_2]\!] \,])(\sigma),$$
$$[\![a]\!](\tilde{\alpha}, [\, K(k) \mid \alpha \cdot 1 \mapsto [\![s_1]\!] \mid \alpha \cdot 2 \mapsto [\![s_2]\!] \,])(\sigma))$$
$$= d(\sigma' \cdot kc[\, K(k) \mid \alpha \mapsto [\![s_1 \parallel s_2]\!] \,](\sigma'),$$
$$\sigma' \cdot kc[\, K(k) \mid \alpha \cdot 1 \mapsto [\![s_1]\!] \mid \alpha \cdot 2 \mapsto [\![s_2]\!] \,](\sigma'))$$
$$= \tfrac{1}{2} \cdot d(kc[\, K(k) \mid \alpha \mapsto [\![s_1 \parallel s_2]\!] \,](\sigma'),$$
$$kc[\, K(k) \mid \alpha \cdot 1 \mapsto [\![s_1]\!] \mid \alpha \cdot 2 \mapsto [\![s_2]\!] \,](\sigma')) \ ^{(5.10.8)}$$

As $(\alpha, k) \in Conf$, by $^{(5.10.1)}$, $\exists s' \in Stat, \alpha' \in Id, k' \in Konf$ such that $P_{\|}(\alpha', \alpha, k')$ and:

$$^{(5.10.8)} \leq \tfrac{1}{2} \cdot d([\![s']\!](\alpha', [\, K(k') \mid \alpha \mapsto [\![s_1 \parallel s_2]\!] \,])(\sigma'),$$
$$[\![s']\!](\alpha', [\, K(k') \mid \alpha \cdot 1 \mapsto [\![s_1]\!] \mid \alpha \cdot 2 \mapsto [\![s_2]\!] \,])(\sigma'))$$

Case $[\tilde{s} = x]$.

$$\begin{aligned}
& d([\![x]\!](\tilde{\alpha}, [\,K(k) \mid \alpha \mapsto [\![s_1 \parallel s_2]\!]\,])(\sigma),\\
& \quad [\![x]\!](\tilde{\alpha}, [\,K(k) \mid \alpha \cdot 1 \mapsto [\![s_1]\!] \mid \alpha \cdot 2 \mapsto [\![s_2]\!]\,])(\sigma))\\
& = d([\![D(x)]\!](\tilde{\alpha}, [\,K(k) \mid \alpha \mapsto [\![s_1 \parallel s_2]\!]\,])(\sigma),\\
& \qquad [\![D(x)]\!](\tilde{\alpha}, [\,K(k) \mid \alpha \cdot 1 \mapsto [\![s_1]\!] \mid \alpha \cdot 2 \mapsto [\![s_2]\!]\,])(\sigma))\ ^{(5.10.9)}
\end{aligned}$$

By the induction hypothesis ($\varsigma(D(x)) < \varsigma(x)$) $\exists s' \in Stat, \sigma' \in \Sigma, \alpha' \in Id, k' \in Konf$ such that $P_{\|}(\alpha', \alpha, k')$ and:

$$\begin{aligned}
{}^{(5.10.9)}\ \ & \leq \tfrac{1}{2} \cdot\ d([\![s']\!](\alpha', [\,K(k') \mid \alpha \mapsto [\![s_1 \parallel s_2]\!]\,])(\sigma'),\\
& \qquad [\![s']\!](\alpha', [\,K(k') \mid \alpha \cdot 1 \mapsto [\![s_1]\!] \mid \alpha \cdot 2 \mapsto [\![s_2]\!]\,])(\sigma'))
\end{aligned}$$

Case $[\tilde{s} = \tilde{s}^1 \| \tilde{s}^2]$.

$$\begin{aligned}
& d([\![\tilde{s}^1 \| \tilde{s}^2]\!](\tilde{\alpha}, [\,K(k) \mid \alpha \mapsto [\![s_1 \parallel s_2]\!]\,])(\sigma),\\
& \quad [\![\tilde{s}^1 \| \tilde{s}^2]\!](\tilde{\alpha}, [\,K(k) \mid \alpha \cdot 1 \mapsto [\![s_1]\!] \mid \alpha \cdot 2 \mapsto [\![s_2]\!]\,])(\sigma))\\
& = d([\![\tilde{s}^1]\!](\tilde{\alpha} \cdot 1, [\,K(k) \mid \alpha \mapsto [\![s_1 \parallel s_2]\!] \mid \tilde{\alpha} \cdot 2 \mapsto [\![\tilde{s}^2]\!]\,])(\sigma) +\\
& \qquad [\![\tilde{s}^2]\!](\tilde{\alpha} \cdot 1, [\,K(k) \mid \alpha \mapsto [\![s_1 \parallel s_2]\!] \mid \tilde{\alpha} \cdot 2 \mapsto [\![\tilde{s}^1]\!]\,])(\sigma),\\
& \qquad [\![\tilde{s}^1]\!](\tilde{\alpha} \cdot 1,\\
& \qquad\qquad [\,K(k) \mid \alpha \cdot 1 \mapsto [\![s_1]\!] \mid \alpha \cdot 2 \mapsto [\![s_2]\!] \mid \tilde{\alpha} \cdot 2 \mapsto [\![\tilde{s}^2]\!]\,])(\sigma) +\\
& \qquad [\![\tilde{s}^2]\!](\tilde{\alpha} \cdot 1,\\
& \qquad\qquad [\,K(k) \mid \alpha \cdot 1 \mapsto [\![s_1]\!] \mid \alpha \cdot 2 \mapsto [\![s_2]\!] \mid \tilde{\alpha} \cdot 2 \mapsto [\![\tilde{s}^1]\!]\,])(\sigma))\\
& \qquad [\ + \text{ is nonexpansive }\ ]\\
& \leq max\{d([\![\tilde{s}^1]\!](\tilde{\alpha} \cdot 1, [\,K(k) \mid \alpha \mapsto [\![s_1 \parallel s_2]\!] \mid \tilde{\alpha} \cdot 2 \mapsto [\![\tilde{s}^2]\!]\,])(\sigma),\\
& \qquad\quad [\![\tilde{s}^1]\!](\tilde{\alpha} \cdot 1,\\
& \qquad\qquad [\,K(k) \mid \alpha \cdot 1 \mapsto [\![s_1]\!] \mid \alpha \cdot 2 \mapsto [\![s_2]\!] \mid \tilde{\alpha} \cdot 2 \mapsto [\![\tilde{s}^2]\!]\,])(\sigma)),\\
& \qquad d([\![\tilde{s}^2]\!](\tilde{\alpha} \cdot 1, [\,K(k) \mid \alpha \mapsto [\![s_1 \parallel s_2]\!] \mid \tilde{\alpha} \cdot 2 \mapsto [\![\tilde{s}^1]\!]\,])(\sigma),\\
& \qquad\quad [\![\tilde{s}^2]\!](\tilde{\alpha} \cdot 1,\\
& \qquad\qquad [\,K(k) \mid \alpha \cdot 1 \mapsto [\![s_1]\!] \mid \alpha \cdot 2 \mapsto [\![s_2]\!] \mid \tilde{\alpha} \cdot 2 \mapsto [\![\tilde{s}^1]\!]\,])(\sigma))\}\\
& [\ P_{\|}(\tilde{\alpha}, \alpha, k) \Rightarrow \tilde{\alpha} \cdot 1, \tilde{\alpha} \cdot 2 \notin \{\alpha, \alpha \cdot 1, \alpha \cdot 2\}; \text{ Lemma 5.4(a)}]\\
& = max\{d([\![\tilde{s}^1]\!](\tilde{\alpha} \cdot 1, [\,K[\,k \mid \tilde{\alpha} \cdot 2 \mapsto \tilde{s}^2\,] \mid \alpha \mapsto [\![s_1 \parallel s_2]\!]\,])(\sigma),\\
& \qquad\quad [\![\tilde{s}^1]\!](\tilde{\alpha} \cdot 1,\\
& \qquad\qquad [\,K[\,k \mid \tilde{\alpha} \cdot 2 \mapsto \tilde{s}^2\,] \mid \alpha \cdot 1 \mapsto [\![s_1]\!] \mid \alpha \cdot 2 \mapsto [\![s_2]\!]\,])(\sigma)),\ ^{(5.10.10)}\\
& \qquad d([\![\tilde{s}^2]\!](\tilde{\alpha} \cdot 1, [\,K[\,k \mid \tilde{\alpha} \cdot 2 \mapsto \tilde{s}^1\,] \mid \alpha \mapsto [\![s_1 \parallel s_2]\!]\,])(\sigma),\\
& \qquad\quad [\![\tilde{s}^2]\!](\tilde{\alpha} \cdot 1,\\
& \qquad\qquad [\,K[\,k \mid \tilde{\alpha} \cdot 2 \mapsto \tilde{s}^1\,] \mid \alpha \cdot 1 \mapsto [\![s_1]\!] \mid \alpha \cdot 2 \mapsto [\![s_2]\!]\,])(\sigma))\ ^{(5.10.11)}\ \}
\end{aligned}$$

$P_{\|}(\tilde{\alpha}, \alpha, k)$ implies $P_{\|}(\tilde{\alpha}\cdot 1, \alpha, [\, k \mid \tilde{\alpha}\cdot 2 \mapsto \tilde{s}^i\,])$ for $i = 1, 2$. Therefore, we can use the induction hypothesis (because $\varsigma(\tilde{s}^i) < \varsigma(\tilde{s}^1 \parallel \tilde{s}^2)$ for $i = 1, 2$) and we infer that (1) $\exists s_1' \in Stat, \sigma_1' \in \Sigma, \alpha_1' \in Id, k_1' \in Konf$ such that $P_{\|}(\alpha_1', \alpha, k_1')$ and (2) $\exists s_2' \in Stat, \sigma_2' \in \Sigma, \alpha_2' \in Id, k_2' \in Konf$ such that $P_{\|}(\alpha_2', \alpha, k_2')$ and:

$$^{(5.10.10)} \leq \tfrac{1}{2}\cdot d([\![s_1']\!](\alpha_1', [\, K(k_1') \mid \alpha \mapsto [\![s_1 \parallel s_2]\!]\,])(\sigma_1'),$$
$$[\![s_1']\!](\alpha_1', [\, K(k_1') \mid \alpha\cdot 1 \mapsto [\![s_1]\!] \mid \alpha\cdot 2 \mapsto [\![s_2]\!]\,])(\sigma_1'))\ ^{(5.10.10')}$$
$$^{(5.10.11)} \leq \tfrac{1}{2}\cdot d([\![s_2']\!](\alpha_2', [\, K(k_2') \mid \alpha \mapsto [\![s_1 \parallel s_2]\!]\,])(\sigma_2'),$$
$$[\![s_2']\!](\alpha_2', [\, K(k_2') \mid \alpha\cdot 1 \mapsto [\![s_1]\!] \mid \alpha\cdot 2 \mapsto [\![s_2]\!]\,])(\sigma_2'))\ ^{(5.10.11')}$$

Therefore, by taking $max\{\ ^{(5.10.10')}\ ,\ ^{(5.10.11')}\ \}$ we obtain the desired result $^{(5.10.2)}$.
■

Lemma 5.11 *For all $\tilde{s}, s_1, s_2 \in Stat, \sigma \in \Sigma, \tilde{\alpha}, \alpha \in Id, k \in Konf$ such that $(\tilde{\alpha}, k) \in Conf, \alpha \notin id(k), \alpha\cdot 1 \notin id(k)$ and $(\neg(\tilde{\alpha} \leq \alpha\cdot 1))$ we have:*

(a) $kc[\, K(k) \mid \alpha \mapsto [\![s_1; s_2]\!]\,](\sigma) = kc[\, K(k) \mid \alpha\cdot 1 \mapsto [\![s_1]\!] \mid \alpha \mapsto [\![s_2]\!]\,](\sigma)$

(b) $[\![\tilde{s}]\!](\tilde{\alpha}, [\, K(k) \mid \alpha \mapsto [\![s_1; s_2]\!]\,])(\sigma) =$

$[\![\tilde{s}]\!](\tilde{\alpha}, [\, K(k) \mid \alpha\cdot 1 \mapsto [\![s_1]\!] \mid \alpha \mapsto [\![s_2]\!]\,])(\sigma)$

Proof: The proof of this Lemma is similar to the proof of Lemma 5.10. In this case the invariant property which is preserved by the computation steps is $P_;$:

$$P_;(\tilde{\alpha}, \alpha, k) \overset{not.}{=} ((\tilde{\alpha}, k) \in Conf) \wedge (\alpha \notin id(k)) \wedge (\alpha\cdot 1 \notin id(k)) \wedge (\neg(\tilde{\alpha} \leq \alpha\cdot 1))$$

We proceed as follows. For 5.11(a) we show that $\forall s_1, s_2 \in Stat, \sigma \in \Sigma, \alpha \in Id, k \in Konf$ such that $\alpha \notin id(k)$ and $\alpha\cdot 1 \notin id(k)$, $\exists s' \in Stat, \alpha' \in Id, k' \in Konf$ such that $P_;(\alpha', \alpha, k')$ such that:

$$d(kc[\, K(k) \mid \alpha \mapsto [\![s_1; s_2]\!]\,](\sigma), kc[\, K(k) \mid \alpha\cdot 1 \mapsto [\![s_1]\!] \mid \alpha \mapsto [\![s_2]\!]\,](\sigma))$$
$$\leq d([\![s']\!](\alpha', [\, K(k') \mid \alpha \mapsto [\![s_1; s_2]\!]\,])(\sigma),$$
$$[\![s']\!](\alpha', [\, K(k') \mid \alpha\cdot 1 \mapsto [\![s_1]\!] \mid \alpha \mapsto [\![s_2]\!]\,])(\sigma))\ ^{(5.11.1)}$$
$$\overset{not.}{=}\ \varepsilon_;(s', \alpha', k', \alpha, s_1, s_2, \sigma)$$

Also, for 5.11(b) we show that $\forall \tilde{s}, s_1, s_2 \in Stat, \sigma \in \Sigma, \tilde{\alpha}, \alpha \in Id, k \in Konf$ such that $P_;(\tilde{\alpha}, \alpha, k)$, $\exists s' \in Stat, \sigma' \in \Sigma, \alpha' \in Id, k' \in Konf$ such that $P_;(\alpha', \alpha, k')$ and:

$$\varepsilon_;(\tilde{s}, \tilde{\alpha}, k, \alpha, s_1, s_2, \sigma)$$
$$= d([\![\tilde{s}]\!](\tilde{\alpha}, [\, K(k) \mid \alpha \mapsto [\![s_1; s_2]\!]\,])(\sigma),$$
$$[\![\tilde{s}]\!](\tilde{\alpha}, [\, K(k) \mid \alpha\cdot 1 \mapsto [\![s_1]\!] \mid \alpha \mapsto [\![s_2]\!]\,])(\sigma))$$

$$\leq \tfrac{1}{2} \cdot d([\![s']\!](\alpha', [\,K(k') \mid \alpha \mapsto [\![s_1; s_2]\!]\,])(\sigma'),$$
$$[\![s']\!](\alpha', [\,K(k') \mid \alpha \cdot 1 \mapsto [\![s_1]\!] \mid \alpha \mapsto [\![s_2]\!]\,])(\sigma')) \quad ^{(5.11.2)}$$
$$\overset{not.}{=} \varepsilon_{;}(s', \alpha', k', \alpha, s_1, s_2, \sigma')$$

If we put $(w \in)W = Stat \times Id \times Konf \times Id \times Stat \times Stat \times \Sigma$ we infer that:

$$sup_{\ w \in W\,:\,P_{;}(\tilde{\alpha},\alpha,k)}\ \ \varepsilon_{;}(w) \leq \tfrac{1}{2} \cdot sup_{\ w' \in W\,:\,P_{;}(\alpha',\alpha,k')}\ \ \varepsilon_{;}(w')$$

where $w = (\tilde{s}, \tilde{\alpha}, k, \alpha, s_1, s_2, \sigma)$ and $w' = (s', \alpha', k', \alpha, s_1, s_2, \sigma')$.
But obviously, $sup_{\ w \in W\,:\,P_{;}(\tilde{\alpha},\alpha,k)}\ \varepsilon_{;}(w) = sup_{\ w' \in W\,:\,P_{;}(\alpha',\alpha,k')}\ \varepsilon_{;}(w')$. So we infer that:

$$sup_{\ w \in W\,:\,P_{;}(\tilde{\alpha},\alpha,k)}\ \ \varepsilon_{;}(w) = 0$$

which implies immediately 5.11(b). Next, by using this result and $^{(5.11.1)}$ we obtain immediately 5.11(a).

The proofs of Lemma 5.11 and Lemma 5.10 are very similar. The main difference is given by the computing invariants $P_{;}$ and $P_{\|}$, which are specific of sequential and parallel composition, respectively. Here we skip the proof of $^{(5.11.1)}$ (which is similar to the proof of $^{(5.10.1)}$). We only give the proof of $^{(5.11.2)}$. We proceed by induction on $\varsigma(\tilde{s})$, using $^{(5.11.1)}$. In the computations given below it is assumed that $P_{;}(\tilde{\alpha}, \alpha, k)$. We treat two subcases.

Case $[\tilde{s} = a]$ when $I(a)(\sigma) = \sigma' \in \Sigma$.

$$\begin{aligned}
& d([\![a]\!](\tilde{\alpha}, [\,K(k) \mid \alpha \mapsto [\![s_1; s_2]\!]\,])(\sigma),\\
& \quad [\![a]\!](\tilde{\alpha}, [\,K(k) \mid \alpha \cdot 1 \mapsto [\![s_1]\!] \mid \alpha \mapsto [\![s_2]\!]\,])(\sigma))\\
& = d(\sigma' \cdot kc[\,K(k) \mid \alpha \mapsto [\![s_1; s_2]\!]\,](\sigma'),\\
& \qquad \sigma' \cdot kc[\,K(k) \mid \alpha \cdot 1 \mapsto [\![s_1]\!] \mid \alpha \mapsto [\![s_2]\!]\,](\sigma'))\\
& = \tfrac{1}{2} \cdot d(kc[\,K(k) \mid \alpha \mapsto [\![s_1; s_2]\!]\,](\sigma'),\\
& \qquad\quad kc[\,K(k) \mid \alpha \cdot 1 \mapsto [\![s_1]\!] \mid \alpha \mapsto [\![s_2]\!]\,](\sigma')) \quad ^{(5.11.3)}
\end{aligned}$$

By $^{(5.11.1)}$ $\exists s' \in Stat, \alpha' \in Id, k' \in Konf$ such that $P_{;}(\alpha', \alpha, k')$ and:

$$\begin{aligned}
^{(5.11.3)}\ & \leq \tfrac{1}{2} \cdot d([\![s']\!](\alpha', [\,K(k') \mid \alpha \mapsto [\![s_1; s_2]\!]\,])(\sigma'),\\
& \qquad [\![s']\!](\alpha', [\,K(k') \mid \alpha \cdot 1 \mapsto [\![s_1]\!] \mid \alpha \mapsto [\![s_2]\!]\,])(\sigma'))
\end{aligned}$$

Case $[\tilde{s} = \tilde{s}^1; \tilde{s}^2]$.

$$\begin{aligned}
& d([\![\tilde{s}^1; \tilde{s}^2]\!](\tilde{\alpha}, [\,K(k) \mid \alpha \mapsto [\![s_1;s_2]\!]\,])(\sigma),\\
& \quad [\![\tilde{s}^1; \tilde{s}^2]\!](\tilde{\alpha}, [\,K(k) \mid \alpha \cdot 1 \mapsto [\![s_1]\!] \mid \alpha \mapsto [\![s_2]\!]\,])(\sigma))\\
& = d([\![\tilde{s}^1]\!](\tilde{\alpha} \cdot 1, [\,K(k) \mid \alpha \mapsto [\![s_1;s_2]\!] \mid \tilde{\alpha} \mapsto [\![\tilde{s}^2]\!]\,])(\sigma),\\
& \qquad [\![\tilde{s}^1]\!](\tilde{\alpha} \cdot 1, [\,K(k) \mid \alpha \cdot 1 \mapsto [\![s_1]\!] \mid \alpha \mapsto [\![s_2]\!] \mid \tilde{\alpha} \mapsto [\![\tilde{s}^2]\!]\,])(\sigma))
\end{aligned}$$

$$[\ P_;(\tilde{\alpha},\alpha,k) \Rightarrow \tilde{\alpha} \neq \alpha, \tilde{\alpha} \neq \alpha\cdot 1;\ \text{Lemma 5.4(a)}\ ]$$
$$= d([\![\tilde{s}^1]\!](\tilde{\alpha}\cdot 1, [\,K[\,k \mid \tilde{\alpha} \mapsto \tilde{s}^2\,] \mid \alpha \mapsto [\![s_1;s_2]\!]\,])(\sigma),$$
$$[\![\tilde{s}^1]\!](\tilde{\alpha}\cdot 1, [\,K[\,k \mid \tilde{\alpha} \mapsto \tilde{s}^2\,] \mid \alpha\cdot 1 \mapsto [\![s_1]\!] \mid \alpha \mapsto [\![s_2]\!]\,])(\sigma))\ ^{(5.11.4)}$$

$P_;(\tilde{\alpha},\alpha,k)$ implies $P_;(\tilde{\alpha}\cdot 1,\alpha,[\,k \mid \tilde{\alpha} \mapsto \tilde{s}^2\,])$. Therefore, we can use the induction hypothesis ($\varsigma(\tilde{s}^1) < \varsigma(\tilde{s}^1;\tilde{s}^2)$) and we infer that $\exists s' \in \mathit{Stat}, \sigma' \in \Sigma, \alpha' \in \mathit{Id}, k' \in \mathit{Konf}$ such that $P_;(\alpha',\alpha,k')$ and

$$^{(5.11.4)}\ \leq \tfrac{1}{2}\cdot\ d([\![s']\!](\alpha', [\,K(k') \mid \alpha \mapsto [\![s_1;s_2]\!]\,])(\sigma'),$$
$$[\![s']\!](\alpha', [\,K(k') \mid \alpha\cdot 1 \mapsto [\![s_1]\!] \mid \alpha \mapsto [\![s_2]\!]\,])(\sigma')) \qquad \blacksquare$$

In the sequel we use the notation

$$s \simeq \overline{s} \qquad (s,\overline{s} \in \mathit{Stat})$$

to express that $[\![C(s)]\!](\alpha,K(k))=[\![C(\overline{s})]\!](\overline{\alpha},K(\overline{k}))$ for all contexts C and for all isomorphic configurations $(\alpha,k) \cong (\overline{\alpha},\overline{k})$ $(\in \mathit{Conf})$.

Theorem 5.12 *For all $s, s_1, s_2, s_3 \in \mathit{Stat}$:*

(a)	$s_1 + s_2$	$\simeq$	$s_2 + s_1$	(*commutativity of* $+$)
(b)	$(s_1 + s_2) + s_3$	$\simeq$	$s_1 + (s_2 + s_3)$	(*associativity of* $+$)
(c)	$s + s$	$\simeq$	s	(*idempotency of* $+$)
(d)	$(s_1 + s_2); s_3$	$\simeq$	$s_1; s_3 + s_2; s_3$	(*right distributivity of* ; over $+$)
(e)	$s_1; (s_2; s_3)$	$\simeq$	$(s_1; s_2); s_3$	(*associativity of* ;)
(f)	$s + \delta$	$\simeq$	s	
(g)	$\delta; s$	$\simeq$	δ	
(h)	$s_1 \parallel s_2$	$\simeq$	$s_1 \lfloor\!\lfloor s_2 + s_2 \lfloor\!\lfloor s_1$	
(i)	$a \lfloor\!\lfloor s$	$\simeq$	$a; s$	
(j)	$(a; s_1) \lfloor\!\lfloor s_2$	$\simeq$	$a; (s_1 \parallel s_2)$	
(k)	$(s_1 + s_2) \lfloor\!\lfloor s_3$	$\simeq$	$s_1 \lfloor\!\lfloor s_3 + s_2 \lfloor\!\lfloor s_3$	(*right distributivity of* $\lfloor\!\lfloor$ over $+$)
(l)	$s_1 \parallel s_2$	$\simeq$	$s_2 \parallel s_1$	(*commutativity of* $\parallel$)
(m)	$s_1 \parallel (s_2 \parallel s_3)$	$\simeq$	$(s_1 \parallel s_2) \parallel s_3$	(*associativity of* $\parallel$)

; *binds stronger than* $\parallel, \lfloor\!\lfloor$. *Also,* $\parallel, \lfloor\!\lfloor$ *bind stronger than* $+$.

Proof: First notice that $[\![s]\!] = [\![\overline{s}]\!] \Rightarrow s \simeq \overline{s}$, for any $s, \overline{s} \in \mathit{Stat}$. Indeed, $[\![s]\!] = [\![\overline{s}]\!] \Rightarrow [\![C(s)]\!] = [\![C(\overline{s})]\!]$ for any context C (by the compositionality of $[\![\cdot]\!]$) and $[\![C(s)]\!] = [\![C(\overline{s})]\!] \Rightarrow s \simeq \overline{s}$, by Corolarry 5.6(a). Therefore, the properties stated by Theorem 5.12 (a)-(d), (f)-(h), (k) and (l) follow immediately by Lemma 5.1.

On the other hand, the properties stated by Theorem 5.12(e), (i), (j) and (m) can be proved for continuations containing *only* denotations of statements (not for arbitrary continuations) and require more involved arguments based on the identification of computing invariants and the use of contraction. By Lemma 5.8 and Corolarry 5.6(a), in order to prove

$s \simeq \overline{s}$ it is enough to show that $[\![s]\!](\alpha, K(k)) = [\![\overline{s}]\!](\alpha, K(k))$ for any $(\alpha, k) \in Conf$ (or that $[\![s]\!](\alpha, K(k))(\sigma) = [\![\overline{s}]\!](\alpha, K(k))(\sigma)$ for any $(\alpha, k) \in Conf$ and an arbitrary $\sigma \in \Sigma$).

Property 5.12(e) follows by using Lemma 5.11.

$$[\![s_1; (s_2; s_3)]\!](\alpha, K(k))(\sigma) = [\![s_1]\!](\alpha \cdot 1, [K(k) \,|\, \alpha \mapsto [\![s_2; s_3]\!]\,])(\sigma)$$

[Lemma 5.4(a)]

$$= [\![s_1]\!](\alpha \cdot 1, K[\,k \mid \alpha \mapsto (s_2; s_3)\,])(\sigma) \ ^{(5.12.1)}$$

It is easy to check that $(\alpha \cdot 1, [\,k \mid \alpha \mapsto (s_2; s_3)\,]) \cong (\alpha \cdot 1 \cdot 1, [\,k \mid \alpha \mapsto (s_2; s_3)\,])$. Thus $^{(5.12.1)} =$ $[\![s_1]\!](\alpha \cdot 1 \cdot 1, K[\,k \mid \alpha \mapsto (s_2; s_3)\,])(\sigma)$, by Corollary 5.6(a), and

$$[\![s_1]\!](\alpha \cdot 1 \cdot 1, K[\,k \mid \alpha \mapsto (s_2; s_3)\,])(\sigma) \qquad \text{[Lemma 5.4(a)]}$$

$$= [\![s_1]\!](\alpha \cdot 1 \cdot 1, [\,K(k) \mid \alpha \mapsto [\![s_2; s_3]\!]\,])(\sigma) \quad \text{[Lemma 5.11(b)]}$$

$$= [\![s_1]\!](\alpha \cdot 1 \cdot 1, [\,K(k) \mid \alpha \cdot 1 \mapsto [\![s_2]\!] \mid \alpha \mapsto [\![s_3]\!]\,])(\sigma)$$

$$= [\![s_1; s_2]\!](\alpha \cdot 1, [\,K(k) \mid \alpha \mapsto [\![s_3]\!]\,])(\sigma)$$

$$= [\![(s_1; s_2); s_3]\!](\alpha, K(k))(\sigma)$$

Property 5.12(i) is an easy consequence of Corollary 5.6. We only treat the subcase when $I(a)(\sigma) = \overline{\sigma} \in \Sigma$.

$$[\![a \,\lfloor\!\lfloor\, s]\!](\alpha, K(k))(\sigma)$$

$$= [\![a]\!](\alpha \cdot 1, [\,K(k) \mid \alpha \cdot 2 \mapsto [\![s]\!]\,])(\sigma)$$

$$= \overline{\sigma} \cdot kc[\,K(k) \mid \alpha \cdot 2 \mapsto [\![s]\!]\,](\overline{\sigma}) \qquad \text{[Lemma 5.4(a)]}$$

$$= \overline{\sigma} \cdot kc(K[\,k \mid \alpha \cdot 2 \mapsto s\,])(\overline{\sigma})$$

$[\ (\alpha, k) \in Conf \Rightarrow [\,k \mid \alpha \cdot 2 \mapsto s\,] \cong [\,k \mid \alpha \mapsto s\,]$, Corollary 5.6(b)]

$$= \overline{\sigma} \cdot kc(K[\,k \mid \alpha \mapsto s\,])(\overline{\sigma}) \qquad \text{[Lemma 5.4(a)]}$$

$$= \overline{\sigma} \cdot kc[\,K(k) \mid \alpha \mapsto [\![s]\!]\,](\overline{\sigma})$$

$$= [\![a]\!](\alpha \cdot 1, [\,K(k) \mid \alpha \mapsto [\![s]\!]\,])(\sigma)$$

$$= [\![a; s]\!](\alpha, K(k))(\sigma)$$

In the proofs of 5.12(j) and (m) one can use Lemma 5.10. For 5.12(j) we only consider the subcase when $I(a)(\sigma) = \overline{\sigma} \in \Sigma$.

$$[\![(a; s_1) \,\lfloor\!\lfloor\, s_2]\!](\alpha, K(k))(\sigma)$$

$= [\![a; s_1]\!](\alpha \cdot 1, [\, K(k) \mid \alpha \cdot 2 \mapsto [\![s_2]\!]\,])(\sigma)$

$= [\![a]\!](\alpha \cdot 1 \cdot 1, [\, K(k) \mid \alpha \cdot 1 \mapsto [\![s_1]\!] \mid \alpha \cdot 2 \mapsto [\![s_2]\!]\,])(\sigma)$

$= \overline{\sigma} \cdot kc[\, K(k) \mid \alpha \cdot 1 \mapsto [\![s_1]\!] \mid \alpha \cdot 2 \mapsto [\![s_2]\!]\,])(\overline{\sigma})$ [Lemma 5.10(a)]

$= \overline{\sigma} \cdot kc[\, K(k) \mid \alpha \mapsto [\![s_1 \parallel s_2]\!]\,](\overline{\sigma})$

$= [\![a]\!](\alpha \cdot 1, [\, K(k) \mid \alpha \mapsto [\![s_1 \parallel s_2]\!]\,])(\sigma)$

$= [\![a; (s_1 \parallel s_2)]\!](\alpha, K(k))(\sigma)$

Next we prove the property stated by Theorem 5.12(m). First, we expand the expressions involved as follows:

$[\![s_1 \parallel (s_2 \parallel s_3)]\!](\alpha, K(k))(\sigma)$

$= [\![s_1]\!](\alpha \cdot 1, [\, K(k) \mid \alpha \cdot 2 \mapsto [\![s_2 \parallel s_3]\!]\,])(\sigma) +$

$[\![s_2 \parallel s_3]\!](\alpha \cdot 1, [\, K(k) \mid \alpha \cdot 2 \mapsto [\![s_1]\!]\,])(\sigma)$ [+ is associative]

$= [\![s_1]\!](\alpha \cdot 1, [\, K(k) \mid \alpha \cdot 2 \mapsto [\![s_2 \parallel s_3]\!]\,])(\sigma)\ ^{(5.12.2)} +$

$[\![s_2]\!](\alpha \cdot 1 \cdot 1, [\, K(k) \mid \alpha \cdot 2 \mapsto [\![s_1]\!] \mid \alpha \cdot 1 \cdot 2 \mapsto [\![s_3]\!]\,])(\sigma)\ ^{(5.12.3)} +$

$[\![s_3]\!](\alpha \cdot 1 \cdot 1, [\, K(k) \mid \alpha \cdot 2 \mapsto [\![s_1]\!] \mid \alpha \cdot 1 \cdot 2 \mapsto [\![s_2]\!]\,])(\sigma)\ ^{(5.12.4)}$

$[\![(s_1 \parallel s_2) \parallel s_3]\!](\alpha, K(k))(\sigma)$

$= [\![s_1 \parallel s_2]\!](\alpha \cdot 1, [\, K(k) \mid \alpha \cdot 2 \mapsto [\![s_3]\!]\,])(\sigma) +$

$[\![s_3]\!](\alpha \cdot 1, [\, K(k) \mid \alpha \cdot 2 \mapsto [\![s_1 \parallel s_2]\!]\,])(\sigma)$

$= [\![s_1]\!](\alpha \cdot 1 \cdot 1, [\, K(k) \mid \alpha \cdot 2 \mapsto [\![s_3]\!] \mid \alpha \cdot 1 \cdot 2 \mapsto [\![s_2]\!]\,])(\sigma)\ ^{(5.12.5)} +$

$[\![s_2]\!](\alpha \cdot 1 \cdot 1, [\, K(k) \mid \alpha \cdot 2 \mapsto [\![s_3]\!] \mid \alpha \cdot 1 \cdot 2 \mapsto [\![s_1]\!]\,])(\sigma)\ ^{(5.12.6)} +$

$[\![s_3]\!](\alpha \cdot 1, [\, K(k) \mid \alpha \cdot 2 \mapsto [\![s_1 \parallel s_2]\!]\,])(\sigma)\ ^{(5.12.7)}$

We show that $^{(5.12.2)} = ^{(5.12.5)}$, $^{(5.12.3)} = ^{(5.12.6)}$ and $^{(5.12.4)} = ^{(5.12.7)}$. First we prove that $^{(5.12.3)} = ^{(5.12.6)}$. Indeed:

$[\![s_2]\!](\alpha \cdot 1 \cdot 1, [\, K(k) \mid \alpha \cdot 2 \mapsto [\![s_1]\!] \mid \alpha \cdot 1 \cdot 2 \mapsto [\![s_3]\!]\,])(\sigma)$ [Lemma 5.4(a)]

$= [\![s_2]\!](\alpha \cdot 1 \cdot 1, K[\, k \mid \alpha \cdot 2 \mapsto s_1 \mid \alpha \cdot 1 \cdot 2 \mapsto s_3\,])(\sigma)$

$[(\alpha, k) \in \mathit{Conf} \Rightarrow$

$(\alpha \cdot 1 \cdot 1, [\, k \mid \alpha \cdot 2 \mapsto s_1 \mid \alpha \cdot 1 \cdot 2 \mapsto s_3\,]) \cong$

$(\alpha \cdot 1 \cdot 1, [\, k \mid \alpha \cdot 2 \mapsto s_3 \mid \alpha \cdot 1 \cdot 2 \mapsto s_1\,]),$ Corollary 5.6(a)]

$= [\![s_2]\!](\alpha \cdot 1 \cdot 1, K[\, k \mid \alpha \cdot 2 \mapsto s_3 \mid \alpha \cdot 1 \cdot 2 \mapsto s_1\,])(\sigma)$

$= [\![s_2]\!](\alpha \cdot 1 \cdot 1, [\, K(k) \mid \alpha \cdot 2 \mapsto [\![s_3]\!] \mid \alpha \cdot 1 \cdot 2 \mapsto [\![s_1]\!]\,])(\sigma)$

Both ${}^{(5.12.2)} = {}^{(5.12.5)}$ and ${}^{(5.12.4)} = {}^{(5.12.7)}$ can be handled by using Lemma 5.10(b). As the proofs are similar, we only treat here ${}^{(5.12.2)} = {}^{(5.12.5)}$.

$[\![s_1]\!](\alpha \cdot 1, [\, K(k) \mid \alpha \cdot 2 \mapsto [\![s_2 \parallel s_3]\!]\,])(\sigma)$ [Lemma 5.10(b)]

$= [\![s_1]\!](\alpha \cdot 1, [\, K(k) \mid \alpha \cdot 2 \cdot 1 \mapsto [\![s_2]\!] \mid \alpha \cdot 2 \cdot 2 \mapsto [\![s_3]\!]\,])(\sigma)$ [Lemma 5.4(a)]

$= [\![s_1]\!](\alpha \cdot 1, K[\, k \mid \alpha \cdot 2 \cdot 1 \mapsto s_2 \mid \alpha \cdot 2 \cdot 2 \mapsto s_3\,])(\sigma)$

$[(\alpha, k) \in Conf \Rightarrow$

$(\alpha \cdot 1, [\, k \mid \alpha \cdot 2 \cdot 1 \mapsto s_2 \mid \alpha \cdot 2 \cdot 2 \mapsto s_3\,]) \cong$

$(\alpha \cdot 1 \cdot 1, [\, k \mid \alpha \cdot 2 \mapsto s_3 \mid \alpha \cdot 1 \cdot 2 \mapsto s_2\,]),$ Corollary 5.6(a)]

$= [\![s_1]\!](\alpha \cdot 1 \cdot 1, K[\, k \mid \alpha \cdot 2 \mapsto s_3 \mid \alpha \cdot 1 \cdot 2 \mapsto s_2\,])(\sigma)$ [Lemma 5.4(a)]

$= [\![s_1]\!](\alpha \cdot 1 \cdot 1, [\, K(k) \mid \alpha \cdot 2 \mapsto [\![s_3]\!] \mid \alpha \cdot 1 \cdot 2 \mapsto [\![s_2]\!]\,])(\sigma)$ ■

Remark 5.13

(a) *As it is well-known, the properties stated by Theorem 5.12 provide a finite axiomatization for the parallel composition operator* $\parallel$*; see, e.g., [5].*[10] *For any non-recursive* $\mathcal{L}$ *program (closed term)* $s \in Stat$ *there is a non-recursive program* $\overline{s} \in Stat$ *that contains only the operators for sequential composition (;) and nondeterministic choice (+) and such that the above set of laws imply* $s \simeq \overline{s}$*. The operators* $\parallel$ *and* $\lfloor\!\lfloor$ *can be eliminated from any non-recursive asynchronous* $\mathcal{L}$ *program.* Such an elimination can always be performed without manipulating continuations explicitly. *For example* $a_1 \| a_2 \simeq a_1 \lfloor\!\lfloor a_2 + a_2 \lfloor\!\lfloor a_1 \simeq a_1; a_2 + a_2; a_1$.

(b) *The semantics of sequential composition (;) is defined using continuations, but it ultimately relies on the prefixing operation* $\sigma \cdot p$*. Notice that if* $\delta \notin p$ *then* $\sigma \cdot (p + \{\delta\}) = \sigma \cdot p \neq \sigma \cdot p + \{\sigma\delta\} = \sigma \cdot p + \sigma \cdot \{\delta\}$*. Therefore we cannot expect to get a model which also satisfies the law* $s_1; (s_2 + s_3) \simeq s_1; s_2 + s_1; s_3$*, i.e. the left distributivity of* ; *over* $+$*.*

Recall that $\mathcal{D}[\![s]\!]\sigma = [\![s]\!](\alpha_0, \kappa_0)(\sigma)$ (see Definition 4.1(c)). For any $a \in Act$, $s, s' \in Stat$ and for any context C one can easily check the following:

- $s \simeq s'$ implies $\mathcal{D}[\![C(s)]\!] = \mathcal{D}[\![C(s')]\!]$,

[10]Strictly speaking, properties 5.12(l) and 5.12(m) are not needed for this purpose.

- $\mathcal{D}[\![a;s]\!]\sigma = \textsf{ if } (I(a)(\sigma) =\uparrow) \textsf{ then } \{\delta\} \textsf{ else } I(a)(\sigma)\cdot\mathcal{D}[\![s]\!](I(a)(\sigma))$[11] and
- $\mathcal{D}[\![s+s']\!]\sigma = \mathcal{D}[\![s]\!]\sigma + \mathcal{D}[\![s']\!]\sigma$.

For example, if $I(a_1)(\sigma)=\sigma_1, I(a_2)(\sigma_1)=\sigma_2$ and $I(a_2)(\sigma)=\uparrow$ then

$$\mathcal{D}[\![a_1 \,\|\, a_2]\!]\sigma = \mathcal{D}[\![a_1 \lfloor\!\lfloor\, a_2 + a_2 \lfloor\!\lfloor\, a_1]\!]\sigma = \mathcal{D}[\![a_1;a_2 + a_2;a_1]\!]\sigma$$
$$= \mathcal{D}[\![a_1;a_2]\!]\sigma + \mathcal{D}[\![a_2;a_1]\!]\sigma = \{\sigma_1\sigma_2\} + \{\delta\} = \{\sigma_1\sigma_2\}$$

6 Concluding remarks and future work

We presented a method of reasoning about the behavior of programs in denotational models designed with metric spaces and continuation semantics for concurrency (CSC) [31]. The method was illustrated on the particular case of a simple asynchronous language [10]. We proved that the semantic operators designed with continuations obey concurrency laws such as the associativity and commutativity of parallel composition. The method is general; we think it could be applied to every language designed by using CSC. The method relies on the identification of computing invariants as relations between continuation structures in combination with arguments of the kind '$\varepsilon \le \frac{1}{2}\cdot\varepsilon \Rightarrow \varepsilon = 0$', which are standard in metric semantics. The significance of the results is given mainly by the *flexibility* provided by the continuations technique which can thus be used to describe concurrent behaviour.

In previous work, we developed CSC-based denotational models for a couple of advanced concepts, including nondeterministic promotion in Andorra-like languages [33], and synchronous communication on multiple channels [32] in the style of Join calculus [17]. In ongoing work, we investigate the possibility to develop a denotational semantics designed with CSC for membrane computing [26]. As far as we know such advanced control concepts have not been modeled denotationally until now without CSC. It should be possible to use the proof method presented in this paper to reason compositionally about the behavior of such advanced concepts. Still, this has to be verified.

In this paper the domain of continuations ($\mathbf{Kont} = \{\!|\, \frac{1}{2}\cdot\mathbf{D} \,|\!\}$) was modelled with the aid of a function space from a set of identifiers (endowed with a partial order) to the domain of computations: $Id \to \frac{1}{2}\cdot\mathbf{D}$ (see Section 3 where the construction $\{\!|\,\cdot\,|\!\}$ was introduced). According to Corollary 5.6, any two isomorphic continuations behave the same. Intuitively, the domain of continuations could be defined in terms of isomorphism classes $[Id \to \frac{1}{2}\cdot\mathbf{D}]$ of such structures. Since the existing models based on isomorphism classes of semantic structures (in particular the metric pomset model [6]) do not involve domains defined by reflexive equations (like $\mathbf{D}$), such a construction also requires further work.

[11]Notice that $\kappa_0 = K(k_0)$, where $k_0 = (\emptyset, t_0)$, with $t_0 \in (Id \to Stat)$, $t_0(\alpha) = \delta$, $\forall\alpha \in Id$. The property follows by using Corollary 5.6 and the fact that $[\,k_0 \mid \alpha' \mapsto s\,] \setminus \alpha' \cong k_0$, $\forall\alpha' \in Id$. In fact $[\,k \mid \alpha' \mapsto s\,] \setminus \alpha' \cong k$, $\forall k \in Konf$ if $\alpha' \notin id(k)$.

References

[1] P. America and J.J.M.M. Rutten, Solving reflexive domain equations in a category of complete metric spaces, *J. of Comp. Syst. Sci* 39(3), 343–375, 1989.

[2] J.W. De Bakker and E.P. De Vink, Rendez-vous with metric semantics, *New Generation Computing* 12, 53–90, 1993.

[3] J.W. De Bakker and E.P. De Vink, *Control Flow Semantics*, MIT Press, 1996.

[4] J.W. De Bakker and J.I. Zucker, Processes and the denotational semantics of concurrency, *Inf. and Control* 54, 70–120, 1982.

[5] J.C.M. Baeten and W.P. Weijland, *Process algebra,* Cambridge Univ. Press, 1990.

[6] J.W. De Bakker and J.H.A. Warmerdam, Metric pomset semantics for a concurrent language with recursion, *LNCS* 469, 21–49, Springer, 1990.

[7] S. Banach, Sur les operations dans les ensembles abstrait et leur applications aux equations integrales, *Fundamenta Matematicae* 3, 133—181, 1922, .

[8] J.A. Bergstra and J.W. Klop, Algebra of communicating processes with abstraction, *Theoretical Computer Science* 37(1), 77–121, 1985.

[9] D.G. Bobrow and B. Wegbreit, A Model and Stack Implementation of Multiple Environments, *Comm. ACM* 16(10), 591–603, 1973.

[10] F.S. De Boer, J.N. Kok, C. Palamidessi and J.J.M.M. Rutten, A paradigm for asynchronous communication and its application to concurrent constraint programming. In Apt, K.R., De Bakker, J.W. and Rutten, J.J.M.M, eds., "Logic Programming Languages: Constraints, Functions and Objects", MIT Press, 82-114 (1993)

[11] S. Brookes, Traces: a unifying semantic framework for parallel programming languages, *MFPS* 18, New Orleans, 2002.

[12] A. de Bruin, Experiments with continuation semantics: jumps, backtracking, dynamic networks, Ph.D. thesis, Vrije Universiteit, Amsterdam, 1986.

[13] A. De Bruin and W. Bohm, The denotational semantics of dynamic networks of processes, *ACM Transactions on Programming Languages and Systems* 7(4), 656–679, 1985.

[14] G. Ciobanu and E.N. Todoran, Continuation semantics for concurrency, Technical Report FML-09-02, Romanian Academy, 2009. (Available at `http://iit.iit.tuiasi.ro/TR/reports/fml0902.pdf`)

[15] G. Ciobanu and E.N. Todoran, A methodology for concurrent languages development based on denotational semantics, *Proc. IEEE SYNASC'09*, 290–298, 2009.

[16] O. Danvy, On evaluation contexts, continuations and the rest of the computation, *4th ACM SIGPLAN Continuations Workshop*, 13–23 2004.

[17] C. Fournet and G. Gonthier, The Join calculus: a language for distributed mobile programming, *LNCS* 2395, 268–332, 2002.

[18] R. van Glabbeek, U. Goltz and J.W. Schicke, On Synchronous and Asynchronous Interaction in Distributed Systems. *LNCS* 5162, 16–35, 2008.

[19] K. Honda, M. Tokoro, An object calculus for asynchronous communication, *LNCS* vol.512, 133–147, Springer (1991)

[20] T. Jech, *Set theory,* Springer, 2003.

[21] J.N. Kok and J.J.M.M. Rutten, Contractions in comparing concurrency semantics, *Theoretical Computer Science* 76, 179–222, 1990.

[22] R. Milner, A calculus of communicating systems, *LNCS* 92, Springer, 1980.

[23] R. Milner, *Communication and concurrency*, Prentice Hall, 1989.

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

[25] C. Palamidessi, Comparing the expressive power of the synchronous and the asynchronous π calculus, *Math. Structures in Computer Science*, 13(5): 685–719, 2003.

[26] Gh. Paun, *Membrane computing. An introduction.* Springer, 2002.

[27] G. Plotkin, A powerdomain construction, *SIAM Journal of Computing* 5(3), 452–487, 1976.

[28] J.J.M.M. Rutten, Semantic correctness for a parallel object oriented language, *SIAM Journal of Computing* 19(2), 341–383, 1990.

[29] V. Saraswat, *Concurrent constraint programming*, MIT Press, 1993.

[30] C. Stratchey and C. Wadsworth, Continuations: a mathematical semantics for handling full jumps, *Higher-Order and Symbolic Comput.* 13, 135–152, 2000.

[31] E.N. Todoran, Metric semantics for synchronous and asynchronous communication: a continuation-based approach, *ENTCS* 28, 119–146, 2000.

[32] E.N. Todoran, Comparative semantics for modern communication abstractions, *Proc. IEEE ICCP'08*, 153–160, 2008.

[33] E.N. Todoran and N. Papaspyrou, Continuations for parallel logic programming, *Proc. ACM PPDP'00*, 257–267, 2000.

[34] E.N. Todoran and N. Papaspyrou, Continuations for prototyping concurrent languages, Technical Report CSD-SW-TR-1-06, National Technical University of Athens, 2006. Available at `http://www.softlab.ntua.gr/research/techrep/CSD-SW-TR-1-06.pdf`

[35] *Synchronous and Asynchronous Interaction in Distributed Systems (SAS)*, Project funded by DFG (German Research Foundation), 2010, `http://concurrency-theory.service.tu-berlin.de/joomla/projects/projects`.