

On the observational theory of the CPS-calculus ^{*}

Massimo Merro

Departimento di Informatica, Università di Verona, Italy

Abstract

We apply powerful proof-techniques of concurrency theory to study the observational theory of Thielecke's *CPS-calculus*, a distillation of the target language of Continuation-Passing Style transforms. We define a *labelled transition system* from which we derive a (weak) *labelled bisimilarity* that completely characterises Morris' context-equivalence. We prove a *context lemma* showing that Morris' context-equivalence coincides with a simpler context-equivalence closed under a smaller class of contexts. Then we profit of the determinism of the CPS-calculus to give a simpler labelled characterisation of Morris' equivalence, in the style of Abramsky's *applicative bisimilarity*. We enhance our bisimulation proof-methods with *up to bisimilarity* and *up to context* proof techniques. We use our bisimulation proof techniques to investigate a few algebraic properties on diverging terms that cannot be proved using the original axiomatic semantics of the CPS-calculus.

1 Introduction

Continuations represent a fundamental concept in the semantics of programming languages. In functional languages, a continuation is a parameter of a function that represents the “rest of the computation” [37, 38]. Functions taking continuations as arguments are called *functions in Continuation-Passing Style* (briefly CPS functions), and have a special syntactic form: they terminate their computation by passing the result to the continuation.

A fairly vast literature on functional programming studies transformations of functions into CPS functions. These transformations are called *CPS transforms*. CPS transforms, as syntactic technique for introducing continuations, were first published by Fisher [6] and studied in detail by Plotkin in his seminal paper on call-by-name and call-by-value λ -calculus [27]. Plotkin showed that the evaluation of CPS programs produces correct outputs and that this evaluation yields the same results under call-by-value and call-by-name. Moreover, he established the soundness and the *incompleteness* of the call-by-value λ -calculus for reasoning about CPS

^{*}An extended abstract appeared in the Proceedings of the 22nd International Conference on the Mathematical Foundations of Program Semantics, May 2006, Elsevier Electronic Notes in Theoretical Computer Science 158.

programs. More recently, Sabry and Felleisen [28] proved the completeness of an optimised version of Fisher’s CPS transform, in an untyped setting. A similar result, but relying on types, has been proved by Führmann and Thielecke [7].

The target language of CPS transforms is usually a simple subset of the λ -calculus that admits a very “imperative” reading in terms of *jumping* [35]. Thielecke [41] proposed a target language called CPS-calculus, similar to the intermediate language of Appel’s compiler [3], designed to bring out the jumping, imperative nature of the continuation-passing. Thielecke showed that the more traditional CPS transforms factorise through his calculus.

The CPS-calculus is a small deterministic name-passing calculus. The calculus comes equipped with an *axiomatic semantics* defined as the congruence induced by four simple axioms. Merro and Sangiorgi [17] proved the soundness of those axioms with respect to Milner and Sangiorgi’s *barbed congruence* [21], a standard contextually-defined program equality. Thielecke provided also a categorical account of the structure inherent in first-class continuations building a term model, from the syntax of the CPS-calculus, as an instance of the categorical framework. A more recent account of the state of the art of the axiomatic and categorical semantics in a simply-typed call-by-value setting can be found in [7].

The CPS-calculus consists of two syntactic constructs and one operational rule. However, this is enough to embody a special programming paradigm. The problem with such calculi of minimal expressiveness is the difficulty in developing an handy behavioural theory. In the current paper, we apply powerful proof-technique of concurrency theory to develop an observational theory of the recursive CPS-calculus (the results can be adapted to other variants of the calculus). More precisely, we are interested in establishing when two CPS-term have the same *observable behaviour*, that is, they are indistinguishable in any context. *Behavioural equivalences* are fundamental for justifying program transformations performed either by programmers, during system development, or by the optimising phases of compilers. While several notions of behavioural equivalences can be found in the literature, most of them share two key properties:

- two terms are equivalent only if they offer identical interactions to any environment, that is, they expose the same *observables*;
- the equivalence is preserved by some key constructs of the calculus, as a consequence, proving the equivalence of two large terms can be reduced to proving the equivalence of their components.

A standard notion of behavioural equality is Morris’ context-equivalence [22]. The definition of Morris’ equivalence is simple and intuitive; in practise, however, it is difficult to use as the quantification on all contexts is a heavy proof obligation. Simpler proof techniques are based on *labelled bisimilarities* [23, 19], which are co-inductive relations that characterise the behaviour of processes using a *labelled transition system* (LTS).

Contribution

- In Section 3 we define a higher-order *labelled transition system* (LTS) for the CPS-calculus which captures the possible interactions of a term with its environment, represented by *external jumps* (i.e. jumps to continuations placed within the environment). We prove that the LTS-based semantics coincides with the *reduction semantics* expressed in terms of *internal jumps*.
- In Section 4.1, we use the LTS to define a (weak) labelled bisimilarity in a delay style [42, 30]. We prove that our bisimilarity completely characterises Morris’ context-equivalence. Notice that, in general, congruence proofs for higher-order bisimulations are hard, in particular when the syntax of the calculus is very rigid. Our proof is relatively simple as it relies on up to bisimilarity proof-techniques. As an easy corollary, we derive a *context lemma* showing that Morris’ context-equivalence coincides with a simpler contextually-based equivalence closed under a smaller class of contexts.
- In Section 4.2, we provide a simpler labelled characterisation of Morris’ equivalence in the style of Abramsky’s *applicative bisimilarity* [1]. The proof is quite simple as we profit of the determinism of the calculus to show that bisimilarity and applicative bisimilarity coincide. Applicative bisimilarity is the natural choice when defining a behavioural equality for a confluent/deterministic calculus as the CPS-calculus. On the other hand, a “standard” definition of bisimilarity allows us to derive easier proofs of the characterisation theorem and the context lemma. Had we worked directly with the applicative bisimilarity, the proofs would have been much harder.
- In Section 4.3, we enhance our proof methods by providing *up to context* proof techniques for both bisimilarities. *up to context* proof techniques are very effective to reduce the size of the candidate bisimulation. In particular, these proof techniques are very useful when working with higher-order bisimulations to factor out the universally quantified processes provided by the environment.
- In Section 5 we investigate a number of properties of divergent terms. We use our bisimulation-based proof techniques to prove a number of algebraic laws that cannot be derived using Thielecke’s axiomatic semantics.

2 The CPS-calculus

2.1 Syntax and reduction semantics

The CPS-calculus is very simple and low-level: only variables can be passed as arguments, and applications are like jumps, with variables as argument. The terms of the CPS-calculus are given by the following grammar:

$$M, N ::= a\langle b \rangle \mid M\{a\langle b \rangle \Leftarrow N\}$$

where lowercase letters a, b, c, \dots range over variables (names) and uppercase letter L, M, N, \dots range over terms. The intended meaning is that $a\langle b \rangle$ is a jump to the continuation a with actual parameter b , while $M\{a\langle b \rangle \Leftarrow N\}$ binds the continuation with body N and formal parameter b to a in M .

We study the *monadic and recursive* variant of the calculus, in that jumps have a single argument, and in a term $M\{a\langle b \rangle \Leftarrow N\}$ the sub-term N may refer to itself under a . More precisely, in a term $M\{a\langle b \rangle \Leftarrow N\}$ the scope of variable a comprehends both M and N , while that of b extends to N only.

Remark 2.1 *The theory developed in this article can be easily extended to the polyadic variant of the CPS-calculus, where a jump may contain several parameters.*

The set of free variables $\text{fv}(M)$ of a CPS term M is defined as follows.

- $\text{fv}(a\langle b \rangle) \stackrel{\text{def}}{=} \{a, b\}$
- $\text{fv}(M\{a\langle b \rangle \Leftarrow N\}) \stackrel{\text{def}}{=} (\text{fv}(M) \setminus \{a\}) \cup (\text{fv}(N) \setminus \{a, b\})$

We write $\text{fv}(M, N)$ as an abbreviation for $\text{fv}(M) \cup \text{fv}(N)$. In a jump $a\langle b \rangle$ we say that a is in *subject* and b in *object* position. We write $M[a/b]$ for the capture avoiding *substitution* of variable of a for each free occurrence of variable b in M . So, alpha-conversion can be formally defined by the following two equations

$$\begin{aligned} M\{a\langle b \rangle \Leftarrow N\} &= M\{a\langle c \rangle \Leftarrow N[c/b]\} && \text{for } c \notin \text{fv}(N) \\ M\{a\langle b \rangle \Leftarrow N\} &= M[c/a]\{c\langle b \rangle \Leftarrow N[c/a]\} && \text{for } c \notin \text{fv}(M) \cup \text{fv}(N). \end{aligned}$$

We will identify terms up to alpha-conversion.

We propose the *reduction semantics* of [17] which is a slight variant of the operational semantics given by Thielecke. For this purpose, notice that every CPS-term is in the form

$$a\langle b \rangle \{a_1\langle b_1 \rangle \Leftarrow M_1\} \dots \{a_n\langle b_n \rangle \Leftarrow M_n\}$$

for some $n \geq 0$. This allows us to model the behaviour of CPS-terms by means of just one (global) reduction rule:

$$\begin{aligned} a_i\langle b \rangle \{a_1\langle b_1 \rangle \Leftarrow M_1\} \dots \{a_i\langle b_i \rangle \Leftarrow M_i\} \dots \{a_n\langle b_n \rangle \Leftarrow M_n\} \\ \rightarrow \\ M_i[b/b_i]\{a_1\langle b_1 \rangle \Leftarrow M_1\} \dots \{a_i\langle b_i \rangle \Leftarrow M_i\} \dots \{a_n\langle b_n \rangle \Leftarrow M_n\} \end{aligned}$$

with $1 \leq i \leq n$ and $a_j \notin \text{fv}(M_i)$, for $1 \leq j < i$. We denote with \rightarrow^* the reflexive and transitive closure of \rightarrow .

2.2 Behavioural semantics

In operational semantics two terms are deemed equivalent if they have the same observable behaviour in all contexts. In the CPS-calculus the notion of observability is represented by the “external” jump that a term can perform to interact with the context. We define an *observability predicate* \Downarrow_a , for each variable a , which detects the possibility of a term to interact with the environment via a . For instance, in a jump of the form $a\langle b \rangle$, we can observe (the occurrence of a jump to) a , whereas the argument b does not play any direct role. More generally, a free variable in the leftmost position can be observed.

Definition 2.2 (Observability/convergence) *Let M be a term of the CPS-calculus and a be a name, we say that M converges to a , written $M \Downarrow_a$, if there are names $b, a_1, b_1, \dots, a_n, b_n$, for some integer $n \geq 0$ with $a \neq a_i$, for all $1 \leq i \leq n$, such that $M = a\langle b \rangle \{a_1\langle b_1 \rangle \Leftarrow M_1\} \dots \{a_n\langle b_n \rangle \Leftarrow M_n\}$. We say that M (weakly) converges to a , written $M \Downarrow_a$, if there exists a CPS-term N such that $M \rightarrow^* N \Downarrow_a$.*

The definition of divergence is straightforward.

Definition 2.3 (Divergence) *A CPS-term M diverges, written $M \Uparrow$, if whenever there is a term M' such that $M \rightarrow^* M'$ then there is also a term M'' such that $M' \rightarrow M''$.*

Obviously, since the calculus is deterministic, a term either diverges or (weakly) converges to some name a .

In order to define contextually-based equivalences we need to specify what is a context. A (monadic) *context* $C[\cdot]$ is a CPS-term with a hole, denoted by $[\cdot]$. CPS-contexts are generated by the following grammar:

$$C[\cdot] ::= [\cdot] \mid C[\cdot]\{a\langle x \rangle \Leftarrow M\} \mid M\{a\langle x \rangle \Leftarrow C[\cdot]\}.$$

A *static context* is a context that can be generated only using the first two productions of the grammar above.

Now, everything is in place to define Morris’ context-equivalences for the CPS-calculus.

Definition 2.4 (Observational equalities) *Let M and N be two CPS terms. We say that M and N are observationally equivalent, written $M \simeq N$, if for all static contexts $C[\cdot]$ and variables a , it holds that $C[M] \Downarrow_a$ iff $C[N] \Downarrow_a$. M and N are observationally congruent, written $M \cong N$, if for all contexts $C[\cdot]$ and variables a , it holds that $C[M] \Downarrow_a$ iff $C[N] \Downarrow_a$.*

The idea is that two terms are equivalent if no amount of programming can tell them apart; obviously, $\cong \subseteq \simeq$.

Barbed equalities are branching-time contextually-based equivalences introduced by Milner and Sangiorgi in the realm of concurrent processes [21]. Their definitions can be easily adapted to the CPS-calculus.

Definition 2.5 (Barbed equalities) A symmetric relation \mathcal{S} on CPS-terms is a barbed bisimulation if $M \mathcal{S} N$ implies:

1. If $M \rightarrow M'$ then there exists N' such that $N \rightarrow^* N'$ and $M' \mathcal{S} N'$.
2. If $M \Downarrow_a$, for some a , then $N \Downarrow_a$.

Two terms M and N are barbed bisimilar, written $M \dot{\approx} N$, if $M \mathcal{S} N$ for some barbed bisimulation \mathcal{S} . Two terms M and N are barbed equivalent if for each static context $C[\cdot]$, it holds that $C[M] \dot{\approx} C[N]$; they are barbed congruent if for each context $C[\cdot]$, it holds that $C[M] \dot{\approx} C[N]$.

However, as the CPS-calculus is deterministic (and hence confluent), it is well-known that observational congruence (respectively, observational equivalence) coincides with barbed congruence (respectively, barbed equivalence) [19].

2.3 Axiomatic semantics

The original semantics [41] of the CPS-calculus is given in terms of an *axiomatic semantics* defined as the congruence induced by the following four axioms:

- (DISTR) $L\{a\langle b \rangle \Leftarrow M\}\{c\langle d \rangle \Leftarrow N\} \equiv L\{c\langle d \rangle \Leftarrow N\}\{a\langle b \rangle \Leftarrow M\{c\langle d \rangle \Leftarrow N\}\}$
where $a \neq c$ and $a, b \notin \text{fv}(N)$
- (GC) $a\langle b \rangle\{c\langle d \rangle \Leftarrow N\} \equiv a\langle b \rangle$ where $c \notin \text{fv}(a\langle b \rangle)$
- (JMP) $a\langle b \rangle\{a\langle c \rangle \Leftarrow N\} \equiv N[b/c]\{a\langle c \rangle \Leftarrow N\}$
- (ETA) $M\{a\langle b \rangle \Leftarrow c\langle b \rangle\} \equiv M[c/a]$ where $a \neq c$.

The (JMP) axiom is in some sense what drives the computation. In fact our reduction rule can be seen as a “contextual” variant of the (JMP) axiom. The axiom (GC) allows us to garbage collect unreachable continuations, whereas (DISTR) is a kind of structural law similar to those for process calculi [20]. Most of the axioms above appear in [3]. We write $\text{CPS} \vdash M \equiv N$ when the axiomatic semantics can be used to derive the equality between M and N .

The axiomatic semantics is sound with respect to barbed congruence, and hence also with respect to observational congruence.

Theorem 2.6 (Merro and Sangiorgi [17]) Let M and N be two CPS-terms. Then,

$$\text{CPS} \vdash M \equiv N \text{ implies } M \cong N .$$

Notice that the axiomatic semantics is meant to be the least equality on CPS-terms that one would wish to impose, while observational equivalence is arguably the greatest such notion one could consider.

Table 1 Labelled Transition System for the CPS-calculus

$$\begin{array}{c}
\text{(Jump)} \frac{x \notin \text{fv}(a\langle b \rangle)}{a\langle b \rangle \xrightarrow{a\langle x \rangle M} M[b/x]\{a\langle x \rangle \Leftarrow M\}} \qquad \text{(Tau)} \frac{M \xrightarrow{a\langle x \rangle N} M'}{M\{a\langle x \rangle \Leftarrow N\} \xrightarrow{\tau} M'} \\
\text{(Cxt Tau)} \frac{M \xrightarrow{\tau} M'}{M\{a\langle x \rangle \Leftarrow N\} \xrightarrow{\tau} M'\{a\langle x \rangle \Leftarrow N\}} \\
\text{(Cxt Jump)} \frac{M \xrightarrow{a\langle x \rangle N} M'\{a\langle x \rangle \Leftarrow N\} \quad a \neq b \quad b \notin \text{fv}(N)}{M\{b\langle y \rangle \Leftarrow O\} \xrightarrow{a\langle x \rangle N} M'\{b\langle y \rangle \Leftarrow O\}\{a\langle x \rangle \Leftarrow N\}}
\end{array}$$

3 A Labelled Transition System

In Table 1 we provide a *labelled transition system* (LTS) for the CPS-calculus. Transitions are of the form $M \xrightarrow{\alpha} M'$ where α can be either τ , to model *internal jumps*, or $a\langle x \rangle N$, for some variable a and CPS-term N , to model *external jumps*. In particular, the observable action $a\langle x \rangle N$ models the capability to perform an external jump $a\langle b \rangle$, for some parameter b . Notice that our actions do not mention the argument of the jump (in this case b), although such argument has its influence on the derivative M' . Intuitively, in a transition

$$M \xrightarrow{a\langle x \rangle N} M'$$

the action $a\langle x \rangle N$ codifies the discriminating context $[\cdot]\{a\langle x \rangle \Leftarrow N\}$ with which M can interact. This context then becomes part of the derivative of the transition. The main inference rules are (Jump) and (Tau), modelling external and internal jumps, respectively. Rules (Cxt Jump) and (Cxt Tau) are their contextual counterparts.

Our LTS is necessarily higher-order to properly model the interaction with the environment while preserving the transition closure of the calculus. In the sequel, we write $\xrightarrow{\tau}^*$ to denote the reflexive and transitive closure of $\xrightarrow{\tau}$.

Proposition 3.1 (Transition closure) *Let M be a CPS-term. If $M \xrightarrow{\alpha} M'$ then M' is a CPS-term.*

Proof By a simple transition induction. □

The calculus is deterministic in the following sense.

Proposition 3.2 (Determinism) *Let M be a CPS-term. Then, only one of the following two cases applies.*

1. M may perform at most one transition of the form $M \xrightarrow{\tau} M'$, for some M' .

2. Fixed an arbitrary term L , the term M may perform at most one transition of the form $M \xrightarrow{a\langle x \rangle L} M'$, for some variable a and term M' .

Proof It follows from the definition of the LTS and because every CPS-term is in the form

$$a\langle b \rangle \{a_1\langle b_1 \rangle \Leftarrow M_1\} \dots \{a_n\langle b_n \rangle \Leftarrow M_n\}.$$

□

The next result shows that the LTS-based semantics coincides with the reduction semantics of Section 2.1.

Theorem 3.3 (Harmony Theorem) *Let M and N be two CPS-terms. Then,*

1. $M \Downarrow_a$ iff $M \xrightarrow{a\langle x \rangle L} M'\{a\langle x \rangle \Leftarrow L\}$ for any CPS-terms L .
2. $M \rightarrow N$ iff $M \xrightarrow{\tau} N$.

Proof See the Appendix. □

The following result says that the operational semantics is preserved under (name) substitution.

Proposition 3.4

1. If $M \xrightarrow{\tau} M'$ then $M\sigma \xrightarrow{\tau} M'\sigma$, for any substitution σ .
2. If $M \Downarrow_a$ and $a \neq x$ then $M[b/x] \Downarrow_a$.

Proof The first item is a standard result of name-passing calculi, like the CPS-calculus. The second part follows from the first one and Theorem 3.3(2). In fact, if $M \Downarrow_a$ then, by Theorem 3.3(2), there is M' such that $M \xrightarrow{\tau}^* M' \Downarrow_a$. As name substitution does not affect silent actions it follows that $M[b/x] \xrightarrow{\tau}^* M'[b/x] \Downarrow_a$. □

4 Bisimulation proof methods

In this section, we propose two labelled characterisations of Morris' context-equivalence. We then prove a context lemma showing that static contexts have the same discriminating power as full contexts. As a consequence, observational congruence and observational equivalence coincide. Finally, we enhance our proof methods with up to context proof techniques.

4.1 A labelled characterisation of Morris' context-equivalence

Starting from the labelled transition system we can define our notion of bisimulation for CPS-terms. We define *weak transitions in delay style* [42, 30] where \Rightarrow denotes the reflexive and transitive closure of $\xrightarrow{\tau}$. We write $\xRightarrow{\alpha}$ for $\Rightarrow \xrightarrow{\alpha}$, and $\xRightarrow{\hat{\alpha}}$ for $\xRightarrow{\alpha}$ if $\alpha \neq \tau$, and for \Rightarrow if $\alpha = \tau$.

Definition 4.1 (Bisimulation) A symmetric relation \mathcal{S} on CPS-terms is a bisimulation if whenever $M \mathcal{S} N$ and $M \xrightarrow{\alpha} M'$ there exists a CPS-term N' such that $N \xrightarrow{\hat{\alpha}} N'$ and $M' \mathcal{S} N'$. Two CPS-terms M and N are bisimilar, written $M \approx N$ if there is some bisimulation \mathcal{S} such that $M \mathcal{S} N$.

It is easy to see that \approx is an equivalence relation.

In order to show that our bisimilarity characterises the observational congruence we first prove the completeness of the bisimilarity with respect to the *observational equivalence* (and not the observational congruence). Notice that in general a result of this kind would not hold in concurrency theory. However, in our case, Lemma 4.2, on the insensitiveness of behavioural equalities to τ -actions, allows us to easily prove the completeness result.

Lemma 4.2 (Insensitiveness to τ -actions) Let M be a CPS-term. If $M \xrightarrow{\tau} M'$ then $M \cong M'$.

Proof By Theorem 3.3(2) the relations \rightarrow and $\xrightarrow{\tau}$ coincide. As a consequence, there are an integer n and variables b, a_i, b_i, M_i , for $1 \leq i \leq n$, such that

$$\begin{aligned} M &= a_i \langle b \rangle \{a_1 \langle b_1 \rangle \Leftarrow M_1\} \dots \{a_i \langle b_i \rangle \Leftarrow M_i\} \dots \{a_n \langle b_n \rangle \Leftarrow M_n\} \\ M' &= M_i [b/b_i] \{a_1 \langle b_1 \rangle \Leftarrow M_1\} \dots \{a_i \langle b_i \rangle \Leftarrow M_i\} \dots \{a_n \langle b_n \rangle \Leftarrow M_n\}. \end{aligned}$$

Now, we prove $\text{CPS} \vdash M \equiv M'$ using the axioms of the previous section. In particular, M' can be derived from M by applying $i-1$ times the axiom (DISTR) to shift the continuation $\{a_i \langle x_i \rangle \Leftarrow M_i\}$ at the extreme left, once axiom (JMP) to reduce along variable a_i , and $i-1$ times the axiom (DISTR) to put back the $\{a_i \langle b_i \rangle \Leftarrow M_i\}$ at its original place. By Theorem 2.6 we obtain $M \cong M'$. \square

Lemma 4.3 (Completeness of \approx w.r.t. \simeq) Let M and N be two CPS-terms. Then $M \simeq N$ implies $M \approx N$.

Proof We prove that the relation \simeq is a bisimulation. Let $M \simeq N$.

- Suppose that $M \xrightarrow{\tau} M'$. By Lemma 4.2 we have $M \cong M'$ and hence also $M \simeq M'$. Now, let us choose as matching transition $N \Rightarrow N$. By transitivity of \simeq it follows that $M' \simeq N$.
- Suppose that $M \xrightarrow{a \langle x \rangle L} M' \{a \langle x \rangle \Leftarrow L\}$. By Theorem 3.3(1) we have $M \downarrow_a$. As $M \simeq N$ we also have $N \downarrow_a$. By several applications of Theorem 3.3(2) and one application of Theorem 3.3(1) we have $N \xrightarrow{a \langle x \rangle L} N' \{a \langle x \rangle \Leftarrow L\}$. As $M \simeq N$ and \simeq is preserved by all static contexts, we also have $M \{a \langle x \rangle \Leftarrow L\} \simeq N \{a \langle x \rangle \Leftarrow L\}$. As $M \xrightarrow{a \langle x \rangle L} M' \{a \langle x \rangle \Leftarrow L\}$ and $N \xrightarrow{a \langle x \rangle L} N' \{a \langle x \rangle \Leftarrow L\}$, by several applications of rules (Tau) and (Cxt Tau) we have $M \{a \langle x \rangle \Leftarrow L\} \xrightarrow{\tau} M' \{a \langle x \rangle \Leftarrow L\}$ and $N \{a \langle x \rangle \Leftarrow L\} \Rightarrow N' \{a \langle x \rangle \Leftarrow L\}$. By applying Lemma 4.2 on the two derivations we obtain $M' \{a \langle x \rangle \Leftarrow L\} \simeq N' \{a \langle x \rangle \Leftarrow L\}$. \square

As to the proof of the soundness, the main difficulty resides in proving that \approx is preserved by all contexts. A direct proof of that is far from trivial, due to the quite rigid syntax of the calculus. To this purpose we define an up to (weak) bisimilarity proof technique. It is well known that, in general, this proof technique is not sound [32, 19]. However, in our case, we use a stronger definition, along the lines of Exercise 2.4.64 of [33].

Definition 4.4 (Bisimulation up to \approx) *A symmetric relation \mathcal{S} over CPS-terms is a bisimulation up to \approx if $M \mathcal{S} N$ implies,*

1. *whenever $M \xrightarrow{\tau} M'$ then, for some N' , $N \Rightarrow N'$ and $M' \mathcal{S} \approx N'$*
2. *whenever $M \xrightarrow{\alpha} M'$, $\alpha \neq \tau$, then, for some N' , $N \xRightarrow{\alpha} N'$ and $M' \approx \mathcal{S} \approx N'$.*

Lemma 4.5 *If \mathcal{S} is a bisimulation up to \approx then $\mathcal{S} \subseteq \approx$.*

Proof See the Appendix. □

Now, everything is in place to prove that \approx is a congruence. Our proof relies on the up to-bisimilarity proof technique, the axiom (DISTR) for permuting continuations, and the axiom (ETA) for encoding substitutions. We then use Theorem 2.6 and Lemma 4.3 to validate these two axioms with respect to bisimilarity.

Lemma 4.6 (\approx is preserved by all contexts) *Let M and N be two CPS-terms such that $M \approx N$. Then,*

1. *$M\{a\langle b \rangle \Leftarrow O\} \approx N\{a\langle b \rangle \Leftarrow O\}$ for all terms O and variables a and b .*
2. *$O\{a\langle b \rangle \Leftarrow M\} \approx O\{a\langle b \rangle \Leftarrow N\}$ for all terms O and variables a and b .*

Proof Let us prove that $M\{a\langle b \rangle \Leftarrow O\} \approx N\{a\langle b \rangle \Leftarrow O\}$. Let \mathcal{S} be the relation defined as:

$$\mathcal{S} \stackrel{\text{def}}{=} \{(M\{a\langle x \rangle \Leftarrow O\}, N\{a\langle x \rangle \Leftarrow O\}) \text{ for all } a \text{ and } O \text{ s.t. } M \approx N\} \cup \approx$$

We prove that \mathcal{S} is a bisimulation up to \approx . We do a case analysis on the transition $M\{a\langle x \rangle \Leftarrow O\} \xrightarrow{\alpha} M'$.

- Let $M\{a\langle x \rangle \Leftarrow O\} \xrightarrow{\tau} M' = M''\{a\langle x \rangle \Leftarrow O\}$, by an application of rule (Cxt Tau), because $M \xrightarrow{\tau} M''$. Since $M \approx N$ there is N'' such that $N \Rightarrow N''$ with $M'' \approx N''$. By several applications of rule (Cxt Tau), we get $N\{a\langle x \rangle \Leftarrow O\} \Rightarrow N''\{a\langle x \rangle \Leftarrow O\}$. By definition of \mathcal{S} it follows that

$$(M''\{a\langle x \rangle \Leftarrow O\}, N''\{a\langle x \rangle \Leftarrow O\}) \in \mathcal{S}.$$

- Let $M\{a\langle x \rangle \Leftarrow O\} \xrightarrow{\tau} M'$, by an application of rule (Tau), because $M \xrightarrow{a\langle x \rangle O} M'$. Since $M \approx N$ there is N' such that $N \xRightarrow{a\langle x \rangle O} N'$ and $M' \approx N'$. By several applications of rule (Cxt Tau) and one application of rule (Tau) we have $N\{a\langle x \rangle \Leftarrow O\} \Rightarrow N'$. By definition of \mathcal{S} it follows that $(M', N') \in \mathcal{S}$.

- Let $M\{a\langle x \rangle \Leftarrow O\} \xrightarrow{b\langle y \rangle O'} M'$, by an application of rule (Cxt Jmp), because $M \xrightarrow{b\langle y \rangle O'} M''\{b\langle y \rangle \Leftarrow O'\}$, with $M' = M''\{a\langle x \rangle \Leftarrow O\}\{b\langle y \rangle \Leftarrow O'\}$. As $M \approx N$ there is N'' such that $N \xrightarrow{b\langle y \rangle O'} N''\{b\langle y \rangle \Leftarrow O'\}$ with

$$M''\{b\langle y \rangle \Leftarrow O'\} \approx N''\{b\langle y \rangle \Leftarrow O'\} .$$

By applying rules (Cxt Tau) and (Cxt Jmp) we get

$$N\{a\langle x \rangle \Leftarrow O\} \xrightarrow{b\langle y \rangle O'} N''\{a\langle x \rangle \Leftarrow O\}\{b\langle y \rangle \Leftarrow O'\} .$$

So, we have to prove that

$$M''\{a\langle x \rangle \Leftarrow O\}\{b\langle y \rangle \Leftarrow O'\} \approx_{\mathcal{S}} N''\{a\langle x \rangle \Leftarrow O\}\{b\langle y \rangle \Leftarrow O'\} .$$

As $M \xrightarrow{b\langle y \rangle O'} M''\{b\langle y \rangle \Leftarrow O'\}$, by applying rule (Tau) we obtain

$$M\{b\langle y \rangle \Leftarrow O'\} \xrightarrow{\tau} M''\{b\langle y \rangle \Leftarrow O'\} .$$

By an application of rule (Cxt Tau) we get

$$M\{b\langle y \rangle \Leftarrow O'\}\{a\langle x \rangle \Leftarrow O\{b\langle y \rangle \Leftarrow O'\}\} \xrightarrow{\tau} M''\{b\langle y \rangle \Leftarrow O'\}\{a\langle x \rangle \Leftarrow O\{b\langle y \rangle \Leftarrow O'\}\} .$$

By Lemmas 4.2, the inclusion of $\cong \subseteq \simeq$, and Lemma 4.3 it follows that

$$M\{b\langle y \rangle \Leftarrow O'\}\{a\langle x \rangle \Leftarrow O\{b\langle y \rangle \Leftarrow O'\}\} \approx M''\{b\langle y \rangle \Leftarrow O'\}\{a\langle x \rangle \Leftarrow O\{b\langle y \rangle \Leftarrow O'\}\} \quad (1)$$

With a similar reasoning, from $N \xrightarrow{b\langle y \rangle O'} N''\{b\langle y \rangle \Leftarrow O'\}$, by several applications of rules (Cxt Tau) and one application of rule (Tau) we get

$$N\{b\langle y \rangle \Leftarrow O'\} \Rightarrow N''\{b\langle y \rangle \Leftarrow O'\} .$$

By several applications of rule (Cxt Tau) we obtain

$$N\{b\langle y \rangle \Leftarrow O'\}\{a\langle x \rangle \Leftarrow O\{b\langle y \rangle \Leftarrow O'\}\} \Rightarrow N''\{b\langle y \rangle \Leftarrow O'\}\{a\langle x \rangle \Leftarrow O\{b\langle y \rangle \Leftarrow O'\}\} .$$

By Lemmas 4.2, the inclusion $\cong \subseteq \simeq$, Lemma 4.3, and the transitivity of \approx it follows that

$$N\{b\langle y \rangle \Leftarrow O'\}\{a\langle x \rangle \Leftarrow O\{b\langle y \rangle \Leftarrow O'\}\} \approx N''\{b\langle y \rangle \Leftarrow O'\}\{a\langle x \rangle \Leftarrow O\{b\langle y \rangle \Leftarrow O'\}\} \quad (2)$$

Now, by axiom (DISTR), Theorem 2.6, the inclusion $\cong \subseteq \simeq$, and Lemma 4.3 we get

$$M''\{a\langle x \rangle \Leftarrow O\}\{b\langle y \rangle \Leftarrow O'\} \approx M''\{b\langle y \rangle \Leftarrow O'\}\{a\langle x \rangle \Leftarrow O\{b\langle y \rangle \Leftarrow O'\}\} \quad (3)$$

$$N''\{a\langle x \rangle \Leftarrow O\}\{b\langle y \rangle \Leftarrow O'\} \approx N''\{b\langle y \rangle \Leftarrow O'\}\{a\langle x \rangle \Leftarrow O\{b\langle y \rangle \Leftarrow O'\}\} \quad (4)$$

Finally, since $M''\{b\langle y \rangle \Leftarrow O'\} \approx N''\{b\langle y \rangle \Leftarrow O'\}$, by definition of \mathcal{S} it follows that

$$M''\{a\langle x \rangle \Leftarrow O\}\{b\langle y \rangle \Leftarrow O'\} \approx_{\mathcal{S}} N''\{a\langle x \rangle \Leftarrow O\}\{b\langle y \rangle \Leftarrow O'\} .$$

This concludes the proof of first part of the statement.

Let us prove now that $M \approx N$ implies $O\{a\langle x \rangle \Leftarrow M\} \approx O\{a\langle x \rangle \Leftarrow N\}$. We show that the relation

$$\mathcal{S} \stackrel{\text{def}}{=} \{(O\{a\langle x \rangle \Leftarrow M\}, O\{a\langle x \rangle \Leftarrow N\}) : \text{for all } a, M, N, O \text{ such that } M \approx N\} \cup \approx$$

is a bisimulation up to \approx . We do a case analysis on the transition $O\{a\langle x \rangle \Leftarrow M\} \xrightarrow{\alpha} M'$.

- Let $O\{a\langle x \rangle \Leftarrow M\} \xrightarrow{\tau} O'\{a\langle x \rangle \Leftarrow M\}$, because $O \xrightarrow{\tau} O'$ by an application of rule (Cxt Tau). This case is easy.
- Let $O\{a\langle x \rangle \Leftarrow M\} \xrightarrow{\tau} M'$, by an application of rule (Tau), because

$$O \xrightarrow{a\langle x \rangle M} C[M[b/x]]\{a\langle x \rangle \Leftarrow M\} = M',$$

for some variable b and some context

$$C[\cdot] = [\cdot]\{a_1\langle x_1 \rangle \Leftarrow M_1\} \dots \{a_n\langle x_n \rangle \Leftarrow M_n\}.$$

As a consequence, there is N' such that

$$O \xrightarrow{a\langle x \rangle N} C[N[b/x]]\{a\langle x \rangle \Leftarrow N\} = N'$$

and hence, by an application of rule (Tau), we have

$$O\{a\langle x \rangle \Leftarrow N\} \xrightarrow{\tau} N'.$$

As $M \approx N$, by applying the first part of the current lemma we derive

$$M\{x\langle y \rangle \Leftarrow b\langle y \rangle\} \approx N\{x\langle y \rangle \Leftarrow b\langle y \rangle\}.$$

By applying in sequence the axiom (ETA), Theorem 2.6, the inclusion $\cong \subseteq \simeq$, and Lemma 4.3 we obtain:

- $M[b/x] \approx M\{x\langle y \rangle \Leftarrow b\langle y \rangle\}$
- $N[b/x] \approx N\{x\langle y \rangle \Leftarrow b\langle y \rangle\}.$

By the transitivity of \approx we derive $M[b/x] \approx N[b/x]$. As $C[\cdot]$ is a static context, by several applications of the first part of the current lemma we obtain

$$C[M[b/x]] \approx C[N[b/x]].$$

Again, by an application of the first part of the current lemma we derive $C[M[b/x]]\{a\langle x \rangle \Leftarrow N\} \approx C[N[b/x]]\{a\langle x \rangle \Leftarrow N\}$. We recall that

$$M' = C[M[b/x]]\{a\langle x \rangle \Leftarrow M\} \text{ and } N' = C[N[b/x]]\{a\langle x \rangle \Leftarrow N\}.$$

This allows us to conclude that $M' \mathcal{S} \approx N'$.

- Let $O\{a\langle x \rangle \Leftarrow M\} \xrightarrow{b\langle y \rangle L} M'$, by an application of rule (Cxt Jmp), because $O \xrightarrow{b\langle y \rangle L} O'\{b\langle y \rangle \Leftarrow L\}$, with $M' = O'\{a\langle x \rangle \Leftarrow M\}\{b\langle y \rangle \Leftarrow L\}$. By an application of rule (Cxt Jmp), we have $O\{a\langle x \rangle \Leftarrow N\} \xrightarrow{b\langle y \rangle L} N'$, with $N' = O'\{a\langle x \rangle \Leftarrow N\}\{b\langle y \rangle \Leftarrow L\}$. By applying in sequence the axiom (DISTR), Theorem 2.6, the inclusion $\cong \subseteq \simeq$, and Lemma 4.3 we obtain:

$$\begin{aligned} & - O'\{a\langle x \rangle \Leftarrow M\}\{b\langle y \rangle \Leftarrow L\} \approx O'\{b\langle y \rangle \Leftarrow L\}\{a\langle x \rangle \Leftarrow M\{b\langle y \rangle \Leftarrow L\}\} \\ & - O'\{a\langle x \rangle \Leftarrow N\}\{b\langle y \rangle \Leftarrow L\} \approx O'\{b\langle y \rangle \Leftarrow L\}\{a\langle x \rangle \Leftarrow N\{b\langle y \rangle \Leftarrow L\}\}. \end{aligned}$$

As $M \approx N$, by the first part of the current lemma we also have $M\{b\langle y \rangle \Leftarrow L\} \approx N\{b\langle y \rangle \Leftarrow L\}$. As a consequence,

$$O'\{a\langle x \rangle \Leftarrow M\}\{b\langle y \rangle \Leftarrow L\} \approx \mathcal{S} \approx O'\{a\langle x \rangle \Leftarrow N\}\{b\langle y \rangle \Leftarrow L\} .$$

□

We can now prove the characterisation result.

Theorem 4.7 (Characterisation of \cong) *Let M and N be two CPS-terms. Then $M \approx N$ iff $M \cong N$.*

Proof As to the implication from left to right, by Lemma 4.6 we have $C[M] \approx C[N]$, for all contexts $C[\cdot]$. By Definition 4.1 and Theorem 3.3 we derive $C[M] \Downarrow_a$ iff $C[N] \Downarrow_a$. The implication from right to left follows from Lemma 4.3 and the fact that $\cong \subseteq \simeq$. □

An easy consequence of the previous result and Lemma 4.3 is the following.

Theorem 4.8 (Context lemma) *The relations \simeq and \cong coincide.*

4.2 Applicative bisimilarity

As our equivalences are insensitive to τ -actions (Lemma 4.2) we can simplify the definition of bisimulation by removing the clause on τ -actions. In this manner, we basically get a kind of *applicative bisimilarity*.

Definition 4.9 (Applicative bisimilarity) *A symmetric relation \mathcal{S} on CPS-terms is an applicative bisimulation if whenever $M \mathcal{S} N$ and $M \xrightarrow{\alpha} M'$, $\alpha \neq \tau$, then there exists a CPS-term N' such that $N \xrightarrow{\alpha} N'$ and $M' \mathcal{S} N'$. Two CPS-terms M and N are applicative bisimilar, written $M \approx_A N$, if there is some applicative bisimulation \mathcal{S} such that $M \mathcal{S} N$.*

In general, applicative bisimulations are smaller in size than bisimulations as they allow us to collapse terms that differ only for τ -actions. It is easy to show that the applicative bisimilarity is an equivalence relation.

Here, we prove that bisimilarity and applicative bisimilarity coincide.

Theorem 4.10 *The relations \approx and \approx_A coincide.*

Proof Let us prove that $\approx \subseteq \approx_A$. We show that the relation \approx is an applicative bisimulation. Let $M \approx N$ and $M \xrightarrow{\alpha} M'$, $\alpha \neq \tau$. Then, by definition there is M'' such that $M \Rightarrow M'' \xrightarrow{\alpha} M'$. By Lemma 4.2 and Lemma 4.3 we have $M'' \approx M \approx N$. As $M'' \approx N$ there is N' such that $N \xrightarrow{\alpha} N'$ and $M' \approx N'$.

Let us prove now that $\approx_A \subseteq \approx$. We show that the relation \approx_A is a bisimulation. Let $M \approx_A N$.

1. If $M \xrightarrow{\tau} M'$, then by Lemma 4.2, Lemma 4.3, and the inclusion $\approx \subseteq \approx_A$ we derive $M \approx_A M'$. As matching transition we choose $N \Rightarrow N$. By $M \approx_A N$ and the transitivity of \approx_A we obtain $M' \approx_A N$.
2. If $M \xrightarrow{\alpha} M'$, as $M \approx_A N$, there is N' such that $N \xrightarrow{\alpha} N'$ and $M' \approx_A N'$.

□

Recapitulating, all behavioural equivalences defined up to now coincide.

Corollary 4.11 *The relations \cong , \simeq , \approx , and \approx_A coincide.*

4.3 up to context proof techniques

In this section we introduce *up to context* proof techniques [34, 31] for both bisimilarity and applicative bisimilarity. When comparing terms in higher-order calculi, (equipped with a higher-order LTS) up to context proof techniques are very useful to reduce the size of the candidate bisimulation. Intuitively, these techniques allow us to strip off a common context from the terms under consideration.

Remark 4.12 *up to context techniques are particularly useful when working with applicative bisimulations. However, it is technically easier to prove the correctness of these techniques with respect to the notion of bisimulation. The correctness of the up to context technique for applicative bisimulation is an easy consequence of that for bisimulation.*

Definition 4.13 (Bisimulation up to context and up to \approx) *A symmetric relation \mathcal{S} over CPS-terms is a bisimulation up to context and up to \approx if whenever $M \mathcal{S} N$ and $M \xrightarrow{\alpha} M''$, there is a term N'' such that $N \xrightarrow{\hat{\alpha}} N''$ and there is a static context $C[\cdot]$, and terms M' and N' such that $M'' \approx C[M']$, $C[N'] \approx N''$ and $M' \mathcal{S} N'$.*

In order to prove the soundness of the above proof technique we need a technical lemma.

Lemma 4.14 *Let \mathcal{R} be a bisimulation up to context and up to \approx . If $M \mathcal{R} N$ and for some static context $C[\cdot]$ and term M'' it holds that $C[M] \xrightarrow{\alpha} M''$, then there exists a term N'' such that $C[N] \xrightarrow{\hat{\alpha}} N''$ and there are a static context $C'[\cdot]$ and terms M' and N' such that $M'' \approx C'[M']$, $C'[N'] \approx N''$ and $M' \mathcal{R} N'$.*

Proof See the Appendix. \square

The previous lemma can be easily generalised to the weak case.

Lemma 4.15 *Let \mathcal{R} be a bisimulation up to context and up to \approx . If $M \mathcal{R} N$ and for some static context $C[\cdot]$ and term M'' it holds that $C[M] \xrightarrow{\alpha} M''$, then there exists a term N'' such that $C[N] \xrightarrow{\hat{\alpha}} N''$ and there are a static context $C'[\cdot]$ and terms M' and N' such that $M'' \approx C'[M']$, $C'[N'] \approx N''$ and $M' \mathcal{R} N'$.*

Proof The result follows by induction on the length of the transition $C[M] \xrightarrow{\alpha} M''$, using Lemma 4.14. \square

Theorem 4.16 *If \mathcal{R} is a bisimulation up to context and up to \approx , then $\mathcal{R} \subseteq \approx$.*

Proof We recall that we only use static contexts. The proof consists in showing that the relation

$$\mathcal{S} \stackrel{\text{def}}{=} \{(M, N) : \exists C[\cdot], M', N', \text{ such that } M \approx C[M'], C[N'] \approx N, \text{ and } M' \mathcal{R} N'\}$$

is a bisimulation.

Suppose $(M, N) \in \mathcal{S}$ and $M \xrightarrow{\alpha} M_1$. Since $(M, N) \in \mathcal{S}$ there exist $C[\cdot], M', N'$ such that $M \approx C[M']$, $C[N'] \approx N$, and $M' \mathcal{R} N'$. As $M \approx C[M']$, the definition of bisimilarity ensures that there exists M'_1 such that $C[M'] \xrightarrow{\hat{\alpha}} M'_1$ and $M_1 \approx M'_1$. As $M' \mathcal{R} N'$, Lemma 4.15 tells us that there exist $N'_1, C'[\cdot], M_2, N_2$ such that $C[N'] \xrightarrow{\hat{\alpha}} N'_1$, $M'_1 \approx C'[M_2]$ and $N'_1 \approx C'[N_2]$, with $M_2 \mathcal{R} N_2$. As $N \approx C[N']$, the definition of bisimilarity ensures that there exists N_1 such that $N \xrightarrow{\hat{\alpha}} N_1$ and $N_1 \approx N'_1$. The transitivity of \approx and the definition of \mathcal{S} ensures that $(M_1, N_1) \in \mathcal{S}$. \square

In deterministic higher-order calculi, as the CPS-calculus, it is more convenient to work with applicative bisimulations up to context and up to \approx_A .

Definition 4.17 (Applicative bisimulation up to context and up to \approx_A) *A symmetric relation \mathcal{S} over CPS-terms is an applicative bisimulation up to context and up to \approx_A if whenever $M \mathcal{S} N$ and $M \xrightarrow{\alpha} M''$, for $\alpha \neq \tau$, there is a term N'' such that $N \xrightarrow{\alpha} N''$ and there is a static context $C[\cdot]$, and terms M' and N' such that $M'' \approx_A C[M']$, $C[N'] \approx_A N''$ and $M' \mathcal{S} N'$.*

The soundness of this proof technique follows from Theorem 4.16 and Theorem 4.10.

Theorem 4.18 *If \mathcal{R} is an applicative bisimulation up to context and up to \approx_A , then $\mathcal{R} \subseteq \approx_A$.*

Proof Let us prove that \mathcal{R} is also a bisimulation up to context and up to \approx . Let $(M, N) \in \mathcal{R}$.

1. If $M \Rightarrow M'$, for some M' , then we can choose $N \Rightarrow N$. Let $C[\cdot] \stackrel{\text{def}}{=} [\cdot]$. By Lemmas 4.2 and 4.3 we have $M' \approx M = C[M]$, $C[N] \approx N$, and $M \mathcal{R} N$.
2. If $M \xRightarrow{\alpha} M'$, with $\alpha \neq \tau$, the result follows from Definition 4.17 and Theorem 4.10.

By Theorem 4.16 it follows that $M \approx N$. By Theorem 4.10 we derive $M \approx_A N$. \square

5 On divergent terms

The axiomatic semantics of Thielecke, reported in Section 2.3, allows us to prove a wide number of equalities. However, none of those axioms deal with divergent terms. We start describing how the divergence of CPS-terms is preserved by the operators of the calculus.

Proposition 5.1 *Let M be a divergent CPS-term.*

1. *For any substitution σ , the term $M\sigma$ diverges.*
2. *For any static context $C[\cdot]$, the term $C[M]$ diverges.*
3. *For any term L such that $L \Downarrow_a$, the term $L\{a\langle x \rangle \Leftarrow M\}$ diverges.*

Proof

1. It follows from Proposition 3.4(1).
2. If $C[\cdot]$ is a static context and $M \Uparrow$, then the context $C[\cdot]$ does not play any role during the computation of $C[M]$.
3. As $L \Downarrow_a$, by Proposition 3.2 $L\{a\langle x \rangle \Leftarrow M\} \Rightarrow C[M\sigma]$, for some static context $C[\cdot]$ and some substitution σ . From the first and the second items of this proposition it follows that $L\{a\langle x \rangle \Leftarrow M\}$ diverges.

\square

Now, let us consider a few algebraic laws dealing with divergent terms. The first one equates two simple terms.

$$a\langle b \rangle \{a\langle x \rangle \Leftarrow a\langle b \rangle\} \cong a\langle c \rangle \{a\langle x \rangle \Leftarrow a\langle c \rangle\} . \quad (5)$$

According to Definition 2.3 the two terms diverge. However, Thielecke's axiomatic semantics cannot be used to prove this equality. In fact, we can only apply axiom (JMP) which leaves the two terms unchanged.

The law above could be slightly complicated to equate two terms diverging in different ways.

$$a_1\langle b \rangle \{a_1\langle x_1 \rangle \Leftarrow a_2\langle a_1 \rangle\} \{a_2\langle x_2 \rangle \Leftarrow x_2\langle b \rangle\} \cong a\langle c \rangle \{a\langle x \rangle \Leftarrow a\langle c \rangle\} \quad (6)$$

where the left hand term contains some kind of mutual recursion.

More generally, it holds a reformulation of the Ω -equation of the λ -calculus (see [4]):

$$M \cong N \text{ if both } M \text{ and } N \text{ diverge.} \quad (7)$$

Notice that, in general, it may be useful to have some instruments to determine whether a term diverges. For instance, the terms appearing in Laws 5 and 6 diverge because they enter a loop.

Proposition 5.2 *Let M and N be CPS-terms such that $M \xRightarrow{\tau} N \xRightarrow{\tau} N$. Then $M \uparrow$.*

Proof It follows from the determinism of the calculus. \square

Theorem 5.3 (Ω -equation) *Let M and N be CPS-terms. If both M and N diverge then for any context $C[\cdot]$ it holds that $C[M] \cong C[N]$.*

Proof By Proposition 3.2, if M and N diverge then M and N are trivially applicative bisimilar. From Corollary 4.11 it follows $M \cong N$. From the definition of \cong it follows $C[M] \cong C[N]$, for any context $C[\cdot]$. \square

Now, let us focus on another algebraic law where divergence plays a crucial role.

$$b\langle a \rangle \{a\langle x \rangle \Leftarrow a\langle c \rangle\} \cong b\langle a \rangle \{a\langle x \rangle \Leftarrow a\langle d \rangle\} \quad (8)$$

Here, the two terms can perform an external jump to the continuation “ b ” passing, as an argument, the address a of two different but diverging, and hence equivalent, continuations.

Before proving an appropriate generalisation of the law above we need a couple of technical results.

Lemma 5.4

1. *If $M\{a\langle x \rangle \Leftarrow M\} \uparrow$ then either $M \uparrow$ or $M \Downarrow_a$.*
2. *If $M\{a\langle x \rangle \Leftarrow M\} \uparrow$ and $L \Downarrow_a$ then $L\{a\langle x \rangle \Leftarrow M\} \uparrow$.*

Proof See the Appendix. \square

Theorem 5.5 *If $M\{a\langle x \rangle \Leftarrow M\} \Uparrow$ and $M'\{a\langle x \rangle \Leftarrow M'\} \Uparrow$ then for any CPS-term L it holds that $L\{a\langle x \rangle \Leftarrow M\} \cong L\{a\langle x \rangle \Leftarrow M'\}$.*

Proof We prove that the binary relation \mathcal{R} defined as:

$$\{(L\{a\langle x \rangle \Leftarrow M\}, L\{a\langle x \rangle \Leftarrow M'\}) \mid \forall a, L, M, M'. M\{a\langle x \rangle \Leftarrow M\} \Uparrow \wedge M'\{a\langle x \rangle \Leftarrow M'\} \Uparrow\}$$

is an applicative bisimulation up to context.

Let $L\{a\langle x \rangle \Leftarrow M\} \xrightarrow{\alpha} \widehat{L}$, with $\alpha \neq \tau$. We can suppose $\alpha = b\langle y \rangle L'$, for some b and L' , with $b \neq a$. By Lemma 5.4(2) it follows that $L \not\Downarrow_a$. As a consequence there cannot be any interaction between L and the continuation $\{a\langle x \rangle \Leftarrow M\}$. This means that the action α must be generated by L . More precisely, there is L'' such that

$$L \xrightarrow{b\langle y \rangle L'} L''\{b\langle y \rangle \Leftarrow L'\} .$$

By rule (Cxt Jmp) it follows that

$$L\{a\langle x \rangle \Leftarrow M\} \xrightarrow{b\langle y \rangle L'} L''\{a\langle x \rangle \Leftarrow M\}\{b\langle y \rangle \Leftarrow L'\} .$$

With a similar reasoning we derive

$$L\{a\langle x \rangle \Leftarrow M'\} \xrightarrow{b\langle y \rangle L'} L''\{a\langle x \rangle \Leftarrow M'\}\{b\langle y \rangle \Leftarrow L'\} .$$

If we factor out the context $[\cdot]\{b\langle y \rangle \Leftarrow L'\}$ we get $(L''\{a\langle x \rangle \Leftarrow M\}, L''\{a\langle x \rangle \Leftarrow M'\}) \in \mathcal{R}$. \square

6 Conclusion and Related work

We have presented two labelled characterisations of Morris' observational equivalence for Thielecke's CPS-calculus. The former resembles Sangiorgi's context bisimulation for Higher-Order π -calculus [30], whereas the latter is in the style of Abramsky's applicative bisimilarity [1], an operational theory for higher-order languages, inspired by bisimulation theories for concurrency [23, 19]. Our LTS has some similarities with that developed by Gordon [8] for PCF plus streams, in particular our higher-order rule (Jmp) has its counterpart in Gordon's rule (Trans Fun) for functions.

Since Abramsky's work, the idea of applicative bisimilarity has been applied to a variety of higher-order sequential languages; see [8, 25] for surveys. Our characterisation proof for the applicative bisimilarity is quite different from that of [2] (due to Stoughton), as we use \approx as an auxiliary relation. In fact, the presence of single arrow transitions on the left hand in the definition of \approx is of great help in the congruence proof. Stoughton's proof uses a variant of Milner's [18] and Berry's [5] Context Lemma. Our congruence proof relies on up to bisimilarity proof techniques.

An immediate consequence of our characterisation result is a context lemma showing that the observational congruence coincides with the observational equivalence. In general, context lemmas are hard to prove. The literature on context

lemmas for functional languages is quite large. Milner [18] showed that contextual equivalence on a combinatory-logic of PCF is unchanged if we restrict attention to ‘applicative contexts’ $[\cdot]a_1 \dots a_n$. Berry [5] extended Milner’s proof to the lambda-calculus form of PCF. Gordon [8] proved a context lemma for PCF plus streams showing that only evaluation contexts need to be considered. Pitts [26] proved a context lemma for a higher-order language with assignable variables that only store first-order values. The proof uses logical relations that are defined in terms of the operational semantics. Sullivan [39] defined a metalanguage based on PCF extended with I/O and dynamic store primitives. A context lemma for this language is proved by showing that an applicative simulation relation is a precongruence. Finally, Talcott [40] investigated a lambda-calculus augmented with primitive operations to manipulate the computation state (store, continuation), and the environment (sending messages, creating processes). She proved a context lemma, called ‘ciu theorem’, that allows us to consider only contexts which correspond to computation states in which the hole is associated to the expression to be evaluated next. The ciu theorem relies on the notion of ‘uniform computation’ which allows computation steps to be carried out on states with missing information.

In the current paper, we have enhanced our bisimulation proof-methods with up to bisimilarity and up to context proof techniques [31, 34]. We have used these techniques to prove a few algebraic laws on divergent terms that cannot be derived by Thielecke’s axiomatic semantics. In higher-order languages, up to context proof techniques are notoriously hard. Sangiorgi’s bisimulation up to context is a powerful bisimulation proof method for process calculi [31, 34]. Unfortunately, his correctness proof does not carry over to applicative bisimilarities for higher-order languages. Pitts [24] extended Howe’s congruence proof to establish an up to context rule for applicative bisimulation. Gordon [8] and Sands [29] defined applicative bisimulations up to bisimilarity and/or up to context. They demonstrated the power of this approach to produce concise proofs of equivalences which are difficult to derive by other operational methods. However, the validity of general applicative bisimulations up to context remains an open problem [11]. Other examples of up to context bisimulation proof techniques in higher-order languages are [10] and [12]. More precisely, Koutavas and Wand [10] introduced a new notion of bisimulation for showing contextual equivalence in an untyped λ -calculus augmented with higher-order procedures and a general store. Lassen [12] provided an operational bisimulation account for Böhm tree equivalence including an elementary congruence proof, from which a bisimulation up to context technique is derived. This work is extended and generalised in [14], where underlying principles from Böhm tree [4] and Lévy-Longo tree equivalences [15, 16] are adapted to the call-by-value λ -calculus. A notion of enf bisimulation is defined using eager normal form (enf) equivalence classes and eager reductions, to reduce function arguments to values before application. It is then shown that enf bisimulation congruences are analogues to Lévy-Longo tree equivalence and that they both coincide on terms in the target of the CPS transforms. An up to η -reduction proof technique for enf bisimulation is also introduced since enf bisimulation does not relate terms induced by the η equation $x = \lambda y.xy$. More

recently, Støvring and Lassen [36] have defined an eager normal form bisimilarity for the untyped call-by-value lambda calculus extended with continuations and mutable references. The bisimilarity is proved to be sound and complete with respect to contextual equivalence.

Finally, Lassen and Levy [13] have developed a normal form bisimulation theory for a different CPS-calculus, Jump-With-Argument, called JWA. The paper makes three important contributions: (i) it extends normal form bisimulation to types; (ii) it provides a novel congruence proof, based on insight from game semantics; (iii) it presents a seamless treatment of η -expansion.

Acknowledgements Many thanks to the anonymous referees who spotted a bug in the proof of Lemma 4.6.

References

- [1] S. Abramsky. The lazy lambda calculus. In D. Turner, editor, *Research Topics in Functional Programming*, pages 65–116. Addison-Wesley, 1989.
- [2] S. Abramsky and L. Ong. Full abstraction in the lazy lambda calculus. *Information and Computation*, 105:159–267, 1993.
- [3] A. Appel. *Compiling with Continuations*. Cambridge University Press, 1992.
- [4] H. Barendregt. *The Lambda Calculus: Its Syntax and Semantics*, volume 103 of *Studies in Logic*. North Holland, 1984. Revised edition.
- [5] G. Berry. Some syntactic and categorical constructions of lambda calculus models. Technical Report RR-80, INRIA-Sophia Antipolis, 1981.
- [6] M. J. Fischer. Lambda-calculus schemata. In *Proceedings of ACM Conference on Proving Assertions about Programs*, pages 104–109. SIGPLAN Notice 7(1), 1972.
- [7] C. Führmann and H. Thielecke. On the Call-by-Value CPS transform and its semantics. *Information and Computation*, 188(2):241–283, 2004.
- [8] A. D. Gordon. Bisimilarity as a theory of functional programming. *Theoretical Computer Science*, 228:5–47, 1999. Also in BRICS Notes Series BRICS-NS-95-3, Department of Computer Science, University of Aarhus, 1995.
- [9] D. J. Howe. Proving congruence of bisimulation in functional programming languages. *Information and Computation*, 124(2):103–112, 1996.
- [10] V. Koutavas and M. Wand. Small bisimulations for reasoning about higher-order imperative programs. In *Proc. 33rd POPL*, pages 141–152. ACM Press, 2006.

- [11] S. B. Lassen. Relational reasoning about contexts. In *Higher Order Operational Techniques in Semantics*, pages 91–135. Cambridge University Press, 1998.
- [12] S. B. Lassen. Bisimulation in untyped lambda calculus: Böhm trees and bisimulation up to context. In *Proc. 15th MFPS*, volume 20 of *Electronic Notes in Theoretical Computer Science*, 1999.
- [13] S. B. Lassen and P. B. Levy. Typed normal form bisimulation. In *Proc. of 21st CSL*, volume 4646 of *Lecture Notes in Computer Science*, pages 283–297, 2007.
- [14] S.B. Lassen. Eager normal form bisimulation. In *Proc. of 20th LICS*, pages 345–354. IEEE Computer Society, 2005.
- [15] J.J. Lévy. An algebraic interpretation of the $\lambda\beta\kappa$ -calculus; and an application of a labelled λ -calculus. *Theoretical Computer Science*, 2(1):97–114, 1976.
- [16] G. Longo. Set theoretical models of lambda calculus: Theory, expansions and isomorphisms. *Annales of Pure and Applied Logic*, 24:153–188, 1983.
- [17] M. Merro and D. Sangiorgi. On asynchrony in name-passing calculi. *Journal of Mathematical Structures in Computer Science*, 14:715–767, 2004. An extended abstract appeared in the Proc. of ICALP’98.
- [18] R. Milner. Fully abstract models of typed lambda calculus. *Theoretical Computer Science*, 4:1–22, 1977.
- [19] R. Milner. *Communication and Concurrency*. Prentice Hall, 1989.
- [20] R. Milner. The polyadic π -calculus: a tutorial. Technical Report ECS–LFCS–91–180, LFCS, Dept. of Comp. Sci., Edinburgh Univ., October 1991. Also in *Logic and Algebra of Specification*, ed. F.L. Bauer, W. Brauer and H. Schwichtenberg, Springer Verlag, 1993.
- [21] R. Milner and D. Sangiorgi. Barbed bisimulation. In *Proc. 19th ICALP*, volume 623 of *LNCS*, pages 685–695. Springer Verlag, 1992.
- [22] J. Morris. *Lambda-Calculus Models of Programming Languages*. PhD thesis, Massachusetts Institute of Technology, 1968.
- [23] D.M. Park. Concurrency on automata and infinite sequences. In P. Deussen, editor, *Conf. on Theoretical Computer Science*, volume 104 of *LNCS*. Springer Verlag, 1981.
- [24] A. M. Pitts. An extension of Howe’s construction to yield simulation-up-to-context results. Unpublished Manuscript, 1995.
- [25] A. M. Pitts. Operationally-based theories of program equivalence. In P. Dybjer and A. M. Pitts, editors, *Semantics and Logics of Computation*, Publications of the Newton Institute, pages 241–298. Cambridge University Press, 1997.

- [26] A. M. Pitts. Reasoning about local variables with operationally-based logical relations. In P. W. O'Hearn and R. D. Tennent, editors, *Algol-Like Languages*, volume 2, chapter 17, pages 173–193. Birkhauser, 1997. Reprinted from *Proceedings Eleventh Annual IEEE Symposium on Logic in Computer Science*, Brunswick, NJ, July 1996, pp 152–163.
- [27] G.D. Plotkin. Call by name, call by value and the λ -calculus. *Theoretical Computer Science*, 1:125–159, 1975.
- [28] A. Sabry and M. Felleisen. Reasoning about programs in continuation-passing style. *Lisp and Symbolic Computation*, 6:289–360, 1993.
- [29] D. Sands. From SOS rules to proof principles: An operational metatheory for functional languages. In *Proc. 24th POPL*, pages 428–441. ACM Press, 1997.
- [30] D. Sangiorgi. Bisimulation for Higher-Order Process Calculi. *Information and Computation*, 131(2):141–178, 1996.
- [31] D. Sangiorgi. On the bisimulation proof method. *Journal of Mathematical Structures in Computer Science*, 8:447–479, 1998.
- [32] D. Sangiorgi and R. Milner. The problem of “Weak Bisimulation up to”. In *Proc. CONCUR '92*, volume 630 of *LNCS*, pages 32–46. Springer Verlag, 1992.
- [33] D. Sangiorgi and D. Walker. *The π -calculus: a Theory of Mobile Processes*. Cambridge University Press, 2001.
- [34] D. Sangiorgi and D. Walker. Some results on barbed equivalences in pi-calculus. In *Proc. CONCUR '01*, volume 2154 of *LNCS*. Springer Verlag, 2001.
- [35] G. Steele. RABBIT: A compiler for SCHEME. Technical Report AITR-474, MIT Artificial Intelligence Laboratory, 1978.
- [36] K. Støvring and S. B. Lassen. A complete, co-inductive syntactic theory of sequential control and state. In *Proc. of 34th POPL*, pages 161–172. ACM Press, 2007.
- [37] C. Strachey and P. Wadsworth. Continuations: A mathematical semantics for handling full jumps. Technical Report PRG-11, Oxford University Computing Laboratory, Programming Research Group, 1974.
- [38] C. Strachey and P. Wadsworth. Continuations: A mathematical semantics for handling full jumps. *Higher-Order and Symbolic Computation*, 13(1/2):135–152, 2000.
- [39] G. T. Sullivan. *An Extensional MetaLanguage with I/O and a Dynamic Store*. PhD thesis, Northeastern University, 1996.

- [40] C. Talcott. Reasoning about functions with effects. In *Higher Order Operational Techniques in Semantics*, pages 347–390. Cambridge University Press, 1998.
- [41] H. Thielecke. *Categorical Structure of Continuation Passing Style*. PhD thesis, University of Edinburgh, 1997. Also available as technical report ECS-LFCS-97-376.
- [42] W.P. Weijland. *Synchrony and Asynchrony in Process Algebra*. PhD thesis, University of Amsterdam, 1989.

A Proofs

Proof of Theorem 3.3

1. Let us consider the implication from left to right. If $M \downarrow_a$ then, by definition, there are names $b, a_1, b_1, \dots, a_n, b_n$, for some integer $n \geq 0$, with $a \neq a_i$ for every $1 \leq i \leq n$, such that $M = a\langle b \rangle \{a_1\langle b_1 \rangle \Leftarrow M_1\} \dots \{a_n\langle b_n \rangle \Leftarrow M_n\}$. By an application of rule (Jmp) and n applications of rule (Cxt Jmp) we get the required derivation.

Let us prove the implication from right to left. We do induction on the length of the derivation of an $a\langle x \rangle L$ action.

- Suppose that $a\langle b \rangle \xrightarrow{a\langle x \rangle L} L[b/x]\{a\langle x \rangle \Leftarrow L\}$ then $a\langle b \rangle \downarrow_a$.
- Suppose that $M\{b\langle y \rangle \Leftarrow O\} \xrightarrow{a\langle x \rangle L} M'\{b\langle y \rangle \Leftarrow O\}\{a\langle x \rangle \Leftarrow L\}$ because $M \xrightarrow{a\langle x \rangle L} M'\{a\langle x \rangle \Leftarrow L\}$, with $a \neq b$ and $b \notin \text{fv}(L)$, by an application of rule (Cxt Jmp). By inductive hypothesis $M \downarrow_a$, and since $a \neq b$ we also have $M\{b\langle y \rangle \Leftarrow O\} \downarrow_a$.

2. Let us prove the implication from left to right. The only reduction rule is

$$\begin{aligned}
 & a_i\langle b \rangle \{a_1\langle b_1 \rangle \Leftarrow M_1\} \dots \{a_i\langle b_i \rangle \Leftarrow M_i\} \dots \{a_n\langle b_n \rangle \Leftarrow M_n\} \\
 & \quad \rightarrow \\
 & M_i[b/b_i]\{a_1\langle b_1 \rangle \Leftarrow M_1\} \dots \{a_i\langle b_i \rangle \Leftarrow M_i\} \dots \{a_n\langle b_n \rangle \Leftarrow M_n\} .
 \end{aligned}$$

Now, by applying in sequence, once the rule (Jmp), $i - 1$ times the rule (Cxt Jmp), one time the rule (Tau), and $n - i$ times the rule (Cxt Tau), we get

$$\begin{aligned}
 & a_i\langle b \rangle \{a_1\langle b_1 \rangle \Leftarrow M_1\} \dots \{a_i\langle b_i \rangle \Leftarrow M_i\} \dots \{a_n\langle b_n \rangle \Leftarrow M_n\} \\
 & \quad \xrightarrow{\tau} \\
 & M_i[b/b_i]\{a_1\langle b_1 \rangle \Leftarrow M_1\} \dots \{a_i\langle b_i \rangle \Leftarrow M_i\} \dots \{a_n\langle b_n \rangle \Leftarrow M_n\} .
 \end{aligned}$$

For the implication from right to left we do rule induction on the derivation $M \xrightarrow{\tau} N$.

- Suppose that $M\{a\langle x \rangle \Leftarrow N\} \xrightarrow{\tau} M'$ by an application of rule (Tau) with premise $M \xrightarrow{a\langle x \rangle N} M'$. By applying the first part of the current theorem we have $M \Downarrow_a$. By Definition 2.2 it follows that

$$M = a\langle b \rangle \{a_1\langle b_1 \rangle \Leftarrow M_1\} \dots \{a_n\langle b_n \rangle \Leftarrow M_n\}$$

with

$$M' = N[b/x] \{a_1\langle b_1 \rangle \Leftarrow M_1\} \dots \{a_n\langle b_n \rangle \Leftarrow M_n\} \{a\langle x \rangle \Leftarrow N\} .$$

As a consequence,

$$a\langle b \rangle \{a_1\langle b_1 \rangle \Leftarrow M_1\} \dots \{a_n\langle b_n \rangle \Leftarrow M_n\} \{a\langle x \rangle \Leftarrow N\} \rightarrow$$

$$N[b/x] \{a_1\langle b_1 \rangle \Leftarrow M_1\} \dots \{a_n\langle b_n \rangle \Leftarrow M_n\} \{a\langle x \rangle \Leftarrow N\}$$

and hence $M\{a\langle x \rangle \Leftarrow N\} \rightarrow M'$.

- Suppose that $M\{a\langle x \rangle \Leftarrow N\} \xrightarrow{\tau} M'\{a\langle x \rangle \Leftarrow N\}$ by an application of rule (Cxt Tau) with premise $M \xrightarrow{\tau} M'$. By inductive hypothesis $M \rightarrow M'$ which is a reduction of the form:

$$a_i\langle b \rangle \{a_1\langle b_1 \rangle \Leftarrow M_1\} \dots \{a_i\langle b_i \rangle \Leftarrow M_i\} \dots \{a_n\langle b_n \rangle \Leftarrow M_n\} \rightarrow$$

$$M_i[b/b_i] \{a_1\langle b_1 \rangle \Leftarrow M_1\} \dots \{a_i\langle b_i \rangle \Leftarrow M_i\} \dots \{a_n\langle b_n \rangle \Leftarrow M_n\} .$$

This implies

$$a_i\langle b \rangle \{a_1\langle b_1 \rangle \Leftarrow M_1\} \dots \{a_i\langle b_i \rangle \Leftarrow M_i\} \dots \{a_n\langle b_n \rangle \Leftarrow M_n\} \{a\langle x \rangle \Leftarrow N\} \rightarrow$$

$$M_i[b/b_i] \{a_1\langle b_1 \rangle \Leftarrow M_1\} \dots \{a_i\langle b_i \rangle \Leftarrow M_i\} \dots \{a_n\langle b_n \rangle \Leftarrow M_n\} \{a\langle x \rangle \Leftarrow N\}$$

and therefore $M\{a\langle x \rangle \Leftarrow N\} \rightarrow M'\{a\langle x \rangle \Leftarrow N\}$.

□

Proof of Lemma 4.5

It suffices to prove that the symmetric relation $\approx_{\mathcal{S}\approx}$ is a bisimulation. Let L and O be two CPS-terms such that $L \approx_{\mathcal{S}\approx} O$. This means that there are M and N such that $L \approx M \mathcal{S} N \approx O$. There are two cases.

1. Suppose $L \xrightarrow{\tau} L'$. As $L \approx M$, there is M' such that $M \Rightarrow M'$ and $L' \approx M'$. As $M \mathcal{S} N$, if $M \Rightarrow M'$ then there is N' such that $N \Rightarrow N'$ and $M' \mathcal{S}\approx N'$. Now, if $N \Rightarrow N'$ then there is O' such that $O \Rightarrow O'$ and $N' \approx O'$. By transitivity of \approx we have $L' \approx_{\mathcal{S}\approx} O'$.

2. Suppose $L \xrightarrow{\alpha} L'$. As $L \approx M$, there are M' and M'' such that $M \Rightarrow M' \xrightarrow{\alpha} M''$ and $L' \approx M''$. As $M \mathcal{S} N$, if $M \Rightarrow M'$ then there is N' such that $N \Rightarrow N'$ and $M' \mathcal{S} N'$. As a consequence, if $M' \xrightarrow{\alpha} M''$ then there are N'' and N''' such that $N' \Rightarrow N'' \xrightarrow{\alpha} N'''$ and $M'' \approx \mathcal{S} \approx N'''$. Finally, as $N \approx O$, if $N \Rightarrow N'''$ then there are O' and O'' such that $O \Rightarrow O' \xrightarrow{\alpha} O''$ and $N''' \approx O''$. By transitivity of \approx we have $L' \approx \mathcal{S} \approx O''$.

□

Proof of Lemma 4.14

Let \mathcal{R} be a bisimulation up to context and up to \approx . Suppose that $(M, N) \in \mathcal{R}$ and that $C[M] \xrightarrow{\alpha} M''$ for some static context

$$C[\cdot] \stackrel{\text{def}}{=} [\cdot]\{a_1\langle x_1 \rangle \Leftarrow M_1\} \dots \{a_n\langle x_n \rangle \Leftarrow M_n\}.$$

We decompose the transition $C[M] \xrightarrow{\alpha} M''$, distinguishing two cases.

1. The transition is performed by M , that is, $M \xrightarrow{\alpha} M'$. There are two sub-cases depending on α .

- (a) Let $M \xrightarrow{\tau} M'$. By several applications of rule (Cxt Tau) we derive $M'' = C[M']$. As $(M, N) \in \mathcal{R}$, there is N' such that $N \Rightarrow N'$ and there is a static context $C'[\cdot]$, and terms M''' and N''' such that $M' \approx C'[M''']$ and $C'[N'''] \approx N'$, with $(M''', N''') \in \mathcal{R}$. Let $D[\cdot] \stackrel{\text{def}}{=} C[C'[\cdot]]$. $D[\cdot]$ is a static context. As $M' \approx C'[M''']$ and \approx is preserved by all contexts, $M'' = C[M'] \approx C[C'[M''']] = D[M''']$. As $N \Rightarrow N'$, by several applications of rule (Cxt Tau) we have $C[N] \Rightarrow C[N']$ with $C[N'] \approx C[C'[N''']] = D[N''']$. To conclude, we recall that $(M''', N''') \in \mathcal{R}$.

- (b) Let $M \xrightarrow{a\langle x \rangle L} M' = \widehat{M}\{a\langle x \rangle \Leftarrow L\}$, for some a, L and \widehat{M} . By several applications of rule (Cxt Jmp) we derive $M'' = C[\widehat{M}]\{a\langle x \rangle \Leftarrow L\}$.

As $(M, N) \in \mathcal{R}$, there is N' such that $N \xrightarrow{a\langle x \rangle L} N' = \widehat{N}\{a\langle x \rangle \Leftarrow L\}$, for some \widehat{N} , and there is a static context $C'[\cdot]$, and terms M''' and N''' such that $M' \approx C'[M''']$ and $C'[N'''] \approx N'$, with $(M''', N''') \in \mathcal{R}$. As $N \xrightarrow{a\langle x \rangle L} N'$, by several applications of rules (Cxt Tau) and (Cxt Jmp), we have $C[N] \xrightarrow{a\langle x \rangle L} N'' = C[\widehat{N}]\{a\langle x \rangle \Leftarrow L\}$.

Let

$$\widehat{C}[\cdot] \stackrel{\text{def}}{=} [\cdot]\{a_1\langle x_1 \rangle \Leftarrow M_1\{a\langle x \rangle \Leftarrow L\}\} \dots \{a_n\langle x_n \rangle \Leftarrow M_n\{a\langle x \rangle \Leftarrow L\}\}.$$

By applying n times the axiom (DISTR) (to shift the continuation $\{a\langle x \rangle \Leftarrow L\}$ at the extreme left), Theorem 2.6, the inclusion $\cong \subseteq \simeq$, Lemma 4.3 and Lemma 4.6 we obtain:

$$C[\widehat{M}]\{a\langle x \rangle \Leftarrow L\} \approx \widehat{C}[\widehat{M}]\{a\langle x \rangle \Leftarrow L\}.$$

As $\widehat{M}\{a\langle x \rangle \Leftarrow L\} = M' \approx C'[M''']$ and \approx is preserved by all contexts, it follows that

$$\widehat{C}[\widehat{M}\{a\langle x \rangle \Leftarrow L\}] \approx \widehat{C}[C'[M''']] .$$

With a similar reasoning we derive:

$$C[\widehat{N}]\{a\langle x \rangle \Leftarrow L\} \approx \widehat{C}[\widehat{N}\{a\langle x \rangle \Leftarrow L\}] \approx \widehat{C}[C'[N''']] .$$

Let $D[\cdot] \stackrel{\text{def}}{=} \widehat{C}[C'[\cdot]]$. $D[\cdot]$ is a static context. By the transitivity of \approx we obtain $M'' \approx D[M''']$, $D[N'''] \approx N''$ and $(M''', N''') \in \mathcal{R}$.

2. The transition is due to an interaction between M and the context $C[\cdot]$. More precisely, $M \xrightarrow{a_i\langle x_i \rangle M_i} M' = \widehat{M}\{a_i\langle x_i \rangle \Leftarrow M_i\}$, for some $1 \leq i \leq n$ and term \widehat{M} . By $i - 1$ applications of rule (Cxt Jmp) we have

$$\begin{aligned} & M\{a_1\langle x_1 \rangle \Leftarrow M_1\} \dots \{a_{i-1}\langle x_{i-1} \rangle \Leftarrow M_{i-1}\} \\ & \xrightarrow{a_i\langle x_i \rangle M_i} \\ & \widehat{M}\{a_1\langle x_1 \rangle \Leftarrow M_1\} \dots \{a_{i-1}\langle x_{i-1} \rangle \Leftarrow M_{i-1}\} \{a_i\langle x_i \rangle \Leftarrow M_i\} . \end{aligned}$$

By an application of rule (Tau) we have

$$M\{a_1\langle x_1 \rangle \Leftarrow M_1\} \dots \{a_i\langle x_i \rangle \Leftarrow M_i\} \xrightarrow{\tau} \widehat{M}\{a_1\langle x_1 \rangle \Leftarrow M_1\} \dots \{a_i\langle x_i \rangle \Leftarrow M_i\} .$$

By $n - i$ applications of rule (Cxt Tau) we have $M'' = C[\widehat{M}]$. As $(M, N) \in \mathcal{R}$ there is N' such that $N \xrightarrow{a_i\langle x_i \rangle M_i} N' = \widehat{N}\{a_i\langle x_i \rangle \Leftarrow M_i\}$, for some \widehat{N} , and there is a static context $C'[\cdot]$, and terms M''' and N''' such that $M' \approx C'[M''']$ and $N' \approx C'[N''']$ with $(M''', N''') \in \mathcal{R}$. As $N \xrightarrow{a_i\langle x_i \rangle M_i} N'$, by $i - 1$ applications of rule (Cxt Jmp), and $n - i$ applications of rule (Cxt Tau) we have $C[N] \Rightarrow N'' = C[\widehat{N}]$.

Let

$$\begin{aligned} \widehat{C}[\cdot] & \stackrel{\text{def}}{=} [\cdot]\{a_1\langle x_1 \rangle \Leftarrow M_1\{a_i\langle x_i \rangle \Leftarrow M_i\}\} \\ & \dots \{a_{i-1}\langle x_{i-1} \rangle \Leftarrow M_{i-1}\{a_i\langle x_i \rangle \Leftarrow M_i\}\}\{a_{i+1}\langle x_{i+1} \rangle \Leftarrow M_{i+1}\} \\ & \dots \{a_n\langle x_n \rangle \Leftarrow M_n\} . \end{aligned}$$

By applying $i - 1$ times the axiom (DISTR) (to shift $\{a_i\langle x_i \rangle \Leftarrow M_i\}$ at the extreme left), Theorem 2.6, the inclusion $\cong \subseteq \simeq$, Lemma 4.3 and Lemma 4.6 we obtain:

$$C[\widehat{M}] \approx \widehat{C}[\widehat{M}\{a_i\langle x_i \rangle \Leftarrow M_i\}] .$$

As $\widehat{M}\{a_i\langle x_i \rangle \Leftarrow M_i\} = M' \approx C'[M''']$ and \approx is preserved by all contexts, it follows that

$$\widehat{C}[\widehat{M}\{a_i\langle x_i \rangle \Leftarrow M_i\}] \approx \widehat{C}[C'[M''']] .$$

With a similar reasoning we derive

$$C[\widehat{N}] \approx \widehat{C}[\widehat{N}\{a_i\langle x_i \rangle \Leftarrow M_i\}] \approx \widehat{C}[C'[N''']] .$$

Let $D[\cdot] \stackrel{\text{def}}{=} \widehat{C}[C'[\cdot]]$. $D[\cdot]$ is a static context. By the transitivity of \approx we obtain $M'' \approx D[M''']$, $N'' \approx D[N''']$ and $(M''', N''') \in \mathcal{R}$. □

Proof of Lemma 5.4

1. There are two possibilities: either $M \Uparrow$ or $M \Downarrow_b$, for some variable b . However, if $M \Downarrow_b$ then it must be $b = a$, otherwise we would have $M\{a\langle x \rangle \Leftarrow M\} \Downarrow_b$.
2. If $L \Downarrow_a$ then

$$L\{a\langle x \rangle \Leftarrow M\} \Rightarrow C[M[b/x]]\{a\langle x \rangle \Leftarrow M\}$$

for some variable b and some static context $C[\cdot]$ that does not capture free occurrences of variable a . As $M\{a\langle x \rangle \Leftarrow M\} \Uparrow$, by the previous item of this lemma there are two possibilities: either $M \Uparrow$ or $M \Downarrow_a$. If $M \Uparrow$ by Proposition 3.4(1) also $M[b/x] \Uparrow$. By Proposition 5.1(2) we obtain $C[M[b/x]]\{a\langle x \rangle \Leftarrow M\} \Uparrow$. If $M \Downarrow_a$ by Proposition 3.4(2) also $M[b/x] \Downarrow_a$. As $C[\cdot]$ does not capture the free occurrences a , it follows that

$$C[M[b/x]]\{a\langle x \rangle \Leftarrow M\} \Rightarrow C'[M[b'/x]]\{a\langle x \rangle \Leftarrow M\}$$

for some variable b' and some static context $C'[\cdot]$ obtained by adding the continuations of $C[\cdot]$ to those of $M[b/x]$. As a consequence also $C'[\cdot]$ does not capture the free occurrences of a . As $M \Downarrow_a$, by Proposition 3.4(2) we have $M[b'/x] \Downarrow_a$ and the reduction sequence above can be reproduced “ad infinitum” showing that the term under investigation diverges. □