Environmental bisimulations for probabilistic higher-order languages

Environmental bisimulations for probabilistic higher-order languages

DAVIDE SANGIORGI, University of Bologna and Inria
VALERIA VIGNUDELLI, Univ. Lyon, CNRS, ENS de Lyon, UCB Lyon 1, LIP

Environmental bisimulations for probabilistic higher-order languages are studied. In contrast with applicative bisimulations, environmental bisimulations are known to be more robust and do not require sophisticated techniques such as Howe's in the proofs of congruence.

As representative calculi, call-by-name and call-by-value λ-calculus, and a (call-by-value) λ-calculus extended with references (i.e., a store) are considered. In each case full abstraction results are derived for probabilistic environmental similarity and bisimilarity with respect to contextual preorder and contextual equivalence, respectively. Some possible enhancements of the (bi)simulations, as 'up-to techniques', are also presented.

Probabilities force a number of modifications to the definition of environmental bisimulations in non-probabilistic languages. Some of these modifications are specific to probabilities, others may be seen as general refinements of environmental bisimulations, applicable also to non-probabilistic languages. Several examples are presented, to illustrate the modifications and the differences.

Davide Sangiorgi and Valeria Vignudelli. 2019. Environmental bisimulations for probabilistic higher-order languages. 1, 1, 62 pages.

1 INTRODUCTION

The general topic of the paper are techniques for proving behavioural equivalence in higher-order probabilistic languages. Checking computer programs for equivalence is a crucial, but challenging, problem. Equivalence between two programs generally means that the programs should behave "in the same manner" under any context. Specifically, two λ-terms are *contextually equivalent* if they have the same convergence behaviour (i.e., they do or do not terminate) in any possible context [32].

Due to the universal quantification on the contexts of the language, it is generally hard to prove that two terms are contextually equivalent. Such proofs are particularly hard to carry out if the language under consideration has higher-order features, which allow contexts to copy and pass around terms.

Bisimulation [34, 43] offers a powerful operational method for proving equivalence of programs in various kinds of languages, relying on the coinduction proof principle. In order to qualify as a proof method, bisimulation relations should be sound with respect to contextual equivalence, i.e., they should induce an equivalence relation - *bisimilarity* - that implies contextual equivalence. Ideally, bisimilarity should be fully abstract with respect to contextual equivalence, i.e., coincide with it.

Bisimulation has been studied in depth in deterministic λ-calculi, e.g., [2, 17, 27, 38, 41]. In contrast, so far, it has been little explored in probabilistic λ-calculi, mainly in the form of *applicative bisimilarity* [8, 11] for the pure call-by-name and call-by-value languages.

Applicative bisimulations are known however to have some significant limitations. First, they do not scale very well to languages richer than the pure λ-calculus. For instance,

they are unsound under the presence of references, or even just generative names, or data abstraction; see [24] for an enlightening discussion. Secondly, congruence proofs of applicative bisimulations are notoriously hard. Such proofs usually rely on Howe's method [19], which is, however, fragile outside the pure λ-calculus. Related to the problems with congruence are also the difficulties of applicative bisimulations with "up-to context" techniques (the usefulness of these techniques in higher-order languages and its problems with applicative bisimulations have been studied by Lassen [27]; see also [26, 39, 41]).

To relieve this burden, *environmental bisimulations* have been proposed [45], refining earlier proposals in [1, 7, 23, 26, 48]. The distinguishing feature of these bisimulations is that the pairs of tested terms are enriched with environments, that intuitively collect the observer's knowledge about values computed during the bisimulation game. The elements of the environment can be used to construct terms to be supplied as inputs during the bisimulation game. The notion has been applied to a variety of languages, including pure λ-calculi [45, 49], extensions of λ-calculi [3, 4, 25, 26, 48], or languages for concurrency or distribution [35, 36, 46]. In environmental bisimulations the proof of congruence goes by induction over the structure of contexts, as in proofs for first-order languages. For this, a 'small-step' reduction relation is more handy than a 'big-step' reduction: the former allows one a tigh control over the syntax of the contexts, which is harder with the latter because in a higher-order language contexts may arbitrarily grow during reduction; moreveor, the former is better suited to bisimulation enhancements of the 'up-to techniques (earlier forms of environmental bisimulation, e.g., [48, 49], place a mild emphasis on up-to techniques and follow the big-step tradition of the λ-calculus).

With probabilities, the drawbacks of applicative bisimilarity are magnified: full abstraction with respect to contextual equivalence may fail also in a pure λ-calculus, and Howe's technique has to be enriched with non-trivial 'disentangling' properties for sets of real numbers, these properties themselves proved by modeling the problem as a flow network and then applying the Max-flow Min-cut Theorem.

In this paper, our goal is understanding environmental bisimulations in probabilistic higher-order languages. As representative calculi we consider call-by-name and call-by-value λ-calculus, and a (call-by-value) λ-calculus extended with higher-order references. The computation is made probabilistic by endowing these calculi with a primitive for binary, fair, probabilistic choice. In each case we derive full abstraction results for probabilistic environmental similarity and bisimilarity with respect to the contextual preorder and contextual equivalence, respectively. Below, we discuss the main differences of our proposals in comparison with ordinary (i.e., non probabilistic) environmental (bi)simulations.

Static and dynamic environments. In ordinary environmental bisimulation the values produced during the bisimulation game are placed into the environment, so that the observer can later play them at will during the bisimulation game. This schema is irrespective of the evaluation strategy (call-by-name or call-by-value), and is *the* distinguishing feature of environmental bisimulations over the applicative ones. 'Playing a term' means copying it. However, in the λ-calculus the copying possibilities for call-by-name and call-by-value are quite different. In call-by-name, evaluation only occurs in functional position and therefore the term resulting from the evaluation may not be copied. In call-by-value, in contrast, a term may be evaluated also in argument position, and then given as input to a function; thus copying is possible also after evaluation. The different copying behaviour is well visible, for instance, in linear logic interpretations of call-by-name and call-by-value [29].

Now, the semantics of probabilistic languages is sensitive to the copying operation; for instance the probability of success of an experiment, if non-trivial, may be lowered by playing the experiment several times. This has a strong impact on behavioural equivalences for call-by-name and call-by-value in probabilistic λ-calculi. As an example, using Ω for a purely divergent term and $\oplus$ for the binary, fair, probabilistic choice operator,

$$A \stackrel{\mathtt{def}}{=} \lambda x.(x \oplus \Omega) \quad \text{and} \quad B \stackrel{\mathtt{def}}{=} (\lambda x.x) \oplus (\lambda x.\Omega) \tag{1}$$

are contextually equivalent in call-by-name: if evaluated alone they always terminate; if evaluated with an argument, they return the argument with the same probability. More generally, in call-by-name abstraction distributes over probabilistic choice. In contrast, distributivity fails in call-by-value, exploiting the possibility of copying evaluated terms; e.g., the probabilities of termination for A and B are different in the context $(\lambda x.x\,(x\,\lambda y.y))[\cdot]$. In a call-by-value strategy, the term in the hole is first evaluated in argument position, then the result of the evaluation is copied and executed twice. When put in the context, term A terminates with probability $\frac{1}{4}$ whereas B terminates with probability $\frac{1}{2}$.

To be able to express such behavioural differences, in our environmental bisimulations the values produced during the bisimulation game are placed into the environment *only* in the call-by-value case. We call such a value environment a *dynamic environment* because it may grow during the bisimulation game. It is precisely the use of the dynamic environment that allows us to separate the two terms A and B above. In probabilistic call-by-name, dynamic environments would break full abstraction for contextual equivalence. The only environment for call-by-name is *static*. The static environment for two compared objects F, G is a pair of λ-terms M, N, which are, intuitively, the initial λ-terms from which, using evaluation and interaction according to the bisimulation game, the objects F, G have been derived. This (small) static environment is sufficient to ensure that the congruence proof of the bisimilarity remains in the style of ordinary environmental bisimulation (i.e., it does not require sophisticated techniques such as Howe's). In short, the static environment reflects the copying possibility for terms before evaluation, whereas the dynamic environment reflects the copying possibility for values resulting from evaluation.

Formal sums. In our probabilistic relations the objects compared are not plain λ-terms but *formal sums*, that are the objects produced by the semantics of a term. These are, intuitively, syntactic representations of probability distributions. As a consequence, environments are not just tuples of values, but formal sums of tuples of values. To see why related objects must be formal sums, consider again the terms A and B in (1): our environmental bisimulation for call-by-name equates A and B by relating A and the formal sum resulting from the evaluation of B. None of the components of the formal sum, $\lambda x.x$ and $\lambda x.\Omega$, could separately be related with A. (A form of bisimulation on formal sums, namely a probabilistic version of *logical bisimulation*, is already defined in [11] for call-by-name; its drawbacks are discussed in Section 6.)

In pure call-by-value λ-calculus, full abstraction for contextual equivalence would also hold without formal sums (i.e., relating plain λ-terms), for the same reason why, in the same language, applicative bisimilarity on plain terms is fully abstract [8]. We do not pursue this simplification of environmental bisimulations because it would be unsound in extensions of the calculus. For instance, consider the following terms of a probabilistic λ-calculus with store (again, an instance of distributivity):

$$H \stackrel{\mathtt{def}}{=} (\boldsymbol{\nu}\, x := 0)(\lambda.(M \oplus N)) \quad K \stackrel{\mathtt{def}}{=} (\boldsymbol{\nu}\, x := 0)((\lambda.M) \oplus (\lambda.N))$$

where $(\boldsymbol{\nu}\, x := 0)$ indicates the creation of a new reference l, initialized with 0 and substituted to x, $\lambda.L$ is a thunk (i.e., $\lambda z.L$ for z not free in L), and where, using $L_1 \,\underline{\texttt{seq}}\, L_2$ for the sequential evaluation of L_1 followed by L_2,

$$M \stackrel{\texttt{def}}{=} \texttt{if}\ !x = 0\ \texttt{then}\ (x := 1 \,\underline{\texttt{seq}}\, \texttt{true})\ \texttt{else}\ \Omega$$
$$N \stackrel{\texttt{def}}{=} \texttt{if}\ !x = 0\ \texttt{then}\ (x := 1 \,\underline{\texttt{seq}}\, \texttt{false})\ \texttt{else}\ \Omega\ .$$

The terms M and N only differ at their first evaluation, when the new reference l that was substituted to x is set to 1 and M produces `true` whereas N produces `false`; thereafter l is 1 and both terms diverge. As a consequence, H and K are contextually equivalent: at their first evaluation they always terminate, each returning `true` and `false` with the same probability, and at later evaluations they always diverge.

To place H and K in a bisimulation, H has to be related with the formal sum obtained from the evaluation of K; again, the single components alone would be distinguished. Once more, this is a copying issue, due to the possibility of copying terms but not stores.

Big-step reduction, term closure, and congruence proof. To achieve full abstraction, in the probabilistic case the bisimulation clauses have to use a big-step, rather than a small-step, reduction relation. Precisely, we give finitary big-step approximants from which the semantics of a term is obtained via a least-fixed point construction (a similar semantics is in [12]). The reason, intuitively, is that while in pure λ-calculi a term converges (i.e., it terminates its computation) in a finite number of reductions, in the probabilistic calculi there may be several terminating computation paths, in which the total number of reductions need not be finitary. An example is given by the terms

$$P \stackrel{\texttt{def}}{=} RR \quad \text{and} \quad Q \stackrel{\texttt{def}}{=} \lambda.\Omega\,, \quad \text{for} \quad R \stackrel{\texttt{def}}{=} \lambda x.((xx) \oplus Q) \tag{2}$$

The terms P and Q are contextually equivalent, intuitively because they both have probability 1 of becoming term Q: after some reductions, P may become Q or may become P again, with equal probability. Only by exploring the whole computation tree produced by P does one find out that the infinite number of leaves in the tree makes a probability 1 of obtaining Q. None of the finite approximants of the infinite tree gives the same information (a formal sum made of a finite subset of the leaves would not be equivalent to Q).

When the reduction relation is small-step (as in the ordinary environmental bisimulations in [45]) the related terms need not be values, because a normalizing term need not produce a value in a single step and bisimulations must be closed under the reduction adopted. In contrast, as our environmental bisimulations are big-steps, the bisimulation game may be confined to values.

A more significant consequence of the adoption of big-step reductions is that the induction over contexts in the proofs of congruence for ordinary environmental bisimulations is replaced by an induction on the number of small-step reductions with which a big-step approximant is derived (possibly coupled with an induction on the size of a context), combined with two levels of continuity arguments. One level stems from the least fixed point construction employed in the definition of the infinitary big-step semantics on terms. The second level stems from a characterization of bisimilarity as the kernel of the similarity preorder and, in turn, as the kernel of a finitary similarity in which (on the challenger side) the big-step reduction relation employed is finite. The proof of the characterization with the finitary similarity makes use of least fixed-points via a saturation construction on formal sums where, intuitively, a formal sum is better than another formal sum if the former conveys more accurate probabilistic information than the latter.

Up-to techniques. Our proofs and examples rely on a few enhancements of the bisimulation proof method ('bisimulations up-to'), some of which are extensions of common (bi)simulation enhancements, others are specific to probabilistic calculi. An example of the latter is 'simulation up-to lifting', whereby it is sufficient, in the coinductive game, that two derivative formal sums are in the probabilistic lifting of the candidate relation, rather than in the candidate relation itself.

While the bisimulations act on formal sums and use infinitary big-step reductions to values, we also explore coinductive games played on plain λ-terms and on finitary multi-step reductions to terms (not necessarily values) as sound proof techniques. In particular, we combine these with up-to context, so to be able to compare terms in the middle of their evaluation when a common context can be isolated and removed.

Structure of the paper. Section 2 introduces some general definitions and notations for the paper. Section 3 presents environmental bisimulations for pure call-by-name, establishes basic properties including full abstraction for bisimilarity and similarity, and develops various up-to techniques. Section 4 considers pure call-by-value, and Section 5 an extension with ML-like references. Section 6 discusses additional related work, and Section 7 concludes, also mentioning possible future work.

2 PRELIMINARIES

We introduce general notations and terminologies for the paper. Familiarity with standard terminologies (such as free/bound variables, and α-conversion) is assumed.

We use meta-variables $M, N, P, Q, \ldots$ for terms, and $V, W, \ldots$ for values. We identify α-convertible terms. We write $M\{^N\!/_x\}$ for the capture-avoiding substitution of N for x in M. A term is *closed* if it contains no free variables. The set of free variables of a term M is $\mathsf{fv}(M)$. A *context* C is an expression obtained from a term by replacing some subterms with *holes* of the form $[\cdot]_i$, where for every i there can be multiple occurrences of hole $[\cdot]_i$. We write $C[M_1, \ldots, M_n]$ for the term obtained by replacing each occurrence of $[\cdot]_i$ in C with M_i.

We use a tilde to denote a tuple; for instance $\widetilde{M}$ is a tuple of terms, and $(\widetilde{M})_i$ is its i-th element. We write tuples as $\{M_i\}_{i\in I}$ when we want to emphasize the indexing set, and we sometimes abbreviate this as $\{M_i\}_i$, implicitly assuming that i ranges over an index set I. All notations are extended to tuples componentwise. Hence, we often write $C[\widetilde{M}]$ for $C[M_1, \ldots, M_n]$ and $\widetilde{M}\mathcal{R}\widetilde{N}$ for $(M_1\mathcal{R}N_1) \wedge \cdots \wedge (M_n\mathcal{R}N_n)$.

By default, the results and definitions in the paper are (implicitly) stated for closed terms. They can be generalized to open terms in a standard way for bisimulations in λ-calculi [26, 45, 48]. Thus the properties between open terms M and N are derived from the corresponding properties between the closed terms $\lambda\widetilde{x}.M$ and $\lambda\widetilde{x}.N$, for $\{\widetilde{x}\} \supseteq \mathsf{fv}(M) \cup \mathsf{fv}(N)$. The caveat is proving that any term M with $\{\widetilde{x}\} = \mathsf{fv}(M)$ is contextually equivalent to its β-expansion $(\lambda\widetilde{x}.M)\widetilde{x}$. This is done by defining the least congruence relating N and $(\lambda y.N)y$, for any N, and then showing that such a relation is preserved by reduction. Adapting the technique, detailed in, e.g., [26, 45, 48], to probabilistic λ-calculi is straightforward, also because both in [26, 45, 48] and in the current paper the reduction relation is 'big-step' (the main difference is that the auxiliary congruence relation has to be extended to formal sums).

The pure λ-calculi will be untyped, whereas we will find types convenient to treat the extension with store.

3 PROBABILISTIC CALL-BY-NAME λ-CALCULUS

The terms of the probabilistic λ-calculus are generated by the following grammar:

$$M, N ::= x \;\Big|\; \lambda x.M \;\Big|\; MN \;\Big|\; M \oplus N$$

We write $\Lambda_\oplus$ for the subset of closed terms. The *values* are the terms of the form $\lambda x.M$ (the abstractions). In call-by-name the *evaluation contexts* (which, in contrast with standard contexts, may have only one occurrence of a single hole $[\cdot]$) are:

$$C := CM \mid [\cdot]$$

In probabilistic languages, the semantics of a term is usually a *(sub)distribution*, that is, a function that specifies the probabilities of all possible outcomes for that term [12]. We prefer, by contrast, syntactic representations of distributions, as *formal sums*, because they allow us a tighter control on the manipulations of the operational semantics, which is important in various places of our coinductive definitions and proofs. Formal sums have the form

$$\textstyle\sum_{i\in I} p_i; M_i$$

where $0 < p_i \leq 1$, for each i, $\sum_{i\in I} p_i \leq 1$, and I is a (possibly infinite) indexing set. In a summand $p_i; M_i$ of a formal sum, p_i is its *probability value* (or *weight*), and M_i is its *term*. The terms of different summands of a formal sum need not be different. The weight $\texttt{weight}(\sum_{i\in I} p_i; M_i)$ of a formal sum is the probability value $\sum_{i\in I} p_i$. We let F, G range over formal sums, and we write the empty formal sum as $\emptyset$ (i.e., the formal sum with no summands). We write $F = G$ if F and G are syntactically equal modulo a permutation of the summands. We use '+' for binary sums, in the usual infix form, and sometimes apply it also to formal sums. Thus, for $F = \sum_{j\in J} q_j; N_j$ and $G = \sum_{j'\in J'} q'_{j'}; N'_{j'}$ with $\sum_{j\in J} q_j + \sum_{j'\in J'} q'_{j'} \leq 1$ we have

$$F + G \stackrel{\texttt{def}}{=} \textstyle\sum_{i\in J\cup J'} p_i; M_i$$

with $p_i; M_i = q_j; N_j$ for $i \in J$ and $p_i; M_i = q'_{j'}; N'_{j'}$ for $i \in J'$.

Value formal sums, ranged over by Y, Z are formal sums in which the term of each summand is a value.

There is an obvious mapping from formal sums to distributions, whereby a formal sum F yields the distribution in which the probability of a term M is the sum of the weights with which M appears in summands of F. The mapping is not injective: in general, infinitely many formal sums yield the same distribution (because of possible duplicates in the terms of the summands of a formal sum). For instance, $\frac{1}{2}; M + \frac{1}{4}; M$ and $\frac{3}{4}; M$ are two different formal sums, that correspond to the same distribution assigning probability $\frac{3}{4}$ to term M and probability 0 to any N such that $N \neq M$.

We sometimes decompose formal sums using a lifting construction. If $F_i = \sum_{j\in J_i} p_{i,j}; M_{i,j}$, for $i \in I$, then

$$\textstyle\sum_{i\in I} p_i{\cdot}F_i \stackrel{\texttt{def}}{=} \sum_{i\in I, j\in J_i} p_i \cdot p_{i,j}; M_{i,j}\ .$$

Note that some F_i can be the empty formal sum $\emptyset$, in which case the corresponding probability value p_i is lost, i.e., $\sum_{i\in I} p_i{\cdot}F_i = \sum_{i\in I\setminus\{j\}} p_i{\cdot}F_i$ if $F_j = \emptyset$. The semantics of a term M, written $[\![M]\!]$, is a value formal sum produced as the supremum of the value formal sums obtained by finite computations starting from M, using a preorder $\leq_{\texttt{apx}}$ on formal sums in which $F_1 \leq_{\texttt{apx}} F_2$ if F_1 is an approximant of F_2 (in other words F_2 conveys more information than F_1); formally, $F_2 = F_1 + G$ for some G. The semantics is obtained in various steps, whose rules are presented in Figure 1:

(1) a single-step reduction relation $\longrightarrow$ from terms to formal sums;
(2) a multi-step reduction relation $\Longrightarrow$ from terms and formal sums to formal sums, from which a relation $\Longmapsto$ to value formal sums is extracted by retaining only the summands whose term is a value via the function $\mathtt{val}$:

$$\mathtt{val}(\textstyle\sum_i p_i; M_i) \stackrel{\mathtt{def}}{=} \sum_{\{i|M_i \text{is a value}\}} p_i; M_i \ ;$$

(3) the semantics $[\![\ ]\!]$, mapping terms and formal sums to value formal sums via the supremum construction.

If $M \Longrightarrow \sum_{i\in I} p_i; M_i$ then I is finite, and each i represents a 'possible world' of the probabilistic run of M, with probability p_i and outcome M_i. The subset of possible worlds in which M_i is a value makes for an approximant of M, and from such approximants the semantics of M is obtained. The definition of the semantics of a term as a supremum requires formal sums with infinitely many summands. As an example, consider the term P defined in 2 in the Introduction, whose semantics is $\sum_{n\geq 1} \frac{1}{2^n}; Q$.

Since value formal sums form an ω-complete partial order with respect to the $\leq_{\mathtt{apx}}$ preorder, and for every M the set of those value formal sums Y such that $M \Longmapsto Y$ is a countable directed set, the semantics $[\![M]\!]$ of a term M exists and is unique.

Relations $\Longrightarrow$ and $\Longmapsto$ are finitary in the sense that a derivation proof where one of such relations appears in the conclusion only contains a finite number of 'small steps' (relation $\longrightarrow$). When reasoning by induction, sometimes we will need to make such number explicit, therefore writing $\Longrightarrow_n$ and $\Longmapsto_n$, respectively.

Rule MulT, in contrast with MulFS, does not need a finitary condition on the indexing set because a formal sum obtained in a small step from a term may have at most two summands.

Remark 1. If we define a semantics based on probability distributions, rather than on formal sums, the definition of the multi-step reduction relation $F \Longrightarrow F$ from distributions to distributions becomes more subtle. In the corresponding rule in Figure 1

$$\textsc{MulFS}\ \frac{M_i \Longrightarrow F_i}{\sum_{i\in I} p_i; M_i + G \Longrightarrow \sum_{i\in I} p_i \cdot F_i + G}\ \ I \text{ finite}$$

we have two possibilities (we use here the notation for formal sums as a notation for probability distributions). If we allow a decomposition of the distribution $F = \sum_{i\in I} p_i; M_i + G$ such that the same term M can be both in the support of $\sum_{i\in I} p_i; M_i$ and in the support of G, then we allow uncountably many different reductions from F. Otherwise, such decomposition is not allowed and a parallel reduction of terms in a probabilistic binary choice $M \oplus M$ is forced by the semantics whenever the terms in the choice coincide, since the formal sum $\frac{1}{2}; M + \frac{1}{2}; M$ is treated as $1; M$. By resorting to formal sums, we thus remain closer to the syntax of the calculus and to standard definitions of reductions over terms.

Additional notation. We introduce here some additional notation, allowing us to easily manipulate terms and formal sums. This simplified notation is only used in the proofs of our results; in the rest of the paper, we use the extended notation defined above. For $Y = \sum_i p_i; \lambda x.M_i$ and $F = \sum_i p_i; M_i$, we define:

- $\lambda x.M \bullet P \stackrel{\mathtt{def}}{=} M\{P/x\}$;
- $Y \bullet P \stackrel{\mathtt{def}}{=} \sum_i p_i; M_i\{P/x\}$;
- $C[F] \stackrel{\mathtt{def}}{=} \sum_i p_i; C[M_i]$;
- $FP \stackrel{\mathtt{def}}{=} \sum_i p_i; M_i P$.

single-step reduction relation from terms to formal sums

$$\textsc{Beta}\ \frac{}{(\lambda x.M)N \longrightarrow 1; M\{^N\!/\!_x\}} \qquad \textsc{Sum}\ \frac{}{M_1 \oplus M_2 \longrightarrow \frac{1}{2}; M_1 + \frac{1}{2}; M_2}$$

$$\textsc{Eval}\ \frac{M \longrightarrow \sum_i p_i; M_i \quad C \text{ is an evaluation context}}{C[M] \longrightarrow \sum_i p_i; C[M_i]}$$

multi-step reduction relation from terms to formal sums

$$\textsc{Mul0}\ \frac{}{M \Longrightarrow 1; M} \qquad \textsc{MulT}\ \frac{M \longrightarrow \sum_i p_i; M_i \qquad M_i \Longrightarrow F_i}{M \Longrightarrow \sum_i p_i{\cdot}F_i}$$

multi-step reduction relation from formal sums to formal sums:

$$\textsc{MulFS}\ \frac{M_i \Longrightarrow F_i}{\sum_{i\in I} p_i; M_i + G \Longrightarrow \sum_{i\in I} p_i{\cdot}F_i + G}\ I \text{ finite}$$

multi-step reduction relation from terms and formal sums to value formal sums

$$\textsc{MulVT}\ \frac{M \Longrightarrow F \qquad \mathtt{val}(F) = Y}{M \Longmapsto Y} \qquad \textsc{MulVFS}\ \frac{F \Longrightarrow F' \qquad \mathtt{val}(F') = Y}{F \Longmapsto Y}$$

the semantic mapping, from terms and formal sums to value formal sums

$$[\![M]\!] \stackrel{\mathtt{def}}{=} \sup\{Y \mid M \Longmapsto Y\} \qquad [\![F]\!] \stackrel{\mathtt{def}}{=} \sup\{Y \mid F \Longmapsto Y\}$$

Fig. 1. Operational semantics for call-by-name

3.1 Environmental bisimulation

In call-by-name, a *probabilistic environmental relation* is a set of elements each of which is of the form (M, N) or $((M, N), Y, Z)$, where M, N, Y, Z are all closed, M, N are $\Lambda_\oplus$-terms and Y, Z value formal sums. Intuitively, in the former elements M and N are terms that we wish to prove equal, and in the latter elements Y and Z are value formal sums obtained from M and N via evaluations and interactions with the environment. We use $\mathcal{R}, \mathcal{S}$ to range over probabilistic environmental relations. In a triple $((M, N), Y, Z)$ the pair component (M, N) is the *static environment*, and Y, Z are the *tested formal sums*. We write $\mathcal{R}_{(M,N)}$ for the relation $\{(Y, Z) \mid ((M, N), Y, Z) \in \mathcal{R}\}$; we accordingly use the infix notation $Y\ \mathcal{R}_{(M,N)}\ Z$, and similarly for $M\ \mathcal{R}\ N$. In the remainder of the paper, when discussing probabilistic environmental relations, bisimulations, simulations, or similar, we abbreviate 'probabilistic environmental' as 'PE', or even omit it when non-ambiguous. Static environments (that is, pairs of $\Lambda_\oplus$-terms) are ranged over by $\mathcal{E}$. If $\mathcal{E} = (M, N)$ then its *context closure*, written $\mathcal{E}^\star$, is the set of all pairs of the form $(C[M], C[N])$. We use a similar notation for the context closure of relations on λ-terms.

Remark 2 (Static environment). Our results would also hold admitting arbitrary sets of pairs of $\Lambda_\oplus$-terms as static environments, rather then single pairs. We have chosen single pairs so to bring up the minimal requirement on static environments for our proofs to hold (notably the congruence for bisimilarity).

Definition 3 (Environmental bisimulation, call-by-name). A PE relation $\mathcal{R}$ is a *(PE) bisimulation* if

(1) $M\ \mathcal{R}\ N$ implies $[\![M]\!]\ \mathcal{R}_{(M,N)}\ [\![N]\!]$;

(2) $\sum_i p_i; \lambda x.M_i \; \mathcal{R}_{\mathcal{E}} \; \sum_j q_j; \lambda x.N_j$ implies:
  (a) $\sum_i p_i = \sum_j q_j$;
  (b) for all $(P, Q) \in \mathcal{E}^\star$, $\sum_i p_i \cdot [\![M_i\{^P/_x\}]\!] \; \mathcal{R}_{\mathcal{E}} \; \sum_j q_j \cdot [\![N_j\{^Q/_x\}]\!]$.

We write $\approx$ for *(PE) bisimilarity*, the union of all PE bisimulations.

While $\approx$ is a PE relation, we are ultimately interested in comparing λ-terms ($M \approx N$ if $M \, \mathcal{R} \, N$ for some bisimulation $\mathcal{R}$).

Remark 4. Using the additional notation defined in Section 3, we can write the bisimulation clauses for formal sums as follows:

(2) $Y \; \mathcal{R}_{\mathcal{E}} \; Z$ implies:
  (a) $\mathtt{weight}(Y) = \mathtt{weight}(Z)$;
  (b) for all $(P, Q) \in \mathcal{E}^\star$, $[\![Y \bullet P]\!] \; \mathcal{R}_{\mathcal{E}} \; [\![Z \bullet Q]\!]$.

Example 5. We have

$$M \stackrel{\mathtt{def}}{=} (\lambda.\lambda.\Omega) \oplus (\lambda.\Omega) \approx \lambda.((\lambda.\Omega) \oplus \Omega) \stackrel{\mathtt{def}}{=} N \, .$$

This is proved noting that $[\![M]\!] = \frac{1}{2}; \lambda.\lambda.\Omega + \frac{1}{2}; \lambda.\Omega$ and $[\![N]\!] = 1; N$, using the bisimulation $\mathcal{R}$ in which $M \, \mathcal{R} \, N$, $[\![M]\!] \; \mathcal{R}_{(M,N)} \; [\![N]\!]$, $\frac{1}{2}; \lambda.\Omega \; \mathcal{R}_{(M,N)} \; \frac{1}{2}; \lambda.\Omega$, and $\emptyset \; \mathcal{R}_{(M,N)} \; \emptyset$. Terms M, N could not be equated by a bisimulation that acted only on terms (ignoring formal sums), as neither $\lambda.\lambda.\Omega$ nor $\lambda.\Omega$ can be equated to N.

Definition 6 (Simulation). In Definition 3, and in the remainder of the paper for other definitions of probabilistic bisimulation, the corresponding *simulation* is obtained by replacing the equality '=' on the weights with '$\leq$'; thus in Definition 3, clause (2a) becomes $\sum_i p_i \leq \sum_j q_j$.

The union of all simulations, *similarity*, is written $\lesssim$.

Theorem 7.

(1) *$\approx$ and $\lesssim$ are the largest bisimulation and simulation, respectively.*
(2) *$\lesssim$ is a preorder, and $\approx$ an equivalence.*
(3) $\approx \; = \; \lesssim \cap \lesssim^{-1}$.

Proof. (1) If $M \approx N$ then there is a bisimulation $\mathcal{R}$ such that $M \; \mathcal{R} \; N$. By the definition of $\mathcal{R}$ and $\approx$ we have $((M, N), [\![M]\!], [\![N]\!]) \in \mathcal{R} \subseteq \approx$. Analogously, if $(\mathcal{E}, Y, Z) \in \approx$ then $(\mathcal{E}, Y, Z) \in \mathcal{R}$ for some bisimulation $\mathcal{R}$ and the formal sums have the same weight and, for all $P, Q \in \mathcal{E}^\star$, $(\mathcal{E}, [\![Y \bullet P]\!], [\![Z \bullet Q]\!]) \in \mathcal{R} \subseteq \approx$. The same holds for simulation.

(2) Identity is a simulation, hence $\lesssim$ is reflexive. If $\mathcal{R}, \mathcal{S}$ are simulations, then their relational composition

$$\begin{aligned} \mathcal{R} \; \mathcal{S} = \{&(M, N) \mid \exists P \text{ such that } M \, \mathcal{R} \, P \, \mathcal{S} \, N\} \\ &\cup \{((M, N), Y, Z) \mid \exists Y', P \text{ such that } Y \, \mathcal{R}_{\,(M,P)} Y' \, \mathcal{S}_{(P,N)} \; Z\} \end{aligned}$$

is a simulation. If $M \; \mathcal{R} \; P \; \mathcal{S} \; N$ then $[\![M]\!] \; \mathcal{R}_{(M,P)} \; [\![P]\!] \; \mathcal{S}_{(P,N)} \; [\![N]\!]$, hence

$$[\![M]\!] (\mathcal{R} \; \mathcal{S})_{(M,N)} [\![N]\!] \, .$$

If $Y \, \mathcal{R}_{\,(M,P)} Y' \, \mathcal{S}_{(P,N)} \; Z$ then $\mathtt{weight}(Y) \leq \mathtt{weight}(Y') \leq \mathtt{weight}(Z)$ and for every C, $[\![Y \bullet C[M]]\!] \; \mathcal{R}_{(M,P)} \; [\![Y' \bullet C[P]]\!] \; \mathcal{S}_{(P,N)} \; [\![Z \bullet C[N]]\!]$. Thus $\lesssim$ is transitive and reflexive. Analogously, $\approx$ is reflexive and transitive, and for any bisimulation $\mathcal{R}$ it holds that

$$\mathcal{R}^{-1} = \{(M, N) \mid N \, \mathcal{R} \, M\} \cup \{((M, N), Y, Z) \mid Z \, \mathcal{R}_{\,(N,M)} Y\}$$

is a bisimulation as well

(3) The result follows from (1) and from the fact that the calculus is deterministic:
  (a) if $\mathcal{R}$ is a bisimulation then both $\mathcal{R}$ and $\mathcal{R}^{-1}$ are simulations;
  (b) to prove that $\lesssim \cap \lesssim^{-1} \subseteq \approx$, we show that $\lesssim \cap \lesssim^{-1}$ is a bisimulation. Let $\mathcal{R}$ and $\mathcal{S}$ be two simulations. If $M\ \mathcal{R}\ N$ and $N\ \mathcal{S}\ M$ then $[\![M]\!]\ \mathcal{R}_{(M,N)}\ [\![N]\!]$ and $[\![N]\!]\ \mathcal{S}_{(N,M)}\ [\![M]\!]$, which implies that $((M,N),[\![M]\!],[\![N]\!]) \in \lesssim\ \cap\ \lesssim^{-1}$. If $((M,N),Y,Z) \in \mathcal{R}$ and $((N,M),Z,Y) \in \mathcal{S}$ then for all C we have $[\![Y \bullet C[M]]\!]\ \mathcal{R}_{(M,N)}\ [\![Z \bullet C[N]]\!]$ and $[\![Z \bullet C[N]]\!]\ \mathcal{S}_{(N,M)}\ [\![Y \bullet C[M]]\!]$. Therefore, $((M,N),[\![Y \bullet C[M]]\!],[\![Z \bullet C[N]]\!]) \in \lesssim \cap \lesssim^{-1}$. Finally the clause on the weights holds by $((M,N),Y,Z) \in \mathcal{R}$ and $((N,M),Z,Y) \in \mathcal{S}$, which imply $\mathtt{weight}(Y) \leq \mathtt{weight}(Z)$ and $\mathtt{weight}(Z) \leq \mathtt{weight}(Y)$, respectively.

□

The bisimilarity, or similarity, is directly defined using the semantics of terms, which is a least-fixed point on top of big-step approximants. When proving properties about bisimilarity and similarity, therefore, we need to reason about such approximants. For this, we introduce a finite-step simulation in which the challenge reductions of the simulation game employ the big-step approximants (the relation $\Longrightarrow$ of Figure 1). We cannot have characterizations of bisimilarity in terms of a finite-step bisimilarity because in general the weights of the approximants of two bisimilar terms are different, as shown in Example 8, and so we should allow related formal sums to have different weights. Hence, to reason about bisimilarity we go through its characterization via similarity (Theorem 7), and then the characterization of similarity via the finite-step similarity (Corollary 12). In this case, since we are defining a preorder rather than an equivalence, we can allow related formal sums to have different weights.

Example 8. Let P and Q be the terms discussed in (2) in Section 1:

$$P \stackrel{\mathtt{def}}{=} RR \quad \text{and} \quad Q \stackrel{\mathtt{def}}{=} \lambda.\Omega\,, \quad \text{for} \quad R \stackrel{\mathtt{def}}{=} \lambda x.((xx) \oplus Q)$$

A bisimulation relating P and Q is

$$\{(P,Q),((P,Q),\textstyle\sum_{n\geq 1} \frac{1}{2^n}; Q, 1; Q),((P,Q),\emptyset,\emptyset)\}\ .$$

We could not prove the equality using finite-step approximants for bisimulation, since those for P are of the form $\sum_{1\leq n\leq m} \frac{1}{2^n}; Q$, for some m, and thus have a smaller total weight than the formal sum $1; Q$ immediately produced by Q.

Definition 9. A PE relation $\mathcal{R}$ is a *finite-step simulation* if

(1) $M\,\mathcal{R}\,N$ and $M \Longmapsto Y$ imply $Y\ \mathcal{R}_{(M,N)}\ [\![N]\!]$;
(2) $\sum_i p_i; \lambda x.M_i\ \mathcal{R}_{\mathcal{E}}\ \sum_j q_j; \lambda x.N_j$ implies:
  (a) $\sum_i p_i \leq \sum_j q_j$;
  (b) for all $(P,Q) \in \mathcal{E}^\star$, if $\sum_i p_i; M_i\{^P/_x\} \Longmapsto Y$ then $Y\ \mathcal{R}_{\mathcal{E}}\ \sum_j q_j \cdot [\![N_j\{^Q/_x\}]\!]$.

We write $\lesssim_{\mathrm{fin}}$ for *finite-step similarity*. In finite-step simulations, the challenges are expressed by finitary reductions, since only a finite number of small-steps reductions $\longrightarrow$ can occur in the derivation of $\Longmapsto$. Moreover, any result about finite-step similarity $\lesssim_{\mathrm{fin}}$ on $\Lambda_\oplus$-terms can be established using a finite-step simulation with only finite formal sums (i.e., formal sums with a finite number of summands) on the challenger side, though this constraint is not required in the definition. Indeed, finite formal sums can only reach finite formal sums via the finitary, multi-step relation.

Remark 10. We do not define clause (2b) of Definition 9 as follows:
for all $(P,Q) \in \mathcal{E}^\star$, *if* $M_i\{^P\!/\!_x\} \Longmapsto Y_i$ *for every* i
then $\sum_i p_i{\cdot}Y_i \; \mathcal{R}_{\mathcal{E}} \; \sum_j q_j{\cdot}[\![N_j\{^Q\!/\!_x\}]\!]$
since we would not know how to prove congruence, and thus soundness, for such proof technique. This is because, as the index set I, *ranged over by* i, *can be infinite, the challenge in the simulation game might not be finitary, i.e., it could allow infinitely many summands to perform a small-step reduction at the same time. By contrast, reduction* $\Longmapsto$ *on formal sums (from Figure 1) is finitary. This allows us to perform proofs by induction on the number of small-steps in a reduction, which is the strategy we use for proving congruence (Lemma 18).* □

We denote by $\texttt{Pairs}(\mathcal{R})$ the set of pairs of terms in a PE relation $\mathcal{R}$. We use two saturation constructions to turn a simulation into a finite-step simulation and conversely. Given a PE relation $\mathcal{R}$, its *saturation by approximants* is

$$\texttt{Pairs}(\mathcal{R}) \cup \{(\mathcal{E}, Y, Z) \mid \text{there is } Y' \text{ with } Y' \; \mathcal{R}_{\mathcal{E}} \; Z \text{ and } Y \leq_{\texttt{apx}} Y' \}$$

and its *saturation by suprema* is $\bigcup_n \mathcal{R}^n$, where

$$\begin{aligned} \mathcal{R}^0 &\stackrel{\texttt{def}}{=} \mathcal{R} \\ \mathcal{R}^{n+1} &\stackrel{\texttt{def}}{=} \mathcal{R}^n \cup \{(\mathcal{E}, Y, Z) \mid \text{there is a countable sequence } \{Y_i\}_{i\geq 0} \\ &\qquad\qquad \text{with } Y_i \; \mathcal{R}^n_{\mathcal{E}} \; Z, \;\; Y_i \leq_{\texttt{apx}} Y_{i+1}, \text{ and } Y = \sup\{Y_i\}_i\}. \end{aligned}$$

The saturation by suprema $\bigcup_n \mathcal{R}^n$ of a relation $\mathcal{R}$ is saturated with respect to triples $(\mathcal{E}, Y, Z)$ where Y is the supremum of an ω-chain $\{Y_i\}_{i\geq 0}$ with respect to $\leq_{\texttt{apx}}$ such that there exists an n such that for all i it holds $Y_i \; \mathcal{R}^n_{\mathcal{E}} \; Z$. However, the relation $\bigcup_n \mathcal{R}^n$ is not itself saturated by suprema in the following sense. There might exist an ω-chain $\{Y_i\}_{i\geq 0}$ with respect to $\leq_{\texttt{apx}}$ such that for all i it holds $Y_i (\bigcup_n \mathcal{R}^n_{\mathcal{E}}) Z$ (i.e., such that for all i there exists an n such that it holds $Y_i \; \mathcal{R}^n_{\mathcal{E}} \; Z$), and yet it does not hold that $\sup\{Y_i\}_i (\bigcup_n \mathcal{R}^n_{\mathcal{E}}) Z$.

Lemma 11.

(1) *The saturation by approximants of a simulation is a finite-step simulation.*
(2) *The saturation by suprema of a finite-step simulation is a simulation.*

Proof. The proof of (1) follows from the definition of $[\![M]\!]$ as the supremum of the set $\{Y \mid M \Longmapsto Y\}$. For (2), the crux is proving by induction on n that if $\sum_i p_i; \lambda x.M_i \; \mathcal{R}^n_{\mathcal{E}} \; \sum_j q_j; \lambda x.N_j$ then:

(1) $\sum_i p_i \leq \sum_j q_j$;
(2) $\sum_i p_i; M_i\{^P\!/\!_x\} \Longmapsto Y$ implies $Y \; \mathcal{R}^n_{\mathcal{E}} \; \sum_j q_j{\cdot}[\![N_j\{^Q\!/\!_x\}]\!]$, for all $(P,Q) \in \mathcal{E}^\star$.

The details of the proof can be found in Appendix A. □

Corollary 12. *The similarity and finite-step similarity preorders,* $\lesssim$ *and* $\lesssim_{\text{fin}}$, *coincide.*

Proof. The result follows from Lemma 11 and from the fact that a simulation (respectively, a finite-step simulation) is included in its saturation by approximants (respectively, by suprema). □

The following example highlights the differences between simulations and finite-step simulations, by proving the equality in Example 8 using finite-step simulations.

Example 13. Terms P and Q of Example 8 can be proved equivalent by exhibiting the following finite-step simulations, where $Y_0 \stackrel{\text{def}}{=} \emptyset$ and $Y_m \stackrel{\text{def}}{=} \sum_{1\leq n\leq m} \frac{1}{2^n}; Q$ for $m \geq 1$:
$\mathcal{R} \stackrel{\text{def}}{=} \{(P,Q), ((P,Q),\emptyset,\emptyset)\} \cup \{((P,Q), Y_m, 1; Q) \mid m \geq 0\}$
$\mathcal{S} \stackrel{\text{def}}{=} \{(Q,P), ((Q,P),\emptyset,\emptyset), ((Q,P), 1; Q, \sum_{n\geq 1} \frac{1}{2^n}; Q)\}$.

By $\mathcal{R}$ we have $P \lesssim_{\text{fin}} Q$, and by $\mathcal{S}$ we have $Q \lesssim_{\text{fin}} P$. Since the saturation constructions used in Lemma 11 allow us to freely move from simulations to finite-step simulations and vice versa, this is equivalent to showing that $P \lesssim Q$ and $Q \lesssim P$, which in turn coincides with proving bisimilarity. Relation $\mathcal{R}$ can be turned into a simulation by saturating it by suprema, i.e., by adding the triple $((P,Q), \sup_m Y_m, 1; Q)$. Indeed, we have $\sup_m Y_m = [\![P]\!]$. Conversely, we can obtain the finite-step simulation $\mathcal{R}$ out of the simulation

$$\{(P,Q), ((P,Q), \textstyle\sum_{n\geq 1} \frac{1}{2^n}; Q, 1; Q), ((P,Q),\emptyset,\emptyset)\}$$

by saturating it by approximants, i.e., by adding to the simulation all pairs

$$((P,Q), \textstyle\sum_{i\in I} \frac{1}{2^i}; Q, 1; Q)$$

where I is a finite subset of natural numbers (we assume that $\sum_{i\in I} \frac{1}{2^i}; Q = \emptyset$ if I is empty). □

Given a relation $\mathcal{R}$, we say that $\mathcal{R}$ is *saturated by approximants* if for every pair $((M,N), Y, Z)$ in $\mathcal{R}$, if $Y' \leq_{\texttt{apx}} Y'$ then $((M,N), Y', Z) \in \mathcal{R}$. Example 13 also shows that finite-step simulations do not have to be saturated by approximants: relation $\mathcal{S}$ is a finite-step simulation but it does not contain, e.g., the triple $((P,Q), \frac{1}{4}; Q, 1; Q)$. However, given a finite-step step simulation, we can always build a finite-step simulation saturated by approximants that contains it, by taking its saturation by approximants.

To derive the substitutivity properties of the similarity, and hence of the bisimilarity, we also need an up-to technique for the finite-step similarity. Specifically, we need an up-to lifting technique whereby, in the simulation game, two derivative formal sums can be decomposed into 'smaller' formal sums and it is then sufficient that these are pairwise related. We write $\texttt{lift}(\mathcal{S})$ for the probabilistic lifting of a relation $\mathcal{S}$ to another relation on formal sums:

$$\begin{aligned}\texttt{lift}(\mathcal{S}) \stackrel{\text{def}}{=} \{(F,G) \mid\ & \text{there are } I, p_i, F_i, G_i, \text{ for } i \in I, \text{ with} \\ & F_i\ \mathcal{S}\ G_i \text{ and } F = \textstyle\sum_i p_i{\cdot}F_i \text{ and } G = \sum_i p_i{\cdot}G_i\}.\end{aligned}$$

Definition 14. A PE relation $\mathcal{R}$ is a *finite-step simulation up-to lifting* if

(1) $M\ \mathcal{R}\ N$ and $M \Longmapsto Y$ imply $Y\ \texttt{lift}(\mathcal{R}_{(M,N)})\ [\![N]\!]$;
(2) $\sum_i p_i; \lambda x.M_i\ \mathcal{R}_{\mathcal{E}}\ \sum_j q_j; \lambda x.N_j$ implies:
  (a) $\sum_i p_i \leq \sum_j q_j$;
  (b) for all $(P,Q) \in \mathcal{E}^\star$, if $\sum_i p_i; M_i\{^P/_x\} \Longmapsto Y$ then $Y\ \texttt{lift}(\mathcal{R}_{\mathcal{E}})\ \sum_j q_j{\cdot}[\![N_j\{^Q/_x\}]\!]$.

Lemma 15. *If $\mathcal{R}$ is a finite-step simulation up-to lifting then $\mathcal{R} \subseteq \lesssim_{\text{fin}}$.*

Proof. Let $\mathcal{R}$ be a finite-step simulation up-to lifting. Then $\mathcal{S}$ is finite-step simulation:

$$\mathcal{S} \stackrel{\text{def}}{=} \texttt{Pairs}(\mathcal{R}) \cup \{((M,N), Y, Z) \mid Y\ \texttt{lift}(\mathcal{R}_{(M,N)})\ Z\}$$

If $M\ \mathcal{S}\ N$ then $M\ \mathcal{R}\ N$, which implies that if $M \Longmapsto Y$ then $Y\ \texttt{lift}(\mathcal{R}_{(M,N)})\ [\![N]\!]$. Hence, $Y\ \mathcal{S}_{(M,N)}\ [\![N]\!]$.
Let $Y\ \mathcal{S}_{(M,N)}\ Z$, i.e., $Y = \sum_i p_i{\cdot}Y_i$, $Z = \sum_i p_i{\cdot}Z_i$ and $Y_i\ \mathcal{R}_{(M,N)}\ Z_i$. For every i, $\texttt{weight}(Y_i) \leq \texttt{weight}(Z_i)$, hence $\texttt{weight}(Y) \leq \texttt{weight}(Z)$.
For every i we have $Y_i \bullet C[M] \Longmapsto Y_i'$ implies $Y_i'\ \texttt{lift}(\mathcal{R}_{(M,N)})\ [\![Z_i \bullet C[N]]\!]$ by the definition of $\mathcal{R}$. Since $Y \bullet C[M] \Longmapsto Y'$ implies that $Y' = \sum_i p_i{\cdot}Y_i'$, for some Y_i' such that

$Y_i \bullet C[M] \Longmapsto Y_i'$, and $[\![Z \bullet C[N]]\!] = \sum_i p_i \cdot [\![Z_i \bullet C[N]]\!]$, then $Y' \,\mathtt{lift}(\,\mathtt{lift}(\,\mathcal{R}_{\,(M,N)})\,)\,[\![Z \bullet C[N]]\!]$. The result follows from $\mathtt{lift}(\,\mathtt{lift}(\,\mathcal{R}_{\,(M,N)})\,) = \mathtt{lift}(\,\mathcal{R}_{\,(M,N)})$. □

Example 16. Let $P \stackrel{\mathtt{def}}{=} \lambda.Q$, $Q \stackrel{\mathtt{def}}{=} \lambda.\Omega$, and

$$M \stackrel{\mathtt{def}}{=} (P \oplus P) \oplus (Q \oplus Q)\,, \quad N \stackrel{\mathtt{def}}{=} (P \oplus Q) \oplus (P \oplus Q)\,.$$

The following finite-step simulation up-to lifting shows $M \lesssim_{\mathrm{fin}} N$:

$$\{(M,N), ((M,N),1;P,1;P), ((M,N),1;Q,1;Q), ((M,N),\emptyset,1;P), ((M,N),\emptyset,1;Q), ((M,N),\emptyset,\emptyset)\}$$

For this example, the 'up-to lifting' technique allows us to have a relation with only empty or Dirac formal sums (i.e., a single summand with probability 1).

Remark 17. We have seen in Example 8 that the terms P and Q defined as follows:

$$P \stackrel{\mathtt{def}}{=} RR \quad \text{and} \quad Q \stackrel{\mathtt{def}}{=} \lambda.\Omega\,, \quad \text{for} \quad R \stackrel{\mathtt{def}}{=} \lambda x.((xx) \oplus Q)$$

cannot be proved equivalent using a bisimulation with small-step, finitary clauses. We could prove the terms equivalent by using a bisimulation with small-step clauses if we allowed the reached formal sums to be decomposed into equally weighted formal sums (formally: using the up-to-distribution-and-lifting technique discussed in Section 3.3). In this case, it would be sufficient to define a bisimulation relating P and Q and their Dirac formal sums, and we would only need to consider the formal sum $\frac{1}{2};P + \frac{1}{2};Q$ (reached by P in one step) and decompose $1;Q$ (the formal sum reached in zero steps by Q) as $\frac{1}{2};Q + \frac{1}{2};Q$.
However, such a bisimulation would not be complete, since the same reasoning would not apply to the terms M and N defined below, where $R_1 \stackrel{\mathtt{def}}{=} \lambda x.((xx) \oplus \lambda.Q)$ and $R_2 \stackrel{\mathtt{def}}{=} \lambda x.((xx) \oplus Q)$

$$M \stackrel{\mathtt{def}}{=} (R_1 R_1) \oplus \lambda.\lambda.Q \quad N \stackrel{\mathtt{def}}{=} \lambda.((R_2 R_2) \oplus \lambda.Q)$$

Terms M and N are contextually equivalent, but cannot be proved equivalent using a small-step, finitary bisimulation. Indeed, $1;N$ is only equivalent to the semantics of M, which is not reachable in a finite number of steps. No decomposition of $1;N$ can be matched with a decomposition of any approximation of the semantics of M.

We can now prove that $\lesssim_{\mathrm{fin}}$ is a precongruence on terms.

Lemma 18. *If $M \lesssim_{\mathrm{fin}} N$ then $C[M] \lesssim_{\mathrm{fin}} C[N]$, for any context C.*

Proof. Given a a finite-step simulation $\mathcal{R}$ saturated by approximants, we prove that the PE relation

$$\begin{array}{l}\{(C[M],C[N]) \mid M\,\mathcal{R}\,N\}\\ \cup\{((C[M],C[N]),1;\lambda x.C'[M],1;\lambda x.C'[N]) \mid M\,\mathcal{R}\,N\}\\ \cup\{((C[M],C[N]),Y,Z) \mid Y\,\mathcal{R}_{\,(M,N)}\,Z\}\\ \cup\{((M,N),\emptyset,Z) \mid \text{ for some } M,N,Z\}\end{array}$$

is a finite-step simulation up-to lifting. The details of the proof can be found in Appendix A. □

Hence, (pre)congruence results for bisimilarity and similarity follow from Lemma 18, Corollary 12 and the fact that $\approx \,=\, \lesssim \cap \lesssim^{-1}$ (Theorem 7).

Corollary 19. *On $\Lambda_\oplus$-terms, $\approx$ is a congruence, and $\lesssim$ a precongruence.*

3.2 Contextual equivalence

Let $M \Downarrow \stackrel{\text{def}}{=} \mathtt{weight}(\llbracket M \rrbracket)$ be the probability of termination of M.

Definition 20 (Contextual preorder and equivalence). M and N are in the *contextual preorder*, written $M \leq_{\mathtt{ctx}} N$, (resp. in *contextual equivalence*, written $M =_{\mathtt{ctx}} N$), if $C[M] \Downarrow \leq C[N] \Downarrow$ (resp. $C[M] \Downarrow = C[N] \Downarrow$), for every context C.

Remark 21. We pointed out in Section 2 that all our definitions and results are for closed terms. The definitions above of contextual preorder and equivalence could actually be used also for open terms. In contrast, the definitions of simulations and bisimulations (and alike relations) are on closed terms only (they are extended to open terms by considering closing λ-abstractions, see the discussion in Section 2).

Lemma 22 (Completeness). *On $\Lambda_{\oplus}$-terms, $\leq_{\mathtt{ctx}} \subseteq \lesssim$.*

Proof. We prove that the following is a simulation:

$$\mathcal{R} \stackrel{\text{def}}{=} (\leq_{\mathtt{ctx}}) \cup \{((M,N), \llbracket C[M] \rrbracket, \llbracket C[N] \rrbracket) \mid M \leq_{\mathtt{ctx}} N\}$$

We have $M \mathcal{R} N$ if and only if $M \leq_{\mathtt{ctx}} N$, which by definition of $\mathcal{R}$ implies $\llbracket M \rrbracket \mathcal{R}_{(M,N)} \llbracket N \rrbracket$. If $Y \mathcal{R}_{(M,N)} Z$ then $Y = \llbracket C[M] \rrbracket$ and $Z = \llbracket C[N] \rrbracket$ for some context C. Hence, for any C' we have $\llbracket Y \bullet C'[M] \rrbracket = \llbracket \llbracket C[M] \rrbracket \bullet C'[M] \rrbracket = \llbracket C[M]C'[M] \rrbracket$ and $\llbracket Z \bullet C'[N] \rrbracket = \llbracket \llbracket C[N] \rrbracket \bullet C'[N] \rrbracket = \llbracket C[N]C'[N] \rrbracket$ (we refer the reader to Appendix A for a proof of these properties) and by the definition of $\mathcal{R}$ we have $\llbracket C[M]C'[M] \rrbracket \mathcal{R}_{(M,N)} \llbracket C[N]C'[N] \rrbracket$.

□

Corollary 23 (Full abstraction). *On $\Lambda_{\oplus}$-terms:*

(1) *relations $\leq_{\mathtt{ctx}}$ and $\lesssim$ coincide.*
(2) *relations $=_{\mathtt{ctx}}$ and $\approx$ coincide.*

Proof. Completeness of the simulation preorder holds by Lemma 22. The converse, i.e., soundness, follows from the fact that $\lesssim$ is a precongruence (Corollary 19) and that $M \lesssim N$ implies $\mathtt{weight}(\llbracket M \rrbracket) \leq \mathtt{weight}(\llbracket N \rrbracket)$ (by clause (2a) of simulation). Hence, $M \lesssim N$ implies $C[M] \lesssim C[N]$ implies $\mathtt{weight}(\llbracket C[M] \rrbracket) \leq \mathtt{weight}(\llbracket C[N] \rrbracket)$.
Completeness and soundness for $\approx$ follow from (1) and the fact that $\approx$ (respectively, $=_{\mathtt{ctx}}$) is the kernel of $\lesssim$ (respectively, $\leq_{\mathtt{ctx}}$), by item (3) of Theorem 7. □

3.3 Up-to techniques

We have pointed out (Example 5 and 8) that our simulations (and bisimulations) have to be based on formal sums and cannot employ finitary reductions, as in ordinary environmental bisimulations, in order to faithfully represent contextual equivalence. However every one of these features is sound and can therefore be used in proof techniques. In this section we show examples of such techniques. These techniques are very limited and we leave for future work the development of more conclusive ones.

Finitary reductions — the possibility of stopping the evaluation of a term after a few β-reductions — are interesting in enhancements with up-to context (the ability of isolating and removing common contexts in derivative terms) because sometimes such common contexts appear in the middle of a reduction. For applicability, up-to context is usually combined with further up-to techniques that allow us to bring up the common contexts. In the first up-to technique, where the coinduction game still uses formal sums, we combine up-to context with up-to lifting, so to be able to decompose related formal sums into pieces with different

common contexts. In the technique, the context closure of the up-to context is only applied onto λ-terms. The closure could probably be made more powerful by applying it also on formal sums, at the price of a more complex proof, but its usefulness is unclear.

In clause (2b) below, and in the remainder of the paper, we use the function `dirac` that takes a set of pairs of λ-terms (M, N) and returns the set of pairs of their (Dirac) formal sums $(1; M, 1; N)$.

Definition 24. A PE relation $\mathcal{R}$ is a *finite-step simulation up-to lifting and context* if:

(1) $M \,\mathcal{R}\, N$ and $M \Longmapsto Y$ imply $Y \,\mathtt{lift}(\,\mathcal{R}_{\,(M,N)})\, [\![N]\!]$;

(2) $\sum_i p_i; \lambda x.M_i \;\mathcal{R}_{\mathcal{E}}\; \sum_j q_j; \lambda x.N_j$ implies:

(a) $\sum_i p_i \leq \sum_j q_j$;

(b) for all $(P, Q) \in \mathcal{E}^\star$, one of the following holds:

- there are F, G such that $\sum_i p_i; M_i\{^P\!/\!_x\} \Longrightarrow F$ and $\sum_j q_j; N_j\{^Q\!/\!_x\} \Longrightarrow G$ with $F \,\mathtt{lift}(\mathtt{dirac}(\mathcal{E}^\star))\, G$;
- if $\sum_i p_i; M_i\{^P\!/\!_x\} \Longmapsto Y$ then $Y \;\mathtt{lift}(\,\mathtt{dirac}(\mathcal{E}^\star) \cup \mathcal{R}_{\mathcal{E}}\,)\; \sum_j q_j \cdot [\![N_j\{^Q\!/\!_x\}]\!]$.

The following lemma proves the soundness of the up-to lifting and context technique.

Lemma 25. *If $\mathcal{R}$ is a finite-step simulation up-to lifting and context then $\mathcal{R} \subseteq \lesssim_{\mathrm{fin}}$.*

Proof. Let $\mathcal{R}$ be a finite-step simulation up-to lifting and context. We prove that

$$\begin{array}{l}\mathcal{R}' \stackrel{\mathsf{def}}{=} \mathtt{Pairs}(\mathcal{R}) \\ \cup \{((M,N), Y, Z) \mid Y' \,\mathcal{R}_{\,(M,N)} Z \text{ and } Y \leq_{\mathtt{apx}} Y', \text{ for some } Y'\} \\ \cup \{((M,N), 1; \lambda x.C[M], 1; \lambda x.C[N]) \mid M \;\mathcal{R}\; N\} \\ \cup \{((M,N), \emptyset, Z) \mid \text{ for some } M, N, Z\}\end{array}$$

is a finite-step simulation up-to lifting, from which the result follows by $\mathcal{R} \subseteq \mathcal{R}'$. The details of the proof can be found in Appendix A. □

Example 26. The up-to lifting and context technique allows us to prove that terms A, B defined in (1) in Section 1 are bisimilar. We prove $A \lesssim B$ using the PE relation

$$\{(A, B), ((A, B), [\![A]\!], [\![B]\!])\} \ .$$

Indeed, $[\![A]\!] = 1; A$ and $[\![B]\!] = \frac{1}{2}; \lambda x.x + \frac{1}{2}; \lambda x.\Omega$ and, for any pair of arguments of the form $(C[A], C[B])$ used to test the formal sums, we have $1; C[A] \oplus \Omega \longrightarrow \frac{1}{2}; C[A] + \frac{1}{2}; \Omega$ and the pair $(\frac{1}{2}; C[A] + \frac{1}{2}; \Omega, \frac{1}{2}; C[B] + \frac{1}{2}; \Omega)$ is in $\mathtt{lift}(\mathtt{dirac}(\{(A, B)\}^\star))$. Analogously, we prove $B \lesssim A$ using the relation $\{(B, A), ((B, A), [\![B]\!], [\![A]\!])\} \cup \{((B, A), \emptyset, Z) \mid \text{for any } Z\}$. In this example, we have only used the option given by the first bullet for case (2b) in the definition of the up-to technique. The option given by the second bullet for case (2b) can be useful for dealing with analogous examples but with terms having an infinitary behavior (see Example 13 and Remark 17), which might force the opponent side in the bisimulation game to directly evaluate to its semantics in order to reply to the challenger.

In the second up-to technique, the game is entirely played on terms, without appeal to formal sums. We present the technique in combination with forms of up-to context, up-to distribution, up-to reduction, and up-to lifting. This technique will allow us to prove the equivalence of two probabilistic fixed-point combinators in Section 3.4.

A *term relation* is a relation $\mathcal{T}_{(M,N)}$ on values of $\Lambda_\oplus$ and the index (M, N) is a pair of $\Lambda_\oplus$-terms. The index corresponds, intuitively, to a static environment of an environmental bisimulation. We use the notation $\mathcal{T}^{\star-}_{(M,N)}$ for $\mathcal{T}_{(M,N)} \cup \{(M, N)\}^\star$.

A term M deterministically reduces to G (notation: $\stackrel{\mathrm{d}}{\Longrightarrow}$) if $M \Longrightarrow G$ and only the last reduction in the sequence may be derived using rule Sum. We write $M \rhd M'$ if M and M' deterministically reduce to the same formal sum, but M' takes fewer steps. That is, there are G, m, m' with $m \geq m'$ and with $M \stackrel{\mathrm{d}}{\Longrightarrow}_m G$, and $M' \stackrel{\mathrm{d}}{\Longrightarrow}_{m'} G$ (where $\stackrel{\mathrm{d}}{\Longrightarrow}_m$ denotes as usual that m 'small steps' are performed, i.e., that m reductions $\longrightarrow$ occur in the derivation of $\stackrel{\mathrm{d}}{\Longrightarrow}$). Thus, in Definition 27, $\rhd^\star \mathcal{T}^{\star -}_{(M,N)}$ is the set

$$\{(P,Q) \mid P \rhd^\star P' \text{ for } P' \text{ with } P'(\mathcal{T}_{(M,N)} \cup \{(M,N)\}^\star)Q\} \ .$$

We write $F =_{\mathtt{dis}} F'$ if F and F' represent the same probability distribution. In the up-to technique below, $\rhd$ gives us the 'up-to reduction', and $=_{\mathtt{dis}}$ the 'up-to distribution'. We use up-to distribution to manipulate formal sums, which are purely syntactic objects. Finally, $\stackrel{\mathrm{d}}{\Longrightarrow}=_{\mathtt{dis}}$ is the composition of the two relations, i.e., $M \stackrel{\mathrm{d}}{\Longrightarrow}=_{\mathtt{dis}} F$ if there is F' with $M \stackrel{\mathrm{d}}{\Longrightarrow} F'$ and $F' =_{\mathtt{dis}} F$.

Definition 27. Let $\mathcal{T}_{(M,N)}$ be a term relation. Then $\{(\mathrm{M},\mathrm{N})\}$ is a *bisimulation up-to context closure, distribution, reduction, and lifting* if

(1) $[\![M]\!]\mathtt{dirac}(\mathcal{T}_{(M,N)})[\![N]\!]$;
(2) if $\lambda x.M' \ \mathcal{T}_{(M,N)} \ \lambda x.N'$ then for all $(P,Q) \in \{(M,N)\}^\star$,

$$M'\{P/x\} \stackrel{\mathrm{d}}{\Longrightarrow}=_{\mathtt{dis}} \mathtt{lift}(\mathtt{dirac}(\rhd^\star \mathcal{T}^{\star -}_{(M,N)})) =_{\mathtt{dis}} \stackrel{\mathrm{d}}{\Longleftarrow} N'\{Q/x\} \ .$$

We first establish the soundness of the up-to distribution and lifting technique. For a relation $\mathcal{R}$ on formal sums, we write $\mathtt{dislift}(\mathcal{R})$ for the set of pairs F, G such that $F =_{\mathtt{dis}} \mathtt{lift}(\mathcal{R}) =_{\mathtt{dis}} G$. The up-to distribution and lifting technique is obtained by substituting $\mathtt{lift}(\cdot)$ with $\mathtt{dislift}(\cdot)$ in definition 14.

Lemma 28. *If $\mathcal{R}$ is a finite-step simulation up-to distribution and lifting then $\mathcal{R} \subseteq \lesssim_{\mathrm{fin}}$.*

The proof follows as the one for the up-to lifting technique (Lemma 15), and exploits the fact that $\mathtt{dislift}(\mathtt{dislift}(\cdot)) = \mathtt{dislift}(\cdot)$. Then, by exploiting this result, we can derive the soundness of the full technique in Definition 27. An environmental relation has two kinds of components, i.e., it contains pairs of terms and it contains triples, each containing a pair of terms and two formal sums. In a bisimulation up-to context closure, distribution, reduction, and lifting we have only allowed the first kind of component, and the soundness proof exhibits the missing triples (the proof can be found in Appendix A).

Lemma 29. *If there is a term relation $\mathcal{T}_{(M,N)}$ such that $\{(M,N)\}$ is a bisimulation up-to context closure, distribution, reduction, and lifting then $(M,N) \in \approx$.*

The restriction to deterministic transitions $\stackrel{\mathrm{d}}{\Longrightarrow}$ (i.e., to transitions where only the last reduction in the sequence may be derived using rule Sum) in the definition of the up-to technique is used in the proof of soundness, in order to carry out an induction over the number of small-steps reductions.

Remark 30. The first clause of the up-to technique in Definition 27 is not sound if $\mathcal{T}_{(M,N)}$ is substituted by $\mathcal{T}^{\star -}_{(M,N)}$: in this case, for any pair of values V, W, relation $\mathcal{T}_{(V,W)} \stackrel{\mathtt{def}}{=} \emptyset$ would satisfy the definition, since $[\![V]\!] = 1;V$, $[\![W]\!] = 1;W$ and $1;V \ \mathtt{dirac}(\{(V,W)\}^\star) \ 1;W$. □

3.4 Fixed-point combinator example

In the reductions of this example, we write a Dirac formal sum $1; M$ as M, so to have reductions between λ-terms. We exploit the up-to technique of Definition 27 to prove the equivalence between two fixed-point combinators. One of the combinators is Υ:

$$\begin{array}{c} \Upsilon \stackrel{\mathtt{def}}{=} \lambda y.y(Dy(Dy)) \\ \text{where } D \stackrel{\mathtt{def}}{=} \lambda y.\lambda x.y(xx)\ . \end{array}$$

For any term L we have

$$\begin{array}{rcl} \Upsilon L & \longrightarrow & L(DL(DL)) \\ \text{and then} \quad DL(DL) & \longrightarrow\!\!\!\!\longrightarrow & L(DL(DL))\ . \end{array} \tag{3}$$

The other combinator at any cycle can probabilistically decide whether to behave differently (i.e., as Turing's fixed-point combinator) or to turn for good into the previous Υ combinator:

$$\begin{array}{rl} & \Upsilon' \stackrel{\mathtt{def}}{=} D'D' \\ \text{where} \quad & D' \stackrel{\mathtt{def}}{=} \lambda x.\lambda y.((y(Dy(Dy))) \oplus (y(xxy)))\ . \end{array}$$

Thus the computation of $\Upsilon' L$ will unveil, for a while, some L's while computing as Turing's combinator, and then will continue unveiling L's by computing as Υ. Indeed, for

$$\Upsilon'_1 \stackrel{\mathtt{def}}{=} \lambda y.((y(Dy(Dy))) \oplus (y(D'D'y)))\,,$$

we have

$$\begin{array}{rcl} \Upsilon' L \longrightarrow \Upsilon'_1 L & \longrightarrow & (L(DL(DL))) \oplus (L(D'D'L)) \\ & \longrightarrow & \frac{1}{2}; L(DL(DL)) + \frac{1}{2}; L(D'D'L)\ . \end{array} \tag{4}$$

We can establish $\Upsilon \approx \Upsilon'$ using the term relation

$$\mathcal{T}_{(\Upsilon,\Upsilon')} \stackrel{\mathtt{def}}{=} \{(\Upsilon, \Upsilon'_1)\}\ .$$

The interesting case is the bisimulation clause for (Υ, Υ'_1). Take any $M\ \{(\Upsilon, \Upsilon')\}^\star\ N$. By (3), we have $\Upsilon M \longrightarrow M(DM(DM))$, whereas by (4), $\Upsilon'_1 N \stackrel{\mathrm{d}}{\Longrightarrow} \frac{1}{2}; N(DN(DN)) + \frac{1}{2}; N(D'D'N)$. Now we could conclude, up-to context closure, distribution, reduction, and lifting, if we can show that the pairs

$$\begin{array}{c} (M(DM(DM)), N(DN(DN))) \\ \text{and } (M(DM(DM)), N(\Upsilon' N)) \end{array}$$

are in $\rhd^\star \mathcal{T}^{\star -}_{(M,N)}$ This holds because: the first pair is in $\{(\Upsilon, \Upsilon')\}^\star$; for the second pair, by (3) we deduce $DM(DM) \rhd \Upsilon M$, and then we have

$$M(DM(DM)) \rhd^\star M(\Upsilon M)\ \{(\Upsilon, \Upsilon')\}^\star\ N(\Upsilon' N).$$

The example also shows the usefulness of static environments (whose terms need not be values) for context closures in 'up-to context' techniques.

4 PROBABILISTIC CALL-BY-VALUE λ-CALCULUS

In call-by-value, the static environments are not anymore sufficient. As in ordinary environmental bisimulations, we need a dynamic environment to record the values produced during the bisimulation game. In ordinary environmental bisimulations we can see such environments as tuples of values. In the probabilistic case formal sums come into the picture. *Environment formal sums* are terms of the form

$$\textstyle\sum_i p_i; \widetilde{V}_i$$

(i.e., sums of weighted tuples) in which all tuples $\widetilde{V}_i$ have the same length and, as for ordinary formal sums, $0 < p_i \leq 1$ for each i and $\sum_i p_i \leq 1$. We call the length of the tuples $\widetilde{V}_i$'s the *length* of the environment formal sum. In case the index set I of the formal sum is empty, i.e., if there are no summands, we assume to have an empty formal sum $\emptyset_n$ for any length n. This index will be generally omitted when clear by the context. The tuples $\widetilde{V}_i$ represent the dynamic environment: the knowledge that an observer has accumulated during the bisimulation game. There may be several such elements $\widetilde{V}_i$, reflecting the possible worlds produced by the probabilistic evaluation. During the bisimulation game, the environment formal sum is updated. Viewing the environment formal sum as a matrix, in which $\widetilde{V}_i$ represents the i-row and the elements $(\widetilde{V}_1)_r, (\widetilde{V}_2)_r, \ldots$ (the r-th element of each row) represent the r-th column, a column is a set of values that the various possible worlds have produced at the same step of the bisimulation game. (This explains why the tuples $\widetilde{V}_i$'s of the sum have the same length.)

More precisely, in the bisimulation game at each possible world i a term M_i (constructed from the $\widetilde{V}_i$'s using a context closure discussed below) is evaluated. The evaluation of M_i yields, probabilistically, a multiset of values (as a formal sum). This multiset is empty when all evaluations from M_i diverge; in this case the whole row i disappears, meaning that in the i-th possible world the observer never receives an answer. When the multiset is non-empty, the row i is split into as many possible worlds as the values in the multiset. For instance if the evaluation of M_i produces V with probability $\frac{1}{2}$ and V' with probability $\frac{1}{3}$ then the row $p_i; \widetilde{V}_i$ is split into the two rows $\frac{1}{2} \cdot p_i; \widetilde{V}_i, V$ and $\frac{1}{3} \cdot p_i; \widetilde{V}_i, V'$.

This splitting operation is captured by the following multiplication of an environment formal sum $\sum_{i \in I} p_i; \widetilde{V}_i$ and a tuple of formal sums $Y_i = \sum_{j \in J_i} p_{i,j}; V_{i,j}$:

$$\textstyle\sum_{i \in I} p_i; \widetilde{V}_i \cdot Y_i \stackrel{\texttt{def}}{=} \sum_{i \in I, j \in J_i} p_i \cdot p_{i,j}; \widetilde{V}_i, V_{i,j} \ .$$

We use $\mathbf{Y}, \mathbf{Z}$ to range over environment formal sums, and we sometimes treat a formal sum as a special case of environment formal sum in which all tuples have length one.

The view of environment formal sums as matrices is illustrated in Figure 2, for an environment formal sum $\mathbf{Y} \stackrel{\texttt{def}}{=} \sum_{1 \leq i \leq 3} p_i; \widetilde{V}_i$ of length 4. The figure also illustrates the extraction of the column r of the formal sum, written $\mathbf{Y}\downarrow_r$, that yields the tuple of values along the same column, and the multiplication of an environment formal sum with formal sums resulting from the semantics of terms, one per row (where $I = \lambda x.x$ is the identity function).

The *dynamic environment* of two environment formal sums of the same length $\sum_{i \in I} p_i; \widetilde{V}_i$ and $\sum_{j \in J} q_j; \widetilde{W}_j$ is the pair of tuples (of tuples of the same length) $(\{\widetilde{V}_i\}_{i \in I}, \{\widetilde{W}_j\}_{j \in J})$. In environmental bisimulations, the input for two higher-order functions is constructed as the context closure of their environments. In call-by-value, the environments have both a static and a dynamic component and the inputs are constructed accordingly. Given a static environment (M, N) and a dynamic environment $(\{\widetilde{V}_i\}_i, \{\widetilde{W}_j\}_j)$, their context closure, written

$$(\{M, \widetilde{V}_i\}_i, \{N, \widetilde{W}_j\}_j)^{\widehat{\star}}$$

is the set of all pairs of tuples $(\{T_i\}_i, \{U_j\}_j)$ for which there is a context C such that for every i we have $T_i = C[M, \widetilde{V}_i]$, and for every j we have $U_j = C[N, \widetilde{W}_j]$. Thus every T_i is obtained from the same context C by filling its holes with the first element M of the static environment and the dynamic environment $\widetilde{V}_i$. Similarly for U_j, using N, the tuple $\widetilde{W}_j$ and

$$\mathbf{Y} = \sum \begin{array}{c} \\ p_1; \\ p_2; \\ p_3; \end{array} \overbrace{\begin{array}{|c|c|c|c|} \hline V_{1,1} & V_{1,2} & V_{1,3} & V_{1,4} \\ \hline V_{2,1} & V_{2,2} & V_{2,3} & V_{2,4} \\ \hline V_{3,1} & V_{3,2} & V_{3,3} & V_{3,4} \\ \hline \end{array}}^{\mathbf{Y}\lfloor_1 \ \mathbf{Y}\lfloor_2 \ \mathbf{Y}\lfloor_3 \ \mathbf{Y}\lfloor_4} \begin{array}{l} \\ \} \ \widetilde{V}_1 \\ \} \ \widetilde{V}_2 \\ \} \ \widetilde{V}_3 \end{array}$$

$$\sum \begin{array}{l} p_1; \\ p_2; \end{array} \begin{array}{|c|c|} \hline V_{1,1} & V_{1,2} \\ \hline V_{2,1} & V_{2,2} \\ \hline \end{array} \begin{array}{l} \cdot[\![I \oplus \Omega]\!] \\ \cdot[\![I \oplus \lambda.\Omega]\!] \end{array} = \sum \begin{array}{l} \frac{p_1}{2}; \\ \frac{p_2}{2}; \\ \frac{p_2}{2}; \end{array} \begin{array}{|c|c|c|} \hline V_{1,1} & V_{1,2} & I \\ \hline V_{2,1} & V_{2,2} & I \\ \hline V_{2,1} & V_{2,2} & \lambda.\Omega \\ \hline \end{array}$$

Fig. 2. Formal sums as matrices

the same context C. Moreover, as we are in call-by-value, C should be a value context, that is, terms T_i and U_j are values for all i, j. Hence, if M or N are not values then $C \neq [\cdot]_1$.

The operational semantics of call-by-value is defined as in call-by-name, provided that the rule for β-reduction and the evaluation contexts are redefined thus:

$$\textsc{BetaV}\ \frac{}{(\lambda x.M)V \longrightarrow 1; M\{^{V}\!/_{x}\}}$$

$$\text{Evaluation contexts} \qquad C = [\cdot] \ \Big| \ CM \ \Big| \ VC$$

The probabilistic lifting of a relation $\mathcal{S}$ on environment formal sums is defined as for call-by-name:

$$\mathtt{lift}(\mathcal{S}) \stackrel{\mathtt{def}}{=} \{(\mathbf{F}, \mathbf{G}) \mid \text{ there are } I,\ p_i, \mathbf{F}_i, \mathbf{G}_i, \text{ for } i \in I, \text{ with } \mathbf{F}_i\ \mathcal{S}\ \mathbf{G}_i \text{ and } F = \textstyle\sum_i p_i{\cdot}\mathbf{F}_i \text{ and } G = \sum_i p_i{\cdot}\mathbf{G}_i\}.$$

4.1 Environmental bisimulation

In call-by-value, a probabilistic environmental relation (that we still abbreviate as PE relation) is like for call-by-name, except that formal sums are replaced by environment formal sums. That is, each element of the relation is either of the form (M, N) (a pair of $\Lambda_{\oplus}$-terms) or $\mathbf{Y}\ \mathcal{R}_{\mathcal{E}}\ \mathbf{Z}$ (two environment formal sums, collecting the dynamic environment, with a static environment).
If $\mathcal{E} = (M, N)$ is a static environment, then $\mathcal{E}_1$ and $\mathcal{E}_2$ denote the projections, i.e., the terms M and N, respectively.

In a PE relation, related environment formal sums are *compatible*, meaning that they have the same length. In the remainder, compatibility of environment formal sums is tacitly assumed.

In clause (1) below we see formal sums as special cases of environment formal sums.

Definition 31 (Environmental bisimulation, call-by-value). A PE relation is a *(PE) bisimulation* if

(1) $M\ \mathcal{R}\ N$ implies $[\![M]\!]\ \mathcal{R}_{(M,N)}\ [\![N]\!]$;
(2) $\sum_i p_i; \widetilde{V}_i\ \mathcal{R}_{\mathcal{E}}\ \sum_j q_j; \widetilde{W}_j$ implies:
  (a) $\sum_i p_i = \sum_j q_j$;

(b) for all r, if $(\widetilde{V}_i)_r = \lambda x.M_i$ and $(\widetilde{W}_j)_r = \lambda x.N_j$ then
for all $(\{T_i\}_i, \{U_j\}_j) \in (\{\mathcal{E}_1, \widetilde{V}_i\}_i, \{\mathcal{E}_2, \widetilde{W}_j\}_j)^{\widehat{\star}}$ we have

$$\textstyle\sum_i p_i; \widetilde{V}_i \cdot [\![M_i\{^{T_i}\!/_x\}]\!] \; \mathcal{R}_{\mathcal{E}} \; \sum_j q_j; \widetilde{W}_j \cdot [\![N_j\{^{U_j}\!/_x\}]\!] \; ;$$

(c) $\sum_i p_i; \widetilde{V}_i \cdot [\![\mathcal{E}_1]\!] \; \mathcal{R}_{\mathcal{E}} \; \sum_j q_j; \widetilde{W}_j \cdot [\![\mathcal{E}_2]\!]$.

(PE) bisimilarity, $\approx$, is the union of all PE bisimulations, and the corresponding *similarity* is $\lesssim$.

The structure of the above definition is similar to that of ordinary environmental bisimulations. There are three main differences: first, the appearance of formal sums and of the probability measures (notably in clause (2a)); second, the use of an (infinitary) big-step semantics, rather than a small-step, which shows up in the function $[\![\ ]\!]$ in clauses (1), (2b) and (2c); thirdly the appearance of a static environment, that is used in the context closure and in clauses (1) and (2c). In clause (2b), the related environment formal sums, viewed as matrices, grow by the addition of a new column resulting, on left-hand side, from the multiplication of each row $p_i; \widetilde{V}_i$ with the formal sum $[\![M_i\{^{T_i}\!/_x\}]\!]$, and similarly on the right-hand side. Thus the compatibility between related environment formal sums is maintained. Clause (2c) allows to re-evaluate the static environment at any time. This clause and other features are necessary in order to achieve full abstraction in the imperative case (see Section 5.1); they could be removed in pure call-by-value, following [8].

Remark 32. Clause (1) could be substituted by

(1′) $M \; \mathcal{R} \; N$ implies $1; \emptyset \; \mathcal{R}_{(M,N)} \; 1; \emptyset$

where $1; \emptyset$ is the Dirac formal sum with empty environment. Clause (2c) then guarantees that $[\![M]\!] \; \mathcal{R}_{(M,N)} \; [\![N]\!]$. We did not use this definition for continuity with the call-by-name case, and since it is not needed for pure calculi. By contrast, a modification of clause (1) analogous to (1′) is used in Definition 42 of bisimulation for imperative calculi (see Example 43).

Example 33. We have seen in Example 5 that the following terms M and N are equivalent in call-by-name:

$$M \stackrel{\mathsf{def}}{=} (\lambda.\lambda.\Omega) \oplus (\lambda.\Omega) \qquad N \stackrel{\mathsf{def}}{=} \lambda.((\lambda.\Omega) \oplus \Omega) \, .$$

In call-by-value, the presence of the dynamic environment in the definition of bisimulation allows us to distinguish the terms. A call-by-value bisimulation relating these terms should contain the formal sums $[\![M]\!] = \frac{1}{2}; \lambda.\lambda.\Omega + \frac{1}{2}; \lambda.\Omega$ and $[\![N]\!] = 1; N$, with static environment $\mathcal{E} = (M, N)$, and thus the triple $(\mathcal{E}, \frac{1}{2}; \lambda.\lambda.\Omega, \lambda.\Omega, \frac{1}{2}; N, \lambda.\Omega)$ would be in the relation as well. However, the values in the first column of the dynamic environment can be tested again, by clause (2b) of bisimulation, leading to the triple

$$(\mathcal{E}, \tfrac{1}{2}; \lambda.\lambda.\Omega, \lambda.\Omega, \lambda.\Omega, \tfrac{1}{4}; N, \lambda.\Omega, \lambda.\Omega) \ ,$$

which does not satisfy clause (2a).

We want the bisimulation to distinguish the terms in call-by-value since M and N are not contextually equivalent in a call-by-value setting, in contrast with call-by-name. In call-by-value, the context $C \stackrel{\mathsf{def}}{=} (\lambda x.x \, (x \, \lambda y.y))[\cdot]$ separates the two terms. The context first evaluates the given term in argument position, and then copies the value resulting from the evaluation and executes it (i.e., feeds it with an argument) twice. We have that $C[M]$ has probability one half of returning a value, and $C[N]$ has probability a quarter. In the first case, both value $\lambda.\lambda.\Omega$ and value $\lambda.\Omega$ are copied with probability one half, and thus

$C[M]$ has probability one half of converging (i.e., one half times the probability that $\lambda.\lambda.\Omega$ converges two times in a row when applied to an argument, which is one) and one half of diverging (i.e., one half times the probability that $\lambda.\Omega$ converges two times in a row when applied to an argument, which is zero). In $C[N]$, the term N itself is copied with probability one, so $C[N]$ converges with probability one quarter (i.e., one times the probability that N converges two times in a row when applied to an argument, which is one half times one half).

The main results for environmental bisimilarity and similarity in call-by-value (congruence and full abstraction with respect to contextual preorder and equivalence) are as for call-by-name, and the structure of the proofs is similar. The details are however different due to the presence of dynamic environments. As for call-by-name, so in call-by-value to reason about bisimilarity and similarity we need a finite-step simulation, with challenges produced by the finitary big-step approximants. To make sure that the challenges are finite-step, we define *extended environment formal sums*, i.e., terms

$$\textstyle\sum_i p_i; \widetilde{V}_i; M_i$$

in which the environment formal sum $\sum_i p_i; \widetilde{V}_i$ is extended with an additional column of arbitrary $\Lambda_\oplus$-term (not necessarily values). Intuitively, an element $\widetilde{V}_i; M_i$ indicates that the λ-term M_i has to be run with an observer whose knowledge is $\widetilde{V}_i$. Extended environment formal sums are ranged over by $\mathbf{F}, \mathbf{G}$ and $\mathtt{val}(\mathbf{F})$ is defined analogously to formal sums:

$$\mathtt{val}(\textstyle\sum_i p_i; \widetilde{V}_i; M_i) \stackrel{\mathtt{def}}{=} \sum_{\{i | M_i \text{is a value}\}} p_i; \widetilde{V}_i, M_i \ .$$

Extended environment formal sums allow us to define the multi-step reduction relation from extended environment formal sums to environment formal sums: for $\mathbf{F} = \sum_{i\in I} p_i; \widetilde{V}_i; M_i + \mathbf{G}$, where I is a finite set, we set

$$\frac{M_i \Longmapsto Y_i \quad \text{for every } i}{\mathbf{F} \Longmapsto \sum_{i\in I} p_i; \widetilde{V}_i \cdot Y_i + \mathtt{val}(\mathbf{G})}$$

This intuitively corresponds to the multi-step reduction relation from formal sums to value formal sums. For an extended environment formal sum $\mathbf{F} = \sum_i p_i; \widetilde{V}_i; M_i$, we let $[\![\mathbf{F}]\!] \stackrel{\mathtt{def}}{=} \sup\{\mathbf{Y} \mid \mathbf{F} \Longmapsto \mathbf{Y}\}$, and we have $\sum_i p_i; \widetilde{V}_i \cdot [\![M_i]\!] = [\![\mathbf{F}]\!]$.

Additional notation. As we shall now describe, we extend to (extended) environment formal sums the notations for β-reduction, contexts, and application on formal sums, and introduce a notation for extending the dynamic environments. We use r to range over indexes of columns of environment formal sums, and we let $\mid \mathbf{Y} \mid$ denote the length of an environment formal sum $\mathbf{Y}$. Abusing notation, we sometimes write $\mid \mathbf{Y} \mid$ also for the index set $\{1, ..., \mid \mathbf{Y} \mid\}$. Let $\mathbf{Y} = \sum_i p_i; \widetilde{V}_i$ and $\mathbf{F} = \sum_i p_i; \widetilde{V}_i; M_i$ and P be a term.

- for any r, if $(\widetilde{V}_i)_r = \lambda x.M_i$ we let $\mathbf{Y}; \mathbf{Y}\!\mid_r \bullet P \stackrel{\mathtt{def}}{=} \sum_i p_i; \widetilde{V}_i; M_i\{P/x\}$;
- for any r, if $(\widetilde{V}_i)_r = \lambda x.M_i$ and C is a context with holes with indexes ranging over $\mid \mathbf{Y} \mid +1$ we let $\mathbf{Y}; \mathbf{Y}\!\mid_r \bullet C[P, \mathbf{Y}] \stackrel{\mathtt{def}}{=} \sum_i p_i; \widetilde{V}_i; M_i\{C[P, \widetilde{V}_i]/x\}$;
- $\mathbf{Y}; P \stackrel{\mathtt{def}}{=} \sum_i p_i; \widetilde{V}_i; P$;
- for any context C, we let $C[\mathbf{F}] \stackrel{\mathtt{def}}{=} \sum_i p_i; \widetilde{V}_i; C[M_i]$;
- for $F_j = \sum_{i\in I_j} p_{j,i}; M_{j,i}$, we let $\sum_j q_j; C_j[F_j] \stackrel{\mathtt{def}}{=} \sum_{j,i\in I_j} q_j \cdot p_{j,i}; C_j[M_{j,i}]$.

Using this notation, the call-by-value bisimulation clauses for environment formal sums become as follows:

(2) $\mathbf{Y}\ \mathcal{R}_{\mathcal{E}}\ \mathbf{Z}$ implies:
  (a) $\mathtt{weight}(\mathbf{Y}) = \mathtt{weight}(\mathbf{Z})$;
  (b) for all r and for all contexts C, $[\![\mathbf{Y}; \mathbf{Y}\lfloor_r \bullet C[\mathcal{E}_1, \mathbf{Y}]]\!]\ \mathcal{R}_{\mathcal{E}}\ [\![\mathbf{Z}; \mathbf{Z}\lfloor_r \bullet C[\mathcal{E}_2, \mathbf{Z}]]\!]$;
  (c) $[\![\mathbf{Y}; \mathcal{E}_1]\!]\ \mathcal{R}_{\mathcal{E}}\ [\![\mathbf{Z}; \mathcal{E}_2]\!]$.

As for call-by-name, we will use this additional notation only for proofs.

We first establish the basic properties of similarity and bisimilarity.

Theorem 34.

(1) $\approx$ *and* $\lesssim$ *are the largest bisimulation and simulation, respectively.*
(2) $\lesssim$ *is a preorder, and* $\approx$ *an equivalence.*
(3) $\approx\ =\ \lesssim \cap \lesssim^{-1}$.

Proof. The proof follows analogously to call-by-name, except for transitivity (item (2)), which requires some modifications. Clause (2b) of (bi)simulation allows M, N to be given as argument to functions (using context $C = [\cdot]_1$) if and only if they are both values; thus when one of them is not a value we cannot use them as arguments. This prevents us from concluding that, given relations $\mathcal{R}_{(M,N)}$ and $\mathcal{R}'_{(N,P)}$ satisfying the simulation clauses, their relational composition is a simulation. Indeed, whenever M, P are values and N is not a value, we want to build a relation $\mathcal{S}_{M,P}$ on environment formal sums such that the context $C = [\cdot]_1$ can be applied in clause (2b). If N is not a value, however, the simulations $\mathcal{R}_{(M,N)}$ and $\mathcal{R}'_{(N,P)}$ do not allow us to use this context. The issue can be solved by noticing that whenever M, P are values we can apply clause (2c) to $\mathcal{R}_{(M,N)}$ and $\mathcal{R}'_{(N,P)}$, which adds M and P as a column of the respective dynamic environments, and then allows us to use them as arguments. In other words, we have to work with a relation that allows environments to be extended with new columns. Adding columns is safe because it means enlarging the dynamic environment: terms that are equal in the larger environment are also so in the smaller environment as the tests that can be built (when playing the bisimulation game) with the latter environment are a subset of those that are obtained from the former environment. Define the following preorder $\leq_{\mathtt{env}}$ on pairs of formal sums (where, for each pair, the formal sums in the pair have the same length): $(\mathbf{Y}, \mathbf{Z})$ are below formal sums $(\mathbf{Y}', \mathbf{Z}')$ if, in the second pair, the dynamic environment of the first pair is extended and the columns have been possibly permuted. Formally, $(\mathbf{Y}, \mathbf{Z}) \leq_{\mathtt{env}} (\mathbf{Y}', \mathbf{Z}')$ if $\mathbf{Y} = \sum_i p_i; \widetilde{V}_i, \mathbf{Z} = \sum_j q_j; \widetilde{W}_j, \mathbf{Y}' = \sum_i p_i; \widetilde{V}'_i, \mathbf{Z}' = \sum_j q_j; \widetilde{W}'_j$ and for every index r in $\mid \mathbf{Y} \mid$ there is an index r' in $\mid \mathbf{Y}' \mid$ such that $\mathbf{Y}\lfloor_r = \mathbf{Y}'\lfloor_{r'}$ and $\mathbf{Z}\lfloor_r = \mathbf{Z}'\lfloor_{r'}$. Let $\mathcal{R}_{(M,N)}$ and $\mathcal{R}'_{(N,P)}$ be relations on environment formal sums satisfying the simulation clauses. We show that relation $\mathcal{S}_{(M,P)} \stackrel{\mathtt{def}}{=} \geq_{\mathtt{env}} (\mathcal{R}_{(M,N)} \circ \mathcal{R}'_{(N,P)})$, that is,

$$\{(\mathbf{Y}, \mathbf{Z}) \mid \exists \mathbf{Y}', \mathbf{X}', \mathbf{Z}' \text{ such that } \mathbf{Y}'\ \mathcal{R}_{(M,N)}\ \mathbf{X}' \text{ and } \mathbf{X}'\ \mathcal{R}'_{(N,P)}\ \mathbf{Z}' \text{ and } (\mathbf{Y}, \mathbf{Z}) \leq_{\mathtt{env}} (\mathbf{Y}', \mathbf{Z}')\}$$

satisfies the simulation clauses. The interesting case is proving clause (2b) when M, P are values and N is not. In this case, if the context used is different from $C = [\cdot]_1$ then the we can directly appeal to clause (2b) for $\mathcal{R}_{(M,N)}$ and $\mathcal{R}'_{(N,P)}$. If $C = [\cdot]_1$ then we have $\mathbf{Y}'\ \mathcal{R}_{(M,N)}\ \mathbf{X}'$ and $\mathbf{X}'\ \mathcal{R}'_{(N,P)}\ \mathbf{Z}'$ and $(\mathbf{Y}, \mathbf{Z}) \leq_{\mathtt{env}} (\mathbf{Y}', \mathbf{Z}')$, and we want to show $[\![\mathbf{Y}; \mathbf{Y}\lfloor_r \bullet M]\!]\ \mathcal{S}_{(M,P)}\ [\![\mathbf{Z}; \mathbf{Z}\lfloor_r \bullet P]\!]$. By applying clause (2c) to $\mathbf{Y}'\ \mathcal{R}_{(M,N)}\ \mathbf{X}'$ and $\mathbf{X}'\ \mathcal{R}'_{(N,P)}\ \mathbf{Z}'$ we have $\mathbf{Y}''\ \mathcal{R}_{(M,N)}\ \mathbf{X}''$ and $\mathbf{X}''\ \mathcal{R}'_{(N,P)}\ \mathbf{Z}''$, for $\mathbf{Y}'' = [\![\mathbf{Y}'; M]\!]$, $\mathbf{X}'' = [\![\mathbf{X}'; N]\!]$, $\mathbf{Z}'' = [\![\mathbf{Z}'; P]\!]$. Since M and P are values, they are respectively the only terms appearing in the last columns of $\mathbf{Y}''$ and $\mathbf{Z}''$, and we can use them to mimic context $C = [\cdot]_1$. Moreover, by $(\mathbf{Y}, \mathbf{Z}) \leq_{\mathtt{env}} (\mathbf{Y}'', \mathbf{Z}'')$ there is a column r' of $(\mathbf{Y}'', \mathbf{Z}'')$ such that $\mathbf{Y}\lfloor_r = \mathbf{Y}''\lfloor_{r'}$ and $\mathbf{Z}\lfloor_r = \mathbf{Z}''\lfloor_{r'}$. Hence,

there are r' and C' (specifically, with C' being the context hole which gives the last column of $\mathbf{Y}''$ and $\mathbf{Z}''$) such that

$$(\llbracket \mathbf{Y}; \mathbf{Y}\lfloor_r \bullet M \rrbracket, \llbracket \mathbf{Z}; \mathbf{Z}\lfloor_r \bullet P \rrbracket) \quad \leq_{\mathtt{env}} \quad (\llbracket \mathbf{Y}''; \mathbf{Y}''\lfloor_{r'} \bullet M \rrbracket, \llbracket \mathbf{Z}''; \mathbf{Z}''\lfloor_{r'} \bullet P \rrbracket)$$
$$= (\llbracket \mathbf{Y}''; \mathbf{Y}''\lfloor_{r'} \bullet C'[M, \mathbf{Y}''] \rrbracket, \llbracket \mathbf{Z}''; \mathbf{Z}''\lfloor_{r'} \bullet C'[P, \mathbf{Z}''] \rrbracket)$$

and, by clause (2b),

$$\llbracket \mathbf{Y}''; \mathbf{Y}''\lfloor_{r'} \bullet C'[M, \mathbf{Y}''] \rrbracket \; \mathcal{R}_{(M,N)} \; \llbracket \mathbf{X}''; \mathbf{X}''\lfloor_{r'} \bullet C'[N, \mathbf{X}''] \rrbracket$$
$$\llbracket \mathbf{X}''; \mathbf{X}''\lfloor_{r'} \bullet C'[N, \mathbf{X}''] \rrbracket \; \mathcal{R}'_{(N,P)} \; \llbracket \mathbf{Z}''; \mathbf{Z}''\lfloor_{r'} \bullet C'[P, \mathbf{Z}''] \rrbracket.$$

Then the results follows, i.e., $\llbracket \mathbf{Y}; \mathbf{Y}\lfloor_r \bullet M \rrbracket \; \mathcal{S}_{(M,P)} \; \llbracket \mathbf{Z}; \mathbf{Z}\lfloor_r \bullet P \rrbracket$. □

Definition 35. A PE relation is a *finite-step simulation* if

(1) $M \, \mathcal{R} \, N$ and $M \Longmapsto Y$ imply $Y \; \mathcal{R}_{(M,N)} \; \llbracket N \rrbracket$;
(2) $\sum_i p_i; \widetilde{V}_i \; \mathcal{R}_{\mathcal{E}} \; \sum_j q_j; \widetilde{W}_j$ implies:
  (a) $\sum_i p_i \leq \sum_j q_j$;
  (b) for all r, if $(\widetilde{V}_i)_r = \lambda x.M_i$ and $(\widetilde{W}_j)_r = \lambda x.N_j$ then
    for all $(\{T_i\}_i, \{U_j\}_j) \in (\{\mathcal{E}_1, \widetilde{V}_i\}_i, \{\mathcal{E}_2, \widetilde{W}_j\}_j)^{\hat{\star}}$ we have
    $\sum_i p_i; \widetilde{V}_i; M_i\{^{T_i}/_x\} \Longmapsto \mathbf{Y}$ implies $\mathbf{Y} \; \mathcal{R}_{\mathcal{E}} \; \sum_j q_j; \widetilde{W}_j \cdot \llbracket N_j\{^{U_j}/_x\} \rrbracket$;
  (c) $\sum_i p_i; \widetilde{V}_i; \mathcal{E}_1 \Longmapsto \mathbf{Y}$ implies $\mathbf{Y} \, \mathcal{R}_{\,\mathcal{E}} \, \sum_j q_j; \widetilde{W}_j \cdot \llbracket \mathcal{E}_2 \rrbracket$.

We write $\lesssim_{\mathrm{fin}}$ for the union of all finite-step simulations. Analogously to call-by-name, we use a saturation by approximants and a saturation by suprema to move from a simulation to a finite-step simulation and conversely, and exploit this to prove that similarity and finite-step similarity coincide.

Lemma 36. *Relations $\lesssim$ and $\lesssim_{\mathrm{fin}}$ coincide.*

The proof follows as in call-by-name. We prove that the saturation by approximants of a simulation is a finite-step simulation and that the saturation by suprema of a finite-step simulation is a simulation. Clause (2c) is treated analogously to clause (2b).

As in call-by-name, we derive congruence for bisimilarity and similarity by first proving the property for finite-step similarity. We also exploit a combination of two up-to techniques for finite-step simulation, namely up-to lifting and up-to environment. Up-to lifting is defined analogously to call-by-name, using the lifting operation on environment formal sums. Up-to environment is based on the preorder $\leq_{\mathtt{env}}$ on pairs of formal sums, as defined in the proof of Theorem 34, which allows us to exchange columns of environment formal sums (when these are viewed as matrices as in Figure 2) and to add new columns. In the up-to lifting and environment technique, we combine this preorder with the probabilistic lifting of a relation. Then, $\mathbf{Y} \; \mathtt{lift}(\geq_{\mathtt{env}} (\mathcal{R})) \; \mathbf{Z}$ holds if there are p_i, $\mathbf{Y}_i$ and $\mathbf{Z}_i$, for i ranging over some index set, such that $\mathbf{Y} = \sum_i p_i \cdot \mathbf{Y}_i$ and $\mathbf{Z} = \sum_i p_i \cdot \mathbf{Z}_i$, and for every i there are $\mathbf{Y}'_i, \mathbf{Z}'_i$ such that $\mathbf{Y}'_i \; \mathcal{R} \; \mathbf{Z}'_i$ and $(\mathbf{Y}_i, \mathbf{Z}_i) \leq_{\mathtt{env}} (\mathbf{Y}'_i, \mathbf{Z}'_i)$.

Definition 37. A PE relation is a *probabilistic finite-step simulation up-to lifting and environment* if:

(1) $M \, \mathcal{R} \, N$ and $M \Longmapsto Y$ imply $Y \; \mathtt{lift}(\geq_{\mathtt{env}} (\mathcal{R}_{(M,N)})) \; \llbracket N \rrbracket$;
(2) $\sum_i p_i; \widetilde{V}_i \; \mathcal{R}_{\mathcal{E}} \; \sum_j q_j; \widetilde{W}_j$ implies:
  (a) $\sum_i p_i \leq \sum_i q_i$;

(b) for all r, if $(\widetilde{V}_i)_r = \lambda x.M_i$ and $(\widetilde{W}_j)_r = \lambda x.N_j$ then
for all $(\{T_i\}_i, \{U_j\}_j) \in (\{\mathcal{E}_1, \widetilde{V}_i\}_i, \{\mathcal{E}_2, \widetilde{W}_j\}_j)^{\widehat{\star}}$ we have
$\sum_i p_i; \widetilde{V}_i; M_i\{T_i/x\} \Longmapsto \mathbf{Y}$ implies $\mathbf{Y}\,\mathtt{lift}(\geq_{\mathtt{env}}(\mathcal{R}_{\mathcal{E}}))\,\sum_j q_j; \widetilde{W}_j \cdot [\![N_j\{U_j/x\}]\!]$;

(c) $\sum_i p_i; \widetilde{V}_i; \mathcal{E}_1 \Longmapsto \mathbf{Y}$ implies $\mathbf{Y}\,\mathtt{lift}(\geq_{\mathtt{env}}(\mathcal{R}_{\mathcal{E}}))\,\sum_j q_j; \widetilde{W}_j \cdot [\![\mathcal{E}_2]\!]$.

We now prove that the up-to lifting and environment technique is sound.

Theorem 38. *If $\mathcal{R}$ is a finite-step simulation up-to lifting and environment then $\mathcal{R} \subseteq \lesssim_{\mathrm{fin}}$.*

Proof. Let $\mathcal{R}$ be a finite-step simulation up-to lifting and environment. Then the following is a finite-step simulation:

$$\mathcal{S} = \mathtt{Pairs}(\mathcal{R}) \cup \bigcup_{\mathcal{S}} \mathtt{lift}(\geq_{\mathtt{env}}(\mathcal{R}_{\mathcal{E}}))$$

If $M\ \mathcal{S}\ N$ then $M\,\mathcal{R}\,N$, which implies that if $M \Longmapsto \mathbf{Y}$ then $\mathbf{Y}\,\mathtt{lift}(\geq_{\mathtt{env}}(\mathcal{R}_{(M,N)}))\,[\![N]\!]$. Hence, $\mathbf{Y}\ \mathcal{S}_{(M,N)}\ [\![N]\!]$.
Let $\mathbf{Y}\ \mathcal{S}_{(M,N)}\ \mathbf{Z}$, i.e., $\mathbf{Y} = \sum_i p_i{\cdot}\mathbf{Y}_i$, $\mathbf{Z} = \sum_i p_i{\cdot}\mathbf{Z}_i$ and for every i there are $\mathbf{Y}'_i\ \mathcal{R}_{(M,N)}\ \mathbf{Z}'_i$ such that $(\mathbf{Y}_i, \mathbf{Z}_i) \leq_{\mathtt{env}} (\mathbf{Y}'_i, \mathbf{Z}'_i)$.
Clause (a) holds since for every i, $\mathtt{weight}(\mathbf{Y}'_i) \leq \mathtt{weight}(\mathbf{Z}'_i)$. Therefore, $\mathtt{weight}(\mathbf{Y}) \leq \mathtt{weight}(\mathbf{Z})$.
To prove clause (b), suppose that $\mathbf{Y}; \mathbf{Y}\lfloor_r \bullet C[M, \mathbf{Y}] \Longmapsto \mathbf{W}$. Then $\mathbf{W} = \sum_i p_i{\cdot}\mathbf{W}_i$, for some $\mathbf{W}_i$ such that $\mathbf{Y}_i; \mathbf{Y}_i\lfloor_r \bullet C[M, \mathbf{Y}_i] \Longmapsto \mathbf{W}_i$. Analogously, we have

$$[\![\mathbf{Z}; \mathbf{Z}\lfloor_r \bullet C[N, \mathbf{Z}]]\!] = \textstyle\sum_i p_i{\cdot}[\![\mathbf{Z}_i; \mathbf{Z}_i\lfloor_r \bullet C[N, \mathbf{Z}_i]]\!].$$

Suppose that $\mathbf{Y}_i; \mathbf{Y}_i\lfloor_r \bullet C[M, \mathbf{Y}_i] = \sum_j q_j; \widetilde{V}_j; M_j$ and $\mathbf{Z}_i; \mathbf{Z}_i\lfloor_r \bullet C[M, \mathbf{Z}_i] = \sum_k q'_k; \widetilde{W}_k; N_k$. Then it follows from the definition of the environment preorder $\leq_{\mathtt{env}}$ that there are r_i and C_i such that $\mathbf{Y}'_i; \mathbf{Y}'_i\lfloor_{r_i} \bullet C_i[M, \mathbf{Y}'_i] = \sum_j q_j; \widetilde{V}'_j; M_j$ and $\mathbf{Z}'_i; \mathbf{Z}'_i\lfloor_{r_i} \bullet C_i[N, \mathbf{Z}'_i] = \sum_k q'_k; \widetilde{W}'_k; N_k$. Therefore, there is a $\mathbf{W}'_i$ such that $\mathbf{Y}'_i; \mathbf{Y}'_i\lfloor_{r_i} \bullet C_i[M, \mathbf{Y}'_i] \Longmapsto \mathbf{W}'_i$ and

$$(\mathbf{W}_i, [\![\mathbf{Z}_i; \mathbf{Z}_i\lfloor_r \bullet C[N, \mathbf{Z}_i]]\!]) \leq_{\mathtt{env}} (\mathbf{W}'_i, [\![\mathbf{Z}'_i; \mathbf{Z}'_i\lfloor_{r_i} \bullet C_i[N, \mathbf{Z}'_i]]\!])$$

with $\mathbf{W}'_i\,\mathtt{lift}(\geq_{\mathtt{env}}(\mathcal{R}_{(M,N)}))\,[\![\mathbf{Z}'_i; \mathbf{Z}'_i\lfloor_{r_i} \bullet C_i[N, \mathbf{Z}'_i]]\!]$, since $\mathcal{R}$ is a finite-step simulation up-to lifting and environment and $\mathbf{Y}'_i\ \mathcal{R}_{(M,N)}\ \mathbf{Z}'_i$. Hence, we have

$$\mathbf{W}\,\mathtt{lift}(\geq_{\mathtt{env}}(\mathtt{lift}(\geq_{\mathtt{env}}(\mathcal{R}_{(M,N)}))))\,[\![\mathbf{Z}; \mathbf{Z}\lfloor_r \bullet C[N, \mathbf{Z}]]\!]$$

and the result follows from $\mathtt{lift}(\geq_{\mathtt{env}}(\mathtt{lift}(\geq_{\mathtt{env}}(\mathcal{R}_{(M,N)})))) = \mathtt{lift}(\geq_{\mathtt{env}}(\mathcal{R}_{(M,N)}))$.
Finally, clause (c) follows analogously to clause (b).

□

Having these up-to techniques, we derive the result by showing that the context closure (which, differently from call-by-name, is now applied both to terms and to the environments of formal sums) of a finite-step simulation saturated by approximants is a finite-step simulation up-to lifting and environment.

Lemma 39. *If $M \lesssim_{\mathrm{fin}} N$ then for every context C we have $C[M] \lesssim_{\mathrm{fin}} C[N]$.*

Proof. We first define, for any pairs of terms (M, N), the preorder $\leq_{\mathtt{cce}(M,N)}$ (context closure of environments) on pairs of environment formal sums: $(\mathbf{Y}, \mathbf{Z}) \leq_{\mathtt{cce}(M,N)} (\mathbf{Y}', \mathbf{Z}')$ if $\mathbf{Y} = \sum_i p_i; \widetilde{V}_i$, $\mathbf{Z} = \sum_j q_j; \widetilde{W}_j$ with $|\ \mathbf{Y}\ | = |\ \mathbf{Z}\ |$, $\mathbf{Y}' = \sum_i p_i; \widetilde{V}'_i$, $\mathbf{Z}' = \sum_j q_j; \widetilde{W}'_j$ with $|\ \mathbf{Y}'\ | = |\ \mathbf{Z}'\ |$ and

- for every index r in $|\ \mathbf{Y}\ |$ there is an index r' in $|\ \mathbf{Y}'\ |$ such that such that $\mathbf{Y}\lfloor_r = \mathbf{Y}'\lfloor_{r'}$ and $\mathbf{Z}\lfloor_r = \mathbf{Z}'\lfloor_{r'}$

- for every index r' in $\mid \mathbf{Y}' \mid$ there is a value-context C whose indexes range over $\mid \mathbf{Y} \mid +1$ such that for every i, j, $(\widetilde{V}_i')_{r'} = C[M, \widetilde{V}_i]$ and $(\widetilde{W}_j')_{r'} = C[N, \widetilde{W}_j]$ (i.e., $(\mathbf{Y}'\!\upharpoonright_{r'}, \mathbf{Z}'\!\upharpoonright_{r'}) \in (\{(M, \widetilde{V}_i\}_i, \{N, \widetilde{W}_j\}_j)^{\star})$.

Intuitively, these requirements corresponds to considering all finite subsets of the context closure of a relation: given formal sums $(\mathbf{Y}, \mathbf{Z})$, related elements are columns of their environments that have the same indexes, and $(\mathbf{Y}', \mathbf{Z}')$ expands $(\mathbf{Y}, \mathbf{Z})$ (up-to permutation of columns of the pair) with columns that are obtained by filling the same context with related columns and with the static environment. The first requirement ensures that the original relation is included in the context closure, and the second requirement says that new elements are obtained by applying the same context.

Let $\mathcal{R}$ be a finite-step simulation saturated by approximants such that $M \mathcal{R} N$. We can assume without loss of generality that this is the only pair of terms in $\mathcal{R}$ and that for any $\mathcal{E}$ such that $\mathcal{R}_{\mathcal{E}} \subseteq \mathcal{R}$ we have $\mathcal{E} = (M, N)$. We can also assume that $\mathcal{R}_{(M,N)}$ contains all pairs in $\{(\emptyset, \mathbf{Y}) \mid \mathbf{Y}$ is an environment formal sum$\}$, and that it contains the pair of Dirac formal sums with empty environment $(1; \emptyset, 1; \emptyset)$, since these pairs trivially satisfy the finite-step simulation clauses.

Let $\mathcal{R}^{\mathtt{cce}}_{(M,N)} \stackrel{\mathtt{def}}{=} \leq_{\mathtt{cce}(M,N)} (\mathcal{R}_{(M,N)})$, which is turn denotes the set

$$\{(\mathbf{Y}, \mathbf{Z}) \mid \exists \mathbf{Y}', \mathbf{Z}' \text{ such that } (\mathbf{Y}', \mathbf{Z}') \leq_{\mathtt{cce}(M,N)} (\mathbf{Y}, \mathbf{Z}) \ \wedge \ \mathbf{Y}' \ \mathcal{R}_{(M,N)} \ \mathbf{Z}'\}$$

We prove that the following is a finite-step simulation up-to lifting and environment:

$$\mathcal{S} \stackrel{\mathtt{def}}{=} \{(C[M], C[N] \mid M \mathcal{R} N\} \cup \{((C[M], C[N]), \mathbf{Y}, \mathbf{Z}) \mid \mathbf{Y} \ \mathcal{R}^{\mathtt{cce}}_{(M,N)} \ \mathbf{Z}\}$$

We require $\mathcal{R}_{(M,N)}$ to contain the pair $(1; \emptyset, 1; \emptyset)$ in order to include triples such as

$$((\lambda x.C[M], \lambda x.C[N]), 1; \lambda x.C[M], 1; \lambda x.C[N])$$

which must be in $\mathcal{S}$ since $\lambda x.C[M] \ \mathcal{S} \ \lambda x.C[N]$.

We assume $\emptyset \ \mathcal{R}_{(M,N)} \ \mathbf{Z}$ for any $\mathbf{Z}$ since, for any C that is not a value context, we have $C[M] \Longmapsto \emptyset$, so we want to derive $\emptyset \ \mathtt{lift}(\geq_{\mathtt{env}} (\mathcal{R}^{\mathtt{cce}}_{(M,N)}))\, [\![C[N]]\!]$.

Finally, relation $\mathcal{R}$ must be saturated by approximants since, as showed in Example 13, a finite-step simulation need not be so, and it might contain, e.g., the triple $((P, Q), \frac{1}{2}; Q + \frac{1}{4}; Q, 1; Q)$ but not the triple $((P, Q), \frac{1}{4}; Q, 1; Q)$. However, if $C = I\,[\cdot]$ then $\frac{1}{2}; C[Q] + \frac{1}{4}; C[Q] \Longmapsto \frac{1}{4}; Q$ and it might be the case that there are no $\mathbf{Y}, \mathbf{Z}$ such that $\mathbf{Y} \ \mathcal{R}_{(P,Q)} \ \mathbf{Z}$ and $(\frac{1}{4}; Q, 1; Q) \leq_{\mathtt{cce}} (\mathbf{Y}, \mathbf{Z})$.

The proof follows the same steps as the congruence proof for the imperative λ-calculus, and we thereby refer the reader to Appendix A, proof of Theorem 52. □

4.2 Contextual equivalence

The definitions of the contextual preorder and equivalence, $\leq_{\mathtt{ctx}}$ and $=_{\mathtt{ctx}}$, are as for call-by-name.

Theorem 40 (Completeness). *If $M \leq_{\mathtt{ctx}} N$ then $M \lesssim N$.*

Proof. We prove that the relation

$$\begin{aligned}\mathcal{R} = \{&((M, N), \textstyle\sum_i p_i; V_1^i, ..., V_n^i \,, \sum_j q_j; W_1^j, ..., W_n^j) \mid \\ &M \leq_{\mathtt{ctx}} N \ \wedge \ \exists C \text{ such that } [\![C[M]]\!] = \textstyle\sum_i p_i; \lambda x.x V_1^i ... V_n^i \\ &\wedge \ [\![C[N]]\!] = \textstyle\sum_j q_j; \lambda x.x W_1^j ... W_n^j\}\end{aligned}$$

is a simulation. Then we derive the result as follows: let $M \leq_{\mathtt{ctx}} N$ and $P = (\lambda y \lambda x.xy)$. Then if $[\![M]\!] = \sum_i p_i; V_i$ and $[\![N]\!] = \sum_j q_j; W_j$ then $[\![PM]\!] = \sum_i p_i; \lambda x.xV_i$ and $[\![PN]\!] = \sum_j q_j; \lambda x.xW_j$, which implies that $[\![M]\!] \mathcal{R}_{(M,N)} [\![N]\!]$. Hence, $M \lesssim N$.

To prove that $\mathcal{R}$ is a simulation, suppose that there are M, N and C such that $M \leq_{\mathtt{ctx}} N$, $[\![C[M]]\!] = \sum_i p_i; \lambda x.xV_1^i...V_n^i$ and $[\![C[N]]\!] = \sum_j q_j; \lambda x.xW_1^j...W_n^j$. Let $\mathbf{Y} = \sum_i p_i; V_1^i, ..., V_n^i$ and $\mathbf{Z} = \sum_j q_j; W_1^j, ..., W_n^j$. We want to prove that for any $r \in \{1, ..., n\}$ and for any context C'',

$$[\![\mathbf{Y}; \mathbf{Y}\lfloor_r \bullet C''[M, \mathbf{Y}]]\!] \mathcal{R}_{(M,N)} [\![\mathbf{Z}; \mathbf{Z}\lfloor_r \bullet C''[N, \mathbf{Z}]]\!]$$

which is equivalent to saying that there is a context D such that

$$\begin{array}{l}[\![D[M]]\!] = \sum_i p_i; \lambda x.(xV_1^i...V_n^i[\![V_r^i \bullet C''[M, V_1^i, ..., V_n^i]]\!]) \\ [\![D[N]]\!] = \sum_j q_j; \lambda x.(xW_1^j...W_n^j[\![W_r^j \bullet C''[N, W_1^j, ..., W_n^j]]\!]).\end{array}$$

Let C' be any context and let

$$P_{M,C'} = \lambda x_1, ..., x_n.(\lambda z, x.xx_1...x_nz)C'[M, x_1, ..., x_n]$$

and $P_{N,C'}$ the same term with M substituted to N. It follows from $M \leq_{\mathtt{ctx}} N$ that $C[M]P_{M,C'} \leq_{\mathtt{ctx}} C[N]P_{N,C'}$. We have:

$$\begin{array}{rl}[\![C[M]P_{M,C'}]\!] & = [\![[\![C[M]]\!]P_{M,C'}]\!] \\ & = [\![\sum_i p_i; (P_{M,C'}V_1^i...V_n^i)]\!] \\ & = [\![\sum_i p_i; (\lambda z, x.xV_1^i...V_n^iz)C'[M, V_1^i, ..., V_n^i]]\!] \\ & = \sum_i p_i; (\lambda z, x.xV_1^i...V_n^iz)[\![C'[M, V_1^i, ..., V_n^i]]\!] \\ & = \sum_i p_i; (\lambda x.xV_1^i...V_n^i[\![C'[M, V_1^i, ..., V_n^i]]\!])\end{array}$$

and analogously for N:

$$[\![C[N]P_{N,C'}]\!] = \sum_j q_j; (\lambda x.xW_1^j...W_n^j[\![C'[N, W_1^j, ..., W_n^j]]\!])$$

Then by the definition of $\mathcal{R}$, for any context C',

$$\sum_i p_i; V_1^i, ..., V_n^i \cdot [\![C'[M, V_1^i, ..., V_n^i]]\!] \mathcal{R}_{(M,N)} \sum_j q_j; W_1^j, ..., W_n^j \cdot [\![C'[N, W_1^j, ..., W_n^j]]\!]$$

This holds in particular for any context of the form $C' = [\cdot]_{r+1}C''$, for $r \in \{1, ..., n\}$ and C'' a value-context, which implies the clause (b) of simulation on formal sums.

Clause (c) is proved using the same result, by taking $C' = [\cdot]_1$.

The first clause of simulation on formal sums follows since $M \leq_{\mathtt{ctx}} N$ implies $C[M] \leq_{\mathtt{ctx}} C[N]$, which in turn implies $\mathtt{weight}([\![C[M]]\!]) \leq \mathtt{weight}([\![C[N]]\!])$. □

Theorem 41 (Full abstraction). *On $\Lambda_\oplus$-terms:*

(1) *relations $\leq_{\mathtt{ctx}}$ and $\lesssim$ coincide;*
(2) *relations $=_{\mathtt{ctx}}$ and $\approx$ coincide.*

Proof. $\lesssim$ is complete by Theorem 40. The soundness follows from the fact that it is a congruence, which is obtained by exploiting the characterization $\lesssim = \lesssim_{\mathrm{fin}}$ and Lemma 39. The result for the equivalences follows from $\approx = \lesssim \cap \lesssim^{-1}$ and $=_{\mathtt{ctx}} = \leq_{\mathtt{ctx}} \cap \leq_{\mathtt{ctx}}^{-1}$. □

5 PROBABILISTIC IMPERATIVE λ-CALCULUS

In this section we add imperative features, namely higher-order references (locations), to the call-by-value calculus, along the lines of the languages in [26, 45]. The syntax of terms and values is:

$$
\begin{array}{rll}
M ::= & x & \text{variables} \\
\mid & c & \text{constants} \\
\mid & \lambda x.M & \text{functions} \\
\mid & M_1 M_2 & \text{applications} \\
\mid & l & \text{locations} \\
\mid & (\boldsymbol{\nu}\, x := M_1) M_2 & \text{new location} \\
\mid & !M & \text{dereferencing} \\
\mid & M_1 := M_2 & \text{assignments} \\
\mid & \mathtt{op}(M_1, ..., M_n) & \text{primitive operations} \\
\mid & \mathtt{if}\ M_1\ \mathtt{then}\ M_2\ \mathtt{else}\ M_3 & \text{if-then-else} \\
\mid & \#_i(M) & \text{projection} \\
\mid & (M_1, ..., M_n) & \text{tuples} \\
\mid & M_1 \oplus M_2 & \text{probabilistic choice}
\end{array}
$$

$$V ::= c \mid \lambda x.M \mid l \mid (V_1, ..., V_n)$$

We use s, t to range over stores, i.e., mappings from locations to closed values, and l, k over locations. Then $s[l \to V]$ is the update of s (possibly an extension of s if l is not in the domain of s). The locations that occur in a term M are $\mathtt{Loc}(M)$. We assume that the set of primitive operations contains the equality function on constants, and write $\mathtt{unit}$ for the unit value (i.e., the nullary tuple).

The language is typed — a simply-typed system with recursive types — to make sure that the values in the summands of a formal sum have the same structure (e.g., they are all abstractions). We allow recursive types to maintain the peculiar possibility of probabilistic languages of having infinite but meaningful computation trees. Whenever possible, we omit any mention of the types. For instance, in any store update $s[l \to V]$ it is intended that V has the type appropriate for l; in this case we say that the type of V is *consistent* with that of l. We require that in a term $(\boldsymbol{\nu}\, x := M_1) M_2$, the variable x cannot occur free in M_1. For similar reasons, we assume that the type syntax is constrained in some way that ensures that for any location type $\mathrm{Ref}\, T$ there exists a closed location-free term M with type T. There are various ways of achieving this — the specific syntax is not relevant for this work.

In examples, $M_1 \,\underline{\mathtt{seq}}\, M_2$ denotes term $(\lambda.M_2)M_1$, i.e., the execution of M_1 and M_2 in sequence. Reduction is defined on terms with a store, i.e., configurations of the form $\langle s\,;\, M\rangle$; hence such configurations appear also in formal sums (where we omit brackets). The small-step reduction and the evaluation contexts are defined in Figure 3, where we assume that the semantics of primitive operations (which take as input n terms of ground type and return a ground type term) is already given by the function $\mathtt{Prim}$. The rules for the semantic mapping, $[\![\ ]\!]$, and the multistep reductions relations, $\Longrightarrow$ and $\Longmapsto$, remain those of Figure 1, with the addition of a store. In all semantic rules, any configuration $\langle s\,;\, M\rangle$ is *well-formed*, in that M is closed and all the locations in M and s are in the domain $\mathsf{dom}(s)$ of s. As in the previous calculi, it is easy to check that the semantics of a term exists and is unique.

$$\textsc{Beta}\ \frac{}{\langle s\,;\,(\lambda x.M)V\rangle \longrightarrow 1; s; M\{^{V}\!/_{x}\}} \qquad \textsc{Sum}\ \frac{}{\langle s\,;\,M_1 \oplus M_2\rangle \longrightarrow \frac{1}{2}; s; M_1 + \frac{1}{2}; s; M_2}$$

$$\textsc{Assign}\ \frac{}{\langle s\,;\,l := V\rangle \longrightarrow 1; s[l \to V]; \mathtt{unit}} \qquad \textsc{Deref}\ \frac{s(l) = V}{\langle s\,;\,!l\rangle \longrightarrow 1; s; V}$$

$$\textsc{New}\ \frac{l \text{ not in the domain of } s}{\langle s\,;\,(\boldsymbol{\nu}\, x{:=}V)M\rangle \longrightarrow 1; s[l \to V]; M\{^{l}\!/_{x}\}}$$

$$\textsc{IfTrue}\ \frac{}{\langle s\,;\ \mathtt{if\ true\ then}\ M_1\ \mathtt{else}\ M_2\rangle \longrightarrow 1; s; M_1}$$

$$\textsc{IfFalse}\ \frac{}{\langle s\,;\ \mathtt{if\ false\ then}\ M_1\ \mathtt{else}\ M_2\rangle \longrightarrow 1; s; M_2}$$

$$\textsc{Proj}\ \frac{}{\langle s\,;\,\#_i(\widetilde{V})\rangle \longrightarrow 1; s; (\widetilde{V})_i} \qquad \textsc{PrimOp}\ \frac{\mathtt{Prim}(\mathtt{op}, \widetilde{c}) = c'}{\langle s\,;\,\mathtt{op}(\widetilde{c})\rangle \longrightarrow 1; s; c'}$$

$$\textsc{Eval}\ \frac{\langle s\,;\,M\rangle \longrightarrow \sum_i p_i; s_i; M_i \quad C \text{ is an evaluation context}}{\langle s\,;\,C[M]\rangle \longrightarrow \sum_i p_i; s_i; C[M_i]}$$

$$\text{Evaluation contexts}\quad C := [\cdot] \mid CM \mid VC \mid \mathtt{if}\ C\ \mathtt{then}\ M_1\ \mathtt{else}\ M_2 \mid \mathtt{op}(\widetilde{c}, C, \widetilde{M}) \mid (\widetilde{V}, C, \widetilde{M}) \mid \#_i C \mid\ !C \mid C := M \mid l := C \mid (\boldsymbol{\nu}\, x{:=}C)M$$

Fig. 3. Single-step reduction relation for imperative probabilistic λ-calculus

Notations and terminology for (environment) formal sums are adapted to the extended syntax in the expected manner. We only recall the multiplication of an environment formal sum $\mathbf{Y} \stackrel{\mathsf{def}}{=} \sum_i p_i; s_i; \widetilde{V}_i$ and formal sums $Y_i \stackrel{\mathsf{def}}{=} \sum_{j\in J_i} p_{i,j}; s_{i,j}; V_{i,j}$ which is defined as:

$$\textstyle\sum_i p_i; \widetilde{V}_i \cdot Y_i \stackrel{\mathsf{def}}{=} \sum_{i,j\in J_i} p_i \cdot p_{i,j}; s_{i,j}; \widetilde{V}_i, V_{i,j}\ .$$

An environment formal sum now has not only a specific length, but it also has an associated sequence of types of that length. These types are the types of the terms in the corresponding columns, i.e., all terms in $\mathbf{Y}\!\upharpoonright_r$ have the r-th type in the type sequence associated to $\mathbf{Y}$. In the definition of dynamic environment we then assume that the pairs of environment formal sums have not only the same length, but also the same associated sequence of types. The context closure of an environment, $(\{M, \widetilde{V}_i\}_i, \{N, \widetilde{W}_j\}_j)^{\widehat{\star}}$, is defined as in the previous section, but now contexts are *location-free*, i.e., no locations occur in the contexts. This constraint, standard in environmental bisimulations for imperative languages, ensures well-formedness of the terms, and is not really a limitation because locations may occur in terms of the environments and may thus end up in the terms of the context closure. Hence, restricting to location-free values in the bisimulation game makes the proof technique simpler while maintaining full abstraction results with respect to contextual equivalence, which allows arbitrary contexts (Section 5.2).

5.1 Environmental bisimulation

The notion of environmental relation is modified to accommodate stores, which are needed to run terms. The elements of an environmental relation are now well-formed pairs of configurations $(\langle s\,;\,M\rangle, \langle t\,;\,N\rangle)$ or well-formed triples

$$(\mathcal{E}, \textstyle\sum_i p_i; s_i; \widetilde{V}_i, \sum_j q_j; t_j; \widetilde{W}_j)\ .$$

Well-formedness on triples ensures that the store s_i of the possible world i defines all locations that appear in (the range of) s_i, in $\widetilde{V}_i$, and in $\mathcal{E}_1$, and similarly for t_j, $\widetilde{W}_j$ and $\mathcal{E}_2$. Further, the triples must be *compatible*: the related environment formal sums should have the same length, and should respect the types, that is, corresponding columns of the environment formal sums should contain terms that have the same type.

Since locations could occur in the terms we want to prove equivalent, we parametrize bisimulations with respect to a set $\{\widetilde{l}\}$ of locations so that the pairs of terms in the relation must have stores with domain $\{\widetilde{l}\}$. This allows us to put these locations in the dynamic environment of the relation (clause (1)), which reflects the fact that the locations occurring in the terms are public (i.e., contexts can access them). In what follows, when we write $\{\widetilde{l}\}$ we assume that no repetitions of the same location occur in the sequence $\widetilde{l}$. For a pair $(\{s_i\}_i, \{t_j\}_j)$ of (tuples of) stores, we say that locations $(\{l_i\}_i, \{k_j\}_j)$ are $(\{s_i\}_i, \{t_j\}_j)$-*fresh* if for every i, j we have $l_i \notin \mathsf{dom}(s_i)$ and $k_j \notin \mathsf{dom}(t_j)$.

Definition 42 (Environmental bisimulation, imperative). A PE relation is a *(PE)* $\{\widetilde{l}\}$-*bisimulation* if

(1) $\langle s\,;\,M\rangle\,\mathcal{R}\,\langle t\,;\,N\rangle$ implies $\mathsf{dom}(s) = \mathsf{dom}(t) = \{\widetilde{l}\}$ and $1; s; \widetilde{l}\;\mathcal{R}_{(M,N)}\;1; t; \widetilde{l}$;

(2) $\sum_i p_i; s_i; \widetilde{V}_i\;\mathcal{R}_{\mathcal{E}}\;\sum_j q_j; t_j; \widetilde{W}_j$ implies:

(a) $\sum_i p_i = \sum_j q_j$;

(b) for all r, if $(\widetilde{V}_i)_r = \lambda x.M_i$ and $(\widetilde{W}_j)_r = \lambda x.N_j$ then
for all $(\{T_i\}_i, \{U_j\}_j) \in (\{\mathcal{E}_1, \widetilde{V}_i\}_i, \{\mathcal{E}_2, \widetilde{W}_j\}_j)^{\widehat{\star}}$,

$$\textstyle\sum_i p_i; \widetilde{V}_i \cdot [\![\langle s_i\,;\,M_i\{^{T_i}\!/_x\}\rangle]\!]\,\mathtt{lift}(\mathcal{R}_{\mathcal{E}})\,\sum_j q_j; \widetilde{W}_j \cdot [\![\langle t_j\,;\,N_j\{^{U_j}\!/_x\}\rangle]\!]\ ;$$

(c) for all r, if $(\widetilde{V}_i)_r = l_i$ and $(\widetilde{W}_j)_r = k_j$ then

- $\sum_i p_i; s_i; \widetilde{V}_i, s_i(l_i)\,\mathtt{lift}(\mathcal{R}_{\mathcal{E}})\,\sum_j q_j; t_j; \widetilde{W}_j, t_j(k_j)$,
- for all $(\{T_i\}_i, \{U_j\}_j) \in (\{\mathcal{E}_1, \widetilde{V}_i\}_i, \{\mathcal{E}_2, \widetilde{W}_j\}_j)^{\widehat{\star}}$,

$$\textstyle\sum_i p_i; s_i[l_i \to T_i]; \widetilde{V}_i\;\mathcal{R}_{\mathcal{E}}\;\sum_j q_j; t_j[k_j \to U_j]; \widetilde{W}_j\ ;$$

(d) for any $(\{s_i\}_i, \{t_j\}_j)$-fresh locations $(\{l_i\}_i, \{k_j\}_j)$,
and for all $(\{T_i\}_i, \{U_j\}_j) \in (\{\mathcal{E}_1, \widetilde{V}_i\}_i, \{\mathcal{E}_2, \widetilde{W}_j\}_j)^{\widehat{\star}}$,

$$\textstyle\sum_i p_i; s_i[l_i \to T_i]; \widetilde{V}_i, l_i\;\mathcal{R}_{\mathcal{E}}\;\sum_j q_j; t_j[k_j \to U_j]; \widetilde{W}_j, k_j\ ;$$

(e) for all r, if $(\widetilde{V}_i)_r = c_i$ and $(\widetilde{W}_j)_r = c_j$ then all constants in the two columns are the same (i.e., there is a c with $c_i = c_j = c$ for all i, j);

(f) for all r, if $(\widetilde{V}_i)_r = (V_{i,1}, ..., V_{i,n})$ and $(\widetilde{W}_j)_r = (W_{j,1}, ..., W_{j,n})$ then

$$\textstyle\sum_i p_i; s_i; \widetilde{V}_i, V_{i,1}, ..., V_{i,n}\,\mathtt{lift}(\mathcal{R}_{\mathcal{E}})\,\sum_j q_j; t_j; \widetilde{W}_j, W_{j,1}, ..., W_{j,n}\,;$$

(g) $\sum_i p_i; \widetilde{V}_i \cdot [\![\langle s_i\,;\,\mathcal{E}_1\rangle]\!]\,\mathtt{lift}(\mathcal{R}_{\mathcal{E}})\,\sum_j q_j; \widetilde{W}_j \cdot [\![\langle t_j\,;\,\mathcal{E}_2\rangle]\!]$.

We let $\approx^{\{\widetilde{l}\}}$ denote the union of all $\{\widetilde{l}\}$-bisimulations.

With respect to the definition for pure call-by-value, the definition above has the additional ingredient of the store, and of clauses (2c) and (2d) to deal with the case in which the values are locations: (2c) gives an observer the possibility of reading and writing the store, and (2d) the possibility of extending the store with fresh locations. Clause (2f) adds all elements of a tuple to the dynamic environment. These aspects are similar to those in ordinary environmental bisimulations for imperative languages [24, 45].

Three further aspects, however, are new. First, by clause (2e), related environment formal sums should be *first-order consistent*, meaning that corresponding columns of constants should contain exactly one constant. This constraint is a consequence of the equality test on constants in the language. To ensure that first-order consistency is maintained in the bisimulation game, most of the clauses use a lifting construction. Thus, when the evaluation of first-order terms may probabilistically yield different constants, lifting allows us to separate the final possible worlds according to the specific constants obtained (since the semantics of a term might be a formal sum assigning non-zero probability to infinitely many different constant, the definition of lifting must allow for a possibly infinite index set in order to ensure first-order consistency). This constraint is further discussed in Examples 46 and 47. A second new aspect is that, since the effect of the evaluation of the terms in the static environment may change depending on the current store, clauses (1) and (2g) allow us to derive a congruence result for arbitrary terms (not necessarily values), as illustrated in the example below. Finally, we parametrize the relation with a set of locations in order to deal with terms where (public) locations may occur.

If the domains of the stores are empty then we can consider bisimulations parametrized by the empty set of locations, and in clause (1) we will have empty sequences of values in the dynamic environment. Anyway, clause (2g) can be applied independently of the presence of values in the dynamic environment.

In what follows, we sometimes omit any reference to the set of locations parametrizing the relation, and simply refer to (bi)simulations and (bi)similarity when the parametrizing set is not relevant.

Example 43. Let $M \stackrel{\text{def}}{=} l := 1$ and $N \stackrel{\text{def}}{=} \texttt{if}\ !l = 0\ \texttt{then}\ l := 1\ \texttt{else}\ \Omega$. Without the static environment, terms $\langle l = 0\,;\, M\rangle$ and $\langle l = 0\,;\, N\rangle$ are bisimilar. However, they are not contextually equivalent: if $C \stackrel{\text{def}}{=} [\cdot]\,\underline{\texttt{seq}}\,[\cdot]$, then $\langle l = 0\,;\, C[M]\rangle$ terminates whereas $\langle l = 0\,;\, C[N]\rangle$ does not. This aspect is determined by the store, probabilities do not really matter. Ordinary environmental bisimulations do not have a static environment, and cannot therefore test repeated runs of given terms that are not values; as a consequence M and N are equated, and bisimulation is not fully substitutive on arbitrary terms (see [45, Section 5.2]).

Clause (1) is also modified with respect to pure call-by-name and call-by-value calculi. Indeed, if we defined the clause as follows:

$$\langle s\,;\, M\rangle\, \mathcal{R}\, \langle t\,;\, N\rangle \text{ implies } [\![\langle s\,;\, M\rangle]\!]\, \texttt{lift}(\mathcal{R}_{(M,N)})\, [\![\langle t\,;\, N\rangle]\!]$$

then the bisimulation would not be sound with respect to terms that diverge at the first run. For instance, the terms $M \stackrel{\text{def}}{=} \texttt{if}\ !l = 1\ \texttt{then true else}\ \Omega$ and $N \stackrel{\text{def}}{=} \texttt{if}\ !l = 1\ \texttt{then false else}\ \Omega$ (that are not contextually equivalent because of contexts such as $l := 1\,\underline{\texttt{seq}}\,[\cdot]$) would be bisimilar with store $l = 0$, by simply considering the relation $\{(\langle l = 0\,;\, M\rangle, \langle l = 0\,;\, N\rangle), ((M, N), \emptyset, \emptyset)\}$.

Even if we make the location l public by starting the bisimulation game from terms (M, l)

and (N, l) and by exploiting clause (2f), so as to allow contexts to use the location (as in [45, Theorem 5.10]), we still have $\langle l = 0\,;\,(M, l)\rangle$ and $\langle l = 0\,;\,(N, l)\rangle$ bisimilar, since $[\![\langle l = 0\,;\,(M, l)\rangle]\!] = [\![\langle l = 0\,;\,(N, l)\rangle]\!] = \emptyset$.
Hence, clause (1) ensures that location l is actually put in the dynamic environment, so that we can use clause (2c) to change the value of l and then evaluate again M and N using clause (2g).

The following examples are meant to further illustrate and motivate the form and the clauses of our bisimulation. The examples only use boolean and integer locations, and we accordingly assume that all locations in the language are of these types. Higher-order locations would not affect the essence of the examples and would complicate the description of the required bisimulations due to the possibility of extending the store (clause (2d)). (The full abstraction results will not rely on the existence of locations of specific types.) Moreover, since the terms compared always have the same locations, we assume that fresh locations for the extensions of the store are the same on both sides. If not specified otherwise, we also assume that thunks take arguments of unit type.

Example 44 shows that in imperative call-by-value, in contrast with pure call-by-value, to achieve full abstraction it is necessary to define bisimulation on formal sums rather than on terms.

Example 44. We have explained in Section 1 why the terms

$$H \stackrel{\text{def}}{=} (\boldsymbol{\nu}\, x := 0)(\lambda.(M \oplus N)) \qquad K \stackrel{\text{def}}{=} (\boldsymbol{\nu}\, x := 0)((\lambda.M) \oplus (\lambda.N))$$

where

$$M \stackrel{\text{def}}{=} \texttt{if } !x = 0 \texttt{ then } x := 1 \,\underline{\texttt{seq}}\, \texttt{true else } \Omega$$
$$N \stackrel{\text{def}}{=} \texttt{if } !x = 0 \texttt{ then } x := 1 \,\underline{\texttt{seq}}\, \texttt{false else } \Omega$$

are contextually equivalent, but would be separated by a bisimulation that acted on terms. With our bisimulation, we can prove H and K equal using a relation that contains the pair $(\langle s\,;\,H\rangle, \langle s\,;\,K\rangle)$, for s the empty store, and all triples $((H, K), \mathbf{Y}, \mathbf{Z})$ in which $\mathbf{Y}, \mathbf{Z}$ are first-order consistent, have the same total weight and, seeing them as matrices, for every column r of the dynamic environments that is not made of constants one of the following properties holds:

(a) there is l such that all terms in $\mathbf{Y}\lfloor_r$ are $\lambda.(M \oplus N)\{^l\!/\!_x\}$, whereas all terms in $\mathbf{Z}\lfloor_r$ are either $\lambda.M\{^l\!/\!_x\}$ or $\lambda.N\{^l\!/\!_x\}$; moreover l does not occur elsewhere in terms of the dynamic environment and its value in the stores is 1;
(b) there is l such that all terms in $\mathbf{Y}\lfloor_r$ are $\lambda.(M \oplus N)\{^l\!/\!_x\}$, whereas $\mathbf{Z}\lfloor_r$ contains both $\lambda.M\{^l\!/\!_x\}$ and $\lambda.N\{^l\!/\!_x\}$; moreover l does not occur elsewhere in the dynamic environment and its value in the stores is 0. The right-hand matrix obtained by erasing all columns that are not of this shape is either $\emptyset$ or (without considering the stores) of the form $(...((Y_1 \cdot Y_2) \cdot Y_3) \cdot ...) \cdot Y_n$, for $Y_i = \frac{1}{2}; \lambda.M\{^{l_i}\!/\!_x\} + \frac{1}{2}; \lambda.N\{^{l_i}\!/\!_x\}$. This clause guarantees that $\lambda.M\{^{l_i}\!/\!_x\}$ and $\lambda.N\{^{l_i}\!/\!_x\}$ have the same probability in every column (if l_i is set to 0 in the stores), and that this property still holds if the matrix is splitted by separating the rows with $\lambda.M\{^{l_i}\!/\!_x\}$ from the rows with $\lambda.N\{^{l_i}\!/\!_x\}$;
(c) there is l such that all terms in $\mathbf{Y}\lfloor_r$ and $\mathbf{Z}\lfloor_r$ are l; moreover l is set to the same value in all the stores.

Finally, we add to the relation the triple $((H, K), 1; s; \emptyset, 1; s; \emptyset)$, where $\emptyset$ denotes the empty dynamic environment, to satisfy clause (1) of Definition 42. Clause (2g) is handled appealing to item (b). The most interesting case is the bisimulation clause (2b) applied to a column r

of functions that satisfy item (b). The result of the evaluation of such functions (with `unit` as argument) is that l is set to 1 and then `true` and `false` are returned, with the same probability. Using the lifting construction we can now split the possible worlds in which `true` has been produced and those in which `false` has been produced, yielding two pairs of environment formal sums both of which are in the bisimulation (note that the lifting splits the original column r so that the corresponding column in the two final pairs satisfies item (a) above).

In this work we sometimes view environment formal sums as matrices (Figure 2). This however is only for representation convenience: our environments are *tuples of rows* (each row representing a possible world originated by the probabilistic evaluation of terms), rather than *tuples of columns*, that is, tuples of formal sums. The next example shows that if the environments were tuples of formal sums, where formal sums are added to the environment following the evaluation of terms during the bisimulation game, then bisimilarity would not be complete. Intuitively this happens because the histories of different possible worlds would not be anymore separated and could interfere.

Example 45. Let

$$A \stackrel{\text{def}}{=} (\boldsymbol{\nu}\, y{:=}0)(L \oplus M) \quad B \stackrel{\text{def}}{=} (\boldsymbol{\nu}\, y{:=}0)(L \oplus N)$$
$$L \stackrel{\text{def}}{=} \lambda.!y \quad M \stackrel{\text{def}}{=} \lambda.(y := 1 \,\underline{\texttt{seq}}\, 2) \quad N \stackrel{\text{def}}{=} \lambda.2\ .$$

Terms A and B create a new location and allow the reading capability on it in the subterm L. The writing capability, in contrast, exists only in the subterm M of A. A behaviour from A that could not be mimicked with B is the run of M, where 1 is assigned to the location x, followed by a run of L, where x is read and 1 is emitted (with B, any value produced by L would be 0). This behaviour, however, is impossible, because L and M are in a probabilistic choice and are therefore obtained in two distinct possible worlds, in one of which x can only be read, in the other x can only be written. Moreover, the writing capability alone is irrelevant, because the location is private; hence it can be omitted from M, resulting in the term N that appears in B. Indeed, A and B are contextually equivalent.

However, the 'wrong' behaviour above for A could be reproduced in the bisimulation if the environments were tuples of formal sums (that is, all possible worlds have the same environment, made of formal sums). The formal sum obtained by the evaluation of A, with summand terms L and M, would be stored in the environment and could then be executed several times, with possible interleaving of evaluations of L and M. (The example could be made more complex so as to obtain a 'wrong' behaviour from the execution of two different formal sums in the environment, rather than by multiple executions of the same formal sum.)

With our bisimulation, we can prove A, B equal using a relation composed by (A, B) (for simplicity, we leave out the store) and by all triples $((A, B), \mathbf{Y}, \mathbf{Z})$ where the environment formal sums $\mathbf{Y} = 1; s; V_1, ..., V_n$ and $\mathbf{Z} = 1; t; W_1, ..., W_n$ are first-order consistent, and for each column r that does not contain constants one of the following holds:

(a) there is l such that $V_r = L\{{}^{l}\!/\!_{y}\} = W_r$; moreover l does not occur elsewhere in the dynamic environment or within a location of the stores, and is set to 0 in both stores;
(b) there is l such that $V_r = M\{{}^{l}\!/\!_{y}\}$ and $W_r = N$; and, again, l does not occur elsewhere in the dynamic environment or within a location of the stores; moreover in the store s we have $s(l) \in \{0, 1\}$ whereas in t we have $t(l) = 0$;
(c) $V_r = W_r = l$ for some l assigned to the same value in both stores.

The proof that this relation is a bisimulation crucially exploits the lifting construction. For instance, using (a) and (b) one shows that the semantics of A and B are in the lifting of the relation, and similarly one proceeds when handling clause (2g) of the bisimulation.

The main purpose of the lifting construct in Definition 42 of environmental bisimulation is to maintain the first-order consistency of related environment formal sums. One may wonder whether something simpler would suffice, namely avoiding the lifting construct altogether and simply requiring that, whenever two first-order terms are evaluated, the probability of obtaining a given constant is the same on both sides (and thus maintaining first-order consistency by avoiding the addition of such values onto the dynamic environments). The example below shows that this would be unsound.

Example 46. We compare the terms $A \stackrel{\text{def}}{=} (\boldsymbol{\nu}\, x{:=}0)(M, N_1)$ and $B \stackrel{\text{def}}{=} (\boldsymbol{\nu}\, x{:=}0)(M, N_2)$ where

$$M \stackrel{\text{def}}{=} \lambda.\ \texttt{if}\ !x = 0\ \texttt{then}\ ((x := 1\ \underline{\texttt{seq}}\ \texttt{true}) \oplus (x := 2\ \underline{\texttt{seq}}\ \texttt{false}))\ \texttt{else}\ \Omega$$
$$N_1 \stackrel{\text{def}}{=} \lambda.\ \texttt{if}\ !x = 2\ \texttt{then}\ x := 3\ \underline{\texttt{seq}}\ n\ \texttt{else}\ \Omega$$
$$N_2 \stackrel{\text{def}}{=} \lambda.\ \texttt{if}\ (!x = 1 \vee !x = 2)\ \texttt{then}\ x := 3\ \underline{\texttt{seq}}\ (n \oplus \Omega)\ \texttt{else}\ \Omega$$

and n is any integer. The terms A and B produce the values $(M, N_1)\{l/x\}$ and $(M, N_2)\{l/x\}$ and l is a location that is accessible only to such values. The definitions of $M\{l/x\}$ and $N_i\{l/x\}$ (for $i = 1, 2$) use conditionals on the content of l in such a way that the only meaningful manipulations with the values $(M\{l/x\}, N_i\{l/x\})$ is to evaluate $M\{l/x\}$ first, and then, possibly, to evaluate $N_i\{l/x\}$. Any other order of evaluation would produce a divergence.

We explain why, intuitively, bisimilarity would equate A and B if, on constants, bisimulation simply checked the probabilities of obtaining each constant (rather than employing the lifting construction). The evaluation of (the body of) $M\{l/x\}$ produces `true` or `false`, with the same probability $\frac{1}{2}$ and with l respectively set to 1 and 2. Then the only meaningful observation is the evaluation of the values $N_i\{l/x\}$. This means evaluating the formal sums

$$F_1 \stackrel{\text{def}}{=} \tfrac{1}{2}; l = 1; N_1\{l/x\}\texttt{unit} + \tfrac{1}{2}; l = 2; N_1\{l/x\}\texttt{unit}$$
$$\text{and}\quad F_2 \stackrel{\text{def}}{=} \tfrac{1}{2}; l = 1; N_2\{l/x\}\texttt{unit} + \tfrac{1}{2}; l = 2; N_2\{l/x\}\texttt{unit}\ .$$

The evaluation of F_1 terminates only when $l = 2$, yielding the value formal sum $Y_1 \stackrel{\text{def}}{=} \frac{1}{2}; l = 3; n$. The evaluation of F_2, in contrast, may terminate under both stores, yielding the value formal sum $Y_2 \stackrel{\text{def}}{=} \frac{1}{4}; l = 3; n + \frac{1}{4}; l = 3; n$. Both in Y_1 and in Y_2 the outcome n has the overall probability $\frac{1}{2}$.

The terms A and B however are not contextually equivalent, because distinguished by a context C that evaluates $M\{l/x\}$ and then proceeds with the evaluation $N_i\{l/x\}$ *only* when the outcome from $M\{l/x\}$ was `true`. Now, $C[A]$ never terminates, whereas $C[B]$ terminates and produces n with probability $\frac{1}{4}$.

Our environmental bisimulation distinguishes A from B because we separately analyze the possible worlds in which the evaluation of $M\{l/x\}$ has produced `true` and the possible worlds in which the evaluation has produced `false`, somehow mimicking the effect of the context C above.

Yet another possibility for avoiding the lifting construct of the Definition 42 of bisimulation might have been to drop the requirement of first-order consistency, allowing environment formal sums in which a first-order column may contain different constants. Thus constants would be added to the dynamic environment as any other type of value, and one would

simply check that, at any time, the weights for the occurrences of a given constant in related columns are the same; formally, replacing clause (2e) with:

(2e′) for every column r and every constant c,
$\sum_{\{i \mid (\widetilde{V}_i)_r = c\}} p_i = \sum_{\{j \mid (\widetilde{W}_j)_r = c\}} q_j$.

The example below shows that this choice would be unsound too. We write $(\boldsymbol{\nu}\, x,\, y{:=}0)M$ for the creation of two locations in which the initialization of the first one is irrelevant.

Example 47. This is a variation of the previous example.
We compare the terms $M_1 \stackrel{\text{def}}{=} (\boldsymbol{\nu}\, x,\, y{:=}0)(A, B_1)$ and $M_2 \stackrel{\text{def}}{=} (\boldsymbol{\nu}\, x,\, y{:=}0)(A, B_2)$ where

$$A \stackrel{\text{def}}{=} \lambda.\ \texttt{if}\ !y = 0\ \texttt{then}\ y := 1\ \underline{\texttt{seq}}\ (\lambda z.(x := z\ \underline{\texttt{seq}}\ z))(\texttt{true} \oplus \texttt{false})\ \texttt{else}\ \Omega$$
$$B_1 \stackrel{\text{def}}{=} \lambda.\ \texttt{if}\ !y = 1\ \texttt{then}\ y := 2\ \underline{\texttt{seq}}\ !x\ \texttt{else}\ \Omega$$
$$B_2 \stackrel{\text{def}}{=} \lambda.\ \texttt{if}\ !y = 1\ \texttt{then}\ y := 2\ \underline{\texttt{seq}}\ (\texttt{true} \oplus \texttt{false})\ \texttt{else}\ \Omega$$

As in the previous example, M_1 and M_2 yield the values $(A, B_1)\{l, l'/x, y\}$ and $(A, B_2)\{l, l'/x, y\}$, and the interactions of the terms with the store is such that the only meaningful experiment is to evaluate $A\{l, l'/x, y\}$ first, and then $B_i\{l, l'/x, y\}$ (indeed, the location l' is only used to this end).

We explain why, intuitively, the variant (2e′) above of the clause for first-order values would incorrectly equate M_1 and M_2. The evaluation of $A\{l, l'/x, y\}$ adds to the dynamic environment the formal sum

$$\tfrac{1}{2}; l = \texttt{true}, l' = 1; \texttt{true} + \tfrac{1}{2}; l = \texttt{false}, l' = 1; \texttt{false}$$

(the values produced are also placed in the location l).

Now, in one case the evaluation of $B_1\{l, l'/x, y\}$ adds to the dynamic environment a column of boolean values identical to the column produced above (because $B_1\{l, l'/x, y\}$ emits the value stored in l, which is identical to the value produced by the evaluation of $A\{l, l'/x, y\}$). This means that we end up with an environment formal sum in which the relevant columns are

$$\begin{array}{llll} \frac{1}{2}\ ; & \texttt{true}\ , & \texttt{true} \\ \frac{1}{2}\ ; & \texttt{false}\ , & \texttt{false} \end{array} \tag{5}$$

In contrast, when evaluating $B_2\{l, l'/x, y\}$ each possible world is split into two, in each of which `true` and `false` have probability $\frac{1}{2}$. Thus the relevant columns of the final environment formal sum are

$$\begin{array}{llll} \frac{1}{4}\ ; & \texttt{true}\ , & \texttt{true} \\ \frac{1}{4}\ ; & \texttt{true}\ , & \texttt{false} \\ \frac{1}{4}\ ; & \texttt{false}\ , & \texttt{true} \\ \frac{1}{4}\ ; & \texttt{false}\ , & \texttt{false} \end{array}$$

In each of these columns, the probabilities for `true` and `false` are the same as in the columns of (5), as required by (2e′).

However the terms are not contextual equivalent. They are separated by a context that evaluates $B_i\{l, l'/x, y\}$ only if the outcome of the evaluation of $A\{l, l'/x, y\}$ is `true`. Thus the overall probability of obtaining `true` at the end is $\frac{1}{2}$ in one case, and $\frac{1}{4}$ in the other. Similarly the terms are distinguished in our bisimulation, reasoning along the lines of Example 46.

The definition of $\{\widetilde{l}\}$-simulation is the same as the definition of $\{\widetilde{l}\}$-bisimulation, but for the first clause on environment formal sums with stores, which becomes: $\sum_i p_i \leq \sum_j q_j$. We let $\lesssim^{\{\widetilde{l}\}}$ denote $\{\widetilde{l}\}$-similarity.

The basic properties and definitions for environmental (bi)simulations in pure call-by-value remain valid, with the due adjustments. In some cases, however, some subtleties arise.

It can be easily proved that, for any set $\{\widetilde{l}\}$, $\{\widetilde{l}\}$-bisimilarity and $\{\widetilde{l}\}$-similarity are an equivalence and a preorder relation respectively. For proving transitivity, in particular, the restriction to parametrized relations, rather than to arbitrary relations, is fundamental. Analogously, we have that $\{\widetilde{l}\}$-(bi)simulations are closed under union, and thus relations $\approx^{\{\widetilde{l}\}}$ and $\lesssim^{\{\widetilde{l}\}}$ are respectively the largest $\{\widetilde{l}\}$-bisimulation and $\{\widetilde{l}\}$-simulation.

In finite-step simulation, clauses (2b) and (2g) are modified so to make sure that only a finite number of reductions are performed on the challenger side. No modification is made to the clauses (1), (2c), (2d), (2e) and (2f) for locations, constants, and tuples, because there is no evaluation of terms involved.

The definition of extended environment formal sum and of the multi-step reduction from extended environment formal sums to environment formal sums is adapted to the imperative case as expected, by assuming that when $\sum_i p_i; s_i; \widetilde{V}_i; M_i \Longmapsto Y$ there is only a finite number of $\langle s_i\,;\, M_i\rangle$ that actually perform some reduction steps.

Definition 48. A PE relation is a *finite-step* $\{\widetilde{l}\}$*-simulation* if it satisfies the same clauses (1), (2a), (2c), (2d), (2e) and (2f) of (the simulation version of) Definition 42; and, in place of clauses (2b) and (2g) we have:

(2) $\sum_i p_i; s_i; \widetilde{V}_i \;\mathcal{R}_{\mathcal{E}}\; \sum_j q_j; t_j; \widetilde{W}_j$ implies:
  (b) for all r, if $(\widetilde{V}_i)_r = \lambda x.M_i$ and $(\widetilde{W}_j)_r = \lambda x.N_j$ then
    for all $(\{T_i\}_i, \{U_j\}_j) \in (\{\mathcal{E}_1, \widetilde{V}_i\}_i, \{\mathcal{E}_2, \widetilde{W}_j\}_j)^{\star}$,
    if $\sum_i p_i; s_i; \widetilde{V}_i; M_i\{T_i/x\} \Longmapsto Y$ then $Y\, \mathtt{lift}(\mathcal{R}_{\mathcal{E}})\, \sum_j q_j; \widetilde{W}_j \cdot [\![\langle t_j\,;\, N_j\{U_j/x\}\rangle]\!]$;
  (g) if $\sum_i p_i; s_i; \widetilde{V}_i; \mathcal{E}_1 \Longmapsto Y$ then $Y\, \mathtt{lift}(\mathcal{R}_{\mathcal{E}})\, \sum_j q_j; \widetilde{W}_j \cdot [\![\langle t_j\,;\, \mathcal{E}_2\rangle]\!]$.

We write $\lesssim^{\{\widetilde{l}\}}_{\mathrm{fin}}$ for finite-step $\{\widetilde{l}\}$-similarity. We prove that $\{\widetilde{l}\}$-similarity and finite-step $\{\widetilde{l}\}$-similarity coincide by exploiting the saturation by approximants and the saturation by suprema of $\{\widetilde{l}\}$-simulations and finite-step $\{\widetilde{l}\}$-simulations, respectively. Since only clauses (2b) and (2g) are modified, we can proceed as in the proofs for the pure calculi.

THEOREM 49. $\lesssim^{\{\widetilde{l}\}} = \lesssim^{\{\widetilde{l}\}}_{\mathrm{fin}}$.

Precongruence is derived for the finite-step similarity using 'up-to lifting and environment' techniques, and then transported to similarity, from which it is transported to bisimilarity using the characterization of bisimilarity as the equivalence induced by the simulation preorder.

The up-to lifting and environment technique is defined analogously to the probabilistic call-by-value case. The environment preorder is as follows: $(\mathbf{Y}, \mathbf{Z}) \leq_{\mathtt{env}} (\mathbf{Y}', \mathbf{Z}')$ if $\mathbf{Y} = \sum_i p_i; s_i; \widetilde{V}_i, \mathbf{Z} = \sum_j q_j; t_j; \widetilde{W}_j$ with $|\ \mathbf{Y}\ | = |\ \mathbf{Z}\ |$ and $\mathbf{Y}' = \sum_i p_i; s_i; \widetilde{V}'_i, \mathbf{Z}' = \sum_j q_j; t_j; \widetilde{W}'_j$ with $|\ \mathbf{Y}'\ | = |\ \mathbf{Z}'\ |$ and for every index r in $|\ \mathbf{Y}\ |$ there is an index r' in $|\ \mathbf{Y}'\ |$ such that for all i, j, $(\widetilde{V}_i)_r = (\widetilde{V}'_i)_{r'}$ and $(\widetilde{W}_j)_r = (\widetilde{W}'_j)_{r'}$.

Definition 50. A PE relation is a *finite-step* $\{\widetilde{l}\}$*-simulation up-to lifting and environment* if:

(1) $\langle s\,;\, M\rangle\, \mathcal{R}\, \langle t\,;\, N\rangle$ implies $\mathsf{dom}(s) = \mathsf{dom}(t) = \{\widetilde{l}\}$ and $1; s; \widetilde{l}\; \mathcal{R}_{(M,N)}\; 1; t; \widetilde{l}$;
(2) $\sum_i p_i; s_i; \widetilde{V}_i \;\mathcal{R}_{\mathcal{E}}\; \sum_j q_j; t_j; \widetilde{W}_j$ implies:
  (a) $\sum_i p_i \leq \sum_i q_i$;

(b) for all r, if $(\widetilde{V}_i)_r = \lambda x.M_i$ and $(\widetilde{W}_j)_r = \lambda x.N_j$ then
for all $(\{T_i\}_i, \{U_j\}_j) \in (\{\mathcal{E}_1, \widetilde{V}_i\}_i, \{\mathcal{E}_2, \widetilde{W}_j\}_j)^{\widehat{\star}}$,
if $\sum_i p_i; s_i; \widetilde{V}_i; M_i\{^{T_i}/_x\} \Longmapsto Y$ then $Y\, \mathtt{lift}(\geq_{\mathtt{env}} (\mathcal{R}_{\mathcal{E}}))\, \sum_j q_j; \widetilde{W}_j \cdot [\![\langle t_j \,;\, N_j\{^{U_j}/_x\}\rangle]\!]$;

(c) for all r, if $(\widetilde{V}_i)_r = l_i$ and $(\widetilde{W}_j)_r = k_j$ then for all $(\{T_i\}_i, \{U_j\}_j) \in (\{\mathcal{E}_1, \widetilde{V}_i\}_i, \{\mathcal{E}_2, \widetilde{W}_j\}_j)^{\widehat{\star}}$,
- $\sum_i p_i; s_i; \widetilde{V}_i, s_i(l_i)\, \mathtt{lift}(\geq_{\mathtt{env}} (\mathcal{R}_{\mathcal{E}}))\, \sum_j q_j; t_j; \widetilde{W}_j, t_j(k_j)$,
- $\sum_i p_i; s_i[l_i \to T_i]; \widetilde{V}_i\, \mathtt{lift}(\geq_{\mathtt{env}} (\mathcal{R}_{\mathcal{E}}))\, \sum_j q_j; t_j[k_j \to U_j]; \widetilde{W}_j$;

(d) for any $(\{s_i\}_i, \{t_j\}_j)$-fresh locations $(\{l_i\}_i, \{k_j\}_j)$,
and for all $(\{T_i\}_i, \{U_j\}_j) \in (\{\mathcal{E}_1, \widetilde{V}_i\}_i, \{\mathcal{E}_2, \widetilde{W}_j\}_j)^{\widehat{\star}}$,

$$\sum_i p_i; s_i[l_i \to T_i]; \widetilde{V}_i, l_i\, \mathtt{lift}(\geq_{\mathtt{env}} (\mathcal{R}_{\mathcal{E}}))\, \sum_j q_j; t_j[k_j \to U_j]; \widetilde{W}_j, k_j\,;$$

(e) for all r, if $(\widetilde{V}_i)_r = c_i$ and $(\widetilde{W}_j)_r = c_j$ then all constants in the two columns are the same (i.e., there is c_a with $c_i = c_j = c_a$ for all i, j);

(f) for all r, if $(\widetilde{V}_i)_r = (V_{i,1}, ..., V_{i,n})$ and $(\widetilde{W}_j)_r = (W_{j,1}, ..., W_{j,n})$ then

$$\sum_i p_i; s_i; \widetilde{V}_i, V_{i,1}, ..., V_{i,n}\, \mathtt{lift}(\geq_{\mathtt{env}} (\mathcal{R}_{\mathcal{E}}))\, \sum_j q_j; t_j; \widetilde{W}_j, W_{j,1}, ..., W_{j,n}\,;$$

(g) $\sum_i p_i; s_i; \widetilde{V}_i; \mathcal{E}_1 \Longmapsto Y$ then $Y\, \mathtt{lift}(\geq_{\mathtt{env}} (\mathcal{R}_{\mathcal{E}}))\, \sum_j q_j; \widetilde{W}_j \cdot [\![\langle t_j \,;\, \mathcal{E}_2\rangle]\!]$.

THEOREM 51. *If $\mathcal{R}$ is a finite-step $\{\widetilde{l}\}$-simulation up-to lifting and environment then $\mathcal{R} \subseteq \lesssim_{\mathrm{fin}}^{\{\widetilde{l}\}}$.*

The soundness of the up-to lifting and environment technique follows as in call-by-value (Lemma 38). Given a finite-step $\{\widetilde{l}\}$-simulation up-to lifting and environment $\mathcal{R}$, we prove that $\mathcal{S} = \mathtt{Pairs}(\mathcal{R}) \cup \bigcup_{\mathcal{E}} \mathtt{lift}(\geq_{\mathtt{env}} (\mathcal{R}_{\mathcal{E}}))$ is a finite-step $\{\widetilde{l}\}$-simulation.

We first prove congruence of finite-step $\{\widetilde{l}\}$-similarity for contexts with locations in $\{\widetilde{l}\}$, and then we show how to derive congruence for general contexts. The proofs of these results are reported in Appendix A. The proof structure for Theorem 52 is as in call-by-value; we define the context closure of a finite-step $\{\widetilde{l}\}$-simulation and we prove that it is a finite-step $\{\widetilde{l}\}$-simulation up-to lifting and environment.

THEOREM 52. *Finite-step $\{\widetilde{l}\}$-similarity is a precongruence for contexts with locations in $\{\widetilde{l}\}$: if $\langle s \,;\, M\rangle \lesssim_{\mathrm{fin}}^{\{\widetilde{l}\}} \langle t \,;\, N\rangle$ then for every context C with $\mathtt{Loc}(C) \subseteq \{\widetilde{l}\}$ we have $\langle s \,;\, C[M]\rangle \lesssim_{\mathrm{fin}}^{\{\widetilde{l}\}} \langle t \,;\, C[N]\rangle$.*

Then we derive precongruence for general contexts by showing how to move from relations parametrized by a set $\{\widetilde{l}\}$ to relations parametrized by $\{\widetilde{l}'\}$, for $\{\widetilde{l}'\}$ a set including $\{\widetilde{l}\}$.

THEOREM 53. *Let $\widetilde{l}' = \widetilde{l}, \widetilde{l}''$ and let $\widetilde{V}' = \widetilde{V}, \widetilde{V}''$ be a sequence of values whose types are consistent with those of $\widetilde{l}'$, and with locations in $\{\widetilde{l}'\}$. Let C be a context with locations in $\{\widetilde{l}'\}$. If $\langle s \,;\, M\rangle \lesssim_{\mathrm{fin}}^{\{\widetilde{l}\}} \langle t \,;\, N\rangle$ then:*

- $\langle s[\widetilde{l}'' \to \widetilde{V}''] \,;\, C[M]\rangle \lesssim_{\mathrm{fin}}^{\{\widetilde{l}'\}} \langle t[\widetilde{l}'' \to \widetilde{V}''] \,;\, C[N]\rangle$;
- $\langle \widetilde{l}' = \widetilde{V}' \,;\, C[M]\rangle \lesssim_{\mathrm{fin}}^{\{\widetilde{l}'\}} \langle \widetilde{l}' = \widetilde{V}' \,;\, C[N]\rangle$.

5.2 Contextual equivalence

We set $\langle s \,;\, M\rangle \Downarrow = \mathtt{weight}([\![\langle s \,;\, M\rangle]\!])$. Contextual equivalence and the contextual preorder are defined by quantifying over all stores and contexts.

Definition 54. M and N are in the *contextual preorder*, written $M \leq_{\mathtt{ctx}} N$, (resp. *contextually equivalent*, written $M =_{\mathtt{ctx}} N$), if, for any store s and context C such that $\langle s\,;\, C[M]\rangle$ and $\langle s\,;\, C[N]\rangle$ are well-formed, $\langle s\,;\, C[M]\rangle\Downarrow \leq \langle s\,;\, C[N]\rangle\Downarrow$ (resp. $\langle s\,;\, C[M]\rangle\Downarrow = \langle s\,;\, C[N]\rangle\Downarrow$).

Theorem 55 (Completeness). *Let* $\mathtt{Loc}(M) \cup \mathtt{Loc}(N) = \{\widetilde{l}\}$. *If* $M \leq_{\mathtt{ctx}} N$ *then* $\langle \widetilde{l} = \widetilde{V}\,;\, M\rangle \lesssim^{\{\widetilde{l}\}} \langle \widetilde{l} = \widetilde{V}\,;\, N\rangle$, *for any* $\widetilde{V}$ *whose types and locations are consistent with* $\widetilde{l}$.

Proof. We prove that the relation

$$\begin{aligned}\mathcal{R} \stackrel{\mathsf{def}}{=} \{&(\langle \widetilde{l} = \widetilde{V}\,;\, M\rangle, \langle \widetilde{l} = \widetilde{V}\,;\, N\rangle) \mid M \leq_{\mathtt{ctx}} N \wedge \{\widetilde{l}\} = \mathtt{Loc}(M) \cup \mathtt{Loc}(N)\} \cup \\ &\{((M, N), \textstyle\sum_i p_i; s_i; V_1^i, ..., V_n^i\,,\, \sum_j q_j; t_j; W_1^j, ..., W_n^j) \mid M \leq_{\mathtt{ctx}} N \\ &\wedge\, \exists C, \widetilde{V} \text{ such that } (\llbracket \langle \widetilde{l} = \widetilde{V}\,;\, C[M]\rangle\rrbracket = \textstyle\sum_i p_i; s_i; \lambda x.xV_1^i...V_n^i \\ &\wedge\, \llbracket \langle \widetilde{l} = \widetilde{V}\,;\, C[N]\rangle\rrbracket = \textstyle\sum_j q_j; t_j; \lambda x.xW_1^j...W_n^j \\ &\text{with } \mathtt{Loc}(C) \subseteq \{\widetilde{l}\} = \mathtt{Loc}(M) \cup \mathtt{Loc}(N) \\ &\wedge \text{ they are } \textit{first-order consistent})\}\end{aligned}$$

is a $\{\widetilde{l}\}$-simulation. The full proof is in Appendix A. □

We can now derive full abstraction for the simulation preorder and bisimilarity. Contextual equivalence and preorder are defined on terms, while bisimilarity and the simulation preorder are defined over configurations of a term and a store. In the full abstraction result we show that congruence on terms corresponds to bisimilarity when an arbitrary store is considered.

Theorem 56 (Full abstraction). *Let* $\mathtt{Loc}(M) \cup \mathtt{Loc}(N) = \{\widetilde{l}\}$. *We have, for any* $\widetilde{V}$ *whose types and locations are consistent with* $\widetilde{l}$*:*

- $M \leq_{\mathtt{ctx}} N$ *if and only if* $\langle \widetilde{l} = \widetilde{V}\,;\, M\rangle \lesssim^{\{\widetilde{l}\}} \langle \widetilde{l} = \widetilde{V}\,;\, N\rangle$ *;*
- $M =_{\mathtt{ctx}} N$ *if and only if* $\langle \widetilde{l} = \widetilde{V}\,;\, M\rangle \approx^{\{\widetilde{l}\}} \langle \widetilde{l} = \widetilde{V}\,;\, N\rangle$ *.*

Proof. Theorem 55 proves completeness. For soundness, suppose $\langle \widetilde{l} = \widetilde{V}\,;\, M\rangle \lesssim^{\{\widetilde{l}\}} \langle \widetilde{l} = \widetilde{V}\,;\, N\rangle$ for some $\widetilde{V}$ consistent with $\widetilde{l}$. Let s be a store and C a context such that $\langle s\,;\, C[M]\rangle$ and $\langle s\,;\, C[N]\rangle$ are well-formed. We want to prove that $\langle s\,;\, C[M]\rangle \Downarrow \leq \langle s\,;\, C[N]\rangle \Downarrow$.
By well-formedness, we know that $s = \emptyset[\widetilde{l'} \to \widetilde{V'}]$, for some $\widetilde{l'}$ such that $\widetilde{l'} = \widetilde{l}, \widetilde{l''}$ and for some consistent $\widetilde{V'}$, and that C has locations in $\{\widetilde{l'}\}$. By $\langle \widetilde{l} = \widetilde{V}\,;\, M\rangle \lesssim^{\{\widetilde{l}\}} \langle \widetilde{l} = \widetilde{V}\,;\, N\rangle$ we have $\langle \widetilde{l} = \widetilde{V}\,;\, M\rangle \lesssim^{\{\widetilde{l}\}}_{\mathrm{fin}} \langle \widetilde{l} = \widetilde{V}\,;\, N\rangle$ and by Theorem 53 we derive $\langle \widetilde{l'} = \widetilde{V'}\,;\, C[M]\rangle \lesssim^{\{\widetilde{l'}\}}_{\mathrm{fin}} \langle \widetilde{l'} = \widetilde{V'}\,;\, C[N]\rangle$. Then $\langle \widetilde{l'} = \widetilde{V'}\,;\, C[M]\rangle \Downarrow \leq \langle \widetilde{l'} = \widetilde{V'}\,;\, C[N]\rangle \Downarrow$.

□

The universal quantification over stores in the full abstraction is outside, and not inside, the double implication, i.e., we do not prove

$$M \leq_{\mathtt{ctx}} N \text{ if and only if for all consistent } \widetilde{V},\ \langle \widetilde{l} = \widetilde{V}\,;\, M\rangle \lesssim^{\{\widetilde{l}\}} \langle \widetilde{l} = \widetilde{V}\,;\, N\rangle$$

but rather

$$\text{for all consistent } \widetilde{V},\ M \leq_{\mathtt{ctx}} N \text{ if and only if } \langle \widetilde{l} = \widetilde{V}\,;\, M\rangle \lesssim^{\{\widetilde{l}\}} \langle \widetilde{l} = \widetilde{V}\,;\, N\rangle.$$

The latter statement implies the former one (assuming that a consistent initialisation of the store is possible). The reason why we can prove the latter one, and thereby consider an arbitrary store when proving contextual equivalence, is that the definition of simulation and bisimulation already includes the universal quantification over different assignments of the locations in $\widetilde{l}$, since the locations are in the dynamic environment and we can apply the second item of clause (2c).

6 ADDITIONAL RELATED WORKS

We discuss here some additional works on probabilistic calculi based on different notions of bisimulations (with respect to applicative or environmental) or different techniques for proving contextual equivalence.

The probabilistic λ-calculus, i.e., a λ-calculus endowed with binary, fair, probabilistic choice, was first presented in [12]. In [11] and [8] probabilistic applicative bisimulations for pure call-by-name and call-by-value λ-calculi are shown to be congruences, using Howe's method coupled with a disentangling technique. Completeness however only holds in call-by-value, while it fails in call-by-name. The reason for this is that distributivity of lambda-abstraction over probabilistic choice holds for contextual equivalence in call-by-name but does not hold in call-by-value, and in order to prove such distributivity laws the bisimulation has to be defined over probability distributions or formal sums, rather than being defined over terms (see Examples 5 and 33). In call-by-name, completeness is obtained using *coupled logical bisimulation* [11], a probabilistic version of the logical bisimilarity for deterministic languages [44]. While applicative bisimulation requires two functions to be related whenever they take as input the same argument, logical bisimulation requires two functions to be related whenever they take as input terms in the contextual closure of the relation itself. Since the contextual closure of a relation includes identity, the set of terms with which related functions are tested is enlarged with respect to applicative bisimilarity. This makes the congruence proof easier by allowing a direct use of the inductive hypothesis, thereby removing the need for Howe's technique. Drawbacks of all forms of logical bisimilarity are a non-monotone functional (which makes it harder to prove that bisimilarity is the largest bisimulation) and a confinement to pure λ-calculi. Further, up-to techniques may be difficult in logical bisimilarity. For instance, Example 26 cannot be proved with the techniques in [11]: the equality fails for applicative bisimilarity, and the up-to context technique provided for logical bisimilarity is not powerful enough (the paper shows a similar example, akin to Example 5, where however the functions employed immediately throw away their input, and this is essential for the proof).

A further difference between applicative bisimulations and environmental bisimulations shows up when one considers the corresponding *simulations*. Even in cases where applicative bisimilarity is fully abstract for contextual equivalence, the corresponding simulation may not be fully abstract for the contextual preorder. Pure call-by-value is such an example [8, 10]. In contrast, in all calculi in the paper, full abstraction for environmental bisimilarity carries over to the corresponding simulation, with a similar proof.

An alternative bisimulation for enriched calculi is *normal form* (or *open*) bisimulation [22, 28, 42, 47]. This is complete (with respect to contextual equivalence) only in certain extensions of the λ-calculus (e.g., call-by-value with *both* state *and* callcc), and would be incomplete in other languages (such as λ-calculus without state or/and callcc).

Another approach to contextual equivalence in higher-order languages is via logical relations (see, e.g., [31, Chapter 8] and [37]). This technique has been applied to probabilistic typed higher-order languages by Bizjak and Birkedal [6]. Their probabilistic logical relation uses biorthogonality, and is defined on terms, rather than on distributions. These features introduce some universal quantification (e.g., on evaluation contexts) which makes it difficult to prove examples such as 44, as discussed in [5, Section 1.5]. Proof techniques combining features of bisimulations and logical relations in the non-probabilistic case are studied in [20, 21, 33].

In denotational semantics, fully abstract models for probabilistic PCF have been studied in [18] using domain theory and adding statistical termination testers, and in [16] using probabilistic coherence spaces. [13] provides a fully abstract game semantics for probabilistic Algol, using a quotienting step.

Finally, in this work we have only considered exact behavioral equivalences, as opposed to approximate behavioral equivalences (allowing the programs to differ up to a certain probability value ϵ) or metrics (measuring the distance between the behaviors of probabilistic programs) [14, 15]. Bisimulation metrics for an affine probabilistic pure λ-calculus have been recently proposed in [9]. Applicative bisimulation metric is proved to be sound with respect to the contextual distance, and a metric for an extensions of the language with tuples is defined. In order to be sound with respect to the contextual metric, the tuple distance is endowed with a notion of environment.

7 CONCLUSIONS AND FUTURE WORK

We have investigated environmental bisimulations in sequential higher-order languages, considering pure λ-calculi, both call-by-name and call-by-value, and an extension with higher-order references.

While we have tried to respect the general schema of environmental bisimulations, our definitions and results present noticeable technical differences. Some differences, such as the appeal to formal sums, are specific to probabilities. Other differences, however, may be seen as insights into environmental bisimulations that were suggested by the study of probabilities. An example is the distinction between a static and a dynamic environment, which reflects the copying facilities of the language on the terms of the environment. This distinction yields sharper congruence results, which show up well in the imperative λ-calculus: with ordinary environmental bisimulations, bisimilarity is fully substitutive only for values, since for general terms substitutivity holds only for evaluation contexts (see Example 43). The example in Section 3.4 shows that static environments can also be useful in context closures of 'up-to context' techniques.

To understand environmental bisimulations for call-by-value calculi, we have found important the study of the imperative extension. Only in the richer language do various aspects of our definitions find a justification: the use of formal sums (Example 44); dynamic environments as formal sums of tuples of values, as opposite to, e.g., tuples of formal sums (Example 45); the lifting construct to handle first-order values (Example 46). The pure call-by-value calculus has allowed us to present the concepts in a simpler setting, as a stepping stone towards the imperative extension, but seems a rather peculiar language, one in which a number of variations of the definitions collapse.

The dynamic environments are used only for the call-by-value calculi. In general, the form of the bisimulation clauses depends on the features of the calculus. It would be interesting to investigate an abstract formulation of bisimulation, of which the concrete definitions presented in this paper would be instances. Possible bases for such a framework could be coalgebras [40] or bigraphs [30].

Related to the problems with congruence of applicative bisimulations are also the difficulties with "up-to context" techniques (the usefulness of these techniques in higher-order languages and its problems with applicative bisimulations have been studied by Lassen [27]; see also [26, 39, 41]). Enhancements of the bisimulation proof method, as up-to techniques, are particularly useful for environmental bisimulations [45] because of the quantification over contexts that appear in the definition and that can make bisimulation proofs tedious. We have explored some basic enhancements, mainly in call-by-name, including new forms of up-to

techniques specific to probabilities such as 'up-to lifting'. The study of powerful enhancements goes beyond the scopes of this work; we regard it as an important and challenging line for future work. The study could be pursued in a number of directions, for instance investigating other forms of up-to and strengthening the 'up-to context' enhancements.

Another interesting direction for future work is the addition of concurrency. A major consequence of this could be the move to semantics that combine probabilities with non-determinism.

ACKNOWLEDGMENTS

This work has been supported by the ERC H2020 project 'CoVeCe' (grant agreement No 678157), by the LABEX MILYON (ANR-10-LABX-0070) of Université de Lyon, within the program 'Investissements d'Avenir' (ANR-11-IDEX-0007), by the project ANR-16-CE25-0011 'REPAS', by the project ANR 12IS02001'PACE', and by H2020-MSCA-RISE project 'Behapi' (ID 778233). We have benefited from discussions with Raphaëlle Crubillé and Ugo Dal Lago. We are deeply grateful to the reviewers for their detailed reading of the paper and their many useful comments and remarks.

A PROOFS

Proof of Lemma 11

(1) The proof of (1) follows from the definition of $[\![M]\!]$ as the supremum of the set $\{Y \mid M \Longmapsto Y\}$ with respect to the $\leq_{\mathtt{apx}}$ preorder. Let $\mathcal{R}$ be a simulation and let $\mathcal{S}$ be its saturation by approximants.
If $M\ \mathcal{S}\ N$ then $[\![M]\!]\ \mathcal{S}_{(M,N)}\ [\![N]\!]$, since $\leq_{\mathtt{apx}}$ is reflexive and $[\![M]\!]\ \mathcal{R}_{(M,N)}\ [\![N]\!]$. If $Y\ \mathcal{S}_{(M,N)}\ Z$ then there is a Y' such that $Y \leq_{\mathtt{apx}} Y'$ and $Y'\ \mathcal{R}_{(M,N)}\ Z$. We have $\mathtt{weight}(Y) \leq \mathtt{weight}(Y')$, by the definition of approximant, and $\mathtt{weight}(Y') \leq \mathtt{weight}([\![Z]\!])$, since $\mathcal{R}$ is a simulation.
Suppose $Y + Y'' = Y'$. Then, for any context C,

$$[\![Y \bullet C[M]]\!] \leq_{\mathtt{apx}} [\![Y \bullet C[M]]\!] + [\![Y'' \bullet C[M]]\!] = [\![Y' \bullet C[M]]\!]\ \mathcal{R}_{(M,N)}\ [\![Z \bullet C[N]]\!]$$

which implies $[\![Y \bullet C[M]]\!]\ \mathcal{S}_{(M,N)}\ [\![Z \bullet C[N]]\!]$.

(2) Let $\mathcal{S} = \bigcup_n \mathcal{R}^n$ be the saturation by suprema of a finite-step simulation $\mathcal{R}$. The clause on λ-terms is immediate, since $M\ \mathcal{S}\ N$ and $M \Longmapsto Y$ implies $Y\ \mathcal{R}^0_{(M,N)}\ [\![N]\!]$ (by the definition of finite-step simulation), which in turn implies $[\![M]\!]\ \mathcal{R}^1_{(M,N)}\ [\![N]\!]$ (since $[\![M]\!] = \sup\{Y \mid M \Longmapsto Y\}$ and thus we can find an ordered sequence of formal sums that satisfies the condition for $\mathcal{R}^1$).
For the clause on formal sums, the crux is proving the following lemma.

LEMMA 57. *If* $\sum_i p_i; \lambda x.M_i\ \mathcal{R}^n_{\mathcal{E}}\ \sum_j q_j; \lambda x.N_j$ *then:*
(a) $\sum_i p_i \leq \sum_j q_j$
(b) $\sum_i p_i; M_i\{^P/_x\} \Longmapsto Y$ *implies* $Y\ \mathcal{R}^n_{\mathcal{E}}\ \sum_j q_j \cdot [\![N_j\{^Q/_x\}]\!]$, *for all* $P, Q \in \mathcal{E}^\star$.

PROOF. The proof is by induction on n.
For the case $n = 0$, the two properties above are immediate consequences of the definition of $\mathcal{R}$.
For the inductive case, if $Y\ \mathcal{R}^{n+1}_{(M,N)} Z$ then either $Y\ \mathcal{R}^{n}_{(M,N)} Z$, and the result follows by the inductive hypothesis, or $Y = \sup S$ for $S = \{Y_k\}_{k\geq 0}$ a set of formal sums such that $Y_k\ \mathcal{R}^{n}_{(M,N)} Z$ for all k and $Y_k \leq_{\mathtt{apx}} Y_{k+1}$. As a consequence, there is a sequence

Y'_k such that $Y_0 = Y'_0$ and $Y_{k+1} = Y_k + Y'_{k+1}$, i.e., $Y_k = \sum_{0\leq h\leq k} Y'_h$. Hence, it follows from $Y = \sup S$ that $Y = \sum_{k\geq 0} Y'_k$.
The first item follows by the inductive hypothesis, since Y is the supremum of S and $Y_k \in S$ implies $\mathtt{weight}(Y_k) \leq \mathtt{weight}(Z)$. As to the second item, we have $Y \bullet C[M] = \sup\{Y_k \bullet C[M]\} = \sum_{k\geq 0} Y'_k \bullet C[M]$. We want to prove that if $\sum_{k\geq 0} Y'_k \bullet C[M] \Longmapsto X$ for some formal sum X then $X \,\mathcal{R}^{n+1}_{(M,N)} [\![Z \bullet C[N]]\!]$.
If $\sum_{k\geq 0} Y'_k \bullet C[M] \Longmapsto X$ then, by the definition of the multi-step reduction relation (which guarantees that only a finite number of terms are evaluated in the formal sum), there is an $m \geq 0$ such that $\sum_{0\leq k\leq m} Y'_k \bullet C[M] \Longmapsto X'$ and $X = X' + \mathtt{val}(\sum_{k>m} Y'_k \bullet C[M])$.
For any $m' \geq 0$ we have

$$\textstyle\sum_{0\leq k\leq m+m'} Y'_k \bullet C[M] \Longmapsto X' + \mathtt{val}(\sum_{m\leq k\leq m+m'} Y'_k \bullet C[M])$$

Since $\sum_{0\leq k\leq m+m'} Y'_k = Y_{m+m'}$ and $Y_{m+m'} \,\mathcal{R}^{n}_{(M,N)} Z$, by the inductive hypothesis $\sum_{0\leq k\leq m+m'} Y'_k \bullet C[M] \Longmapsto X' + \mathtt{val}(\sum_{m\leq k\leq m+m'} Y'_k \bullet C[M])$ implies

$$X' + \mathtt{val}(\textstyle\sum_{m\leq k\leq m+m'} Y'_k \bullet C[M]) \;\mathcal{R}^{n}_{(M,N)} [\![Z \bullet C[N]]\!].$$

Hence, by the definition of $\mathcal{R}^{n+1}_{(M,N)}$ and by

$$X = \sup\{X' + \mathtt{val}(\textstyle\sum_{m\leq k\leq m+m'} Y'_k \bullet C[M])\,\}_{m'\geq 0}$$

we derive $X \,\mathcal{R}^{n+1}_{(M,N)} [\![Z \bullet C[N]]\!]$.

□

Then the result follows since $Y = \sum_i p_i; \lambda x.M_i \;\mathcal{S}_{\mathcal{E}}\; \sum_j q_j; \lambda x.N_j = Z$ implies $Y \;\mathcal{R}^n_{\mathcal{E}}\; Z$ for some n, and we have $\sum_i p_i \leq \sum_j q_j$ (by the first item of lemma 57), and

$$\textstyle\sum_i p_i; [\![M_i\{^P\!/\!_x\}]\!] = \sup\{Y \mid \sum_i p_i; M_i\{^P\!/\!_x\} \Longmapsto Y\} \;\mathcal{R}^{n+1}_{\mathcal{E}}\; \sum_j q_j; [\![N_j\{^Q\!/\!_x\}]\!],$$

for all $P, Q \in \mathcal{E}^\star$ (by the second item of lemma 57 and by the definition of relation $\mathcal{R}^{n+1}$), which implies $\mathcal{R}^{n+1}_{\mathcal{E}} \subseteq \mathcal{S}_{\mathcal{E}}$.

Proof of Lemma 18

Let $\mathcal{R}$ be a finite-step simulation saturated by approximants and let

$$\mathcal{S} = \{(C[M], C[N]) \;\mid\; M \,\mathcal{R}\, N\} \cup \{((C[M], C[N]), Y, Z) \;\mid\; Y \;\mathcal{S}'_{(M,N)}\; Z\}$$

with

$$\begin{aligned}\mathcal{S}' = {} & \{((M,N),1;\lambda x.C'[M],1;\lambda x.C'[N]) \mid M \;\mathcal{R}\; N\} \\ & \cup\{((M,N),Y,Z) \mid Y \;\mathcal{R}_{(M,N)}\; Z\} \\ & \cup\{((M,N),\emptyset,Z) \mid \text{ for some } M,N,Z\}\end{aligned}$$

We show that $\mathcal{S}$ is a finite-step simulation up-to lifting. Note that the last set of triples is added to the relation since for any term M that is not a value we have $M \Longmapsto_0 \emptyset$, and $\emptyset$ is simulated by any formal sum Z. We first prove the following result:

Lemma 58. *For any context C, if $M \;\mathcal{R}\; N$ and $C[M] \Longmapsto Y$, then we have $Y \,\mathtt{lift}(\mathcal{S}'_{(M,N)})\, [\![C[N]]\!]$.*

Proof. We prove by induction on the length n of the reduction that $M \,\mathcal{R}\, N$ and $C[M] \Longmapsto_n Y$ imply $Y \,\mathtt{lift}(\mathcal{S}'_{(M,N)})\, [\![C[N]]\!]$. If $Y = \emptyset$ then the result follows by the third set of $\mathcal{S}'$. Suppose that $Y \neq \emptyset$. If $n = 0$ then we have two cases:

- $C = [\cdot]$ and M is a value. Then $M \Longmapsto Y$ and since $\mathcal{R}$ is a finite-step simulation we have $Y \,\mathcal{R}_{(M,N)} [\![N]\!] = [\![C[N]]\!]$, which implies that they are in $\mathcal{S}'_{(M,N)}$ as well.

- $C = \lambda x.C'$. Then $Y = 1; \lambda x.C'[M]\ \mathcal{S}'_{(M,N)}\ 1; \lambda x.C'[N] = [\![C[N]]\!]$, by the first set of $\mathcal{S}'$.

Suppose now that $C[M] \Longmapsto_{n+1} Y$.

- $C = [\cdot]$ and $M \Longmapsto_{n+1} Y$. The result follows from the fact that $\mathcal{R}$ is a finite-step simulation, as in the first case of $n = 0$.
- $C = C_1 \oplus C_2$ and $C[M] \longrightarrow \frac{1}{2}; C_1[M] + \frac{1}{2}; C_2[M] \Longmapsto_n Y$. Then $C_1[M] \Longmapsto_{n_1} Y_1$, $C_2[M] \Longmapsto_{n_2} Y_2$ and $Y = \frac{1}{2}; Y_1 + \frac{1}{2}; Y_2$. We have $[\![C[N]]\!] = \frac{1}{2}; [\![C_1[N]]\!] + \frac{1}{2}; [\![C_2[N]]\!]$ and it follows from the inductive hypothesis on n_1 and n_2 that $Y_1\ \mathtt{lift}(\mathcal{S}'_{(M,N)})\ [\![C_1[N]]\!]$ and $Y_2\ \mathtt{lift}(\mathcal{S}'_{(M,N)})\ [\![C_2[N]]\!]$. Hence, $Y\ \mathtt{lift}(\mathcal{S}'_{(M,N)})\ [\![C[N]]\!]$.
- $C = C_1C_2$. Then $C[M] \Longmapsto_{n+1} Y$ implies $C[M] \Longrightarrow_{n_1} Y_1C_2[M] \Longmapsto_{n_2} Y$, where $C_1[M] \Longmapsto_{n_1} Y_1 = \sum_i p_i; \lambda x.P_i$. Since $n_1 \leq n$ (by $Y \neq \emptyset$), we can apply the inductive hypothesis and derive $Y_1\ \mathtt{lift}(\mathcal{S}'_{(M,N)})\ [\![C_1[N]]\!]$, i.e., $Y_1 = \sum_j r_j \cdot Y'_j$ and $[\![C_1[N]]\!] = \sum_j r_j \cdot Z'_j$ for $Y'_j\ \mathcal{S}'_{(M,N)}\ Z'_j$. Hence, $Y_1C_2[M] \Longmapsto_{n_2} Y$ implies $Y'_jC_2[M] \Longmapsto_{n'_j} Y''_j$, with $\sum_j n'_j = n_2$ and $Y = \sum_j r_j \cdot Y''_j$, and $[\![C[N]]\!] = [\![[\![C_1[N]]\!]C_2[N]]\!] = [\![\sum_j r_j \cdot Z'_jC_2[N]]\!] = \sum_j r_j \cdot [\![Z'_jC_2[N]]\!] = \sum_j r_j; [\![Z'_j \bullet C_2[N]]\!]$.
  Since $\mathtt{lift}(\mathtt{lift}(\mathcal{R})) = \mathtt{lift}(\mathcal{R})$ for any relation $\mathcal{R}$, the result follows if we can prove that for every j $Y'_jC_2[M] \Longmapsto_{n'_j} Y''_j$ implies $Y''_j\ \mathtt{lift}(\mathcal{S}'_{(M,N)})\ [\![Z'_j \bullet C_2[N]]\!]$. If $Y'_j\ \mathcal{S}'_{(M,N)}\ Z'_j$ then either $Y'_j = \emptyset$ and the result trivially follows, or one of these cases hold:
  - $Y'_j = 1; \lambda x.C'[M]$ and $Z'_j = 1; \lambda x.C'[N]$. Hence, either $Y''_j = \emptyset$, in which case the result follows by the third set of $\mathcal{S}'$, or $Y'_jC_2[M] \longrightarrow 1; C'[M]\{^{C_2[M]}/_x\} \Longmapsto_{n''_j} Y''_j$, with $n''_j \leq n$. Terms $C'[M]\{^{C_2[M]}/_x\}$ and $C'[N]\{^{C_2[N]}/_x\}$ are respectively of the form $C''[M]$ and $C''[N]$, so we can apply the inductive hypothesis to derive $Y''_j\ \mathtt{lift}(\mathcal{S}'_{(M,N)})\ Z'_j \bullet C_2[N]$.
  - $Y'_j\ \mathcal{R}_{(M,N)}\ Z'_j$. It is easy to check that $Y'_jC_2[M] \Longmapsto Y''_j$ iff $Y'''_j \bullet C_2[M] \Longmapsto Y''_j$ for some $Y'''_j \leq_{\mathtt{apx}} Y'_j$. Since $\mathcal{R}$ is finite-step simulation saturated by approximants, $Y'''_j \bullet C_2[M] \Longmapsto Y''_j$ implies $Y''_j\ \mathcal{R}_{(M,N)}\ [\![Z'_j \bullet C_2[N]]\!]$, and the result follows from $\mathcal{R}_{(M,N)} \subseteq \mathtt{lift}(\mathcal{S}'_{(M,N)})$.

□

We derive from lemma 58 that $\mathcal{S}$ is a finite-step simulation up-to lifting as follows:

- Let $C[M]\ \mathcal{S}\ C[N]$ with $M\ \mathcal{R}\ N$. If $C[M] \Longmapsto Y$ then, by Lemma 58, $Y\ \mathtt{lift}(\mathcal{S}'_{(M,N)})\ [\![C[N]]\!]$. Therefore, $Y\ \mathtt{lift}(\mathcal{S}_{(C[M],C[N])})\ [\![C[N]]\!]$.
- Let $1; \lambda x.C'[M]\ \mathcal{S}_{C[M],C[N]}\ 1; \lambda x.C'[N]$ with $M\ \mathcal{R}\ N$. Then for any C'' there is a context C''' such that $1; \lambda x.C'[M] \bullet C''[C[M]] = 1; C'''[M]$ and $1; \lambda x.C'[N] \bullet C''[C[N]] = 1; C'''[N]$. It is easy to check that $1; P \Longmapsto Y$ iff $P \Longmapsto Y$ and that $[\![1; P]\!] = [\![P]\!]$ for any term P. Therefore, by Lemma 58, $1; C'''[M] \Longmapsto Y$ implies $Y\ \mathtt{lift}(\mathcal{S}'_{(M,N)})\ [\![1; C'''[M]]\!]$, which implies $Y\ \mathtt{lift}(\mathcal{S}_{C[M],C[N]})\ [\![1; C'''[M]]\!]$.
- Let $Y\ \mathcal{R}_{C[M],C[N]}\ Z$ with $Y\ \mathcal{R}_{(M,N)}\ Z$. Then, as $\mathcal{R}$ is a finite-step simulation, for any C' $Y \bullet C'[C[M]] \Longmapsto Y'$ implies $Y'\ \mathcal{R}_{(M,N)}\ [\![Z \bullet C'[C[N]]]\!]$, which in turn implies $Y'\ \mathcal{S}_{C[M],C[N]}\ [\![Z \bullet C'[C[N]]]\!]$.
- Let $\emptyset\ \mathcal{R}_{C[M],C[N]}\ Z$. Then for any P we have $\emptyset \bullet P \Longmapsto Y$ implies $Y = \emptyset$ and we stay in the third set.

Proof of Lemma 22

To complete the proof, it remains to prove that for any C' we have $[\![[\![C[M]]\!] \bullet C'[M]]\!] = [\![C[M]\, C'[M]]\!]$. This result can be derived by Lemma 59 below, since

$$\begin{aligned} [\![[\![C[M]]\!] \bullet C'[M]]\!] &= [\![[\![C[M]]\!] C'[M]]\!] && \text{(by 59(2))} \\ &= [\![C[M]\, C'[M]]\!] && \text{(by 59(1))} \end{aligned}$$

Lemma 59. *For any M, N, Y it holds:*

(1) $[\![MN]\!] = [\![[\![M]\!]N]\!]$
(2) $[\![YN]\!] = [\![Y \bullet N]\!]$

Proof. (1) For the left to right inequality (i.e., $[\![MN]\!] \leq [\![[\![M]\!]N]\!]$), we prove that $MN \Longmapsto Z$ implies $[\![M]\!]N \Longmapsto Z'$ for some $Z \leq_{\mathtt{apx}} Z'$. Since $MN \Longmapsto Z$ implies $M \Longmapsto Z''$ and $Z''N \Longmapsto Z$ for some Z''. By definition $Z'' \leq_{\mathtt{apx}} [\![M]\!]$, so we derive $[\![M]\!]N \Longmapsto Z'$ for some Z' such that $Z \leq_{\mathtt{apx}} Z'$.
For the converse inequality, we analogously prove that $[\![M]\!]N \Longmapsto Z$ implies $MN \Longmapsto Z'$ for some $Z \leq_{\mathtt{apx}} Z'$. Since relations $\Longrightarrow$ and $\Longmapsto$ are small-step and finitary, although $[\![M]\!]$ could have an infinite support (i.e., be a sum of infinitely many values) only a finite number of β-reductions substituting N in a value of $[\![M]\!]$ can be performed in the reduction $[\![M]\!]N \Longmapsto Z$. Hence, there is some formal sum Z'' with finite support such that $Z'' \leq_{\mathtt{apx}} [\![M]\!]$ and $Z''N \Longmapsto Z$. This implies that $M \Longmapsto Z'''$ for some Z''' such that $Z'' \leq_{\mathtt{apx}} Z'''$, which in turn implies $MN \Longmapsto Z'$ for some Z' such that $Z \leq_{\mathtt{apx}} Z'$.
(2) The left to right inequality (i.e., $[\![YN]\!] \leq [\![Y \bullet N]\!]$) follows trivially, since $YN \Longmapsto Z$ implies $Y \bullet N \Longmapsto Z'$ for some Z' such that $Z \leq_{\mathtt{apx}} Z'$.
In order to prove $[\![Y \bullet N]\!] \leq [\![YN]\!]$ we show that $Y \bullet N \Longmapsto Z$ implies that there is a sequence of formal sums $\{Z_j\}_{j\geq 0}$ (for j ranging over natural numbers) ordered with respect to $\leq_{\mathtt{apx}}$ such that $YN \Longmapsto Z_j$ for every j, and with $\sup\{Z_j\}_{j\geq 0} = Z$. Hence, the result follows since

$$[\![Y \bullet N]\!] = \sup\{Z \mid Y \bullet N \Longmapsto Z\} \leq \sup\{(\sup\{Z_j\}_{j\geq 0}) \mid Y \bullet N \Longmapsto Z\}$$

and, since for every Z we have $\sup\{Z_j\}_{j\geq 0} \leq [\![YN]\!]$, we derive

$$\sup\{(\sup\{Z_j\}_{j\geq 0}) \mid Y \bullet N \Longmapsto Z\} \leq [\![YN]\!].$$

Hence, it remains to prove that $Y \bullet N \Longmapsto Z$ implies that there is a (infinite) sequence of formal sums $\{Z_j\}_{j\geq 0}$ ordered with respect to $\leq_{\mathtt{apx}}$ such that $YN \Longmapsto Z_j$ for every j, and with $\sup\{Z_j\}_{j\geq 0} = Z$. Let $Y = \sum_{i\in I} p_i; \lambda x.P_i$ and $Y \bullet N \Longmapsto Z$. It follows by the definition of $\Longmapsto$ that there is a finite $I' \subseteq I$ and a formal sum Z' such that $\sum_{i\in I'} p_i; P_i\{^N\!/\!_x\} \Longmapsto Z'$ and $Z = Z' + \mathtt{val}(\sum_{i\in I\setminus I'} p_i; P_i\{^N\!/\!_x\})$ (only a finite number of steps can be taken, and thus only a finite number of summands can move). Hence, by allowing all and only the summands with index $i \in I'$ to perform the β-reduction substituting N in P_i we derive $YN \Longmapsto Z'$. Then, by allowing one at a time the summands of Y with index $i \in I\setminus I'$ to perform the β-reduction substituting N in P_i, we obtain a sequence $\{Z_j\}_{j\geq 0}$, that starts from $Z_0 = Z'$ and progressively adds to it the values in $\mathtt{val}(\sum_{i\in I\setminus I'} p_i; P_i\{^N\!/\!_x\})$ one by one. This sequence is ordered with respect to $\leq_{\mathtt{apx}}$, it is such that $YN \Longmapsto Z_j$ for every j, and it has $\sup\{Z_j\}_{j\geq 0} = Z$.
□

Proof of Lemma 25

Let $\mathcal{R}$ be a finite-step simulation up-to lifting and context. We prove that

$$\begin{array}{l}\mathcal{R}' \stackrel{\text{def}}{=} \mathtt{Pairs}(\mathcal{R}) \\ \quad \cup \{((M,N),1;\lambda x.C[M],1;\lambda x.C[N]) \mid M\ \mathcal{R}\ N\} \\ \quad \cup \{((M,N),Y,Z) \mid Y'\,\mathcal{R}_{(M,N)}Z \text{ and } Y \leq_{\mathtt{apx}} Y', \text{ for some } Y'\} \\ \quad \cup \{((M,N),\emptyset,Z) \mid \text{ for some } M,N,Z\}\end{array}$$

is a finite-step simulation up-to lifting, from which the result follows by $\mathcal{R} \subseteq \mathcal{R}'$.
Note that the relation $\mathtt{lift}(\mathcal{R}'_{(M,N)})$ is saturated by approximants, i.e., the following property holds: if $Y \leq_{\mathtt{apx}} Y'\,\mathtt{lift}(\mathcal{R}'_{(M,N)})\,Z$ then $Y\,\mathtt{lift}(\mathcal{R}'_{(M,N)})\,Z$. We first prove the following result:

Lemma 60. *For any context C, if $M\ \mathcal{R}\ N$ and $C[M] \Longmapsto Y$, then we have $Y\,\mathtt{lift}(\mathcal{R}'_{(M,N)})\,[\![C[N]]\!]$.*

Proof. We prove that $M\ \mathcal{R}\ N$ and $C[M] \Longmapsto_n Y$ imply $Y\,\mathtt{lift}(\mathcal{R}'_{(M,N)})\,[\![C[N]]\!]$, by induction on n. If $Y = \emptyset$ then the result holds by the last set of $\mathcal{R}'$. Suppose that $Y \neq \emptyset$. If $n = 0$ then we have two cases:

- $C = [\cdot]$ and M is a value. Then $M \Longmapsto Y$ and we have $Y\,\mathtt{lift}(\mathcal{R}_{(M,N)})\,[\![N]\!] = [\![C[N]]\!]$.
- $C = \lambda x.C'$. Then $Y \leq_{\mathtt{apx}} 1;\lambda x.C'[M]\mathtt{dirac}((M,N)^\star)1;\lambda x.C'[N] = [\![C[N]]\!]$.

Suppose now that $C[M] \Longmapsto_{n+1} Y$.

- $C = [\cdot]$ and $M \Longmapsto_{n+1} Y$. The result follows from the fact that $\mathcal{R}$ is a finite-step simulation up-to lifting and context, as in the first case of $n = 0$.
- $C = C_1 \oplus C_2$ and $C[M] \longrightarrow \frac{1}{2};C_1[M] + \frac{1}{2};C_2[N] \Longmapsto_n Y$. Then $C_1[M] \Longmapsto_{n_1} Y_1$, $C_2[N] \Longmapsto_{n_2} Y_2$ with $n_1 + n_2 \leq n$ and $Y = \frac{1}{2};Y_1 + \frac{1}{2};Y_2$. We have $[\![C[N]]\!] = \frac{1}{2};[\![C_1[N]]\!] + \frac{1}{2};[\![C_2[N]]\!]$ and it follows from the inductive hypothesis on n_1 and n_2 that $Y_1\,\mathtt{lift}(\mathcal{R}'_{(M,N)})\,[\![C_1[N]]\!]$ and $Y_2\,\mathtt{lift}(\mathcal{R}'_{(M,N)})\,[\![C_2[N]]\!]$. Hence, we derive $Y\,\mathtt{lift}(\mathcal{R}'_{(M,N)})\,[\![C[N]]\!]$.
- $C = C_1C_2$. Then $C[M] \Longmapsto_{n+1} Y$ implies $C[M] \Longrightarrow_{n_1} Y_1C_2[M] \Longmapsto_{n_2} Y$, where $C_1[M] \Longmapsto_{n_1} Y_1 = \sum_i p_i;\lambda x.P_i$. Since $n_1 \leq n$ (by $Y \neq \emptyset$), we can apply the inductive hypothesis and derive $Y_1\,\mathtt{lift}(\mathcal{S}'_{(M,N)})\,[\![C_1[N]]\!]$, i.e., $Y_1 = \sum_j r_j{\cdot}Y'_j$ and $[\![C_1[N]]\!] = \sum_j r_j{\cdot}Z'_j$ for $Y'_j\mathcal{R}'_{(M,N)}Z'_j$. Hence, $Y_1C_2[M] \Longmapsto_{n_2} Y$ implies $Y'_jC_2[M] \Longmapsto_{n'_j} Y''_j$, with $\sum_j n'_j = n_2$ and $Y = \sum_j r_j{\cdot}Y''_j$, and

$$[\![C[N]]\!] = [\![[\![C_1[N]]\!]C_2[N]]\!] =$$

$$[\![\textstyle\sum_j r_j{\cdot}Z'_jC_2[N]]\!] = \sum_j r_j{\cdot}[\![Z'_jC_2[N]]\!] = \sum_j r_j;[\![Z'_j \bullet C_2[N]]\!].$$

  Since $\mathtt{lift}(\mathtt{lift}(\mathcal{R})) = \mathtt{lift}(\mathcal{R})$ for any relation $\mathcal{R}$, the result follows if we can prove that for every j $Y'_jC_2[M] \Longmapsto_{n'_j} Y''_j$ implies $Y''_j\,\mathtt{lift}(\mathcal{R}'_{(M,N)})\,[\![Z'_j \bullet C_2[N]]\!]$. If $Y'_j\ \mathcal{R}'_{(M,N)}\ Z'_j$ then one of the following cases hold:
  - $Y'_j = 1;\lambda x.C'[M]$ and $Z'_j = 1;\lambda x.C'[N]$. Hence, either $Y''_j = \emptyset$, in which case the result follows by the last set of $\mathcal{R}'$, or $Y'_jC_2[M] \longrightarrow 1;C'[M]\{^{C_2[M]}\!/\!_x\} \Longmapsto_{n''_j} Y''_j$, with $n''_j \leq n$. Terms $C'[M]\{^{C_2[M]}\!/\!_x\}$ and $C'[N]\{^{C_2[N]}\!/\!_x\}$ are respectively of the form $C''[M]$ and $C''[N]$ and we can apply the inductive hypothesis to derive $Y''_j\,\mathtt{lift}(\mathcal{R}'_{(M,N)})\,[\![Z'_j \bullet C_2[N]]\!]$.
  - $Y'_j \leq_{\mathtt{apx}} X\,\mathcal{R}_{(M,N)}Z'_j$. If $Y'_jC_2[M] \Longrightarrow Y''_j$ then there is a X' such that $XC_2[M] \Longmapsto_{n''_j} X'$ and $Y''_j \leq_{\mathtt{apx}} X'$, for some $n''_j \leq n'_j$. It is easy to check that there is exists a X'' such that $X \bullet C_2[M] \Longmapsto_{n'''_j} X''$ and $X' \leq_{\mathtt{apx}} X''$, for some $n'''_j < n''_j$. Since $X\,\mathcal{R}_{(M,N)}Z'_j$, there are two cases:

$*$ $X \bullet C_2[M] \Longrightarrow F$ and $Z'_j \bullet C_2[N] \Longrightarrow G$ with $F\,\mathtt{lift}(\mathtt{dirac}((M,N)^\star))\,G$.
Hence, $F \Longmapsto_{n'''_j} X'''$ with $X'' \leq_{\mathtt{apx}} X'''$, for some $n''''_j \leq n'''_j < n$. We can apply the inductive hypothesis to the pairs in $(M,N)^\star$ whose projections respectively compose F and G through the lifting, and derive $X'''\,\mathtt{lift}(\,\mathtt{lift}(\mathcal{R}'_{(M,N)})\,)\,[\![G]\!]$. It follows from $Y''_j \leq_{\mathtt{apx}} X'''$ that $Y''_j\,\mathtt{lift}(\mathcal{R}'_{(M,N)})\,[\![G]\!] = [\![Z'_j \bullet C_2[N]]\!]$.

$*$ $X''\,\mathtt{lift}(\mathtt{dirac}((M,N)^\star) \cup \mathcal{R}_{(M,N)})\,[\![Z'_j \bullet C_2[N]]\!]$.
As $Y''_j \leq_{\mathtt{apx}} X''$, we then derive $Y''_j\,\mathtt{lift}(\mathcal{R}'_{M,N})\,[\![Z'_j \bullet C_2[N]]\!]$, since $\mathtt{lift}(\mathcal{R}'_{(M,N)})$ is saturated by approximants.

– the result is immediate if $Y'_j = \emptyset$.

□

From Lemma 60, we can now derive that $\mathcal{R}'$ is a finite-step simulation up-to lifting:

(1) if $M\ \mathcal{R}'\ N$ then $M\ \mathcal{R}\ N$, and $M \Longmapsto Y$ implies $Y\,\mathtt{lift}(\mathcal{R}_{(M,N)})\,[\![N]\!]$, by the definition of $\mathcal{R}$. Then the result follows by $\mathcal{R} \subseteq \mathcal{R}'$.

(2) if $Y\ \mathcal{R}'_{(M,N)}\ Z$ then:

- if $Y \leq_{\mathtt{apx}} Y'\ \mathcal{R}_{(M,N)}\ Z$ then for all $P, Q \in (M,N)^\star$ we have two cases:
  – $Y' \bullet P \Longrightarrow F$ and $Z \bullet Q \Longrightarrow G$ with $F\,\mathtt{lift}(\mathtt{dirac}((M,N)^\star))\,G$.
  If $Y \bullet P \Longmapsto Y''$ then there is a Y''' such that $Y' \bullet P \Longrightarrow F \Longmapsto Y'''$ and $Y'' \leq_{\mathtt{apx}} Y'''$. It follows from $F\,\mathtt{lift}(\mathtt{dirac}((M,N)^\star))\,G$ and from Lemma 60 that $F \Longmapsto Y'''$ implies $Y'''\,\mathtt{lift}(\,\mathtt{lift}(\mathcal{R}'_{(M,N)})\,)\,[\![G]\!]$, which is equivalent to saying that $Y'''\,\mathtt{lift}(\mathcal{R}'_{(M,N)})\,[\![Z \bullet Q]\!]$. Since $F'' \leq_{\mathtt{apx}} Y'''$ and $\mathtt{lift}(\mathcal{R}'_{(M,N)})$ is saturated by approximants, we derive $Y'''\,\mathtt{lift}(\mathcal{R}'_{(M,N)})\,[\![Z \bullet Q]\!]$.
  – $Y' \bullet P \Longmapsto X$ and $X\,\mathtt{lift}(\mathtt{dirac}((M,N)^\star) \cup \mathcal{R}_{(M,N)})\,[\![Z \bullet Q]\!]$.
  If $Y \bullet P \Longmapsto Y''$ then there is a Y''' such that $Y' \bullet P \Longmapsto Y'' + Y''' = X$ and, by the definition of $\mathcal{R}$, $Y'' + Y'''\,\mathtt{lift}(\mathtt{dirac}((M,N)^\star) \cup \mathcal{R}_{(M,N)})\,[\![Z \bullet Q]\!]$, which implies $Y'' + Y'''\,\mathtt{lift}(\mathcal{R}'_{(M,N)})\,[\![Z \bullet Q]\!]$. Since $Y'' \leq_{\mathtt{apx}} Y'' + Y'''$, we have $Y''\,\mathtt{lift}(\mathcal{R}'_{(M,N)})\,[\![Z \bullet Q]\!]$.
- If $Y = 1; \lambda x.C[M]$ and $Z = 1; \lambda x.C[N]$ with $M\ \mathcal{R}\ N$ then for all $P, Q \in (M,N)^\star$ we have $Y \bullet P = 1; C'[M]$ and $Z \bullet Q = 1; C'[N]$ for some C'. Then, by Lemma 60, having $Y \bullet P \Longmapsto Y'$ we derive $Y'\,\mathtt{lift}(\mathcal{R}'_{(M,N)})\,[\![Z \bullet Q]\!]$.
- If $Y = \emptyset$ then the simulation clause holds and we stay in the last set of $\mathcal{R}'$.

Proof of Lemma 29

Let $\rhd^{\star t}$ denote the transitive closure of $\rhd^\star$, i.e., $P \rhd^{\star t} P'$ if there are $P = P_1, P_2, ..., P_k = P'$ such that for every $1 \leq i < k$ there are a context C_i and tuples $\widetilde{P_i}, \widetilde{P'_i}$ with $P_i = C_i[\widetilde{P_i}]$ and $P_{i+1} = C_i[\widetilde{P'_i}] = C_{i+1}[\widetilde{P}_{i+1}]$ and $\widetilde{P_i} \rhd \widetilde{P'_i}$. Define the following term relation

$$\mathcal{R}_{(M,N)} \stackrel{\mathtt{def}}{=} (\rhd^{\widehat{\star t}}\ (M,N)^{\widehat{\star}}) \cup (\rhd^{\widehat{\star t}} \mathcal{T}_{(M,N)})$$

We consider $\rhd^{\star t}$, instead of just considering $\rhd^\star$, since otherwise the induction hypothesis would not suffice in the proof of Lemma 61(**), for the application case.
Let $\mathtt{liftd}(\cdot)$ denote $\mathtt{lift}(\mathtt{dirac}(\cdot))$.

Lemma 61. *It holds that*

(*) $P \rhd^{\star t} P'$ and $P \Longmapsto_n Y$ imply $P' \Longmapsto_{n'} Y'$ with $n \geq n'$ and $Y\,\mathtt{liftd}(\rhd^{\widehat{\star t}}) \leq_{\mathtt{apx}} Y'$
(**) $P(\rhd^{\star t}\ (M,N)^\star)Q$ and $P \Longmapsto Y$ imply $Y\,\mathtt{disliftd}(\mathcal{R}_{(M,N)}) \leq_{\mathtt{apx}} [\![Q]\!]$

Proof. We first prove (*). Let $P = P_1, P_2, ..., P_k = P'$ be such that for every i, for $1 \leq i < k$, there are a context C_i and tuples $\widetilde{P}_i, \widetilde{P'_i}$ with $P_i = C_i[\widetilde{P}_i]$ and $P_{i+1} = C_i[\widetilde{P'_i}] = C_{i+1}[\widetilde{P}_{i+1}]$ and $\widetilde{P}_i \rhd \widetilde{P'_i}$. Suppose that $P \Longmapsto_n Y$. We prove by induction on k that for every i such that $1 \leq i < k$ we have $P_i \Longmapsto_{m_i} Y_i$ and $m_i \geq m_{i+1}$ and $Y_i\, \mathtt{liftd}(\rhd^{\widehat{\star}t}) \leq_{\mathtt{apx}} Y_{i+1}$. The result is trivial for $k = 1$. Suppose that $k = k' + 1$. The result follows from the inductive hypothesis and from

$$Y\, \mathtt{liftd}(\rhd^{\widehat{\star}t}) \leq_{\mathtt{apx}} Y'\, \mathtt{liftd}(\rhd^{\widehat{\star}t}) \leq_{\mathtt{apx}} Y'' \text{ implies } Y\, \mathtt{liftd}(\rhd^{\widehat{\star}t}) \leq_{\mathtt{apx}} Y'',$$

if we can prove that for any C and $\widetilde{P}, \widetilde{P'}$ such that $\widetilde{P} \rhd \widetilde{P'}$, whenever $C[\widetilde{P}] \Longmapsto_m Y$, then $C[\widetilde{P'}] \Longmapsto_{m'} Y'$ with $m \geq m'$ and $Y\, \mathtt{liftd}(\rhd^{\widehat{\star}t}) \leq_{\mathtt{apx}} Y'$. This is proved by induction on m. If $m = 0$ then $Y = Y'$ and the result follows. Suppose $C[\widetilde{P}] \Longmapsto_{m+1} Y$. We have three cases:

- if $C = [\cdot]$ then the result immediately follows by $\widetilde{P} = P \rhd P' = \widetilde{P'}$.
- $C = C_1 \oplus C_2$. Then there are Y_1, Y_2 such that $Y = \frac{1}{2}; Y_1 + \frac{1}{2}; Y_2$ and $C_i[\widetilde{P}] \Longmapsto_{m_i} Y_i$ for $i = 1, 2$ and $m_i < m + 1$. The result follows from the inductive hypothesis
- $C = C_1 C_2$. If $C_1[\widetilde{P}]$ is a value then, by the definition of $\rhd$, $C_1[\widetilde{P'}]$ is exactly the same value. Then the terms resulting after the β-reduction are in $\rhd^{\widehat{\star}t}$, and we conclude by the inductive hypothesis.
  If $C_1[\widetilde{P}] \Longmapsto_{m_1} Y_1$ then we can apply the inductive hypothesis to m_i and the result follows.

We can now prove (**) by showing by induction on n that $P(\rhd^{\star t}\ (M, N)^{\star})Q$ and $P \Longmapsto_n Y$ imply $Y\, \mathtt{disliftd}(\mathcal{R}_{(M,N)}) \leq_{\mathtt{apx}} [\![Q]\!]$.
Let $P \rhd^{\star t} P'$, $P' = C[M]$ and $Q = C[N]$. If $n = 0$ and $Y \neq \emptyset$ then P is a value, $Y = 1; P$ and by (*) P' is a value too, with $P \rhd^{\widehat{\star}t} P'$. We have two cases:

- $C = [\cdot]$, with $P' = M$ a value, and $Q = N$.
  By the definition of $\mathcal{T}_{(M,N)}$ we have $1; P' = [\![M]\!]\mathtt{dirac}(\mathcal{T}_{(M,N)})[\![N]\!]$ and it follows from $P \rhd^{\widehat{\star}t} P'$ that $Y\, \mathtt{disliftd}(\rhd^{\widehat{\star}t}\mathcal{T}_{(M,N)})[\![Q]\!]$.
- $P' = \lambda x.C'[M]$ and $Q = \lambda x.C'[N]$.
  Then $Y\, \mathtt{disliftd}(\rhd^{\widehat{\star}t}\ (M, N)^{\widehat{\star}})[\![Q]\!]$.

If $P \Longmapsto_{n+1} Y$ then by (*) we have $P' \Longmapsto_m Y'$ with $m \leq n + 1$ and $Y\, \mathtt{liftd}(\rhd^{\widehat{\star}t}) \leq_{\mathtt{apx}} Y'$. Suppose that $Y' \neq \emptyset$. We have three cases:

- $C = [\cdot]$. If $M \Longmapsto_m Y'$ then $Y' = [\![M]\!]\mathtt{dirac}(\mathcal{T}_{(M,N)})[\![N]\!]$. For any term relations $\mathcal{S}, \mathcal{S}'$ we have
  – $\mathtt{liftd}(\mathcal{S})\, \mathtt{dirac}(\mathcal{S}') \subseteq \mathtt{disliftd}(\mathcal{S}\mathcal{S}')$
  – $\leq_{\mathtt{apx}} \mathtt{disliftd}(\mathcal{T}_{(M,N)}) \subseteq \mathtt{disliftd}(\mathcal{T}_{(M,N)}) \leq_{\mathtt{apx}}$
  Hence, we derive $Y\, \mathtt{disliftd}(\rhd^{\widehat{\star}t}\mathcal{T}_{(M,N)}) \leq_{\mathtt{apx}} [\![N]\!]$.
- $C = C_1 + C_2$. Then there are Y_1, Y_2 such that $Y' = \frac{1}{2}; Y_1 + \frac{1}{2}; Y_2$ and $C_i[M] \Longmapsto_{m_i} Y_i$ for $i = 1, 2$ and $m_i \leq n$. By the inductive hypothesis, we have

  $$Y'\, \mathtt{lift}(\mathtt{disliftd}(\mathcal{R}_{(M,N)})) \leq_{\mathtt{apx}} [\![Q]\!]$$

  As $\mathtt{lift}(\mathtt{disliftd}(\cdot)) = \mathtt{disliftd}(\cdot)$ and $\mathtt{liftd}(\mathcal{S})\, \mathtt{disliftd}(\mathcal{S}') \subseteq \mathtt{disliftd}(\mathcal{S}\mathcal{S}')$ for any $\mathcal{S}$ and $\mathcal{S}'$, from $\rhd^{\star t}\rhd^{\star t} = \rhd^{\star t}$ we derive

  $$Y\, \mathtt{disliftd}(\mathcal{R}_{(M,N)}) \leq_{\mathtt{apx}} [\![Q]\!].$$

- $C = C_1 C_2$. We consider two cases:

– $C_1[M] = M$ and $M = \lambda x.M'$ and $N = \lambda x.N'$ are values.
Then $M\ \mathcal{T}_{(M,N)}\ N$ and we have $MC_2[M] \Longrightarrow_m Y'$ iff $M'\{^{C_2[M]}\!/_x\} \stackrel{\mathrm{d}}{\Longrightarrow} F \Longrightarrow_{m'} Y'$, with $m' < m \leq n{+}1$ and $[\![NC_2[N]]\!] = [\![N'\{^{C_2[N]}\!/_x\}]\!] = [\![G]\!]$, where $F\mathtt{disliftd}(\rhd^\star \mathcal{T}^{\star -}_{(M,N)})G$ (note that here the determinism of the reduction to F is used in order to guarantee that $F \Longrightarrow_{m'} Y'$ and that $m' < m \leq n+1$). Hence, we have $Y'\mathtt{dislift}(\{(Y_i, Z_i)\}_i)[\![G]\!]$, with either $Y_i\mathtt{dirac}(\rhd^{\widehat{\star}} \mathcal{T}^{\star -}_{(M,N)})Z_i$ (which implies $Y_i\mathtt{dirac}(\mathcal{R}_{(M,N)})Z_i$) or $M_i \Longrightarrow_{m_i} Y_i$ and $[\![N_i]\!] = Z_i$ for $M_i \rhd^\star (M,N)^\star N_i$ and $m_i \leq m' < n+1$, and by applying the inductive hypothesis we derive $Y_i\mathtt{disliftd}(\mathcal{R}_{(M,N)}) \leq_{\mathtt{apx}} Z_i$. Therefore, $Y\,\mathtt{liftd}(\rhd^{\star t}) \leq_{\mathtt{apx}} Y'\mathtt{disliftd}(\mathcal{R}_{(M,N)}) \leq_{\mathtt{apx}} [\![Q]\!]$, and the result follows.
– $C_1[M] = \lambda x.C'_1[M]$ and $C_1[N] = \lambda x.C'_1[N]$.
We can apply the inductive hypothesis to $C'_1[M]\{^{C_2[M]}\!/_x\} \Longrightarrow_{m'} Y'$ to derive

$$Y'\mathtt{disliftd}(\mathcal{R}_{(M,N)}) \leq_{\mathtt{apx}} [\![C'[N]\{^{C_2[N]}\!/_x\}]\!],$$

then we have $Y\,\mathtt{liftd}(\rhd^{\star t}) \leq_{\mathtt{apx}} Y'\mathtt{disliftd}(\mathcal{R}_{(M,N)}) \leq_{\mathtt{apx}} [\![C'[N]\{^{C_2[N]}\!/_x\}]\!]$, which implies the result.

□

We can now show that the relation

$$\mathcal{S} \stackrel{\mathsf{def}}{=} \{(M,N)\} \cup \{((M,N),1;V,1;W) \mid V\ \mathcal{R}_{(M,N)}\ W\} \cup \{((M,N),\emptyset,Z) \mid \text{ for any } Z\}$$

is a finite-step simulation up-to distribution and lifting:

- since $M(\rhd^{\star t} (M,N)^\star)N$, by Lemma 61(**), if $M \Longrightarrow Y$ then

$$Y\mathtt{disliftd}(\mathcal{R}_{(M,N)})Y' \leq_{\mathtt{apx}} [\![N]\!],$$

which implies $Y\mathtt{dislift}(\mathcal{S}_{(M,N)})[\![N]\!]$ (using the last set of relation $\mathcal{S}_{(M,N)}$ to eliminate the approximation preorder $\leq_{\mathtt{apx}}$).
- the weight of the formal sums is the same.
- if $1;V\ \mathcal{S}_{(M,N)}\ 1;W$ and $V \rhd^{\widehat{\star}t} V'(M,N)^{\widehat{\star}}W$ then for any $(P,Q) \in (M,N)^\star$ we have $VP \rhd^{\star t} V'P(M,N)^\star WQ$. Then clause (2b) of finite-step simulation holds by Lemma 61(**).
If $V \rhd^{\widehat{\star}t} V'\ \mathcal{T}_{(M,N)}\ W$ then for any $(P,Q) \in (M,N)^\star$ we have $VP \rhd^{\star t} V'P$. Hence, $VP \Longrightarrow Y$ implies, by Lemma 61(*), that $V'P \Longrightarrow Y'$ with $Y\,\mathtt{liftd}(\rhd^{\widehat{\star}t}) \leq_{\mathtt{apx}} Y'$. By the definition of $\mathcal{T}_{(M,N)}$ and since $\stackrel{\mathrm{d}}{\Longrightarrow}$ is deterministic, from $V'P \Longrightarrow Y'$ and Lemma 61(**) we derive $Y'\mathtt{disliftd}(\mathcal{R}_{(M,N)}) \leq_{\mathtt{apx}} [\![WQ]\!]$ (see the proof of Lemma 61(**), application case, for more details). Then $Y\mathtt{disliftd}(\mathcal{R}_{(M,N)}) \leq_{\mathtt{apx}} [\![WQ]\!]$, which implies $Y\mathtt{disliftd}(\mathcal{S}_{(M,N)})[\![WQ]\!]$.

Hence, we have $M \lesssim_{\mathrm{fin}} N$.

It remains to prove that $N \lesssim_{\mathrm{fin}} M$ (then it follows from Corollary 12 and Theorem 7(3) that $M \approx N$). To do so, we first prove the following lemma.

Lemma 62. *It holds that*

(*) $P \rhd^{\star t} P'$ and $P' \Longrightarrow Y'$ imply $P \Longrightarrow Y$ with $Y\,\mathtt{liftd}(\rhd^{\widehat{\star}t})\,Y'$
(**) $P(\rhd^{\star t} (M,N)^\star)Q$ and $Q \Longrightarrow Z$ imply $P \Longrightarrow Y$ for some Y such that $Y\mathtt{disliftd}(\mathcal{R}_{(M,N)})Z$

The results in Lemma 62 are symmetric to the results in Lemma 61, and are proved analogously. For item (*), we proceed first by induction on the length k of the sequence $P = P_k \rhd .... \rhd P_1 = P'$, and then, for the inductive step, we show by induction on the

length of m that $P \rhd P'$ and $P' \Longrightarrow_m Y'$ imply $P \Longrightarrow Y$ with Y $\mathtt{liftd}(\rhd)$ Y'. Item (**) is proved by induction on the length of the reduction $Q \Longrightarrow Z$.

There are two main differences with respect to Lemma 61. In item (*), since the direction of $\rhd^{\star t}$ is now reversed, the length of the reduction $P' \Longrightarrow Y'$ is not greater than or equal to the length of the reduction from P. Secondly, in item (**), instead of considering the semantics of terms (on one side) we only relate formal sums reached via the multi-step reduction relation. This is a stronger condition, that is needed in the remainder of the proof. Finally, we prove that relation

$$\mathcal{S}' \stackrel{\mathsf{def}}{=} \{(N,M)\} \cup \{((N,M),1;W,1;V) \mid V\ \mathcal{R}_{(M,N)}\ W\} \cup \{((N,M),\emptyset,Z) \mid \text{ for any } Z\}$$

is a finite-step simulation up-to distribution and lifting. The proof proceeds as the proof for $\mathcal{S}$. The conditions on the first and the last set of $\mathcal{S}'$ are immediate; for the second set of $\mathcal{S}'$, the simulation clauses follow by deriving from Lemma 62(*) and (**) the following property: if $V\ \mathcal{R}_{(M,N)}\ W$ and $P(M,N)^\star Q$ then $WQ \Longrightarrow Z$ implies $VP \Longrightarrow Y$ and $Y\ \mathtt{disliftd}(\mathcal{R}_{(M,N)})Z$.

Proof of Theorem 52

Let $\mathcal{R}$ be a finite-step $\{\widetilde{l}\}$-simulation saturated by approximants. We first define, for any pair of terms (M,N), the preorder $\leq_{\mathsf{cce}(M,N)}$ (context closure of environments) on pairs of environment formal sums with store: $(\mathbf{Y},\mathbf{Z}) \leq_{\mathsf{cce}(M,N)} (\mathbf{Y}',\mathbf{Z}')$ if $\mathbf{Y} = \sum_i p_i; s_i; \widetilde{V}_i, \mathbf{Z} = \sum_j q_j; t_j; \widetilde{W}_j$ with $|\ \mathbf{Y}\ | = |\ \mathbf{Z}\ |$ and $\mathbf{Y}' = \sum_i p_i; s_i; \widetilde{V}'_i, \mathbf{Z}' = \sum_j q_j; t_j; \widetilde{W}'_j$ with $|\ \mathbf{Y}'\ | = |\ \mathbf{Z}'\ |$ and

- for every index r in $|\ \mathbf{Y}\ |$ there is an index r' in $|\ \mathbf{Y}'\ |$ such that $\mathbf{Y}\lfloor_r = \mathbf{Y}'\lfloor_{r'}$ and $\mathbf{Z}\lfloor_r = \mathbf{Z}'\lfloor_{r'}$;
- for every index $r' \leq |\ \mathbf{Y}'\ |$ there is a location-free value-context C such that for every i,j it holds that $(\widetilde{V}'_i)_{r'} = C[M,\widetilde{V}_i]$ and $(\widetilde{W}'_j)_{r'} = C[N,\widetilde{W}_j]$ (i.e., $(\mathbf{Y}'\lfloor_{r'}, \mathbf{Z}'\lfloor_{r'}) \in (\{(M,\widetilde{V}_i\}_i, \{N,\widetilde{W}_j\}_j)^{\widehat{\star}}$).

For any indexed relation $\mathcal{R}_{(M,N)}$ on environment formal sums, we write $\mathcal{R}^{\mathsf{cce}}_{(M,N)}$ for relation $\leq_{\mathsf{cce}(M,N)}(\mathcal{R}_{M,N})$, i.e.,

$$\mathcal{R}^{\mathsf{cce}}_{(M,N)} = \{(\mathbf{Y},\mathbf{Z}) \mid \exists \mathbf{Y}',\mathbf{Z}' \text{ such that } (\mathbf{Y}',\mathbf{Z}') \leq_{\mathsf{cce}(M,N)} (\mathbf{Y},\mathbf{Z}) \ \wedge\ \mathbf{Y}'\ \mathcal{R}_{(M,N)}\ \mathbf{Z}'\}$$

Note that we can add to a finite-step $\{\widetilde{l}\}$-simulation $\mathcal{R}$ saturated by approximants the set

$$\{((M,N),\emptyset,\mathbf{Y}) \mid \mathbf{Y} \text{ is an environment formal sum}\}$$

for any (M,N), and we still have that $\mathcal{R}$ is a finite-step $\{\widetilde{l}\}$-simulation saturated by approximants, since the presence of the empty formal sum on the left is irrelevant and trivially satisfies the simulation clauses. Hence, we assume that finite-step simulations have this property, that we refer to as $\mathcal{R}$ *is saturated by* $\emptyset$.

The following result is used in the proof of Lemma 64 (in Lemma 63 and 64, the set of locations parametrizing the relations is not relevant, hence we omit it).

Lemma 63. *Let $\mathcal{R}$ be a finite-step simulation and let*

$$\mathbf{Y} = \textstyle\sum_i p_i; s_i; \widetilde{V}_i\ \mathcal{R}^{\mathsf{cce}}_{(M,N)}\ \sum_j q_j; t_j; \widetilde{W}_j = \mathbf{Z}$$

Let $(\{\lambda x.P_i\}_i, \{\lambda x.Q_j\}_j) \in (\{M,\widetilde{V}_i\}_i, \{N,\widetilde{W}_j\}_j)^{\widehat{\star}}$ and $(\{T_i\}_i,\{U_j\}_j) \in (\{M,\widetilde{V}_i\}_i, \{N,\widetilde{W}_j\}_j)^{\widehat{\star}}$. Then one of the following holds:

- $(\{P_i\{T_i/i\}\}_i, \{Q_j\{U_j/x\}\}_j) \in (\{M, \widetilde{V}_i\}_i, \{N, \widetilde{W}_j\}_j)^\star$
- *for every* $\mathbf{W}$, $\sum_i p_i; s_i; \widetilde{V}_i, \lambda x.P_i; P_i\{T_i/i\} \Longmapsto \mathbf{W}$ *implies*

$$\mathbf{W}\,\mathtt{lift}(\mathcal{R}^{\mathtt{cce}}_{(M,N)})\,[\![\textstyle\sum_j q_j; t_j; \widetilde{W}_j, \lambda x.Q_j; Q_j\{U_j/x\}]\!]$$

Proof. We have the three cases below:

- If $\lambda x.P_i = \lambda x.C'[M, \widetilde{V}_i]$ and $\lambda x.Q_j = \lambda x.C'[N, \widetilde{W}_j]$ for some location-free context C', then there is a location-free context C'' such that

$$\textstyle\sum_i p_i; s_i; \widetilde{V}_i; P_i\{T_i/x\} = \sum_i p_i; s_i; \widetilde{V}_i; C''[M, \widetilde{V}_i]$$
$$\textstyle\sum_j q_j; t_j; \widetilde{W}_j; Q_j\{U_j/x\} = \sum_j q_j; t_j; \widetilde{W}_j; C''[N, \widetilde{W}_j]$$

  and the first item holds.
- If there is an r such that $\lambda x.P_i = (\widetilde{V}'_i)_r$ and $\lambda x.Q_j = (\widetilde{W}'_j)_r$ for some

$$\mathbf{Y}' = \textstyle\sum_i p_i; s_i; \widetilde{V}'_i\ \mathcal{R}_{(M,N)}\ \sum_j q_j; t_j; \widetilde{W}'_j = \mathbf{Z}'$$

  such that $(\mathbf{Y}', \mathbf{Z}') \leq_{\mathtt{cce}(M,N)} (\mathbf{Y}, \mathbf{Z})$, then it follows from the fact that $\mathcal{R}$ is a finite-step simulation that $\sum_i p_i; s_i; \widetilde{V}'_i; P_i\{T_i/x\} \Longmapsto \mathbf{W}'$ implies

$$\mathbf{W}'\,\mathtt{lift}(\mathcal{R}_{(M,N)})\,[\![\textstyle\sum_j q_j; t_j; \widetilde{W}'_j; Q_j\{U_j/x\}]\!].$$

  Since $\sum_i p_i; s_i; \widetilde{V}_i, \lambda x.P_i; P_i\{T_i/i\} \Longmapsto \mathbf{W}$ implies

  $(\mathbf{W}', [\![\sum_j q_j; t_j; \widetilde{W}'_j; Q_j\{U_j/x\}]\!]) \leq_{\mathtt{cce}(M,N)} (\mathbf{W}, [\![\sum_j q_j; t_j; \widetilde{W}_j, \lambda x.Q_j; Q_j\{U_j/x\}]\!])$,

  we derive

$$\mathbf{W}\,\mathtt{lift}(\mathcal{R}^{\mathtt{cce}}_{(M,N)})\,[\![\textstyle\sum_j q_j; t_j; \widetilde{W}_j, \lambda x.Q_j; Q_j\{U_j/x\}]\!]$$

- If $\lambda x.P_i = M$ and $\lambda x.Q_j = N$ then M and N are values and by clause (2g) applied to $\mathcal{R}_{(M,N)}$,

$$\textstyle\sum_i p_i; s_i; \widetilde{V}'_i, M\ \mathcal{R}_{(M,N)}\ \sum_j q_j; t_j; \widetilde{W}'_j, N$$

  for some $(\sum_i p_i; s_i; \widetilde{V}'_i, \sum_j q_j; t_j; \widetilde{W}'_j) \leq_{\mathtt{cce}(M,N)} (\mathbf{Y}, \mathbf{Z})$. By clause (2b) applied to $\mathcal{R}_{(M,N)}$, from $\sum_i p_i; s_i; \widetilde{V}'_i, M; P_i\{T_i/x\} \Longmapsto \mathbf{W}'$ we derive

$$\mathbf{W}'\,\mathtt{lift}(\mathcal{R}_{(M,N)})\,[\![\textstyle\sum_j q_j; t_j; \widetilde{W}'_j, N; Q_j\{U_j/x\}]\!].$$

  By the assumptions $\lambda x.P_i = M$ and $\lambda x.Q_j = N$ there are already columns in the dynamic environments of $\mathbf{Y}$ and $\mathbf{Z}$ composed by M and N respectively; thus

  $(\mathbf{W}', [\![\sum_j q_j; t_j; \widetilde{W}'_j, N; Q_j\{U_j/x\}]\!]) \leq_{\mathtt{cce}(M,N)} \geq_{\mathtt{env}} (\mathbf{W}, [\![\sum_j q_j; t_j; \widetilde{W}_j; Q_j\{U_j/x\}]\!])$,

  from which the result follows.

□

In what follows, we sometimes use relations $\leq_{\mathtt{lift}}$ and $\leq_{\mathtt{lift}}\leq_{\mathtt{env}}$, defined as follows:

- $(\mathbf{Y}, \mathbf{Z}) \leq_{\mathtt{lift}} \{(\mathbf{Y}_g, \mathbf{Z}_g)\}_g$ if there are probability values p_g such that $\mathbf{Y} = \sum_g p_g \cdot \mathbf{Y}_g$ and $\mathbf{Z} = \sum_g p_g \cdot \mathbf{Z}_g$;
- $(\mathbf{Y}, \mathbf{Z}) \leq_{\mathtt{lift}}\leq_{\mathtt{env}} \{(\mathbf{Y}_g, \mathbf{Z}_g)\}_g$ if there are probability values p_g such that $\mathbf{Y} = \sum_g p_g \cdot \mathbf{Y}'_g$ and $\mathbf{Z} = \sum_g p_g \cdot \mathbf{Z}'_g$ with $(\mathbf{Y}'_g, \mathbf{Z}'_g) \leq_{\mathtt{env}} (\mathbf{Y}_g, \mathbf{Z}_g)$ for every g.

The notation, as for relations $\leq_{\mathtt{env}}$ and $\leq_{\mathtt{cce}(M,N)}$, is extended to environment formal sums with running terms by requiring the running term to be the same everywhere.

LEMMA 64. *Suppose that $\mathcal{R}_{(M,N)}$ is a finite-step simulation saturated by approximants (only defined on formal sums). For any location-free context C if $\mathbf{Y}\ \mathcal{R}^{\mathsf{cce}}_{(M,N)}\ \mathbf{Z}$ and $\mathbf{Y}; C[M,\mathbf{Y}] \Longrightarrow \mathbf{W}$ then $\mathbf{W}\,\mathtt{lift}(\geq_{\mathtt{env}}(\mathcal{R}^{\mathsf{cce}}_{(M,N)}))\,[\![\mathbf{Z}; C[N,\mathbf{Z}]]\!]$.*

PROOF. Suppose that $\mathbf{Y} = \sum_i p_i; s_i; \widetilde{V}_i\ \mathcal{R}^{\mathsf{cce}}_{(M,N)}\ \sum_j q_j; t_j; \widetilde{W}_j = \mathbf{Z}$, with $\mathbf{Y}' = \sum_i p_i; s_i; \widetilde{V}'_i$ and $\mathbf{Z}' = \sum_j q_j; t_j; \widetilde{W}'_j$ related by $\mathcal{R}_{(M,N)}$ and such that $(\mathbf{Y}', \mathbf{Z}') \leq_{\mathsf{cce}(M,N)} (\mathbf{Y}, \mathbf{Z})$.

We prove by induction on n and then by induction on the structure of C that

$$\textstyle\sum_i p_i; s_i; \widetilde{V}_i; C[M, \widetilde{V}_i] \Longrightarrow_n \mathbf{W} \text{ implies } \mathbf{W}\,\mathtt{lift}(\geq_{\mathtt{env}}(\mathcal{R}^{\mathsf{cce}}_{(M,N)}))\,[\![\textstyle\sum_j q_j; t_j; \widetilde{W}_j; C[N, \widetilde{W}_j]]\!].$$

(In the proof we sometimes assume that $\widetilde{V}_i = \widetilde{V}'_i, \widetilde{V}''_i$ and $\widetilde{W}_j = \widetilde{W}'_j, \widetilde{W}''_j$. This does not affect the results since the context closure of environments allows us to permute the columns in the formal sums.)

If $\mathbf{W} = \emptyset$ then the result follows by the fact that $\mathcal{R}$ is saturated by $\emptyset$. Suppose that $\mathbf{W} \neq \emptyset$ and $n = 0$. We have one of the following cases:

- $C = [\cdot]_1$ and M is a value. Then
$$\textstyle\sum_i p_i; \widetilde{V}'_i \cdot (1; s_i; M)\,\mathtt{lift}(\mathcal{R}_{(M,N)})\ \sum_j q_j; \widetilde{W}'_j \cdot [\![\langle t_j\ ;\ N\rangle]\!]$$
and we derive
$$\textstyle\sum_i p_i; \widetilde{V}_i \cdot (1; s_i; M)(\,\mathtt{lift}(\mathcal{R}^{\mathsf{cce}}_{(M,N)})\,)\sum_j q_j; \widetilde{W}_j \cdot [\![\langle t_j\ ;\ N\rangle]\!]\,.$$
- $C \neq [\cdot]_1$ and for every i, j, $C[M, \widetilde{V}_i]$ and $C[N, \widetilde{W}_j]$ are values. Then it follows from the definition of $\leq_{\mathsf{cce}(M,N)}$ that

$$\mathbf{W} = \textstyle\sum_i p_i; s_i; \widetilde{V}_i, C[M, \widetilde{V}_i]\ \mathcal{R}^{\mathsf{cce}}_{(M,N)}\ \sum_j q_j; t_j; \widetilde{W}_j, C[N, \widetilde{W}_j] = [\![\sum_j q_j; t_j; \widetilde{W}_j, C[N, \widetilde{W}_j]]\!]\,.$$

Suppose now that $\sum_i p_i; s_i; \widetilde{V}_i; C[M, \widetilde{V}_i] \Longrightarrow_{n+1} \mathbf{W}$, with $\mathbf{W} \neq \emptyset$. We have the following cases:

- $C = [\cdot]_1$ and M is not a value. Then the result follows analogously to the $n = 0$ case.
- $C = C_1 \oplus C_2$. In this case $\sum_i p_i; s_i; \widetilde{V}_i; (C_1 \oplus C_2)[M, \widetilde{V}_i] \Longrightarrow \mathbf{W}$ implies

$$\textstyle\sum_i p_i; s_i; \widetilde{V}_i; (C_1 \oplus C_2)[M, \widetilde{V}_i] \longrightarrow \frac{1}{2} \cdot \sum_i p_i; s_i; \widetilde{V}_i; C_1[M, \widetilde{V}_i] + \frac{1}{2} \sum_i p_i; s_i; \widetilde{V}_i; C_2[M, \widetilde{V}_i]$$

  and $\mathbf{W} = \frac{1}{2} \cdot \mathbf{W}_1 + \frac{1}{2} \cdot \mathbf{W}_2$, with $\sum_i p_i; s_i; \widetilde{V}_i; C_h[M, \widetilde{V}_i] \Longrightarrow_{n_h} \mathbf{W}_h$, for $h = 1, 2$ and $n_h \leq n$. Since
$$\begin{aligned}&[\![\textstyle\sum_j q_j; t_j; \widetilde{W}_j; (C_1 \oplus C_2)[N, \widetilde{W}_j]]\!] = \\ &\tfrac{1}{2} \cdot [\![\textstyle\sum_j q_j; t_j; \widetilde{W}_j; C_1[N, \widetilde{W}_j]]\!] + \tfrac{1}{2} \cdot [\![\textstyle\sum_j q_j; t_j; \widetilde{W}_j; C_1[N, \widetilde{W}_j]]\!]\end{aligned}$$
  we can apply the inductive hypothesis to n_1 and n_2 and derive
$$\mathbf{W}\,\mathtt{lift}(\,\mathtt{lift}(\geq_{\mathtt{env}}(\mathcal{R}^{\mathsf{cce}}_{(M,N)}))\,)\,[\![\textstyle\sum_j q_j; t_j; \widetilde{W}_j; (C_1 \oplus C_2)[N, \widetilde{W}_j]]\!]$$
  The result follows from $\mathtt{lift}(\,\mathtt{lift}(\mathcal{R})\,) = \mathtt{lift}(\mathcal{R})$.
- $C = C_1 C_2$.
  Suppose that there exists some i such that $C_1[M, \widetilde{V}_i]$ is not a value, and such that $C_1[M, \widetilde{V}_i]$ performs some small steps in the reduction leading to $\mathbf{W}$. Hence, there is a set $I' \subseteq I$ such that

$$\textstyle\sum_{i\in I} p_i; s_i; \widetilde{V}_i; C_1[M, \widetilde{V}_i]C_2[M, \widetilde{V}_i] \Longrightarrow_{n_1} \sum_{i\in I' \subseteq I,k} p_{i,k}; s_{i,k}; \widetilde{V}_i; V_{i,k}C_2[M, \widetilde{V}_i] \Longrightarrow_{n_2} \mathbf{W}$$

with $n_1 > 1$ and

$$\sum_{i\in I} p_i; s_i; \widetilde{V}_i; C_1[M, \widetilde{V}_i] \Longrightarrow_{n_1} \sum_{i\in I',k} p_{i,k}; s_{i,k}; \widetilde{V}_i, V_{i,k}\,.$$

We have

$$[\![\textstyle\sum_{j\in J} q_j; t_j; \widetilde{W}_j; C_1[N, \widetilde{W}_j]C_2[N, \widetilde{W}_j]]\!] = [\![\sum_{j\in J'\subseteq J,h} q_{j,h}; t_{j,h}; \widetilde{W}_j; W_{j,h}C_2[N, \widetilde{W}_j]]\!]$$

for $[\![\sum_{j\in J} q_j; t_j; \widetilde{W}_j; C_1[N, \widetilde{W}_j]]\!] = \sum_{j\in J',h} q_{j,h}; t_{j,h}; \widetilde{W}_j, W_{j,h}$. Then we can apply the inductive hypothesis to n_1 and derive

$$\textstyle\sum_{i\in I',k} p_{i,k}; s_{i,k}; \widetilde{V}_i, V_{i,k}\, \mathtt{lift}(\geq_{\mathtt{env}} (\mathcal{R}^{\mathsf{cce}}_{(M,N)}))\ \sum_{j\in J',h} q_{j,h}; t_{j,h}; \widetilde{W}_j, W_{j,h}$$

which is equivalent to saying

$$(\textstyle\sum_{i\in I',k} p_{i,k}; s_{i,k}; \widetilde{V}_i, V_{i,k}, \sum_{j\in J',h} q_{j,h}; t_{j,h}; \widetilde{W}_j, W_{j,h}) \leq_{\mathtt{lift}}\leq_{\mathtt{env}} \{(\mathbf{Y}_g, \mathbf{Z}_g)\}_g \subseteq \mathcal{R}^{\mathsf{cce}}_{(M,N)}$$

with $\mathbf{Y}_g = \sum_{i,k\in I_g} p'_{i,k}; s_{i,k}; \widetilde{V}_{i,k}$, and $\mathbf{Z}_g = \sum_{j,h\in J_g} q'_{j,h}; t_{j,h}; \widetilde{W}_{j,h}$ such that for every g there is a context C_g such that for all $i, k \in I_g$ and for all $j, h \in J_g$ we have $V_{i,k}C_2[M, \widetilde{V}_i] = C_g[M, \widetilde{V}_{i,k}]$ and $W_{j,h}C_2[N, \widetilde{W}_j] = C_g[N, \widetilde{W}_{j,h}]$.
If $\sum_{i\in I'\subseteq I,k} p_{i,k}; s_{i,k}; \widetilde{V}_i; V_{i,k}C_2[M, \widetilde{V}_i] \Longrightarrow_{n_2} \mathbf{W}$ then for every g there is a $\mathbf{W}_g$ such that $\sum_{i,k\in I_g} p'_{i,k}; s_{i,k}; \widetilde{V}_{i,k}; C_g[M, \widetilde{V}_{i,k}] \Longrightarrow_{n'_g} \mathbf{W}_g$ and $\sum_g n'_g = n_2$ and

$$\begin{aligned}&(\mathbf{W}, [\![\textstyle\sum_{j\in J} q_j; t_j; \widetilde{W}_j; C_1[N, \widetilde{W}_j]C_2[N, \widetilde{W}_j]]\!])\\ &\leq_{\mathtt{lift}}\leq_{\mathtt{env}} \{(\mathbf{W}_g, [\![\textstyle\sum_{j,h\in J_g} q'_{j,h}; t_{j,h}; \widetilde{W}_{j,h}; C_g[N, \widetilde{W}_{j,h}]]\!])\}_g\end{aligned}$$

By the inductive hypothesis on each of the n'_g we derive

$$\{(\mathbf{W}_g, [\![\textstyle\sum_{j,h\in J_g} q_{j,h}; t_{j,h}; \widetilde{W}_{j,h}; C_g[N, \widetilde{W}_{j,h}]]\!])\}_g \subseteq \mathtt{lift}(\geq_{\mathtt{env}} (\mathcal{R}^{\mathsf{cce}}_{(M,N)}))$$

from which the result follows by $\mathtt{lift}(\geq_{\mathtt{env}} (\mathtt{lift}(\geq_{\mathtt{env}} (\mathcal{R})))) = \mathtt{lift}(\geq_{\mathtt{env}} (\mathcal{R}))$.
We now consider the case when no $C_1[M, \widetilde{V}_i]$ contributes to the multi-step reduction to $\mathbf{W}$. If there exist some i such that $C_2[M, \widetilde{V}_i]$ contributes to the multi-step reduction to $\mathbf{W}$, then we can derive the result using the same reasoning as above.
Otherwise, there is a set $I' \subseteq I$ such that $C_1[M, \widetilde{V}_i]$ and $C_2[M, \widetilde{V}_i]$ are values for every $i \in I'$, and

$$\begin{aligned}\textstyle\sum_{i\in I'} p_i; s_i; \widetilde{V}_i; C_1[M, \widetilde{V}_i]C_2[M, \widetilde{V}_i] &= \textstyle\sum_{i\in I'} p_i; s_i; \widetilde{V}_i; (\lambda x.P_i)C_2[M, \widetilde{V}_i]\\ &\Longrightarrow_{n_1} \textstyle\sum_{i\in I'} p_i; s_i; \widetilde{V}_i; P_i\{C_2[M, \widetilde{V}_i]/x\}\\ &\Longrightarrow_{n_2} \mathbf{W}\end{aligned}$$

with $n_1 \leq n + 1$ and $n_2 \leq n$.
Suppose that $C_1[N, \widetilde{W}_j] = \lambda x.Q_j$ is a value, for all j, and $C_2[N, \widetilde{W}_j]$ is a value. Then,

$$\begin{aligned}&[\![\textstyle\sum_j q_j; t_j; \widetilde{W}_j; C_1[N, \widetilde{W}_j]C_2[N, \widetilde{W}_j]]\!] =\\ &[\![\textstyle\sum_j q_j; t_j; \widetilde{W}_j; \lambda x.Q_jC_2[N, \widetilde{W}_j]]\!] =\\ &[\![\textstyle\sum_j q_j; t_j; \widetilde{W}_j; Q_j\{C_2[N, \widetilde{W}_j]/x\}]\!]\end{aligned}$$

Since $\mathcal{R}$ is saturated by approximants,

$$\textstyle\sum_{i\in I'} p_i; s_i; \widetilde{V}_i\ \mathcal{R}^{\mathsf{cce}}_{(M,N)}\ \sum_j q_j; t_j; \widetilde{W}_j$$

which implies, by Lemma 63, that one of the following holds:

– there is a context C' such that

$$\sum_{i\in I'} p_i; s_i; \widetilde{V}_i; P_i\{C_2[M,\widetilde{V}_i]/x\} = \sum_{i\in I'} p_i; s_i; \widetilde{V}_i; C'[M,\widetilde{V}_i]$$

and

$$\sum_j q_j; t_j; \widetilde{W}_j; Q_j\{C_2[N,\widetilde{W}_j]/x\} = \sum_j q_j; t_j; \widetilde{W}_j; C'[N,\widetilde{W}_j].$$

Then we can apply the inductive hypothesis on the reduction

$$\sum_{i\in I'} p_i; s_i; \widetilde{V}_i; C'[M,\widetilde{V}_i] \Longrightarrow_{n_2} \mathbf{W}$$

and derive the result.

– for any $\mathbf{W}'$, if $\sum_{i\in I'} p_i; s_i; \widetilde{V}_i, \lambda x.P_i; P_i\{C_2[M,\widetilde{V}_i]/x\} \Longrightarrow \mathbf{W}'$ then

$$\mathbf{W}'\ \mathtt{lift}(\mathcal{R}^{\mathsf{cce}}_{(M,N)})\ [\![\textstyle\sum_j q_j; t_j; \widetilde{W}_j, \lambda x.Q_j; Q_j\{C_2[N,\widetilde{W}_j]/x\}]\!].$$

Then we have

$$\begin{aligned}&(\mathbf{W}, [\![\textstyle\sum_j q_j; t_j; \widetilde{W}_j; Q_j\{C_2[N,\widetilde{W}_j]/x\}]\!])\\ &\leq_{\mathtt{env}} (\mathbf{W}', [\![\textstyle\sum_j q_j; t_j; \widetilde{W}_j, \lambda x.Q_j; Q_j\{C_2[N,\widetilde{W}_j]/x\}]\!])\end{aligned}$$

and the result follows by $\geq_{\mathtt{env}}\ \mathtt{lift}(\mathcal{R}) \subseteq \mathtt{lift}(\geq_{\mathtt{env}}(\mathcal{R}))$.

It remains to consider the case when $C_1[N,\widetilde{W}_j]$ or $C_2[N,\widetilde{W}_j]$ are not values for some j. If $C_1[N,\widetilde{W}_j]$ is not a value for some j then $C_1 = [\cdot]_1$ and N is not a value. Suppose $C_2[N,\widetilde{W}_j]$ is a value for all j. Then we have

$$\begin{aligned}&[\![\textstyle\sum_{j\in J} q_j; t_j; \widetilde{W}_j; NC_2[N,\widetilde{W}_j]]\!] =\\ &[\![\textstyle\sum_{j\in J'\subseteq J,h} q_{j,h}; t_{j,h}; \widetilde{W}_j; \lambda x.Q_{j,h}C_2[N,\widetilde{W}_j]]\!] =\\ &[\![\textstyle\sum_{j\in J'\subseteq J,h} q_{j,h}; t_{j,h}; \widetilde{W}_j; Q_{j,h}\{C_2[N,\widetilde{W}_j]/x\}]\!]\end{aligned}$$

for $[\![\sum_{j\in J} q_j; t_j; \widetilde{W}_j; N]\!] = \sum_{j\in J',h} q_{j,h}; t_{j,h}; \widetilde{W}_j, \lambda x.Q_{j,h}$. Since $\mathcal{R}$ is a finite-step simulation saturated by approximants, by clause (2g) we derive

$$\textstyle\sum_{i\in I'} p_i; s_i; \widetilde{V}'_i, \lambda x.P_i\ \mathtt{lift}(\mathcal{R}_{(M,N)})\ \sum_{j\in J',h} q_{j,h}; t_{j,h}; \widetilde{W}'_j, \lambda x.Q_{j,h}$$

which implies

$$\textstyle\sum_{i\in I'} p_i; s_i; \widetilde{V}_i, \lambda x.P_i\ \mathtt{lift}(\mathcal{R}^{\mathsf{cce}}_{(M,N)})\ \sum_{j\in J',h} q_{j,h}; t_{j,h}; \widetilde{W}_j, \lambda x.Q_{j,h}\,.$$

Then

$$(\textstyle\sum_{i\in I'} p_i; s_i; \widetilde{V}_i, \lambda x.P_i, \sum_{j\in J',h} q_{j,h}; t_{j,h}; \widetilde{W}_j, \lambda x.Q_{j,h}) \leq_{\mathtt{lift}} \{(\mathbf{Y}_g, \mathbf{Z}_g)\}_g \subseteq \mathcal{R}^{\mathsf{cce}}_{(M,N)}$$

with $\mathbf{Y}_g = \sum_{i\in I_g} p'_i; s_i; \widetilde{V}_i, \lambda x.P_i$, and $\mathbf{Z}_g = \sum_{j,h\in J_g} q'_{j,h}; t_{j,h}; \widetilde{W}_j, \lambda x.Q_{j,h}$ such that for every g we have

$$\textstyle\sum_{i\in I_g} p'_i; s_i; \widetilde{V}_i, \lambda x.P_i; \lambda x.P_iC_2[M,\widetilde{V}_i]$$

and

$$\textstyle\sum_{j,h\in J_g} q'_{j,h}; t_{j,h}; \widetilde{W}_j, \lambda x.Q_{j,h}; \lambda x.Q_{j,h}C_2[N,\widetilde{W}_j]$$

satisfy the premises of Lemma 63. Then we can proceed as in the previous case and derive that $\sum_{i\in I_g} p'_i; s_i; \widetilde{V}_i, \lambda x.P_i; P_i\{C_2[M,\widetilde{V}_i]/x\} \Longrightarrow \mathbf{W}_g$ implies

$$\mathbf{W}_g\ \mathtt{lift}(\geq_{\mathtt{env}}(\mathcal{R}^{\mathsf{cce}}_{(M,N)}))\ [\![\textstyle\sum_{j,h\in J_g} q'_{j,h}; t_{j,h}; \widetilde{W}_j, \lambda x.Q_{j,h}; Q_{j,h}\{C_2[N,\widetilde{W}_j]/x\}]\!]\,,$$

from which the result follows.

Finally, if there are some j such that $C_2[N, \widetilde{W_j}]$ is not a value then we can proceed as in the previous case, exploiting clause (2g) on $\mathcal{R}_{(M,N)}$ in order to evaluate N in argument position as well.

- case $C = !C'$.
  The interesting case is when $C'[M, \widetilde{V_i}]$ does not contribute to the reduction, for every i (otherwise, we can apply the inductive hypothesis analogously to the application case), i.e., there is a subset $I' \subseteq I$ such that

$$\sum_{i\in I'} p_i; s_i; \widetilde{V_i}; !C'[M, \widetilde{V_i}] = \sum_{i\in I'} p_i; s_i; \widetilde{V_i}; !l_i \Longrightarrow_{n+1} \sum_{i\in I'} p_i; s_i; \widetilde{V_i}, s_i(l_i)$$

  Since $\mathcal{R}$ is saturated by approximants, $\sum_{i\in I'} p_i; s_i; \widetilde{V_i'}\ \mathcal{R}_{(M,N)}\ \sum_j q_j; t_j; \widetilde{W_j'}$.
  Since contexts are location-free, there are two cases:
  - $C'[M, \widetilde{V_i}] \in \widetilde{V_i'}$. Then $C'[N, \widetilde{W_j}] = l_j' \in \widetilde{W_j'}$ and by clause (2c) on $\mathcal{R}$ we have

$$\sum_{i\in I'} p_i; s_i; \widetilde{V_i'}, s_i(l_i)\, \mathtt{lift}(\mathcal{R}_{(M,N)})\ \sum_j q_j; t_j; \widetilde{W_j'}, t_i(l_j')$$

    and the result follows from

$$[\![\sum_j q_j; t_j; \widetilde{W_j'}; !l_j']\!] = \sum_j q_j; t_j; \widetilde{W_j'}, t_i(l_j')$$

  - $C = [\cdot]_1$ and $M = l$.
    Suppose that $N = l'$. Then by clause (2g) we have

$$\sum_{i\in I'} p_i; s_i; \widetilde{V_i'}, M\ \mathcal{R}_{(M,N)}\ \sum_j q_j; t_j; \widetilde{W_j'}, N$$

    and by clause (2c)

$$\sum_{i\in I'} p_i; s_i; \widetilde{V_i'}, M, s_i(l_i)\, \mathtt{lift}(\mathcal{R}_{(M,N)})\ \sum_j q_j; t_j; \widetilde{W_j'}, N, t_j(l_j')$$

    Then

$$\sum_{i\in I'} p_i; s_i; \widetilde{V_i}, s_i(l_i)\, \mathtt{lift}(\geq_{\mathtt{env}} (\mathcal{R}^{\mathtt{cce}}_{(M,N)}))\ \sum_j q_j; t_j; \widetilde{W_j}, t_j(l_j')$$

    The case when N is not a value follows analogously.

- $C = C_1 := C_2$.
  Again, the interesting case is when for every i neither $C_1[M, \widetilde{V_i}]$ nor $C_2[M, \widetilde{V_i}]$ contributes to the reduction, i.e., there is a subset $I \subseteq I'$ such that

$$\begin{aligned}&\textstyle\sum_{i\in I'} p_i; s_i; \widetilde{V_i}; C_1[M, \widetilde{V_i}] := C_2[M, \widetilde{V_i}]\\ &= \textstyle\sum_{i\in I'} p_i; s_i; \widetilde{V_i}; l_i := T_i\\ &\Longrightarrow_{n+1} \textstyle\sum_{i\in I'} p_i; s_i[l_i \to T_i]; \widetilde{V_i}, \mathtt{unit}\end{aligned}$$

  As above, since $\mathcal{R}$ is saturated by approximants, $\sum_{i\in I'} p_i; s_i; \widetilde{V_i'}\ \mathcal{R}_{(M,N)}\ \sum_j q_j; t_j; \widetilde{W_j'}$ and, since contexts are location-free, there are two cases:
  - $C_1[M, \widetilde{V_i}] \in \widetilde{V_i'}$. Then $C_1[N, \widetilde{W_j}] = l_j' \in \widetilde{W_j'}$. Suppose that $C_2[N, \widetilde{W_j}] = U_j$ is a value for every j. Then $(\{T_i\}_i, \{U_j\}_j) \in (\{M, \widetilde{V_i'}\}_i, \{N, \widetilde{W_j'}\}_j)^{\widehat{\star}}$ and by clause (2c) on $\mathcal{R}$ we have

$$\sum_{i\in I'} p_i; s_i[l_i \to T_i]; \widetilde{V_i'}\, \mathtt{lift}(\mathcal{R}_{(M,N)})\ \sum_j q_j; t_j[l_j' \to U_j]; \widetilde{W_j'}$$

    and the result follows.
    If $C_2[N, \widetilde{W_j}]$ is not a value for some j then $C_2[M, \widetilde{V_j}] = M$ and $C_2[N, \widetilde{W_j}] = N$. Then we derive from clause (2g) that

$$\sum_{i\in I'} p_i; s_i; \widetilde{V_i'}, M\ \mathcal{R}_{(M,N)}\ [\![\sum_j q_j; t_j; \widetilde{W_j'}; N]\!] = \sum_{j,h} q_{j,h}; t_{j,h}; \widetilde{W_j'}; U_{j,h}$$

and we can derive, as in the previous case, that

$$\textstyle\sum_{i\in I'} p_i; s_i[l_i \to M]; \widetilde{V}_i', M \,\mathtt{lift}(\mathcal{R}_{(M,N)})\, \sum_{j,h} q_j; t_j[l'_j \to U_{j,h}]; \widetilde{W}'_j, U_{j,h}$$

from which the result follows.

– $C = [\cdot]_1$ and $M = l$.
Suppose that $N = l'$ and $C_2[N, \widetilde{W}_j] = U_j$ is a value for every j. Then $(\{T_i\}_i, \{U_j\}_j) \in (\{M, \widetilde{V}_i'\}_i, \{N, \widetilde{W}'_j\}_j)^{\widehat{\star}}$. Then by clause (2g) we have

$$\textstyle\sum_{i\in I'} p_i; s_i; \widetilde{V}_i', M \;\mathcal{R}_{(M,N)}\; \sum_j q_j; t_j; \widetilde{W}'_j, N$$

and by clause (2c) on $\mathcal{R}$ we have

$$\textstyle\sum_{i\in I'} p_i; s_i[l \to T_i]; \widetilde{V}_i', M \,\mathtt{lift}(\mathcal{R}_{(M,N)})\, \sum_j q_j; t_j[l' \to U_j]; \widetilde{W}'_j, N$$

and the result follows.
If N or $C_2[N, \widetilde{W}_j]$ are not values then we proceed analogously to the previous cases, by proving that we can add to the environment the values they evaluate to while staying in relation $\mathcal{R}_{(M,N)}$.

- $C = (\boldsymbol{\nu}\, x{:=}C_1)C_2$, with C_2 a context with free variable x.
We consider the case when

$$\begin{array}{l}\textstyle\sum_{i\in I'} p_i; s_i; \widetilde{V}_i; (\boldsymbol{\nu}\, x{:=}C_1[M, \widetilde{V}_i])C_2[M, \widetilde{V}_i] \\ = \textstyle\sum_{i\in I'} p_i; s_i; \widetilde{V}_i; (\boldsymbol{\nu}\, x{:=}T_i)C_2[M, \widetilde{V}_i] \\ \Longrightarrow_{n_1} \textstyle\sum_{i\in I'} p_i; s_i[l_i \to T_i]; \widetilde{V}_i; C_2[M, \widetilde{V}_i]\{^{l_i}\!/_x\} \\ \longmapsto_{n_2} \mathbf{W}\end{array}$$

and $C_1[M, \widetilde{V}_i] = U_j$ is a value for every j, thus

$$[\![\textstyle\sum_j q_j; t_j; \widetilde{W}_j; (\boldsymbol{\nu}\, x{:=}C_1[N, \widetilde{W}_j])C_2[N, \widetilde{W}_j]]\!] = [\![\textstyle\sum_j q_j; t_j[k_j \to U_j]; \widetilde{W}_j; C_2[N, \widetilde{W}_j]\{^{k_j}\!/_x\}]\!]$$

for $(\{T_i\}_i, \{U_j\}_j) \in (\{M, \widetilde{V}_i'\}_i, \{N, \widetilde{W}'_j\}_j)^{\widehat{\star}}$ and for locations $(\{l_i\}_i, \{k_j\}_j)$ which are $(\{s_i\}_i, \{t_j\}_j)$-fresh. By clauses (2d) and (2c) we have

$$\textstyle\sum_{i\in I'} p_i; s_i[l_i \to T_i]; \widetilde{V}_i', l_i \;\mathcal{R}_{(M,N)}\; \sum_j q_j; t_j[k_j \to U_j]; \widetilde{W}'_j, k_j$$

which implies

$$\textstyle\sum_{i\in I'} p_i; s_i[l_i \to T_i]; \widetilde{V}_i, l_i; C_2[M, \widetilde{V}_i]\{^{l_i}\!/_x\}\; \mathcal{R}^{\mathsf{cce}}_{(M,N)}\; \sum_j q_j; t_j[k_j \to U_j]; \widetilde{W}_j, k_j; C_2[N, \widetilde{W}_j]\{^{k_j}\!/_x\}\,.$$

Then the result follows from the inductive hypothesis on n_2.

- case $C = \mathtt{if}\ C_1\ \mathtt{then}\ C_2\ \mathtt{else}\ C_3$ and case $C = \mathtt{op}(C_1, ..., C_m)$.
The result follows from the fact that $\mathcal{R}^{\mathsf{cce}}_{(M,N)}$ satisfies clause (2e) for constants and then from the inductive hypothesis.
- $C = (C_1, ..., C_m)$.
Since the multi-step reduction to $\mathbf{W}$ has length strictly greater than one, there is some z such that $1 \le z \le m$ and some i such that $C_z[M, \widetilde{V}_i]$ contributes to the multi-step reduction to $\mathbf{W}$. Then we can apply the inductive hypothesis on the contexts to C_z and derive that:

$$\textstyle\sum_{i\in I'} p_i; s_i; \widetilde{V}_i; C_z[M, \widetilde{V}_i] \Longrightarrow_{n'} \sum_{i\in I',k} p_{i,k}; s_{i,k}; \widetilde{V}_i, V_{i,k}$$

for $n' \le n + 1$ implies

$$\begin{array}{l}\textstyle\sum_{i\in I',k} p_{i,k}; s_{i,k}; \widetilde{V}_i, V_{i,k}\ \mathtt{lift}(\geq_{\mathsf{env}}(\mathcal{R}^{\mathsf{cce}}_{(M,N)}))\ \sum_{j\in J',h} q_{j,h}; t_{j,h}; \widetilde{W}_j, W_{j,h} \\ \qquad\qquad = [\![\textstyle\sum_j q_j; t_j; \widetilde{W}_j; C_z[N, \widetilde{W}_j]]\!]\end{array}$$

i.e.,

$$(\sum_{i\in I',k} p_{i,k}; s_{i,k}; \widetilde{V}_i, V_{i,k}, \sum_{j\in J',h} q_{j,h}; t_{j,h}; \widetilde{W}_j, W_{j,h}) \leq_{\mathtt{lift}} \leq_{\mathtt{env}} \{(\mathbf{Y}_g, \mathbf{Z}_g)\}_g \subseteq \mathcal{R}^{\mathtt{cce}}_{(M,N)}$$

with $\mathbf{Y}_g = \sum_{i,k\in I_g} p'_{i,k}; s_{i,k}; \widetilde{V}_{i,k}$, and $\mathbf{Z}_g = \sum_{j,h\in J_g} q'_{j,h}; t_{j,h}; \widetilde{W}_{j,h}$ and for every g there is a context C_g such that for every $i, k \in I_g$ and $j, h \in J_g$:

$$C[M, \widetilde{V}_i]\{^{V_{i,k}}/_{C_z[M,\widetilde{V}_i]}\} = C_g[M, \widetilde{V}_{i,k}]$$
$$C[N, \widetilde{W}_j]\{^{W_{j,h}}/_{C_z[N,\widetilde{W}_j]}\} = C_g[N, \widetilde{W}_{j,h}]$$

(where the substitution only concerns the specific instance of $C_z[M, \widetilde{V}_i]$ in $C_z[M, \widetilde{V}_i]$ that occurs as the z-th element of the tuple $C[M, \widetilde{V}_i]$, and the same for $C_z[N, \widetilde{W}_i]$). Finally, we derive the result by applying the inductive hypothesis on the number of reductions from

$$\sum_{i,k\in I_g} p'_{i,k}; s_{i,k}; \widetilde{V}_{i,k}; C_g[M, \widetilde{V}_{i,k}]\,.$$

- $C = \#_z(C')$.
  We consider the case when $C'[M, \widetilde{V}_i] = (V_{i,1}, ..., V_{i,m})$ is a value for every i and

  $$\sum_{i\in I} p_i; s_i; \widetilde{V}_i; \#_z(C'[M, \widetilde{V}_i]) \longrightarrow \mathbf{W} = \sum_{i\in I} p_{i,k}; s_{i,k}; \widetilde{V}_i, V_{i,a}$$

  We have three cases:
  - $C'[M, \widetilde{V}_i] = (V_{i,1}, ..., V_{i,m}) \in \widetilde{V}'_i$.
    Then $C'[N, \widetilde{W}_j] = (U_{j,1}, ..., U_{j,m}) \in \widetilde{W}'_j$ and we have, by clause (2f) on $\mathcal{R}$,

    $$\sum_i p_i; s_i; \widetilde{V}'_i, V_{i,1}, ..., V_{i,m}\ \mathtt{lift}(\mathcal{R}_{(M,N)})\ \sum_j q_j; t_j; \widetilde{W}'_j, U_{j,1}, ..., U_{j,m}$$

    Therefore,

    $(\mathbf{W}, [\![\sum_j q_j; t_j; \widetilde{W}_j, \#_z(U_{j,1}, ..., U_{j,m})]\!])$
    $\leq_{\mathtt{env}} (\sum_i p_i; s_i; \widetilde{V}_i, V_{i,1}, ..., V_{i,m}, \sum_j q_j; t_j; \widetilde{W}_j, U_{j,1}, ..., U_{j,m}) \in \mathtt{lift}(\mathcal{R}^{\mathtt{cce}}_{(M,N)})$
  - $C'[M, \widetilde{V}_i] = (C_1[M, \widetilde{V}_i], ..., C_m[M, \widetilde{V}_i])$ and $C'[N, \widetilde{W}_j] = (C_1[N, \widetilde{W}_j], ..., C_m[N, \widetilde{W}_j])$.
    The result directly follows from the definition of the preorder $\leq_{\mathtt{cce}(M,N)}$.
  - $C' = [\cdot]_1$.
    The result follows from clauses (2g) and (2f) for $\mathcal{R}$.

□

Lemma 65. *Suppose that $\mathcal{R}_{(M,N)}$ is a finite-step $\{\widetilde{l}\}$-simulation saturated by approximants (only defined on formal sums), and that C is a context with locations in $\{\widetilde{l}\}$. The following relation satisfies the clauses on formal sums for finite-step simulations up-to lifting and environment:*

$$\mathcal{S} \stackrel{\mathtt{def}}{=} \{((C[M], C[N]), \mathbf{Y}, \mathbf{Z}) \mid \mathbf{Y}\ \mathcal{R}^{\mathtt{cce}}_{(M,N)}\ \mathbf{Z}\}$$

Proof. We can assume that the relation $\mathcal{R}_{(M,N)}$ is *closed by* $\{\widetilde{l}\}$, i.e., each location $l \in \{\widetilde{l}\}$ occurs in corresponding columns in the dynamic environment of the formal sums (formally: for any $\mathbf{Y}, \mathbf{Z}$ in the relation and for every $l \in \{\widetilde{l}\}$ there is an index r, for $1 \leq r \leq |\ \mathbf{Y}\ |$, such that both $\mathbf{Y}\lfloor_r$ and $\mathbf{Z}\lfloor_r$ are tuples composed by location l). This assumption simplifies our proof, while not affecting the results. Indeed, we can eliminate all the pairs that do not satisfy the requirement of $\{\widetilde{l}\}$-closure and we still have a finite-step $\{\widetilde{l}\}$-simulation saturated by approximants.

The proof exploits the fact that if $\mathcal{R}_{(M,N)}$ is closed by $\{\widetilde{l}\}$ then $\mathcal{R}^{\mathsf{cce}}_{(M,N)}$ is closed by $\{\widetilde{l}\}$ and, for any C such that $\mathtt{Loc}(C) \subseteq \{\widetilde{l}\}$, if it holds that $\sum_i p_i; s_i; \widetilde{V_i}\ \mathcal{R}^{\mathsf{cce}}_{(M,N)}\ \sum_j q_j; t_j; \widetilde{W_j}$ and $(\{T_i\}_i, \{U_j\}_j) \in (\{C[M], \widetilde{V_i}\}_i, \{C[N], \widetilde{W_j}\}_j)^{\widehat{\star}}$, then $(\{T_i\}_i, \{U_j\}_j) \in (\{M, \widetilde{V_i}\}_i, \{N, \widetilde{W_j}\}_j)^{\widehat{\star}}$, since the locations in $\{\widetilde{l}\}$ are guaranteed to occur at corresponding columns in $\{\widetilde{V_i}\}_i$ and $\{\widetilde{W_j}\}_j$ and then C can be turned into a location-free context.

The condition (2a) on the weights is immediately satisfied by the definition of $\mathcal{R}^{\mathsf{cce}}_{(M,N)}$, since $\mathcal{R}$ is a simulation.

Then we prove that the conditions from (2b) to (2g) of finite-step simulation up-to lifting and environment are satisfied by $\mathcal{S}$. Suppose that $\mathbf{Y} = \sum_i p_i; s_i; \widetilde{V_i}\ \mathcal{S}_{(C[M],C[N])}\ \sum_j q_j; t_j; \widetilde{W_j} = \mathbf{Z}$ with $\mathbf{Y}' = \sum_i p_i; s_i; \widetilde{V_i}', \mathbf{Z}' = \sum_j q_j; t_j; \widetilde{W_j}'$ related by $\mathcal{R}_{(M,N)}$ and such that $(\mathbf{Y}', \mathbf{Z}') \leq_{\mathsf{cce}(M,N)} (\mathbf{Y}, \mathbf{Z})$.

(2b) for all r, if $(\widetilde{V_i})_r = \lambda x.M_i$ and $(\widetilde{W_j})_r = \lambda x.N_j$ then
for all $(\{T_i\}_i, \{U_j\}_j) \in (\{C[M], \widetilde{V_i}\}_i, \{C[N], \widetilde{W_j}\}_j)^{\widehat{\star}}$,
if $\sum_i p_i; s_i; \widetilde{V_i}; M_i\{^{T_i}\!/\!_x\} \Longmapsto \mathbf{W}$ then

$$\mathbf{W}\,\mathtt{lift}(\geq_{\mathsf{env}} (\mathcal{S}_{(C[M],C[N])}))\ \textstyle\sum_j q_j; \widetilde{W_j} \cdot [\![\langle t_j\ ;\ N_j\{^{U_j}\!/\!_x\}\rangle]\!]$$

Proof. We have three cases:

– if $(\widetilde{V_i})_r = \lambda x.C'[M, \widetilde{V_i'}]$ and $(\widetilde{W_j})_r = \lambda x.C'[N, \widetilde{W_j'}]$ for some location-free context C' then, since $\mathcal{R}_{(M,N)}$ is closed with respect to $\widetilde{l}$, there is a location-free context C'' such that

$$\textstyle\sum_i p_i; s_i; \widetilde{V_i}; M_i\{^{T_i}\!/\!_x\} = \sum_i p_i; s_i; \widetilde{V_i}; C''[M, \widetilde{V_i}]$$
$$\textstyle\sum_j q_j; t_j; \widetilde{W_j}; N_j\{^{U_j}\!/\!_x\} = \sum_j q_j; t_j; \widetilde{W_j}; C''[N, \widetilde{W_j}]$$

and we derive the result by Lemma 64.

– if $(\widetilde{V_i})_r = (\widetilde{V_i'})_{r'}$ and $(\widetilde{W_j})_r = (\widetilde{W_j'})_{r'}$ for some r' then, since $\mathcal{R}_{(M,N)}$ is closed with respect to $\widetilde{l}$, there is a location-free context C'' such that

$$\textstyle\sum_i p_i; s_i; \widetilde{V_i}; M_i\{^{T_i}\!/\!_x\} = \sum_i p_i; s_i; \widetilde{V_i}; M_i\{^{C''[M, \widetilde{V_i'}]}\!/\!_x\}$$
$$\textstyle\sum_j q_j; t_j; \widetilde{W_j}; N_j\{^{U_j}\!/\!_x\} = \sum_j q_j; t_j; \widetilde{W_j}; N_j\{^{C''[N, \widetilde{W_j'}]}\!/\!_x\}$$

Since $\mathcal{R}_{(M,N)}$ is a finite-step simulation,

$$\textstyle\sum_i p_i; s_i; \widetilde{V_i}'; M_i\{^{C''[M, \widetilde{V_i'}]}\!/\!_x\} \Longmapsto \mathbf{W}$$

implies $\mathbf{W}\,\mathtt{lift}(\mathcal{R}_{M,N})\,[\![\sum_j q_j; t_j; \widetilde{W_j}'; N_j\{^{C''[N, \widetilde{W_j'}]}\!/\!_x\}]\!]$, which in turn implies the result analogously to Lemma 63.

– if $(\widetilde{V_i})_r = M$ and $(\widetilde{W_j})_r = N$ then M and N are values and by clause (2g) applied to $\mathcal{R}_{(M,N)}$ we have

$$\textstyle\sum_i p_i; s_i; \widetilde{V_i}', M\ \mathtt{lift}(\mathcal{R}_{(M,N)})\ \sum_j q_j; t_j; \widetilde{W_j}', N$$

Hence, by clause (2b) applied to $\mathcal{R}_{(M,N)}$ we have

$$\textstyle\sum_i p_i; s_i; \widetilde{V_i}', M; M_i\{^{C''[M, \widetilde{V_i'}]}\!/\!_x\} \Longmapsto \mathbf{W}$$

implies $\mathbf{W}\,\mathtt{lift}(\mathcal{R}_{(M,N)})\,[\![\sum_j q_j; t_j; \widetilde{W_j}', N; N_j\{^{U_j}\!/\!_x\}]\!]$, which in turn implies the result (see the proof of Lemma 63).

□

(2c) for all r, if $(\widetilde{V_i})_r = l_i$ and $(\widetilde{W_j})_r = k_j$ then
- $\sum_i p_i; s_i; \widetilde{V_i}, s_i(l_i)\,\mathtt{lift}(\mathcal{S}_{(C[M],C[N])})\ \sum_j q_j; t_j; \widetilde{W_j}, t_j(k_j)$,
- for all $(\{T_i\}_i, \{U_j\}_j) \in (\{C[M], \widetilde{V_i}\}_i, \{C[N], \widetilde{W_j}\}_j)^{\widehat{\star}}$ we have

$$\sum_i p_i; s_i[l_i \to T_i]; \widetilde{V_i}\ \mathcal{S}_{(C[M],C[N])}\ \sum_j q_j; t_j[k_j \to U_j]; \widetilde{W_j}\,.$$

Proof. Since contexts used in $\leq_{\mathsf{cce}(M,N)}$ are location-free and $\mathcal{R}_{(M,N)}$ is closed with respect to the locations in $\widetilde{l}$ (and thereby the locations in $C[M], C[N]$ are in the dynamic environment of the formal sums in $\mathcal{R}_{(M,N)}$ in corresponding columns), there is r' such that $(\widetilde{V_i'})_{r'} = l_i$ and $(\widetilde{W_j'})_{r'} = k_j$, then:
- $\sum_i p_i; s_i; \widetilde{V_i'}, s_i(l_i)\,\mathtt{lift}(\mathcal{R}_{(M,N)})\ \sum_j q_j; t_j; \widetilde{W_j'}, t_j(k_j)$ and we have

$$\sum_i p_i; s_i; \widetilde{V_i}, s_i(l_i)\,\mathtt{lift}(\mathcal{R}^{\mathsf{cce}}_{(M,N)})\ \sum_j q_j; t_j; \widetilde{W_j}, t_j(k_j),$$

  from which the result follows.
- by $(\{T_i\}_i, \{U_j\}_j) \in (\{M, \widetilde{V_i'}\}_i, \{N, \widetilde{W_j'}\}_j)^{\widehat{\star}}$ we derive

$$\sum_i p_i; s_i[l_i \to T_i]; \widetilde{V_i'}\ \mathcal{R}_{(M,N)}\ \sum_j q_j; t_j[k_j \to U_j]; \widetilde{W_j'}\,,$$

  which implies

$$\sum_i p_i; s_i[l_i \to T_i]; \widetilde{V_i}\ \mathcal{S}_{(C[M],C[N])}\ \sum_j q_j; t_j[k_j \to U_j]; \widetilde{W_j}\,.$$

□

(2d) for any $(\{s_i\}_i, \{t_j\}_j)$-fresh locations $(\{l_i\}_i, \{k_j\}_j)$,
and for all $(\{T_i\}_i, \{U_j\}_j) \in (\{C[M], \widetilde{V_i}\}_i, \{C[N], \widetilde{W_j}\}_j)^{\widehat{\star}}$,

$$\sum_i p_i; s_i[l_i \to T_i]; \widetilde{V_i}, l_i\ \mathcal{S}_{(C[M],C[N])}\ \sum_j q_j; t_j[k_j \to U_j]; \widetilde{W_j}, k_j\,.$$

Proof. The result follows from $(\{T_i\}_i, \{U_j\}_j) \in (\{M, \widetilde{V_i'}\}_i, \{N, \widetilde{W_j'}\}_j)^{\widehat{\star}}$ as in the previous clause, by exploiting clause (2d) on $\mathcal{R}$. □

(2e) for all r, if $(\widetilde{V_i})_r = c_i$ and $(\widetilde{W_j})_r = c_j$ then all constants in the two columns are the same (i.e., there is c_a with $c_i = c_j = c_a$ for all i, j).

Proof. For every r, there are three cases: either $\mathbf{Y}\lfloor_r = \mathbf{Y}'\lfloor_{r'}$ and $\mathbf{Z}\lfloor_r = \mathbf{Z}'\lfloor_{r'}$ for some r', in which case both columns are composed by the same constant, since $\mathcal{R}$ is a finite-step simulation; or the value-context is a constant and $\mathbf{Y}\lfloor_r = \mathbf{Z}\lfloor_r = c$; or the constants are respectively M and N, in which case

$$\sum_i p_i; s_i; \widetilde{V_i'}, M\,\mathtt{lift}(\mathcal{R}_{(M,N)})\ \sum_j q_j; t_j; \widetilde{W_j'}, N$$

(by clause (2g)) and thus they have to be the same constant, otherwise $\mathcal{R}$ would not respect condition (2e). □

(2f) for all r, if $(\widetilde{V_i})_r = (V_{i,1}, ..., V_{i,n})$ and $(\widetilde{W_j})_r = (W_{j,1}, ..., W_{j,n})$ then

$$\sum_i p_i; s_i; \widetilde{V_i}, V_{i,1}, ..., V_{i,n}\,\mathtt{lift}(\mathcal{S}_{(C[M],C[N])})\ \sum_j q_j; t_j; \widetilde{W_j}, W_{j,1}, ..., W_{j,n}\,.$$

Proof. If $\mathbf{Y}\lfloor_r = \mathbf{Y}'\lfloor_{r'}$ and $\mathbf{Z}\lfloor_r = \mathbf{Z}'\lfloor_{r'}$ for some r' then it follows from the definition of $\mathcal{R}$ that

$$\sum_i p_i; s_i; \widetilde{V_i}, V_{i,1}, ..., V_{i,n}\,\mathtt{lift}(\mathcal{S}_{(C[M],C[N])})\ \sum_j q_j; t_j; \widetilde{W_j}, W_{j,1}, ..., W_{j,n}.$$

Otherwise, for every $1 \leq h \leq n$ we have $(\{V_{i,h}\}_i, \{U_{j,h}\}_j) \in (\{M, \widetilde{V}_i'\}_i, \{N, \widetilde{W}_j'\}_j)^{\widehat{\star}}$, which implies

$$(\mathbf{Y}', \mathbf{Z}') \leq_{\mathsf{cce}(M,N)} (\textstyle\sum_i p_i; s_i; \widetilde{V}_i, V_{i,1}, ..., V_{i,n}, \sum_j q_j; t_j; \widetilde{W}_j, W_{j,1}, ..., W_{j,n}),$$

i.e., the formal sums are in relation $\mathtt{lift}(\mathcal{S}_{(C[M],C[N])})$.

Finally, if $(\widetilde{V}_i)_r = M$ and $(\widetilde{W}_j)_r = N$ then the clause follows as in the previous cases from $\sum_i p_i; s_i; \widetilde{V}_i', M$ $\mathtt{lift}(\mathcal{R}_{(M,N)})$ $\sum_j q_j; t_j; \widetilde{W}_j', N$, by clause (2g). □

(2g) $\sum_i p_i; \widetilde{V}_i \cdot [\![\langle s_i \,;\, C[M]\rangle]\!]$ $\mathtt{lift}(\mathcal{S}_{(C[M],C[N])})$ $\sum_j q_j; \widetilde{W}_j \cdot [\![\langle t_j \,;\, C[N]\rangle]\!]$.

Proof. Since $\mathcal{R}_{(M,N)}$ is closed with respect to $\widetilde{l}$, there is a location-free context C' such that

$$\mathbf{Y}; C[M] = \mathbf{Y}; C'[M, \mathbf{Y}]$$
$$\mathbf{Z}; C[N] = \mathbf{Z}; C'[N, \mathbf{Z}]$$

and the result follows from Lemma 64. □

□

We can now derive from Lemma 65 the congruence result. Let C be a context such that $\mathtt{Loc}(C) \subseteq \{\widetilde{l}\}$. Let $\mathcal{R}$ be a finite-step $\{\widetilde{l}\}$-simulation such that $\langle s \,;\, M\rangle \;\mathcal{R}\; \langle t \,;\, N\rangle$. Then the relation

$$\{(\langle s \,;\, C[M]\rangle, \langle t \,;\, C[N]\rangle)\} \cup \{((C[M], C[N]), \mathbf{Y}, \mathbf{Z}) \mid \mathbf{Y}\; \mathcal{R}^{\mathsf{cce}}_{(M,N)}\; \mathbf{Y}\}$$

is a finite step simulation up-to lifting and environment, since:

- clause (1) on terms follows since $\mathtt{Loc}(C) \subseteq \{\widetilde{l}\}$ and by clause (1) for $\mathcal{R}$ we have $(1; s; \widetilde{l}, 1; t; \widetilde{l}) \in \mathcal{R}_{(M,N)} \subseteq \mathcal{R}^{\mathsf{cce}}_{(M,N)}$;
- the clauses for formal sums follow by Lemma 65.

Proof of Theorem 53

Let $\widetilde{l}' = \widetilde{l}, \widetilde{l}''$ and let $\widetilde{V}' = \widetilde{V}, \widetilde{V}''$ be a sequence of values whose types are consistent with those of $\widetilde{l}'$ and with locations in $\{\widetilde{l}'\}$. Let C be a context with locations in $\{\widetilde{l}'\}$ and let $\mathcal{R}$ be a finite-step $\{\widetilde{l}\}$-simulation (saturated by approximants) relating $\langle s \,;\, M\rangle$ and $\langle t \,;\, N\rangle$. Then $1; s; \widetilde{l}\; \mathcal{R}_{(M,N)}\; 1; t; \widetilde{l}$ and by repeatedly applying clause (2d) we derive

$$1; s[\widetilde{l}'' \to \widetilde{W}]; \widetilde{l}'\; \mathcal{R}_{(M,N)}\; 1; t[\widetilde{l}'' \to \widetilde{W}]; \widetilde{l}'$$

for a consistent sequence of values $\widetilde{W}$. (Note that we cannot guarantee by just using clause (2d) that the tuple of values $\widetilde{V}''$ is assigned to $\widetilde{l}''$, since locations in $\widetilde{l}''$ might occur in any value in $\widetilde{V}''$. Hence, we first have to put all the locations in $\{\widetilde{l}''\}$ in the dynamic environment.) Then by repeatedly applying clause (2c) we derive

$$1; s[\widetilde{l}'' \to \widetilde{V}'']; \widetilde{l}'\; \mathcal{R}_{(M,N)}\; 1; t[\widetilde{l}'' \to \widetilde{V}'']; \widetilde{l}'\,.$$

It is easy to see that if we restrict $\mathcal{R}_{(M,N)}$ to those pairs of formal sums whose dynamic environments begin with the sequence $\widetilde{l}'$ of locations then the clauses of finite-step $\{\widetilde{l}'\}$-simulation are satisfied. Let $\mathcal{R}'_{(M,N)}$ be such a restriction of $\mathcal{R}_{(M,N)}$. Then we can apply Lemma 65 (see the proof of Theorem 52) and derive that relation

$$\mathcal{S} \stackrel{\mathsf{def}}{=} \{((C[M], C[N]), \mathbf{Y}, \mathbf{Z}) \mid \mathbf{Y}\; \mathcal{R}'^{\mathsf{cce}}_{(M,N)}\; \mathbf{Z}\}$$

is a finite-step $\{\widetilde{l'}\}$-simulation (up-to lifting and environment). Since

$$(1; s[\widetilde{l''} \to \widetilde{V''}]; \widetilde{l'}, 1; t[\widetilde{l''} \to \widetilde{V''}]; \widetilde{l'}) \in \mathcal{R}'_{(M,N)} \subseteq \mathcal{R}'^{\mathsf{cce}}_{(M,N)} = \mathcal{S}_{(C[M],C[N])}$$

we conclude $\langle s[\widetilde{l''} \to \widetilde{V''}]\,;\, C[M]\rangle \lesssim^{\{\widetilde{l'}\}}_{\text{fin}} \langle t[\widetilde{l''} \to \widetilde{V''}]\,;\, C[N]\rangle$.

Finally, by repeatedly applying clause (2c) to locations $\widetilde{l}$ in the pair

$$1; s[\widetilde{l''} \to \widetilde{V''}]; \widetilde{l'}\ \mathcal{S}_{(C[M],C[N])}\ 1; t[\widetilde{l''} \to \widetilde{V''}]; \widetilde{l'}$$

we derive

$$1; \widetilde{l'} = \widetilde{V'}; \widetilde{l'}\ \mathcal{S}_{(C[M],C[N])}\ 1; \widetilde{l'} = \widetilde{V'}; \widetilde{l'}$$

which in turn implies $\langle \widetilde{l'} = \widetilde{V'}\,;\, C[M]\rangle \lesssim^{\{\widetilde{l'}\}}_{\text{fin}} \langle \widetilde{l'} = \widetilde{V'}\,;\, C[N]\rangle$.

Proof of Theorem 55

We prove that the relation

$$\begin{array}{l}\mathcal{R} \stackrel{\mathsf{def}}{=} \{(\langle \widetilde{l} = \widetilde{V}\,;\, M\rangle, \langle \widetilde{l} = \widetilde{V}\,;\, N\rangle) \mid M \leq_{\mathtt{ctx}} N \ \wedge\ \{\widetilde{l}\} = \mathtt{Loc}(M) \cup \mathtt{Loc}(N)\} \cup \\ \quad \{((M,N), \sum_i p_i; s_i; V_1^i, ..., V_n^i\,, \sum_j q_j; t_j; W_1^j, ..., W_n^j) \mid M \leq_{\mathtt{ctx}} N \\ \quad \wedge\ \exists C, \widetilde{V} \text{ such that } (\llbracket \langle \widetilde{l} = \widetilde{V}\,;\, C[M]\rangle \rrbracket = \sum_i p_i; s_i; \lambda x.x V_1^i ... V_n^i \\ \quad \wedge\ \llbracket \langle \widetilde{l} = \widetilde{V}\,;\, C[N]\rangle \rrbracket = \sum_j q_j; t_j; \lambda x.x W_1^j ... W_n^j \\ \quad \text{with } \mathtt{Loc}(C) \subseteq \{\widetilde{l}\} = \mathtt{Loc}(M) \cup \mathtt{Loc}(N) \\ \quad \wedge\ \text{ they are } \textit{first-order consistent})\}\end{array}$$

satisfies the clauses of $\{\widetilde{l}\}$-simulation. Let $\widetilde{l} = l_1, ...l_n = \mathtt{Loc}(M) \cup \mathtt{Loc}(N)$ and $\langle \widetilde{l} = \widetilde{V}\,;\, M\rangle\ \mathcal{R}\ \langle \widetilde{l} = \widetilde{V}\,;\, N\rangle$. Hence, $M \leq_{\mathtt{ctx}} N$ and clause (1) holds since, using context $C = \lambda x.x l_1 ... l_n$ (with no holes) we derive from $M \leq_{\mathtt{ctx}} N$ that $1; \widetilde{l} = \widetilde{V}; l_1, ..., l_n\ \mathcal{R}_{(M,N)}\ 1; \widetilde{l} = \widetilde{V}; l_1, ..., l_n$.

To prove that $\mathcal{R}$ satisfies the clauses of simulation for formal sums, we first show the following lemma.

LEMMA 66. *If* $\mathbf{Y}\ \mathcal{R}_{(M,N)}\ \mathbf{Z}$ *then for any* C *with* $\mathtt{Loc}(C) \subseteq \mathtt{Loc}(M) \cup \mathtt{Loc}(N)$*:*

- *If* $\llbracket \mathbf{Y}; C[M, \mathbf{Y}] \rrbracket$ *and* $\llbracket \mathbf{Z}; C[N, \mathbf{Z}] \rrbracket$ *are first-order consistent then*

$$\llbracket \mathbf{Y}; C[M, \mathbf{Y}] \rrbracket\ \mathcal{R}_{(M,N)}\ \llbracket \mathbf{Z}; C[N, \mathbf{Z}] \rrbracket\,;$$

- *If* $\llbracket \mathbf{Y}; C[M, \mathbf{Y}] \rrbracket$ *and* $\llbracket \mathbf{Z}; C[N, \mathbf{Z}] \rrbracket$ *are not first-order consistent then*

$$\llbracket \mathbf{Y}; C[M, \mathbf{Y}] \rrbracket\, \mathtt{lift}(\mathcal{R}_{(M,N)})\, \llbracket \mathbf{Z}; C[N, \mathbf{Z}] \rrbracket\,.$$

PROOF. If $\mathbf{Y} = \sum_i p_i; s_i; V_1^i, ..., V_n^i\ \mathcal{R}_{(M,N)}\ \sum_j q_j; t_j; W_1^j, ..., W_n^j = \mathbf{Z}$ then they are first-order consistent environment formal sums and there are C, s such that $\llbracket \langle s\,;\, C[M]\rangle \rrbracket = \sum_i p_i; s_i; \lambda x.x V_1^i ... V_n^i$ and $\llbracket \langle s\,;\, C[N]\rangle \rrbracket = \sum_j q_j; t_j; \lambda x.x W_1^j ... W_n^j$ and $\mathtt{Loc}(C) \subseteq \{\widetilde{l}\} = \mathtt{Loc}(M) \cup \mathtt{Loc}(N)$.

Let C' be any context with $\mathtt{Loc}(C) \subseteq \{\widetilde{l}\} = \mathtt{Loc}(M) \cup \mathtt{Loc}(N)$ and let

$$P_{M,C'} = \lambda x_1, ..., x_n.(\lambda z, x.x x_1 ... x_n z) C'[M, x_1, ..., x_n]$$

and $P_{N,C'}$ the same term with M substituted to N. It follows from $M \leq_{\mathtt{ctx}} N$ that $C[M] P_{M,C'} \leq_{\mathtt{ctx}} C[N] P_{N,C'}$.

We have:

$$\begin{aligned}\llbracket\langle s\,;\, C[M]P_{M,C'}\rangle\rrbracket &= \llbracket\llbracket\langle s\,;\, C[M]\rangle\rrbracket P_{M,C'}\rrbracket\\ &= \llbracket\textstyle\sum_i p_i; s_i; (P_{M,C'}V_1^i...V_n^i)\rrbracket\\ &= \llbracket\textstyle\sum_i p_i; s_i; (\lambda z, x.xV_1^i...V_n^i z)C'[M, V_1^i, ..., V_n^i]\rrbracket\\ &= \textstyle\sum_{i,k} p_{i,k}; s_{i,k}; (\lambda z, x.xV_1^i...V_n^i V_{i,k})\end{aligned}$$

for $\llbracket\sum_i p_i; s_i; C'[M, V_1^i, ..., V_n^i]\rrbracket = \sum_{i,k} p_{i,k}; s_{i,k}; V_{i,k}$ and analogously for N:

$$\llbracket\langle s\,;\, C[N]P_{N,C'}\rangle\rrbracket = \textstyle\sum_{j,h} q_{j,h}; t_{j,h}; \lambda x.xW_1^j...W_n^j W_{j,h}$$

for $\llbracket\sum_j q_j; t_j; C'[N, W_1^j, ..., W_n^j]\rrbracket = \sum_{j,k} q_{j,h}; t_{j,h}; W_{j,h}$.
If $\llbracket\sum_i p_i; s_i; C'[M, V_1^i, ..., V_n^i]\rrbracket$ and $\llbracket\sum_j q_j; t_j; C'[N, W_1^j, ..., W_n^j]\rrbracket$ are first order consistent, then we can conclude, by the definition of $\mathcal{R}$, that

$$\begin{aligned}&\textstyle\sum_i p_i; V_1^i, ..., V_n^i \cdot \llbracket\langle s_i\,;\, C'[M, V_1^i, ..., V_n^i]\rangle\rrbracket\\ &\mathcal{R}_{(M,N)}\ \textstyle\sum_j q_j; W_1^j, ..., W_n^j \cdot \llbracket\langle t_j\,;\, C'[N, W_1^j, ..., W_n^j]\rangle\rrbracket\end{aligned}$$

If $\llbracket\sum_i p_i; s_i; C'[M, V_1^i, ..., V_n^i]\rrbracket$ and $\llbracket\sum_j q_j; t_j; C'[N, W_1^j, ..., W_n^j]\rrbracket$ are not first order consistent, which means that the dynamic environment is composed of different constants, then for any constant c we can use the term

$$P_{M,C',c} = \lambda x_1, ..., x_n.(\lambda z, x.xx_1...x_n z)\ \texttt{if}\ \ C'[M, x_1, ..., x_n] = c\ \texttt{then}\ c\ \texttt{else}\ \Omega$$

to derive, analogously as above, that

$$\textstyle\sum_{\{i,k|V_{i,k}=c\}} p_{i,k}; s_{i,k}; V_1^i, ..., V_n^i, V_{i,k}\ \mathcal{R}_{(M,N)}\ \sum_{\{j,h|W_{j,h}=c\}} q_{j,h}; t_{j,h}; W_1^j, ..., W_n^j, W_{j,h}$$

and thus

$$\begin{aligned}\mathbf{Y}; C[M, \mathbf{Y}] = &\textstyle\sum_c \sum_{\{i,k|V_{i,k}=c\}} p_{i,k}; s_{i,k}; V_1^i, ..., V_n^i, V_{i,k}\\ &\texttt{lift}(\mathcal{R}_{(M,N)})\\ &\textstyle\sum_c \sum_{\{j,h|W_{j,h}=c\}} q_{j,h}; t_{j,h}; W_1^j, ..., W_n^j, W_{j,h} = \mathbf{Z}; C[N, \mathbf{Z}]\end{aligned}$$

□

Let $\mathbf{Y} = \sum_i p_i; s_i; V_1^i, ..., V_n^i\ \mathcal{R}_{(M,N)}\ \sum_j q_j; t_j; W_1^j, ..., W_n^j = \mathbf{Z}$ be first-order consistent environment formal sums and let C, s be such that $\llbracket\langle s\,;\, C[M]\rangle\rrbracket = \sum_i p_i; s_i; \lambda x.xV_1^i...V_n^i$ and $\llbracket\langle s\,;\, C[N]\rangle\rrbracket = \sum_j q_j; t_j; \lambda x.xW_1^j...W_n^j$. We can now prove that $\mathcal{R}$ satisfies the simulation clauses on formal sums.

(2a) It follows from the definition of $\leq_{\texttt{ctx}}$ that $M \leq_{\texttt{ctx}} N$ implies $C[M] \leq_{\texttt{ctx}} C[N]$, which implies $\texttt{weight}(\mathbf{Y}) = \texttt{weight}(\llbracket\langle s\,;\, C[M]\rangle\rrbracket) \leq \texttt{weight}(\llbracket\langle s\,;\, C[N]\rangle\rrbracket) = \texttt{weight}(\mathbf{Z})$.

(2b) Let $V_r^i = \lambda x.M_i$ and $W_r^j = \lambda x.N_j$. The result follows from Lemma 66, using context $C = [\cdot]_{r+1}C'$, for any value context C'.

(2c) Let $V_r^i = l_i$ and $W_r^j = k_j$. The result follows from Lemma 66, respectively using contexts $C_1 = ![\cdot]_{r+1}$ and $C_2 = [\cdot]_{r+1} := C'$, for any value context C'. In the latter case, since the formal sums are first-order consistent we can use directly relation $\mathcal{R}_{(M,N)}$, without the lifting construction (by the first item of Lemma 66).

(2d) The result follows from the first item of Lemma 66, using context $C = (\boldsymbol{\nu}\, x := C_1)C_2$, for C_1 a value context and C_2 a context with free variable x.

(2e) The result directly follows from the definition of $\mathcal{R}$.

(2f) For all r, if $V_r^i = (V_{i,1}, ..., V_{i,n})$ and $W_r^j = (W_{j,1}, ..., W_{j,n})$ then the result follows by iteratively applying Lemma 66, using contexts $C_1 = \#_1([\cdot]_{r+1})$,..., $C_n = \#_n([\cdot]_{r+1})$.

(2g) The result follows from Lemma 66, using context $C = [\cdot]_1$.

REFERENCES

[1] Martín Abadi and Andrew D. Gordon. 1998. A Bisimulation Method for Cryptographic Protocols. In *Proc. ESOP'98 (LNCS)*, Chris Hankin (Ed.), Vol. 1381. Springer, 12–26.

[2] Samson Abramsky. 1990. The Lazy Lambda Calculus. In *Research topics in functional programming*, David A. Turner (Ed.). Addison-Wesley, 65–116.

[3] Andrés Aristizábal, Dariusz Biernacki, Sergueï Lenglet, and Piotr Polesiuk. 2016. Environmental Bisimulations for Delimited-Control Operators with Dynamic Prompt Generation. In *Proc. FSCD'16*. 9:1–9:17.

[4] Dariusz Biernacki and Sergueï Lenglet. 2013. Environmental Bisimulations for Delimited-Control Operators. In *Proc. APLAS'13 (LNCS)*, Vol. 8301. Springer, 333–348.

[5] Aleš Bizjak. 2016. *On Semantics and Applications of Guarded Recursion*. Ph.D. Dissertation. Aarhus University.

[6] Ales Bizjak and Lars Birkedal. 2015. Step-Indexed Logical Relations for Probability. In *Proc. FoSSaCS'15*. 279–294.

[7] M. Boreale and D. Sangiorgi. 1998. Bisimulation in name-passing calculi without matching. In *Proc. LICS'98*. IEEE Computer Society Press.

[8] Raphaëlle Crubillé and Ugo Dal Lago. 2014. On Probabilistic Applicative Bisimulation and Call-by-Value λ-Calculi. In *Proc. ESOP'14 (LNCS)*, Vol. 8410. Springer, 209–228.

[9] Raphaëlle Crubillé and Ugo Dal Lago. 2015. Metric Reasoning about λ-Terms: The Affine Case. In *Proc. LICS'15*. IEEE, 633–644.

[10] Raphaëlle Crubillé, Ugo Dal Lago, Davide Sangiorgi, and Valeria Vignudelli. 2015. On Applicative Similarity, Sequentiality, and Full Abstraction. In *Correct System Design*, Roland Meyer, André Platzer, and Heike Wehrheim (Eds.). Springer, 65–82.

[11] Ugo Dal Lago, Davide Sangiorgi, and Michele Alberti. 2014. On Coinductive Equivalences for Higher-order Probabilistic Functional Programs. In *Proc. POPL'14*. ACM, 297–308.

[12] Ugo Dal Lago and Margherita Zorzi. 2012. Probabilistic Operational Semantics for the Lambda Calculus. *RAIRO - Theoretical Informatics and Applications* 46, 3 (2012), 413–450.

[13] Vincent Danos and Russell S. Harmer. 2002. Probabilistic Game Semantics. *ACM Trans. Comput. Logic* 3, 3 (2002), 359–382.

[14] Josee Desharnais, Radha Jagadeesan, Vineet Gupta, and Prakash Panangaden. 2002. The Metric Analogue of Weak Bisimulation for Probabilistic Processes. In *Proc. LICS'02*. IEEE Computer Society, 413–422.

[15] Josée Desharnais, François Laviolette, and Mathieu Tracol. 2008. Approximate Analysis of Probabilistic Processes: Logic, Simulation and Games. In *Proc QEST'08*. IEEE Computer Society, 264–273.

[16] Thomas Ehrhard, Christine Tasson, and Michele Pagani. 2014. Probabilistic Coherence Spaces Are Fully Abstract for Probabilistic PCF. In *Proc. POPL'14*. 309–320.

[17] Andrew D. Gordon. 1995. Bisimilarity as a theory of functional programming. *Electr. Notes Theor. Comput. Sci.* 1 (1995), 232–252.

[18] Jean Goubault-Larrecq. 2015. Full abstraction for non-deterministic and probabilistic extensions of PCF I: The angelic cases. *Journal of Logical and Algebraic Methods in Programming* 84, 1 (2015), 155 – 184.

[19] Douglas J. Howe. 1996. Proving Congruence of Bisimulation in Functional Programming Languages. *Information and Computation* 124, 2 (1996), 103–112.

[20] Chung-Kil Hur, Derek Dreyer, Georg Neis, and Viktor Vafeiadis. 2012. The marriage of bisimulations and Kripke logical relations. In *Proc. POPL'12*. 59–72.

[21] Guilhem Jaber and Nicolas Tabareau. 2015. Kripke Open Bisimulation - A Marriage of Game Semantics and Operational Techniques. In *Proc. APLAS'15*. 271–291.

[22] Radha Jagadeesan, Corin Pitcher, and James Riely. 2009. Open Bisimulation for Aspects. *T. Aspect-Oriented Software Development* 5 (2009), 72–132.

[23] Alan Jeffrey and Julian Rathke. 1999. Towards a Theory of Bisimulation for Local Names. In *Proc. LICS'99*. 56–66.

[24] V. Koutavas, P. B. Levy, and E. Sumii. 2011. From Applicative to Environmental Bisimulation. *Electronic Notes in Theoretical Computer Science* 276 (2011), 215–235.

[25] Vassileios Koutavas and Mitchell Wand. 2006. Bisimulations for Untyped Imperative Objects. In *Proc. ESOP'06*. 146–161.

[26] Vassileios Koutavas and Mitchell Wand. 2006. Small Bisimulations for Reasoning About Higher-Order Imperative Programs. In *Proc. POPL'06*. 141–152.

[27] S. B. Lassen. 1998. *Relational Reasoning about Functions and Nondeterminism*. Ph.D. Dissertation. University of Aarhus.
[28] Søren B. Lassen and Paul Blain Levy. 2007. Typed Normal Form Bisimulation. In *Proc. CSL'07 (LNCS)*, Vol. 4646. Springer, 283–297.
[29] John Maraist, Martin Odersky, David N. Turner, and Philip Wadler. 1999. Call-by-name, Call-by-value, Call-by-need and the Linear lambda Calculus. *Theoretical Computer Science* 228, 1-2 (1999), 175–210.
[30] Robin Milner. 2006. Pure bigraphs: Structure and dynamics. *Inf. Comput.* 204, 1 (2006), 60–122.
[31] John C. Mitchell. 1996. *Foundations for Programming Languages*. MIT Press.
[32] J. H., Jr. Morris. 1968. *Lambda-calculus models of programming languages*. Ph.D. Dissertation. Massachusetts Institute of Technology.
[33] Georg Neis, Chung-Kil Hur, Jan-Oliver Kaiser, Craig McLaughlin, Derek Dreyer, and Viktor Vafeiadis. 2015. Pilsner: A Compositionally Verified Compiler for a Higher-order Imperative Language. In *Proc. ICFP'15*. ACM, New York, NY, USA, 166–178.
[34] D. Park. 1981. A New Equivalence notion for Communicating Systems. In *Bulletin EATCS*, G. Maurer (Ed.), Vol. 14. 78–80.
[35] Adrien Piérard and Eijiro Sumii. 2011. Sound Bisimulations for Higher-Order Distributed Process Calculus. In *Proc. FoSSaCS'11*. 123–137.
[36] Adrien Piérard and Eijiro Sumii. 2012. A Higher-Order Distributed Calculus with Name Creation. In *Proc. LICS'12*. 531–540.
[37] Andrew Pitts. 2005. Typed Operational Reasoning. In *Advanced Topics in Types and Programming Languages*, Benjamin C. Pierce (Ed.). MIT Press, Chapter 7, 245–289.
[38] Andrew Pitts. 2012. Howe's Method for Higher-order Languages. In *Advanced Topics in Bisimulation and Coinduction*, D. Sangiorgi and J. Rutten (Eds.). Cambridge Tracts in Theoretical Computer Science, Vol. 52. Cambridge University Press, Cambridge, Chapter 5, 197–232.
[39] Damien Pous and Davide Sangiorgi. 2012. Enhancements of the bisimulation proof method. In *Advanced Topics in Bisimulation and Coinduction*, Davide Sangiorgi and Jan Rutten (Eds.). Cambridge University Press.
[40] Jan Rutten and Bart Jacobs. 2012. (Co)Algebras and (Co)Induction. In *Advanced Topics in Bisimulation and Coinduction*, Davide Sangiorgi and Jan Rutten (Eds.). Cambridge University Press.
[41] David Sands. 1997. From SOS Rules to Proof Principles: An Operational Metatheory for Functional Languages. In *Proc. POPL'97*. 428–441.
[42] Davide Sangiorgi. 1994. The Lazy Lambda Calculus in a Concurrency Scenario. *Information and Computation* 111, 1 (May 1994), 120–153.
[43] D. Sangiorgi. 2012. *Introduction to Bisimulation and Coinduction*. Cambridge University Press, Cambridge.
[44] Davide Sangiorgi, Naoki Kobayashi, and Eijiro Sumii. 2007. Logical Bisimulations and Functional Languages. In *Proc. FSEN'07 (LNCS)*. Springer, 364–379.
[45] Davide Sangiorgi, Naoki Kobayashi, and Eijiro Sumii. 2011. Environmental Bisimulations for Higher-Order Languages. *ACM Transactions on Programming Languages and Systems* 33, 1:5 (2011).
[46] Nobuyuki Sato and Eijiro Sumii. 2009. The Higher-Order, Call-by-Value Applied Pi-Calculus. In *Proc. APLAS'09*. 311–326.
[47] Kristian Støvring and Soren B. Lassen. 2007. A Complete, Co-inductive Syntactic Theory of Sequential Control and State. In *Proc. POPL'07*. 161–172.
[48] Eijiro Sumii and Benjamin C. Pierce. 2007. A Bisimulation for Dynamic Sealing. *Theoretical Computer Science* 375, 1-3 (2007), 169–192. A preliminary version in *Proc. POPL'04*, 2004.
[49] Eijiro Sumii and Benjamin C. Pierce. 2007. A bisimulation for type abstraction and recursion. *J. ACM* 54, 5 (2007). A preliminary version in *Proc. POPL'05*, 2005.