Alma Mater Studiorum - Università di Bologna

DOTTORATO DI RICERCA IN INFORMATICA
Ciclo: XXVII

Settore Concorsuale di afferenza: 01/B1
Settore Scientifico disciplinare: INF01

Coinductive Techniques on a Linear Quantum λ–Calculus

Presentata da: Alessandro Rioli

Coordinatore Dottorato:
Paolo Ciaccia

Relatore:
Ugo Dal Lago

Esame finale anno 2016

Acknowledgements

I should first thank my parents, which educated me to the curiosity and to joy to get to know and discover new things.

Then let me thank my supervisor Ugo dal Lago and my tutor Simone Martini who guided me along the sometimes difficult path which finally brought me to finish my PhD thesis, especially during the days when for personal reasons I lived some discouraging moments, and I felt the impression that I could not finish the work successfully.

Finally I want to remember my colleagues and among them particularly Giulio Pellitta, Francesco Poggi and Rajesh Sharma for the help, the encouragement, the suggestions they gave me above all in the beginning of my PhD adventure, and my friends Giacomo Presutti for the contribution which he gave in patiently listening to my monologues and Elisa Turrini, for having disclosed me the world of computer science.

Contents

Chapter 1

Introduction

Programming languages theory, which has among its purposes to investigate the logical foundations of computer science, finds in λ calculus an optimal tool of analysis. The λ calculus, which was invented in 1930 by A. Church as a formal system to capture the computational power of functional theories, is in many senses considered as the first programming language and is currently the main instrument to study the properties of the class of higher order functional languages, namely those where functions are permitted as values for procedures. In this work the lambda calculus is used to investigate the issue of equivalence among programs from a formal point of view.

Program equivalence is one of the fundamental notions in the theory of programming languages. Studying the nature of program equivalence is not only interesting from a purely foundational point of view, but can also be the first step towards defining (semi)automatic techniques for program verification, or for validating compiler optimizations. The most widely accepted notion of equivalence among programs, namely Morris's context equivalence [42], leans on the concept of observational behaviour: two programs are contextually equivalent if they may be exchanged for one another in any possible larger program – which is precisely the definition of context – without affecting its evaluation, hence the potentiality to converge. As a

prerequisite, a well working relation to compare programs has indeed to be *compatible* with the language, namely it should commute with its syntactic constructors and it is relatively easy to show that context equivalence matches this condition. Context equivalence relation is an effective tool to prove two programs *not* to be equivalent, since this merely amounts to finding *one* context which separates them. On the other hand, proving two terms to be equivalent requires one to examine their behaviour in *every possible* context.

Various ways to alleviate the burden of proving the quantification over *all* contexts have been proposed in the literature. The proof of context equivalence can be relieved for example by introducing the so called context lemmas, which have the aim to reduce the class of contexts which are needed to show contextual equivalence [41, 44]. Context lemmas ensure that the context equivalence between two programs actually holds if they show to behave the same in a more restricted class of contexts: thus the quantification over all possible contexts, required in proving context equivalence, is replaced by proving the equivalence of two programs on a smaller class of them. Among the possible classes it is relevant that of contexts which are *Uses of Closed Instantiations*: the equivalence of programs within this restricted set of contexts – the so called *evaluation contexts* – is called CIU equivalence and can be proved to coincide with the general context equivalence. *Denotational semantics* methods differ from those of operational semantics, where a program is figured as a sequence of computational steps, because they aim to make programs independent of the abstract machine by finding a bijective relation between programs and some mathematical structures easier to compare. Here two terms are considered equivalent if their semantics correspond to the same mathematical structure. With logical relations [45], programs are compared by giving a family of relations which connect contextually equivalent terms on the set of programs. More recently, trace equivalence [17] has been considered as possible method of investigation: here two programs are compared if they accept the same set of traces, a trace being a sequence of actions that an external observer can perform on the system. We are here especially interested in bisimilarity [1, 39], which is a technique of comparison

among systems defined in a coinductive way and to its applications in the fields of probabilistic and quantum programming languages.

1.1 Coinduction and Bisimulation

It is well known that a set can be defined in an inductive way starting by the simpler elements – usually included in the set by an axiom – and adding, with a sequence of steps, the more complex ones thereby using inference rules from the premises to the conclusions: a new element is added if it is somehow related to the old elements which enjoy a property. The coinductive techniques, as duals of inductive ones [47, 48] , are used to build sets starting from a biggest one, where all elements are supposed to be included – hence postulating that all of them belong to the set – and removing those which don't fulfil the condition expressed by an inference rule, which is used *backward*, namely from the conclusion toward the premises. Bisimilarity is one of the most pervasive techniques for checking equivalence among procedures, it is based on the idea that two processes are equivalent when they behave the same when they interact with the external environment.

Among the various notions of bisimulation which are known to be amenable to higher-order programs, the simplest one is certainly Abramsky's applicative bisimulation [1, 25], in which terms are seen as interactive objects where the interaction with their environment consists in taking input arguments or outputting observable results. Remarkably, the concept of bisimulation is not univocal, since many relations of bisimulation can be arranged on the same set of objects, therefore the union of all the bisimulation relations is taken as well-founded comparison relation among terms, and it is called bisimilarity. In deterministic languages, when used as an equivalence relation among programs, bisimilarity has been proved to be a very powerful tool, since it has been shown to have both the properties of soundness (which means that it is included) and completeness (which is understood as to include) with respect to the context equivalent relation (e.g. [44]). Applicative bisimulation is therefore well-known to be *fully-abstract*, hence sound and complete,

w.r.t. context equivalence when instantiated on plain, untyped, *deterministic* λ-calculi [1, 4].

Even though the first attempts to extend the concept of bisimilarity toward *non-deterministic* higher–order languages have been successfully accomplished since the latest nineties [40], it is somehow underwhelming that in such nondeterministic environments bisimilarity, even if it is sound with respect to context equivalence, doesn't fulfill the criterium to be fully abstract. When extended to *probabilistic* systems [39], the notion of bisimilarity necessarily requires to define a more sophisticated topological structure as the Labelled Marcov Chains (LMC). Probability is inserted into the λ calculus by means of a choice operator, which allows many possible paths in the calculation procedure and the LMC provides the way to manage the set of possible transitions undergone by each program toward other ones when some action of the system is performed. As for the assessment between context equivalence and bisimilarity in probabilistic languages, the situation is more complicated: while applicative bisimilarity is invariably a congruence, thus sound for context equivalence, completeness generally fails [44, 40], even if some unexpected positive results have recently been obtained on this subject [10, 54].

The previous theme of equivalence overlaps with linearity, which is the requirement to use exactly once every variable declared in a program. Linearity in computer language theory, especially in the presence of typed environments, is a straight derivation of linear logic conceived by Girard as a refinement of intuitionistic logic [24]. Connections between linear logic and linear typed languages are given by Curry–Howard correspondence: whatever type judgement finds its analogous in a logical statement. Does applicative bisimulation work well when the underlying calculus has linear types? The question has been replied positively, but only for deterministic λ-calculi [9, 8]. The soundness of the bisimulation in the frame of the contextual equivalence relation, fails also for different, sligtly complex, definitions of bisimulation such as the *environmental bisimulation* [33, 49]. In this thesis, the constraint of linearity is introduced from the very beginning, in view of the purpose to extend the results obtained for the deterministic and probabilistic languages, to a

quantum calculi where the impossibility to clone variables becomes a crucial bond. This is the so called no cloning theorem which expresses the impossibility to create a copy of a quantum state – specifically a qubit – without observing it and hence destroying superposition [32, 22].

1.2 On Quantum Computation

The increasing credit paid to quantum languages is justifiable because of its potentiality to overcome classical limits, improve the efficiency and decrease the time of computation by exploiting the parallelism intrinsically embedded in quantum mechanical processes, which allows to explore at the same time, with a certain probability, several computation paths. At a logical level, a quantum computer consists of a set of operators, the so called *quantum gates* which are assigned to the elaboration and manipulation of quantum data, stored in the computer memory in form of quantum bits (qubit): thus a quantum algorithm is a sequence of quantum gates, but since the qubits are physically comparable to vectors rather than to numbers, the quantum gates act in a more complex way than their classic equivalent, by exploring simultaneously, during the calculus, a plurality of possibilities. The structure of a quantum algorithm is such that, during its execution, there are basically two kinds of allowed operations: unitary transformations – which have as classical correspondent the sequence of gate operations performed on the bits by classical circuits – and the measurement, which is the observation of the final result.

Among the other quantum algorithms, we recall here Shor's algorithm [55] for the factorization of natural numbers, that given an integer finds its prime factors, and Grover algorithm [28] to search an item in a list, which has improved the classical one. The first one is mostly important because security protocols for the privacy across the network communications, encrypt data exploiting the factorization of a given number in primes to encode the sent data [46]. Shor's algorithm requires a polynomial time in the size of the input number entailing an exponential speedup with respect to the classical ones: indeed no classical algorithm is known that can

factor an integer in polynomial time. Grover's algorithm gives a quadratic speedup.

Quantum computation is traditionally introduced at low level, presenting the programs as an ordered series of quantum gates [43], or modelling it as a quantum Turing Machine [20], where both data and control are treated as quantum systems, writing them as a superposition of classical states. Parallely to these purely quantum patterns, some attempts to build quantum programming languages endowing the computation with a set of operational semantics rules have been done [53, 52]. There quantum variables as well classical ones, are permitted but they are controlled and processed by classical devices and programs, represented by the terms of the language. Thus various extensions of classical λ calculus have been used to give the operational semantics rules for first–order quantum calculus [35, 34] in a typed frame. These methods of analysis have been efficiently summarized by the slogan "quantum data, classical control" [50].

Whenever the analysis is limited to first order languages, quantum algorithms and procedures may be compared as linear operators in a linear vector space [6], claiming their equivalence if, by executing them on the finite number of space basis vectors they give the same result. Various other techniques for comparing terms of a quantum language have been studied and adopted for higher order quantum languages, as denotational semantics and context equivalence [52]. In quantum environment too, the notion of context equivalence leans on the demand that two programs have the same observational behaviour whenever they dived inside a whichever context of an *observable* type. This means that the analysis is focused on "ground" types contexts. The concept of quantum context equivalence is then compared with those of *bisimulation* and *denotational* equivalence [52] and trace equivalence as well. On the other hand, a number of notions of quantum bisimulation have been introduced and studied as an efficient means to compare quantum procedures in the framework of process algebra [23, 21, 17] modelling the equivalence between procedures for the communications and the concurrency in quantum systems.

1.3 Contributions

The rest of this thesis is organized as follows: in the next chapter a simply typed, purely deterministic and linear language called ℓST_λ is introduced, giving a set of typing rules and a set of operational semantics rules, in a call–by–value reduction strategy. After having proved the normalization of this calculus, the notions of context equivalence and applicative bisimulation are given: notice that context equivalence is defined on a set of *linear* contexts, where indeed the marker *must* appear only once. Subsequently, the basics of applicative bisimulation are presented, instantiated on ℓST_λ. Within this scope we show that, when instantiated on linear λ-calculi, bisimulation is both sound and complete with respect to linear context equivalence.

Afterwards, in chapter three, the language is enriched with a probabilistic choice operator with the purpose of discussing the impact of probabilities to equivalences and bisimilarity. Keeping the linearity hypothesis, a set of semantics rules is given introducing the notion of probabilistic context equivalence for *linear* contexts. The probabilistic variation on ℓST_λ is called ℓPST_λ: hereby a definition of probabilistic similarity is introduced, where newly added features in the language are shown to correspond to mild variations in the underlying transitions system, which in presence of probabilistic choice becomes a LMC. Exploiting Howe's techniques, the property of compatibility for bisimilarity is shown to be valid also in probabilistic environment: the main contributions in this chapter are congruence results for applicative bisimilarity in probabilistic linear λ-calculi, with soundness with respect to context equivalence as an easy corollary.

In the last part, the ℓST_λ is extended introducing the syntactic elements and operational tools to implement a quantum language. In particular we enrich the former deterministic language with a set of unitary operators, which are a mathematical representation of the quantum gates necessary for the implementation of the quantum algorithms, with a measurement operator $\texttt{meas}_i$, antagonist with respect to the operator `new`, which is entrusted to the creation of quantum variables. Each

term of the quantum variation on ℓST_λ, dubbed ℓQST_λ, *always* requires to be used together with its quantum register, which keeps track of the position of variables, that appear into the term as a linear superposition of "classical" configurations: this is the notion of quantum closure, which is a pair built with the quantum register as first component and the term as second one. Subsequently, we give a set of operational rules for quantum closures, resorting to the results attained for the linear probabilistic case and we introduce the notion of bisimilarity for ℓQST_λ, showing that it is a congruence. A final section of this part is devoted to the discussion about full-abstraction with respect to quantum context equivalence.

We see this thesis as the first successful attempt to apply coinductive techniques to quantum, higher-order calculi. The literature offers some ideas and results about bisimulation and simulation in the context of quantum process algebras [23, 17, 14]. Deep relations between quantum computation and coalgebras have recently been discovered [31]. None of the cited works, however, deals with higher-order functions, this is the main novelty of this work [11, 36].

Chapter 2

Setting the Deterministic Framework

2.1 Linear λ-Calculi: A Minimal Core

In this section, a simple linear λ-calculus called ℓST_λ will be introduced, together with the basics of its operational semantics. *Terms* and *values* are generated by the following grammar:

$$e, f, g ::= v \;\Big|\; ef \;\Big|\; \texttt{if}\ e\ \texttt{then}\ f\ \texttt{else}\ g \;\Big|\; \texttt{let}\ e\ \texttt{be}\ \langle x, x\rangle\ \texttt{in}\ f \;\Big|\; \Omega;$$

$$v, u ::= x \;\Big|\; \texttt{tt} \;\Big|\; \texttt{ff} \;\Big|\; \lambda x.e \;\Big|\; \langle v, u\rangle. \tag{2.1}$$

Here `tt` and `ff` are the usual *boolean constants*, the term $\lambda\, x.e$ is the symbol for a λ *abstraction* namely for the name of a generic function of argument x, whilst ef, said to be an *application*, represents a function which has the term f as argument, `if` e `then` f `else` g is as usual the constructor for *conditional choice*, $\langle v, u\rangle$ – whose components are values – is called a *pair*. ℓST_λ gives, however, the possibility to built an arbitrary pair using the semantic equivalence

$$\langle e, f\rangle = \lambda\, x.\lambda\, y.\langle x, y\rangle ef. \tag{2.2}$$

Observe the presence not only of abstractions and applications, but also of pairs, and of basic constructions for booleans. Finally, terms include a constant Ω for divergence. The symbol `b` is a metavariable for truth values, i.e. `b` stands for either `tt` or `ff`.

Terms of the language whether they are constants, variables or expressions, are defined within the scope of a number greater than or equal to zero of distinct assignments of the form $x_1 : A_1, \dots x_N : A_N$, where for each variable x_i the corresponding type A_i is declared: such a set of assignments is called a *typing context* or *environment* and generally denoted by Γ or Δ, or by another capital Greek letter. More precisely, Γ may be seen as a partial function which assigns a type to each variable which belongs to a given domain $dom(\Gamma)$, which is a list of distinct variables of type $A_i = \Gamma(x_i)$. By the notation $\Gamma, y : B$ we mean the function obtained extending the domain of Γ to the new variable y.

A *typing judgement*, or *assignment*, is a statement of the form

$$\Gamma \vdash e : A,$$

which means that in the typing context Γ it is possible to derive, applying the rule of the language displayed in Table 2.1, the type of the term e to be A. A typing judgement is assumed valid if it is derived applying exclusively these rules. The list $dom(\Gamma)$ is the set of the free variables of e, sometimes denoted by $\mathtt{fv}(e)$. A term is said to be *closed* if it doesn't contain free variables, hence if $\mathtt{fv}(e) = \emptyset$. A closed term is also called a program. Since we need a way to enforce linearity, i.e., the fact that functions use their arguments *exactly* once, we operate in the framework of a linear type system whose language of *types* is the following:

$$A, B ::= \mathtt{bool} \;\Big|\; B \multimap A \;\Big|\; A \otimes B. \tag{2.3}$$

$\mathcal{Y}$ is the set of all types. Typing rules are standard, even if, since the linearity constraint forces the same variable to appear exactly once, in the rules the domains of typing contexts referring to different subterms are disjoint. Rules are listed in Figure 2.1: observe the presence of the same typing context in both branches of the conditioned choice, in rule $(tj - if)$. The set $\mathcal{T}^{\ell ST_\lambda}_{\Gamma,A}$ contains all terms e such that $\Gamma \vdash e : A$, $\mathcal{T}^{\ell ST_\lambda}_{\emptyset,A}$ is usually written as $\mathcal{T}^{\ell ST_\lambda}_{A}$. Notations like $\mathcal{V}^{\ell ST_\lambda}_{\Gamma,A}$ or $\mathcal{V}^{\ell ST_\lambda}_{A}$ are the analogues for values of the corresponding notations for terms.

The divergence is treated apart with a special rule $(tj - div)$. A term is called *divergent* if, both, it doesn't belong to the set $\mathcal{V}^{\ell ST_\lambda}$ and it can't reduce. The set of

Type judgement rule	Name
$\dfrac{}{\emptyset \vdash \mathtt{b} : \mathtt{bool}}$	$(tj - con)$
$\dfrac{}{x : A \vdash x : A}$	$(tj - var)$
$\dfrac{\Gamma, x : A \vdash e : B}{\Gamma \vdash \lambda x.e : A \multimap B}$	$(tj - abs)$
$\dfrac{\Gamma \vdash e : A \multimap B \qquad \Delta \vdash f : A}{\Gamma, \Delta \vdash ef : B}$	$(tj - app)$
$\dfrac{\Gamma \vdash e : \mathtt{bool} \qquad \Delta \vdash f : A \qquad \Delta \vdash g : A}{\Gamma, \Delta \vdash \mathtt{if}\ e\ \mathtt{then}\ f\ \mathtt{else}\ g : A}$	$(tj - if)$
$\dfrac{\Gamma \vdash e : A \qquad \Delta \vdash f : B}{\Gamma, \Delta \vdash \langle e, f \rangle : A \otimes B}$	$(tj - pai)$
$\dfrac{\Gamma, x : X, y : Y \vdash e : A \qquad \Delta \vdash f : X \otimes Y}{\Gamma, \Delta \vdash \mathtt{let}\ f\ \mathtt{be}\ \langle x, y \rangle\ \mathtt{in}\ e :}$	$(tj - let)$
$\dfrac{}{\Gamma \vdash \Omega : A}$	$(tj - div)$

Figure 2.1: Typing Rules: since we are in a linear language Γ and Δ have disjoint domains, as well as the variables x and y, appearing in rules $(tj - abs)$ and $(tj - let)$ don't belong to $dom(\Gamma)$ and $dom(\Delta)$.

divergent terms is generated by the syntax tree

$$o ::= \Omega \mid vo \mid oe \mid \mathtt{if}\ o\ \mathtt{then}\ e\ \mathtt{else}\ e \mid \mathtt{let}\ o\ \mathtt{be}\ \langle x, y \rangle\ \mathtt{in}\ e. \tag{2.4}$$

Following [8], we chose to characterize the divergence with the constant term Ω, rather than through the standard notion of fixed point operator $\mathtt{fix}\, x.e$. This choice could be motivated by the sake of simplicity, since it reduces both the number of semantics rules (see Figure 2.3) and the the number of cases which must be treated in the proofs of the lemmas and theorems. Moreover, depicting the convergence through a fixed point operator, requires to allow, in the last step, that a term does not use a variables appearing in the typing context and this will force to give up

to the linearity requirement. Endowing ℓST_λ with call-by-value small-step or big-step semantics poses no significant problem. With regard to small-step reduction one formally introduces a binary relation $\rightarrow \subseteq \mathcal{T}_A^{\ell ST_\lambda} \times \mathcal{T}_A^{\ell ST_\lambda}$ between closed terms of any type by the usual rule for β-reduction, the natural rule for the conditional operator, and the following rule:

$$\texttt{let } \langle v, u\rangle \texttt{ be } \langle x, y\rangle \texttt{ in } e \rightarrow e\{v/x, u/y\}.$$

Terms are evaluated by mean of the call–by–value reduction strategy, defined by structural induction as displayed in Figure 2.2. Similarly, one can define a big-

Small step semantics rule	Name
$\dfrac{}{(\lambda x.e)v \rightarrow e\{v/x\}}$	(app_β)
$\dfrac{e_1 \rightarrow f}{e_1 e_2 \rightarrow f e_2}$	(app_L)
$\dfrac{e \rightarrow f}{ve \rightarrow vf}$	(app_R)
$\dfrac{}{\texttt{if tt then } e_1 \texttt{ else } e_2 \rightarrow e_1}$	$(if - ax_{\texttt{tt}})$
$\dfrac{}{\texttt{if ff then } e_1 \texttt{ else } e_2 \rightarrow e_2}$	$(if - ax_{\texttt{ff}})$
$\dfrac{e_1 \rightarrow f}{\texttt{if } e_1 \texttt{ then } e_2 \texttt{ else } e_3 \rightarrow \texttt{if } f \texttt{ then } e_2 \texttt{ else } e_3}$	(if)
$\dfrac{}{\texttt{let } \langle v, u\rangle \texttt{ be } \langle x, y\rangle \texttt{ in } e \rightarrow e\{v/x, u/y\}}$	$(let - ax)$
$\dfrac{e_1 \rightarrow f}{\texttt{let } e_1 \texttt{ be } \langle x, y\rangle \texttt{ in } e_2 \rightarrow \texttt{let } f \texttt{ be } \langle x, y\rangle \texttt{ in } e_2}$	(let)

Figure 2.2: Operational semantics rules of ℓST_λ.

step evaluation relation $\Downarrow \subseteq \mathcal{T}_A^{\ell ST_\lambda} \times \mathcal{V}_A^{\ell ST_\lambda}$, between closed terms and values by a completely standard set of rules, shown in Table 2.3. Here the semantics of each

term is fully determined by the knowledge of the semantics of its parts, where the semantics of a term in this deterministic approach is intended to be the unique value the term evaluates to. In Table 2.3 the big–step evaluation rules are displayed. A

BIG STEP SEMANTICS RULE	NAME
$\dfrac{}{v \Downarrow v}$	$(v \Downarrow)$
$\dfrac{e_1 \Downarrow \lambda x.f \qquad e_2 \Downarrow u \qquad f\{u/x\} \Downarrow v}{e_1 e_2 \Downarrow v}$	$(app \Downarrow)$
$\dfrac{e_1 \Downarrow \mathtt{tt} \qquad e_2 \Downarrow v}{(\mathtt{if}\ e_1\ \mathtt{then}\ e_2\ \mathtt{else}\ e_3) \Downarrow v}$	$(if_{\mathtt{tt}} \Downarrow)$
$\dfrac{e_1 \Downarrow \mathtt{ff} \qquad e_3 \Downarrow v}{(\mathtt{if}\ e_1\ \mathtt{then}\ e_2\ \mathtt{else}\ e_3) \Downarrow v}$	$(if_{\mathtt{ff}} \Downarrow)$
$\dfrac{e_1 \Downarrow \langle u_1, u_2\rangle \qquad e_2\{u_1/x, u_2/y\} \Downarrow v}{(\mathtt{let}\ e_1\ \mathtt{be}\ \langle x, y\rangle\ \mathtt{in}\ e_2) \Downarrow v}$	$(let \Downarrow)$

Figure 2.3: Big–step semantics of the language ℓST_λ.

program of a computing abstract machine finds its correspondent on closed λ–terms of the language – possibly nested – defined by the grammar (2.1). Moreover, the control flow of an abstract machine, namely the sequence of instructions as the data entry and the operations on them, finds its analogous in the derivation rules of the operational semantics, listed in Figure 2.2. Thus the execution of a program is simulated by a derivation tree built with the operational semantics of the language itself.

As it has been remarked, linearity is a peculiar characteristic of ℓST_λ, entailing that each variable appearing within the domain of each typing context is used in the scope of the terms exactly once as in the following examples:

$$\begin{aligned} \mathtt{not} &= \lambda x.\mathtt{if}\ x\ \mathtt{then}\ \mathtt{ff}\ \mathtt{else}\ \mathtt{tt} \\ \mathtt{and} &= \lambda x.\lambda y.\mathtt{if}\ x\ \mathtt{then}\ y\ \mathtt{else}\ (\mathtt{if}\ y\ \mathtt{then}\ \mathtt{ff}\ \mathtt{else}\ \mathtt{ff}) \\ \mathtt{or} &= \lambda x.\lambda y.\mathtt{if}\ x\ \mathtt{then}\ (\mathtt{if}\ y\ \mathtt{then}\ \mathtt{tt}\ \mathtt{else}\ \mathtt{tt})\ \mathtt{else}\ y \end{aligned} \quad (2.5)$$

This is necessary in view to be able to exploit the language in a quantum computing framework. The expressive power of the just-introduced calculus is rather poor. Nonetheless, by virtue of the fact that every boolean formula can be written in the conjunctive normal form, namely as a conjunction of disjunctions, it can be proved that the language is complete for first-order computation over booleans, in the following sense: for every function $F : \{\mathtt{tt}, \mathtt{ff}\}^n \to \{\mathtt{tt}, \mathtt{ff}\}$, there is a term which *computes* F, i.e. a term e_F such that $e_F\langle b_1, \ldots, b_n\rangle \to^* F(b_1, \ldots, b_n)$ for every $b_1, \ldots, b_n \in \{\mathtt{tt}, \mathtt{ff}\}^n$. Indeed, even if copying and erasing bits is not in principle allowed, one could anyway encode, e.g., duplication as the following combinator of type $\mathtt{bool} \multimap \mathtt{bool} \otimes \mathtt{bool}$: $\lambda x.\mathtt{if}\ x\ \mathtt{then}\ \langle \mathtt{tt}, \mathtt{tt}\rangle\ \mathtt{else}\ \langle \mathtt{ff}, \mathtt{ff}\rangle$. Similarly, if $\Gamma \vdash e : A$ and x is a fresh variable, one can easily find a term $\mathtt{weak}\ x\ \mathtt{in}\ e$ such that $\Gamma, x : \mathtt{bool} \vdash \mathtt{weak}\ x\ \mathtt{in}\ e : A$ and $\mathtt{weak}\ \mathtt{b}\ \mathtt{in}\ e$ behaves like e for every $\mathtt{b} \in \{\mathtt{ff}, \mathtt{tt}\}$; the term is defined as

$$\mathtt{weak}\ x\ \mathtt{in}\ e \stackrel{\text{def}}{=} \mathtt{if}\ x\ \mathtt{then}\ e\ \mathtt{else}\ e.$$

2.2 Normalization of Closed Terms.

We say a closed term to be in *normal form* when it can not reduce anymore. Intuitively it is clear that the idea of normal form of a term is tightly related with that of value, as indeed the following lemma shows.

Lemma 2.1 (Progress)**.** *Every term that can not be reduced in an empty typing context is either a value, thus it belongs to the set $\mathcal{V}^{\ell PST_\lambda}$, or it is a divergent term.*

Proof. By induction on the structure of the terms of the set $\mathcal{T}^{\ell ST_\lambda}$. If $e = \mathtt{tt}$, $e = \mathtt{ff}$, $e = \lambda x.f$, there is nothing to prove since the term can not reduce and it is indeed already a value. Besides, if $e = \Omega$, it can't reduce by definition and it is a divergence according to the definition (2.4) then there is nothin to prove.

$-e = f_1 f_2-$ Then we have the following derivation for the type judgement

$$\frac{\emptyset \vdash f_1 : B \multimap A \qquad \emptyset \vdash f_2 : B}{\emptyset \vdash f_1 f_2 : A}\,(tj - app).$$

Now several possibilities can occur:

- if f_1 is a value then
  - if $f_2 = v$ is a value in turn then $f_1 f_2$ can reduce by application of (app_β);
  - if $f_2 = o$ is a divergence then also $f_1 f_2$ is a divergence according to definition (2.4);
  - if $f_2 \to g$ then applying (app_R) one finds $f_1 f_2 \to f_1 g$ and the term reduces, therefore it is not a value.
- if $f_1 = o$ is a divergence the the term itself is divergent according to (2.4);
- if f_1 is not a value and neither a divergence, then by induction hypothesis the reduction $f_1 \to g$ can occur and by application of (app_L) one finds $f_1 f_2 \to g f_2$, therefore the term can not be a value. In each one of these cases which have been examined, $f_1 f_2$ can reduce, unless it is a divergence: thus it never can be a value.

$-e = (\texttt{if}\ f_1\ \texttt{then}\ f_2\ \texttt{else}\ f_3) -$ Here the typing judgement has the derivation tree

$$\frac{\emptyset \vdash f_1 : \texttt{bool} \qquad \emptyset \vdash f_2 : A \qquad \emptyset \vdash f_3 : A}{\emptyset \vdash \texttt{if}\ f_1\ \texttt{then}\ f_2\ \texttt{else}\ f_3 : A}\,(tj - if).$$

Here three cases must be distinguished:

- if f_1 is a value then, according to the typing judgement above, necessarily it must be a boolean value: if $f_1 = \texttt{tt}$ the rule $(if - ax_{\texttt{tt}})$ can be applied and we get $(\texttt{if}\ f_1\ \texttt{then}\ f_2\ \texttt{else}\ f_3) \to f_2$, while if $f_1 = \texttt{ff}$, applying $(if - ax_{\texttt{ff}})$ one obtains $(\texttt{if}\ f_1\ \texttt{then}\ f_2\ \texttt{else}\ f_3) \to f_2$;
- if $f = o$ then the whole term is a divergence according to the definition (2.4);
- finally if f_1 is not a value and neither a divergence, then by induction hypothesis $f_1 \to g$ and the small step reduction rule (if) tells us that the reduction $(\texttt{if}\ f_1\ \texttt{then}\ f_2\ \texttt{else}\ f_3) \to (\texttt{if}\ g\ \texttt{then}\ f_2\ \texttt{else}\ f_3)$ occurs.

Anyway, following the previous analysis ($\texttt{if}\ f_1\ \texttt{then}\ f_2\ \texttt{else}\ f_3$) is a reducible term or a divergence, thus it can't be a value.

$-e = (\texttt{let}\ f_1\ \texttt{be}\ \langle x, y\rangle\ \texttt{in}\ f_2)-$ Then the derivation three for typing judgement is

$$\frac{\emptyset \vdash f_1 : B \otimes E \qquad x : B, z : E \vdash f_2 : A}{\emptyset \vdash (\texttt{let}\ f_1\ \texttt{be}\ \langle x, y\rangle\ \texttt{in}\ f_2) : A}\,(tj - let).$$

and the following three cases are possible:

- f_1 is a value, whence the type inference says that it must be in the form $f_1 = \langle v, u\rangle$ and by application of the small step reduction rule $(let - ax)$ the following reduction occurs $(\texttt{let}\ f_1\ \texttt{be}\ \langle x, y\rangle\ \texttt{in}\ f_2) \to f_2\{v/x, u/y\}$.
- f_1 is a divergence and thus the whole term is.
- f_1 is not a value neither a divergence and, recalling induction hypothesis we get $f_1 \to g$, whence by application of (let) we have $(\texttt{let}\ f_1\ \texttt{be}\ \langle x, y\rangle\ \texttt{in}\ f_2) \to (\texttt{let}\ g\ \texttt{be}\ \langle x, y\rangle\ \texttt{in}\ f_2)$.

Therefore also the term $e = (\texttt{let}\ f_1\ \texttt{be}\ \langle x, y\rangle\ \texttt{in}\ f_2)$ is anyway a reducible or divergent. □

If a program can be put in normal form by a (finite) sequence of reduction steps we say that it normalizes. Since in nondeterministic languages a different set of the semantics rules can lead to a multiplicity of reduction paths, possibly evaluating to different values [7], two different ways to normalize can be distinguished:

weak normalization A closed term $\emptyset \vdash e : A$ is weakly normalizable if *at least* a reduction path exists which leads the term in normal form.

strong normalization e such that $\emptyset \vdash e : A$ is strong normalizable if *every* possible reduction sequence terminates in a normal form with a finite number of steps.

It is not particularly difficult to show that in ℓST_λ every terms strongly normalizes: the intuitive argument is that in every linear language every reduction step decreases the size of terms involved. With the purpose to prove it, it is necessary to define the

e		$\mid e \mid$
constants and variables	x, c	1
λ abstractions	$\lambda x.f$	$\mid f \mid +1$
applications	$f_1 f_2$	$\mid f_1 \mid + \mid f_2 \mid +1$
if	$\texttt{if } f_1 \texttt{ then } f_2 \texttt{ else } f_3$	$\mid f_1 \mid + \max(\mid f_2 \mid, \mid f_3 \mid)$
let	$\texttt{let } f_1 \texttt{ be } \langle x, y \rangle \texttt{ in } f_2$	$\mid f_1 \mid + \mid f_2 \mid +1$
pairs	$\langle f_1, f_2 \rangle$	$\mid f_1 \mid + \mid f_2 \mid +1$

Figure 2.4: Definition of size.

notion of size of a term: the size of e is denoted by $\mid e \mid$ and recursively defined on the structure of e itself with a set of rules shown in Figure 2.4. The notion of size enters fully into the statement of the following substitution lemma.

Lemma 2.2 (Substitution)**.** *Let $e \in \mathcal{T}^{\ell ST_\lambda}_{\Gamma,A}$ be a term such that $\Gamma, z : E \vdash e : A$ and let $\Delta \vdash g : E$ be a valid type judgement, then the two following results are both valid:*

2.2.1 ▶ $\Gamma, \Delta \vdash e\{g/z\} : A$

2.2.2 ▶ $\mid e\{g/z\} \mid = \mid e \mid + \mid g \mid -1$

Proof. The proof is by induction on the structure of e.

If $e \equiv c$, due to the typing rules for the constants, necessarily $dom(\Gamma) = \emptyset$. Thus, this case is impossible.

$- e \equiv x -$

2.2.1 we are under the hypothesis $\Gamma, z : E \vdash x : A \wedge \Delta \vdash g : E$. Since we are in a linear framework, necessarily $dom(\Gamma) = \emptyset$ and $z \equiv x$, as well as E coincides with A. Therefore it holds the relationship $\Delta \vdash z\{g/z\} : A$.

2.2.2 Since here the type assignment is $z : E \vdash x : A \wedge \Delta \vdash g : E$, in this case A and E must necessarily be the same type; moreover $\mid e \mid = 1$ and $\mid e\{g/z\} \mid = \mid g \mid = 1 + \mid g \mid -1 = \mid x \mid + \mid g \mid -1$.

$-e \equiv \lambda x.f -$

2.2.1 The hypothesis is $\Gamma, z : E \vdash \lambda x.f : B \multimap A \wedge \Delta \vdash g : E$ and its first statement may be derived only by the typing rule $(tj - abs)$ of Table 2.1, whence we have

$$\frac{\Gamma, z : E, x : B \vdash f : A}{\Gamma, z : E \vdash \lambda x.f : B \multimap A}\,(tj - abs). \tag{2.6}$$

On the premise of the previous rule (2.6) we can apply the induction hypothesis, which is $\Gamma, z : E, x : B \vdash f : A \wedge \Delta \vdash g : E \Rightarrow \Gamma, x : B, \Delta \vdash f\{g/z\} : A$. Therefore, taking this result as a premise in (2.6) we can rewrite

$$\frac{\Gamma, \Delta, x : B \vdash f\{g/z\} : A}{\Gamma, \Delta \vdash \lambda x.f\{g/z\} : B \multimap A}\,(tj - abs), \tag{2.7}$$

which proves the thesis.

2.2.2 By induction hypothesis $\mid f\{g/z\} \mid = \mid f \mid + \mid g \mid -1$, moreover, by definition of size for λ–abstractions $\mid \lambda x.f\{g/z\} \mid = \mid f\{g/z\} \mid +1$. Thus using inductive hypothesis we have

$$\mid e\{z/g\} \mid = \mid \lambda x.f\{g/z\} \mid = \mid f \mid + \mid g \mid = \mid \lambda x.f \mid + \mid g \mid -1.$$

$-e \equiv f_1 f_2 -$

2.2.1 We write the hypothesis as $\Gamma_1, \Gamma_2, z : E \vdash f_1 f_2 : A$ and it has $(tj - app)$ as last rule. Since because of the linearity hypothesis, z belongs either to f_1 or f_2, but not to both of them, we can suppose that z belongs to f_1 without loosing generality, so the rule becomes

$$\frac{\Gamma_1, z : E \vdash f_1 : B \multimap A \qquad \Gamma_2 \vdash f_2 : B}{\Gamma_1, \Gamma_2, z : E \vdash f_1 f_2 : A}\,(tj - app). \tag{2.8}$$

Now the induction hypothesis on f_1 gives $\Gamma_1, z : E \vdash f_1 : A \wedge \Delta \vdash g : E \Rightarrow \Gamma_1, \Delta \vdash f_1\{g/z\} : A$ and taking it as a premise in the rule $(tj - app)$ we get

$$\frac{\Gamma_1, \Delta \vdash f_1\{g/z\} : B \multimap A \qquad \Gamma_2 \vdash f_2 : B}{\Gamma_1, \Gamma_2, \Delta \vdash f_1 f_2\{g/z\} : A}\,(tj - app)$$

which is the thesis.

2.2.2 For we are under the hypothesis of linearity, only one between f_1 and f_2 can depend on the free variable z. Let suppose that f_1 depends on z and f_2 doesn't contain it: by induction hypothesis $\mid f_1\{g/x\} \mid = \mid f_1 \mid + \mid g \mid -1$, then using the definition of size for the application given in Figure 2.4 together with induction hypothesis on f_1 we find $\mid e\{z/g\} \mid = \mid f_1 f_2\{g/z\} \mid = \mid f_1\{g/z\} \mid + \mid f_2 \mid +1 = \mid f_1 \mid + \mid g \mid -1 + \mid f_2 \mid +1 = \mid f_1 f_2 \mid + \mid g \mid -1$.

$-\ e \equiv (\texttt{if}\ f_1\ \texttt{then}\ f_2\ \texttt{else}\ f_3)\ -$

2.2.1 We start from the hypothesis $\Gamma_1, \Gamma_2 \vdash (\texttt{if}\ f_1\ \texttt{then}\ f_2\ \texttt{else}\ f_3) : A \wedge \Delta \vdash g : E$ observing that the first part must have the rule $(tj - if)$ as last derivation in the typing tree.

Supposing, without loss of generality, that z is a free variable of both of the subterms f_2 and f_3, since by linearity it can't belong to both of the domains of typing environments Γ_1 and Γ_2, we write

$$\frac{\Gamma_1 \vdash f_1 : \texttt{bool} \qquad \Gamma_2, z : E \vdash f_2 : A \qquad \Gamma_2, z : E \vdash f_3 : A}{\Gamma_1, \Gamma_2, z : E \vdash (\texttt{if}\ f_1\ \texttt{then}\ f_2\ \texttt{else}\ f_3) : A}\,(tj - if). \tag{2.9}$$

On the subterms f_2 and f_3 the induction hypothesis can be applied. For example for f_3 we may write the statement: $\Gamma_2, z : E \vdash f_3 : A \wedge \Delta \vdash g : E \Rightarrow \Gamma_2, \Delta \vdash f_3\{g/z\} : A$, thus inserting this result in the premises of (2.9) yields:

$$\frac{\Gamma_1 \vdash f_1 : \texttt{bool} \qquad \Gamma_2, \Delta \vdash f_2\{g/z\} : A \qquad \Gamma_2, \Delta \vdash f_3\{g/z\} : A}{\Gamma_1\Gamma_2, \Delta \vdash (\texttt{if}\ f_1\ \texttt{then}\ f_2\ \texttt{else}\ f_3)\,\{g/z\} : A}\,(tj - if), \tag{2.10}$$

as it must be proved.

2.2.2 Under hypothesis of linearity again we must choose which subterm of e should depend on the variable x. Let us suppose that f_1 is such a subterm and let us apply the inductive hypothesis to f_1 obtaining: $\mid f_1\{g/x\} \mid = \mid f_1 \mid + \mid g \mid -1$. Thus the length is, by definition (Figure 2.4):

$$\mid (\texttt{if}\ f_1\ \texttt{then}\ f_2\ \texttt{else}\ f_3)\,\{g/z\} \mid = \mid f_1\{g/z\} \mid + \max(\mid f_2 \mid, \mid f_3 \mid) =$$
$$\mid f_1 \mid + \mid g \mid -1 + \max(\mid f_2 \mid, \mid f_3 \mid) = \mid \texttt{if}\ f_1\ \texttt{then}\ f_2\ \texttt{else}\ f_3 \mid + \mid g \mid -1$$

thus the statement (2.2.2), $\mid e\{g/z\} \mid = \mid e \mid + \mid g \mid -1$ has been proved.

$-e \equiv (\texttt{let}\ f_1\ \texttt{be}\ \langle x, y\rangle\ \texttt{in}\ f_2) -$

2.2.1 The hypothesis is $\Gamma_1, \Gamma_2, z : E \vdash (\texttt{let}\ f_1\ \texttt{be}\ \langle x, y\rangle\ \texttt{in}\ f_2) : A \wedge \Delta \vdash g : E$. Supposing, by linearity, that only f_1 depends on z we write the typing judgement referring us to Table 2.1

$$\frac{\Gamma_1, z : E \vdash f_1 : B \otimes B' \qquad \Gamma_2, x : B, y : B' \vdash f_2 : A}{\Gamma_1, \Gamma_2, z : E \vdash (\texttt{let}\ f_1\ \texttt{be}\ \langle x, y\rangle\ \texttt{in}\ f_2) : A}\,(tj - let),. \tag{2.11}$$

and we apply the induction hypothesis to obtain $\Gamma_1, z : E \vdash f_1 : B \otimes B' \wedge \Delta \vdash g : E \Rightarrow \Gamma_1, \Delta \vdash f_1\{g/z\} : B \otimes B'$. Thus taking this statement as a premise in (2.11) we get:

$$\frac{\Gamma_1, \Delta \vdash f_1\{g/z\} : B \otimes B' \qquad \Gamma_2, x : B, y : B' \vdash f_2 : A}{\Gamma_1, \Gamma_2, \Delta \vdash (\texttt{let}\ f_1\ \texttt{be}\ \langle x, y\rangle\ \texttt{in}\ f_2)\,\{g/z\} : A}\,(tj - let) \tag{2.12}$$

as it should have had to be proved.

2.2.2 Again we suppose that f_1 is the only subterm depending on z, then by inductive hypothesis $\mid f_1\{g/z\} \mid = \mid f_1 \mid + \mid g \mid -1$ and by definition of size for term of this form we have

$$\mid (\texttt{let}\ f_1\ \texttt{be}\ \langle x, y\rangle\ \texttt{in}\ f_2)\,\{g/z\} \mid = \mid f_1\{g/z\} \mid + \mid f_2 \mid +1 =$$
$$\mid f_1 \mid + \mid g \mid -1 + \mid f_2 \mid +1 = \mid \texttt{let}\ f_1\ \texttt{be}\ \langle x, y\rangle\ \texttt{in}\ f_2 \mid + \mid g \mid -1.$$

and again one gets the statement (2.2.2), namely $\mid e\{g/z\} \mid = \mid e \mid + \mid g \mid -1$ which should have been proved.

□

The previous lemma enables us to prove the following result which is the base of strong normalization in linear languages.

Lemma 2.3 (Determinism of one–step reduction operator and effect of reduction on size of terms in ℓST_λ)**.** *If $e \in \mathcal{T}_A^{\ell ST_\lambda}$ is such that $e \to h$, then h is unique and $\mid h \mid < \mid e \mid$.*

Proof. By induction on the structure of e and analysis of small–step semantics reduction rules.

By hypothesis e is not a value, then $e \neq x$, $e \neq c$, $e \neq \lambda\, x.f$, $e \neq \langle u_1, u_2 \rangle$; besides, e is not a divergence, therefore it can reduce.

If $e = f_1 f_2$ then we have some possibilities:

- f_1 is not a value, then rule (app_L) must be applied since we are in a leftmost reduction framework. Thus $f_1 \to g$ and $f_1 f_2 \to g f_2$ which is unequivocally defined. Besides since by induction hypothesis $\mid g \mid < \mid f_2 \mid$, we have $\mid h \mid = \mid g \mid + \mid f_2 \mid +1 < \mid f_1 \mid + \mid f_2 \mid +1 = \mid e \mid$ and the statement is proved.

- $f_1 = v$ is a value and f_2 is reducible. Then rule (app_R) must be applied which gives $f_2 \to g$ and $v f_2 \to v g$ as unique result of reduction. Moreover since by induction hypothesis $\mid g \mid < \mid f_2 \mid$ we get $\mid h \mid < \mid e \mid$.

- $f_1 f_2$ are both values, therefore by Lemma 2.1 they can not reduce. A simple type analysis shows that the term f_1 is a λ-abstraction, hence it has the structure $f_1 = \lambda\, x.g$, whence rule (app_β) will be applied to get $f_1 f_2 \to g\{f_2/x\} = h$. Let us point out that in a linear environment the variable x must appear exactly once. Now the substitution Lemma 2.2 ensures that $\mid h \mid = \mid g\{f_2/x\} \mid = \mid g \mid + \mid f_2 \mid -1 = \mid \lambda\, x.g \mid + \mid f_2 \mid -2 = \mid f_1 f_2 \mid -3 = \mid e \mid -3 < \mid e \mid$

If $e = (\texttt{if}\ f_1\ \texttt{then}\ f_2\ \texttt{else}\ f_3)$ then there are two possibilities

- f_1 is not a value, then we are in the scope of the rule (if) and e reduces unequivocally to $h = \texttt{if}\ g\ \texttt{then}\ f_2\ \texttt{else}\ f_3$ and since by induction hypothesis $\mid g \mid < \mid f_1 \mid$, we have $\mid h \mid = \mid g \mid + \max(\mid f_2 \mid, \mid f_3 \mid) < \mid f_1 \mid + \max(\mid f_2 \mid, \mid f_3 \mid) = \mid e \mid$.

- f_1 is a value, and being a boolean constant it must be $\texttt{tt}$ or $\texttt{ff}$. Supposing $f_1 = \texttt{tt}$ we must apply $(if - ax_{\texttt{tt}})$ obtaining $e \to f_2$ and of course $\mid h \mid = \mid f_2 \mid < \mid f_1 \mid + \max(\mid f_2 \mid, \mid f_3 \mid) = \mid e \mid$. Analogous is the case $f_1 = \texttt{ff}$.

If $e = (\texttt{let}\ f_1\ \texttt{be}\ \langle x, y\rangle\ \texttt{in}\ f_2)$ again some cases must be distinguished with regard whether f_1 is a value or it is not.

- If f_1 is not a value, then (let) have to be applied. In all these cases the form of h is unequivocally determined and induction hypothesis grants that $\mid h \mid < \mid e \mid$.
- If $f_1 = \langle u_1, u_2\rangle$ is a value then only rule $(let{-}ax)$ may be applied thus obtaining $e \to f_2\{u_1/x, u_2/y\}$ and the form of h is determined. Invoking substitution lemma here we have $\mid h \mid = \mid f_2 \mid + \mid f_1 \mid -2 = \mid e \mid -3 < \mid e \mid$.

This concludes the proof. □

Theorem 2.1 (Strong normalization in linear case.)**.** *Each term $e \in \mathcal{T}^{\ell ST_\lambda}$ has a normal form.*

Proof. By induction on the size of e. If the term is a value or a divergent term, then it is irreducible and there is nothing to prove, otherwise Lemma 2.3 ensures that $e \to h$ with $\mid h \mid < \mid e \mid$. Under this assumption, by induction hypothesis there must exist a normal form g such that $h \to \ldots \to g$ in at most N reduction steps, with $N < \mid h \mid$, whence one gets $M = N + 1 < \mid h \mid + 1 \leq \mid e \mid$. Besides, while the reduction relation is deterministic, also the reduction sequence is uniquely determined. □

Lemma 2.4 (Reduction is deterministic in ℓST_λ)**.** *If $e \Downarrow v$ then there exixts, unique, a (finite) sequence of one–step reductions such that $e \to f \to \cdots \to v$*

Proof. By induction on the structure of e.

$-e = v-$ If $e = x$, $e = c$, $e = \lambda x.f$, $e = \langle e_1, e_2\rangle$, there is nothing to prove since the term is a value already.

$-e = (\lambda x.f)u-$ By application of (app_β) one get $e \to f\{u/x\}$ and we obtain the thesis by linearity hypothesis (which ensures that the size of the reduced term is less) and by induction hypothesis.

$-e = (\lambda\, x.f)e_2-$ Here, since by application of $(app \Downarrow)$ we get $e_2 \Downarrow v$, the induction hypothesis on the smaller subterm and and Theorem 2.1 tell us that for a finite number of one step reduction we must have $e_2 \to g_1 \to g_2 \cdots \to \cdots \to v$, whence by application of (app_R)

$$e \to (\lambda\, x.f)g_1 \to (\lambda\, x.f)g_2 \to \cdots \to (\lambda\, x.f)v, \tag{2.13}$$

and we are reduced to the previous case.

$-e = e_1e_2-$ Since the term must reduce, it is not a divergence, thus the application of the rule $(app \Downarrow)$ leads to the existence of a $\lambda-$abstraction such that $e \Downarrow \lambda\, x.f$, thus induction hypothesis and Theorem 2.1 ensure for the existence of a finite number of one-step reductions such that $e_1 \to g_1 \to g_2 \ldots \to \ldots \to \lambda\, x.f$, whence, by application of (app_L)

$$e \to g_1e_2 \to g_2e_2 \ldots \to \ldots \to (\lambda\, x.f)e_2, \tag{2.14}$$

which bring us to previous case.

$-e = \texttt{if b then } e_2 \texttt{ else } e_3-$ Using the one–step reduction axioms $(if - ax_{\texttt{tt}})$ and $(if - ax_{\texttt{ff}})$, we find $e \to e_2$ or $e \to e_3$ depending on whether $\texttt{b} = \texttt{tt}$ or $\texttt{b} = \texttt{ff}$. Then we get thesis by induction hypothesis on the smaller terms e_2 and e_3.

$-e = \texttt{if } e_1 \texttt{ then } e_2 \texttt{ else } e_3-$ In this case, the rules $(if_{\texttt{tt}} \Downarrow)$ and $(if_{\texttt{ff}} \Downarrow)$ forecast, for the smaller term e_1, that $e_1 \Downarrow \texttt{tt}$ or $e_1 \Downarrow \texttt{ff}$. Thus, applying the induction hypothesis on e_1 we get $e \to g_1 \to g_2 \to \cdots \to \cdot\texttt{tt}$, or $e \to g_1 \to g_2 \to \cdots \to \cdot\texttt{ff}$, whence, by application of the rule (if), we obtain

$$e \to \texttt{if } g_1 \texttt{ then } e_2 \texttt{ else } e_3 \to \texttt{if } g_2 \texttt{ then } e_2 \texttt{ else } e_3 \to \cdots \to \texttt{if b then } e_2 \texttt{ else } e_3 \tag{2.15}$$

and we find in the previous case.

$-e = \texttt{let } \langle u_1, u_2\rangle \texttt{ be } \langle x, y\rangle \texttt{ in } e_2-$ In this case, applying the rule (let), we find $e \to e_2\{u_1/x, u_2/y\}$ and we get the result by induction hypothesis.

$-e = \texttt{let}\ e_1\ \texttt{be}\ \langle x, y\rangle\ \texttt{in}\ e_2-$ Here we use the rule $(let \Downarrow)$ and induction hipothesis, which ensure that $e_1 \Downarrow \langle u_1, u_2\rangle$ in a finite sequence of steps $e_1 \to g_1 \ldots \to \langle u_1, u_2\rangle$. Then the (let) rule for this term, forecast the unique path

$$e \to \texttt{let}\ g_1\ \texttt{be}\ \langle x, y\rangle\ \texttt{in}\ e_2 \to \texttt{let}\ g_2\ \texttt{be}\ \langle x, y\rangle\ \texttt{in}\ e_2 \to \cdots \to \texttt{let}\ \langle u_1, u_2\rangle\ \texttt{be}\ \langle x, y\rangle\ \texttt{in}\ e_3 \quad (2.16)$$

and we are left within the previous case. □

In the end of this section we show that the subject reduction, namely the property to preserve the type under both reduction and evaluation, holds in ℓST_λ. The following lemma will be proved:

Lemma 2.5 (Subject Reduction)**.** *If $\emptyset \vdash e : A$, $e \to f$, and $e \Downarrow v$, then both $\emptyset \vdash f : A$ and $\emptyset \vdash v : A$,*

Proof. By analysis of small-step and big–step semantics rules with induction on the structure of the terms.

If $e = x$, $e = c$, $e = \lambda x.g$, $e = \langle v, u\rangle$ there is nothing to prove as e is already a value and $f \equiv v$.

If $e = g_1 g_2$ and $\Gamma \vdash e : A$, from $(tj - app)$ we get $\Delta_1 \vdash g_1 : B \multimap A$ and $\Delta_2 \vdash g_2 : B$, whence $\Gamma \equiv \Delta_1\Delta_2$.

(⇓) Applying the evaluation rule $(app_\Downarrow)$ $\dfrac{g_1 \Downarrow \lambda x.h \qquad g_2 \Downarrow u \qquad h\{u/x\} \Downarrow v}{g_1 g_2 \Downarrow v}$, by induction hypothesis one obtains that u has the same type B of g_2, while $\lambda x.h$ has the arrow type of g_1 whence $\Delta_1, x : B \vdash h : A$, therefore, by substitution lemma 2.2 we obtain that v has type A too.

(→) The induction hypothesis used on the rule (app_L) $\dfrac{g_1 \to h}{g_1 g_2 \to h g_2}$ allows to evince the type of h which is the same as g_1, namely $B \multimap A$. Thus invoking $(tj - app)$ again we conclude that hg_2 has the same type A of e. Similar conclusions we get by the analysis of (app_R) which must be used when g_1 is a value and g_2 is not. In the last case g_1 and g_2 are both values, being $g_1 = \lambda x.h$ and $g_2 = u$, the

type of $g_1 g_2$ is preserved for the same argument used in ($\Downarrow$) case, by application of (app_β).

If $e = (\texttt{if}\ g_1\ \texttt{then}\ g_2\ \texttt{else}\ g_3)$ and $\Gamma \vdash e : A$, we may evince the structure of Γ from the typing rule $(tj - if)$ since for some typing context Δ_1 and Δ_2 it holds $\Delta_1 \vdash g_1 : \texttt{bool}$ and $\Delta_2 \vdash g_2 : A$, $\Delta_2 \vdash g_3 : A$ whence $\Gamma \equiv \Delta_1, \Delta_2$.

($\Downarrow$) From $(if_{\texttt{tt}}\ \Downarrow)$ rule we infer $\dfrac{g_1 \Downarrow \texttt{tt} \qquad g_2 \Downarrow v}{\texttt{if}\ g_1\ \texttt{then}\ g_2\ \texttt{else}\ g_3 \Downarrow v}$ whilst from the $(if_{\texttt{ff}}\ \Downarrow)$ rule it comes $\dfrac{g_1 \Downarrow \texttt{ff} \qquad g_3 \Downarrow v}{\texttt{if}\ g_1\ \texttt{then}\ g_2\ \texttt{else}\ g_3 \Downarrow v}$, and since we may apply the induction hypothesis to the premises we deduce that the type of v is A, as well as the type of both g_2 and g_3. In both cases v has the same type A of e.

($\rightarrow$) Similar conclusions we get by the analysis of small–step reduction rules $(if_{\texttt{tt}})$, $(if_{\texttt{ff}})$ and (if), whence we derive that, if $e \rightarrow f$, in all cases the type of f is the same of e.

If $e = (\texttt{let}\ g_1\ \texttt{be}\ \langle x, y\rangle\ \texttt{in}\ g_2)$ and $\Gamma \vdash e : A$, the typing judgement must be a consequence of the application of typing rule $(tj - let)$ and for some Δ_1 and Δ_2 typing contexts it holds $\Delta_1 \vdash g_1 : B \otimes E$ and $\Delta_2, x : B, y : E \vdash g_2 : A$, where $\Gamma \equiv \Delta_1, \Delta_2$.

($\Downarrow$) Applying rule $(let_\Downarrow)$, $\dfrac{g_1 \Downarrow \langle u, \nu\rangle \qquad g_2\{u/x, \nu/y\} \Downarrow v}{\texttt{let}\ g_1\ \texttt{be}\ \langle x, y\rangle\ \texttt{in}\ g_2 \Downarrow v}$ and using both the substitution lemma and the induction hypothesis, we obtain that the type of the term $g_2\{u/x, \nu/y\}$ is indeed A; thus by induction hypothesis we conclude that the type of v is A too, as it had to be proved.

($\rightarrow$) The type analysis of the terms f which appears in the statement and may be the result of the application of (let), leads to the conclusion that it must have the same type A of term e, taking into account the induction hypothesis that must be used on the subterms of the premises. The analysis of rule $(let - ax)$ leads to the same conclusion using the substitution lemma (2.2): here the term f is $g_2\{v/x, u/y\}$.

□

2.3 Context Preorder

Now it is time to introduce the notion of equivalence among terms, always referring to the elements of ℓST_λ: how could one capture the idea of equivalence for higher-order languages like the one we are examining? The canonical answer goes back to Morris, who proposed *context* equivalence (also known as *observational* equivalence) as the right way to compare terms. Roughly, two terms are context equivalent iff they behave the same when observed in any possible *context*, i.e. when tested against any possible *observer*. Formally, a context is nothing more than a term with a single occurrence of a special marker called the *hole* and denoted as $[\cdot]$. The special feature of operational contexts in a linear language as ℓST_λ, is that the marker must be used exactly once, that's why it appears, e.g., in both the branches of a conditioned choice. The contexts set, being part of the terms of the language, is recursively defined by distinguishing the context which are values from the other ones:

$$V[\cdot] ::= [\cdot] \mid \lambda\, x.C[\cdot] \mid \langle V[\cdot], u\rangle \mid \langle u, V[\cdot]\rangle, \tag{2.17a}$$

$$C[\cdot] ::= [\cdot] \mid V[\cdot] \mid fC[\cdot] \mid C[\cdot]f \mid \texttt{if}\ C[\cdot]\ \texttt{then}\ f\ \texttt{else}\ g \mid \texttt{if}\ f\ \texttt{then}\ C[\cdot]\ \texttt{else}\ D[\cdot] \mid$$
$$\mid \texttt{let}\ f\ \texttt{be}\ \langle x, y\rangle\ \texttt{in}\ C[\cdot] \mid \texttt{let}\ C[\cdot]\ \texttt{be}\ \langle x, y\rangle\ \texttt{in}\ f. \tag{2.17b}$$

Given a context $C[\cdot]$ and a term e, $C[e]$ is the term obtained by filling the single occurrence of $[\cdot]$ in $C[\cdot]$ with e: the situation for the contexts appearing in syntax tree (2.17b) is resumed in Figure 2.5. Among the elements of the syntax tree (2.17b) those contexts which have boolean type are a particular class often defined in the literature as *ground contexts.*

The typing rules for contexts, which are displayed, for ℓST_λ, in Figure 2.6, while they provide a reliable way to correctly build contexts in a typing language, are refined to specify, besides the type of the object which may be sheltered in place of the hole, also if it must be a value or a generic term. Notice that, in each rule of Figure 2.6, both the subscripts which appear immediately beside the symbol $\vdash$, can take the value e or v depending if the object which is typed is a term or a value. A typing judgement for a context assumes generally the structure

$C[\cdot]$	$C[e]$
$[\vdash_v \cdot]$	v
$[\vdash_e \cdot]$	e
$\lambda\, x.C[\vdash_e \cdot]$	$\lambda\, x.C[e]$
$fC[\vdash_e \cdot]$	$fC[e]$
$C[\vdash_e \cdot]f$	$C[e]f$
$\texttt{if}\ C[\vdash_e \cdot]\ \texttt{then}\ f\ \texttt{else}\ g$	$\texttt{if}\ C[e]\ \texttt{then}\ f\ \texttt{else}\ g$
$\texttt{if}\ f\ \texttt{then}\ C[\vdash_e \cdot]\ \texttt{else}\ D[\vdash_e \cdot]$	$\texttt{if}\ f\ \texttt{then}\ C[e]\ \texttt{else}\ D[e]$
$\texttt{let}\ f\ \texttt{be}\ \langle x, y\rangle\ \texttt{in}\ C[\vdash_e \cdot]$	$\texttt{let}\ f\ \texttt{be}\ \langle x, y\rangle\ \texttt{in}\ C[e]$
$\texttt{let}\ C[\vdash_e \cdot]\ \texttt{be}\ \langle x, y\rangle\ \texttt{in}\ f$	$\texttt{let}\ C[e]\ \texttt{be}\ \langle x, y\rangle\ \texttt{in}\ f$
$\langle V[\vdash_e \cdot], u\rangle$	$\langle V[e], u\rangle$
$\langle u, V[\vdash_e \cdot]\rangle$	$\langle u, V[e]\rangle$

Figure 2.5: Filling a linear context with a term.

$\Gamma \vdash_e C[\Delta \vdash_e A] : B$, which can be read informally as saying that whenever the term e is such that $\Delta \vdash e : A$, it holds that $\Gamma \vdash C[e] : B$.

For subsequent uses, let here give the symbol $\mathtt{CTX}_B\,(\Gamma \vdash A)$ which defines the class of all (not necessarily ground) linear contexts such that $\emptyset \vdash C[\Gamma \vdash A] : B$.

Type judgement context rule	Name
$\dfrac{}{\Gamma \vdash_v [\Gamma \vdash_v A] : A}$	$(tjc - ax_v)$
$\dfrac{}{\Gamma \vdash_e [\Gamma \vdash_e A] : A}$	$(tjc - ax_t)$
$\dfrac{\Gamma \vdash_v [\Gamma \vdash_v A] : A}{\Gamma \vdash_e [\Gamma \vdash_e A] : A}$	$(tjc - vt)$
$\dfrac{\Gamma, x : B \vdash_e C[\Theta \vdash_e E] : A}{\Gamma \vdash_v \lambda x.C[\Theta \vdash_e E] : B \multimap A}$	$(tjc - abs)$
$\dfrac{\Gamma \vdash_e C[\Theta \vdash_e E] : B \multimap A \qquad \Delta \vdash f : B}{\Gamma, \Delta \vdash_e C[\Theta \vdash_e E]f : A}$	$(tjc - app_L)$
$\dfrac{\Gamma \vdash f : B \multimap A \qquad \Delta \vdash_e C[\Theta \vdash_e E] : B}{\Gamma, \Delta \vdash_e fC[\Theta \vdash_e E] : A}$	$(tjc - app_R)$
$\dfrac{\Gamma \vdash_e C[\Theta \vdash_e E] : \mathtt{bool} \qquad \Delta \vdash f : A \qquad \Delta \vdash g : A}{\Gamma, \Delta \vdash_e \mathtt{if}\ C[\Theta \vdash_e E]\ \mathtt{then}\ f\ \mathtt{else}\ g : A}$	$(tjc - if_L)$
$\dfrac{\Gamma \vdash f : \mathtt{bool} \qquad \Delta \vdash_e C[\Theta \vdash_e E] : A \qquad \Delta \vdash_e D[\Theta \vdash_e E] : A}{\Gamma, \Delta \vdash_e \mathtt{if}\ f\ \mathtt{then}\ C[\Theta \vdash_e E]\ \mathtt{else}\ D[\Theta \vdash_e E] : A}$	$(tjc - if_R)$
$\dfrac{\Gamma \vdash_e C[\Theta \vdash_e E] : B \otimes F \qquad \Delta, x : B, y : F \vdash f : A}{\Gamma, \Delta \vdash_e \mathtt{let}\ C[\Theta \vdash_e E]\ \mathtt{be}\ \langle x, y\rangle\ \mathtt{in}\ f : A}$	$(tjc - let_L)$
$\dfrac{\Gamma \vdash f : B \otimes F \qquad \Delta, x : B, y : F \vdash_e C[\Theta \vdash_e E] : A}{\Gamma, \Delta \vdash_e \mathtt{let}\ f\ \mathtt{be}\ \langle x, y\rangle\ \mathtt{in}\ C[\Theta \vdash_e E] : A}$	$(tjc - let_R)$
$\dfrac{\Gamma \vdash_v V[\Theta \vdash_e E] : A \qquad \Delta \vdash u : B}{\Gamma, \Delta \vdash_v \langle V[\Theta \vdash_e E], u\rangle : A \otimes B}$	$(tjc - pai_L)$
$\dfrac{\Gamma \vdash u : A \qquad \Delta \vdash_v V[\Theta \vdash_e E] : B}{\Gamma, \Delta \vdash_v \langle u, V[\Theta \vdash_e E]\rangle : A \otimes B}$	$(tjc - pai_R)$

Figure 2.6: Context typing rules for contexts in a typed language: in a *linear* frame, typing context Γ, Δ have disjoint domains.

Lemma 2.6 (On the filled contexts)**.** *Given a context* $C[\cdot] \in \mathtt{CTX}_A(\Delta \vdash B)$ *and a term* $\Delta \vdash e : B$ *then*

$$\Gamma, \Delta \vdash C[e] : A \tag{2.18}$$

is a correct type judgement.

Proof. By induction on the structure of operational contexts, $\emptyset \vdash C[\vdash_B] : A$.

If $C[\vdash_B] = [\vdash_B]$ and $\Delta \vdash e : B$ then necessarily $A \equiv B$ and the typing judgement (2.18) holds.

If $C[\vdash_B] = \lambda x.D[\vdash_B]$ the last rule in the typing judgement tree must be $(tjc - abs)$ whence we get $\Gamma, x : F \vdash D[\vdash_B] : E$, and hence $A = F \multimap E$. From the induction hypothesis one gets $\Gamma, x : F, \Delta \vdash D[e] : E$ we immediately deduce the thesis from the rule $(tjc - abs)$, namely $\Gamma, \Delta \vdash \lambda x.D[e] : F \multimap E$.

If $C[\vdash_B] = fD[\vdash_B]$ let us start from the hypothesis $\Gamma_1, \Gamma_2 \vdash C[\vdash_B] : A$ which must have been derived by $(tjc - app_R)$ to get the type judgements $\Gamma_1 \vdash f : E \multimap A$ and $\Gamma_2 \vdash D[\vdash_B] : E$. By induction hypothesis on the premises it follows $\Gamma_2, \Delta \vdash D[e] : E$ and therefore, by application of $(tjc - app_R)$ we get $\Gamma_1, \Gamma_2, \Delta \vdash fD[e] : A$

If $C[\vdash_B] = D[\vdash_B]f$ then the typing assertion $\Gamma_1, \Gamma_2 \vdash D[\vdash_B]f : A$ must come from the application of $(tjc - app_L)$ and then we deduce $\Gamma_1 \vdash D[\vdash_B] : E \multimap A$ and $\Gamma_2 \vdash f : E$. By induction hypothesis it can be derived the typing judgement $\Gamma_1, \Delta \vdash D[e] : E \multimap A$, then from $(tjc - app_L)$ with this new premise, we get the thesis $\Gamma_1, \Gamma_2, \Delta \vdash D[e]f : A$.

If $C[\vdash_B] = (\mathtt{if}\ D[\vdash_B]\ \mathtt{then}\ f\ \mathtt{else}\ g)$, the type assertion must come from the application of $(tjc - if_L)$. Therefore it must hold that $\Gamma_1 \vdash D[\vdash_B] : \mathtt{bool}$ and $\Gamma_2 \vdash f : A$, $\Gamma_2 \vdash g : A$. By induction hypothesis we get the validity of the statement $\Gamma_1, \Delta \vdash D[e] : \mathtt{bool}$, whence immediately it follows the thesis $\Gamma_1, \Gamma_2, \Delta \vdash (\mathtt{if}\ D[e]\ \mathtt{then}\ f\ \mathtt{else}\ g) : A$ by application of $(tjc - if_L)$.

If $C[\vdash_B] = \big(\mathtt{if}\ f\ \mathtt{then}\ D[\vdash_B]\ \mathtt{else}\ G[\vdash_B]\big)$ then going back of one step in the type derivation tree through $(tjc - if_R)$ rule we get $\Gamma_1 \vdash f : \mathtt{bool}$ and $\Gamma_2 \vdash D[\vdash_B] : A$, $\Gamma_2 \vdash G[\vdash_B] : A$. Here we can rely on the double induction hypothesis $\Gamma_2 \vdash D[e] : A$

and $\Gamma_2 \vdash G[e] : A$. Therefore whichever is the value the guard evaluates to, by application of $(tjc - if_R)$ we must deduce $\Gamma_1, \Gamma_2, \Delta \vdash (\texttt{if}\ f\ \texttt{then}\ D[e]\ \texttt{else}\ G[e]) : A$

If $C[\vdash_B] = (\texttt{let}\ D[\vdash_B]\ \texttt{be}\ \langle x, y\rangle\ \texttt{in}\ f)$ the typing assertion comes necessarily from $(tjc - let_L)$ and we get $\Gamma_1 \vdash D[\vdash_B] : E \otimes F$ and $\Gamma_2, x : E, y : F \vdash f : A$ as valid statements. Moreover the induction hypothesis ensures that $\Gamma_1, \Delta \vdash D[e] : E \otimes F$ holds, and therefore applying (let_L) the thesis $\Gamma_1, \Gamma_2, \Delta \vdash (\texttt{let}\ D[e]\ \texttt{be}\ \langle x, y\rangle\ \texttt{in}\ f) : A$ follows.

If $C[\vdash_B] = (\texttt{let}\ f\ \texttt{be}\ \langle x, y\rangle\ \texttt{in}\ D[\vdash_B])$ going back through the rule $(tjc - let_R)$ we get $\Gamma_1 \vdash f : E \otimes F$ and $\Gamma_2, x : E, y : F \vdash D[\vdash_B] : A$ and exploiting inductive hypothesis we find $\Gamma_2, x : E, y : F, \Delta \vdash D[e] : A$. With this premise, by applying $(tjc - let_R)$ it follows immediately the thesis $\Gamma_1, \Gamma_2, \Delta \vdash (\texttt{let}\ f\ \texttt{be}\ \langle x, y\rangle\ \texttt{in}\ D[e]) : A$

If $C[\vdash_B] = \langle V[\vdash_B], u\rangle$ (and also for the symmetric configuration) one must use $(tjc - pai)$ as last derivation rule thus obtaining $\Gamma_1 \vdash V[\vdash_B] : E$ and $\Gamma_2 \vdash u : F$.

Therefore $A = E \otimes F$ and using induction we find $\Gamma_1, \Delta \vdash V[e] : E$, then by $(tjc - pai)$ with this premise one gets $\Gamma_1, \Gamma_2, \Delta \vdash \langle V[e], u\rangle : A$, which is what it had to be proved. This concludes the proof. □

We are now in a position to define the context preorder: given two terms e and f such that $\Gamma \vdash e, f : A$, we write $e \leq_{\Gamma,A} f$ iff for every context $C[\cdot] \in \mathtt{CTX}_B(\Gamma \vdash A)$, $C[e] \Downarrow v \Rightarrow C[f] \Downarrow u$, where $v, u \in \mathcal{V}_B^{\ell ST_\lambda}$. If $e \leq_{\Gamma,A} f$ and $f \leq_{\Gamma,A} e$, then e and f are said to be *context equivalent*, and we write $e \equiv_{\Gamma,A} f$. For our future purposes it will be found useful to define a function $\mathbf{Obs} : \mathcal{T}_{\ell ST_\lambda}^{\Gamma,A} \to \mathbb{R}$ on the terms set. In a deterministic environment simply choose $\mathbf{Obs}(e) = 1$ if $e \Downarrow v$, $\mathbf{Obs}(e) = 0$ in case of divergent terms. Therefore, the previous relations may be restated as follows:

$$\Gamma \vdash e \leq_{\Gamma,A} f : A \text{ iff } \forall C[\cdot] \in \mathtt{CTX}_B(\Gamma \vdash A), \mathbf{Obs}(C[e]) \leq \mathbf{Obs}(C[f]) \tag{2.19}$$

$$\Gamma \vdash e \equiv_{\Gamma,A} f : A \text{ iff } \forall C[\cdot] \in \mathtt{CTX}_B(\Gamma \vdash A), \mathbf{Obs}(C[e]) = \mathbf{Obs}(C[f]). \tag{2.20}$$

What we have just defined, indeed, are two *typed relations* $\leq$ and $\equiv$, that is to say two families of relations indexed by contexts and types, i.e. $\leq$ is the family $\{\leq_{\Gamma,A}\}_{\Gamma,A}$, while $\equiv$ is $\{\equiv_{\Gamma,A}\}_{\Gamma,A}$. If in the scheme above the type B is restricted

so as to be `bool`, then the obtained relations are the *ground* context preorder and *ground* context equivalence, respectively.

It can be easily proved that $\leq_{\Gamma,A}$ is a preorder – namely a reflexive and transitive relation – on $\mathcal{T}^{\Gamma,A}_{\ell ST_\lambda}$, and $\equiv_{\Gamma,A}$ an equivalence relation likewise. Among the preorders which can be defined over the terms of an higher order language, a particular interest is given to those relations which are *compatible* with the constructors of the language. A relation R in ℓST_λ is called compatible if it respects the following constraints

$$(\mathbf{c-1}) \qquad \forall x,\, x : A \vdash x \,R\, x : A \tag{2.21a}$$

$$(\mathbf{c-2}) \qquad \Gamma, x : B \vdash e \,R\, h : A \;\Rightarrow\; \Gamma \vdash \lambda x.e \,R\, \lambda x.h : B \multimap A \tag{2.21b}$$

$$(\mathbf{c-3}) \qquad \Gamma \vdash e \,R\, h : B \multimap A,\, \Delta \vdash f \,R\, \ell : B \;\Rightarrow\; \Gamma, \Delta \vdash e f \,R\, h \ell : A \tag{2.21c}$$

$$(\mathbf{c-4}) \qquad \Gamma \vdash e \,R\, h : \mathtt{bool},\, \Delta \vdash f \,R\, \ell : A,\, \Delta \vdash g \,R\, a : A \;\Rightarrow$$
$$\Gamma, \Delta \vdash (\mathtt{if}\ e\ \mathtt{then}\ f\ \mathtt{else}\ g) \,R\, (\mathtt{if}\ h\ \mathtt{then}\ \ell\ \mathtt{else}\ a) : A \tag{2.21d}$$

$$(\mathbf{c-5}) \qquad \Gamma \vdash e \,R\, h : A \otimes B,\, \Delta, x : A, y : B \vdash f \,R\, \ell : E \;\Rightarrow$$
$$\Gamma, \Delta \vdash (\mathtt{let}\ e\ \mathtt{be}\ \langle x, y\rangle\ \mathtt{in}\ f) \,R\, (\mathtt{let}\ h\ \mathtt{be}\ \langle x, y\rangle\ \mathtt{in}\ \ell) : E \tag{2.21e}$$

$$(\mathbf{c-6}) \qquad \Gamma \vdash v \,R\, w : A,\, \Delta \vdash u \,R\, v : B \;\Rightarrow\; \Gamma, \Delta \vdash \langle v, u\rangle \,R\, \langle w, v\rangle : A \otimes B. \tag{2.21f}$$

A compatible preorder is said to be a *precongruence*, likewise a compatible equivalence relation is called *congruence*. Thus a congruence is a reflexive, symmetric, transitive and compatible relation. We are going to show that $\leq_{\Gamma,A}$ and $\equiv_{\Gamma,A}$ are a precongruence and a congruence respectively over the set $\mathcal{T}^{\Gamma,A}_{\ell ST_\lambda}$, through the following two lemmas.

Lemma 2.7 (Compatibility of context preorder)**.** *$\leq_{\Gamma,A}$ is a precongruence on $\mathcal{T}^{\Gamma,A}_{\ell ST_\lambda}$ (and $\equiv_{\Gamma,A}$ a congruence likewise.)*

Proof. By examination of every constructor of ℓST_λ, relying on the definition of context preorder (2.19).

$(\mathbf{c-1})$ $\forall x,\, x \leq_{x:A,A} x$ this is obviously true as a special case of reflexivity of $\leq_{\Gamma,A}$.

$(\mathbf{c-2})$ $e \leq_{\Gamma,x:E,A} h \;\Rightarrow\; \lambda x.e \leq_{\Gamma,E\multimap A} \lambda x.h$. From hypothesis we evince $\forall C[\cdot] \in \mathtt{CTX}_B(\Gamma, x : E \vdash A),\ C[e] \Downarrow v \Rightarrow C[h] \Downarrow w$. Thus denoting by $\{x_i\}_{i\in\mathcal{I}}$ the set of variables such that $\{x_i\}_{i\in\mathcal{I}} = dom(\Gamma)$, for every generic context $C'[\cdot] \in \mathtt{CTX}_B(\Gamma \vdash E \multimap A)$ we set $C[\cdot] = C'[\lambda\{x_i\}_{i\in\mathcal{I}}.\lambda x.[\cdot]]$. In fact, the hypothesis $e \leq_{\Gamma,x:E,A} h$ entails $C[e] \Downarrow v \Rightarrow C[h] \Downarrow u$ whence thesis $\lambda x.e \leq_{\Gamma,E\multimap A} \lambda x.h$.

$(\mathbf{c-3})$ $(e_1 \leq_{\Gamma_1,E\multimap A} h_1 \;\wedge\; e_2 \leq_{\Gamma_2,E} h_2) \Rightarrow\; e_1e_2 \leq_{\Gamma_1\Gamma_2,A} h_1h_2$.

The statement for these terms can be written as follows:

$$\begin{aligned}(\forall C[\cdot] \in \mathtt{CTX}_B(\Gamma_1 \vdash E \multimap A),\ \forall D[\cdot] \in \mathtt{CTX}_B(\Gamma_2 \vdash E)\\ \mathbf{Obs}(C[e_1]) \leq \mathbf{Obs}(C[h_1]) \wedge \mathbf{Obs}(D[e_2]) \leq \mathbf{Obs}(D[h_2])) \Rightarrow\\ (\forall C'[\cdot] \in \mathtt{CTX}_B(\Gamma \vdash A)\, \mathbf{Obs}(C'[e_1e_2]) \leq \mathbf{Obs}(C'[h_1h_2])).\end{aligned} \tag{2.22}$$

The hypothesis of contextual preorder for the subterms of e and f, can be written as $\forall C[\cdot] \in \mathtt{CTX}_{B_1}(\Gamma_1 \vdash E \multimap A),\ C[e_1] \Downarrow v \Rightarrow C[h_1] \Downarrow u$ and $\forall D[\cdot] \in \mathtt{CTX}_{B_2}(\Gamma_2 \vdash E),\ D[e_2] \Downarrow \nu \Rightarrow D[h_2] \Downarrow w$. Thus for each generic context $C'[\cdot] \in \mathtt{CTX}_B(\Gamma \vdash A)$, let us denote by $\{x_i\}_{i\in\mathcal{I}}$ the set $\{x_i\}_{i\in\mathcal{I}} = dom(\Gamma_1) \cup dom(\Gamma_2)$, choosing thereby $C[\cdot] = C'[\lambda\{x_i\}_{i\in\mathcal{I}}.[\vdash_{E\multimap A}]e_2] \in \mathtt{CTX}_A(\Gamma_1 \vdash E \multimap A)$ and $D[\cdot] = C'[\lambda\{x_i\}_{i\in\mathcal{I}}.h_1[\vdash_E]] \in \mathtt{CTX}_A(\Gamma_2 \vdash E)$.

Necessarily $B_1 = B_2 = A$, and since $C[h_1] = C'[\lambda\{x_i\}_{i\in\mathcal{I}}.h_1e_2] = D[e_2]$, one gets the chain $C[e_1] \Downarrow v \Rightarrow C[h_1] = D[e_2] \Downarrow u \Rightarrow D[h_2] \Downarrow w$, whence the thesis, being $C[e_1] = C'[\lambda\{x_i\}_{i\in\mathcal{I}}.e_1e_2]$ and $D[h_2] = C'[\lambda\{x_i\}_{i\in\mathcal{I}}.h_1h_2]$.

$(\mathbf{c-4})$ $(e_1 \leq_{\Gamma_1,\mathtt{bool}} h_1 \;\wedge\; e_2 \leq_{\Gamma_2,A} h_2 \;\wedge e_3 \leq_{\Gamma_2,A} h_3) \Rightarrow (\mathtt{if}\ e_1\ \mathtt{then}\ e_2\ \mathtt{else}\ e_3) \leq_{\Gamma_1,\Gamma_2,A} (\mathtt{if}\ h_1\ \mathtt{then}\ h_2\ \mathtt{else}\ e_3)$.

Now by the hypotheses of context preorder between the subterms we know that $\forall C[\cdot] \in \mathtt{CTX}_{B_1}(\Gamma_1 \vdash \mathtt{bool}),\ C[e_1] \Downarrow v_1 \Rightarrow C[h_1] \Downarrow w_1$, and similarly $\forall D \in \mathtt{CTX}_{B_2}(\Gamma_2 \vdash A),\ D[e_2] \Downarrow v_2 \Rightarrow D[h_2] \Downarrow w_2$ and $D[e_3] \Downarrow v_3 \Rightarrow D[h_3] \Downarrow w_3$. Then, for each $C'[\cdot] \in \mathtt{CTX}_B(\Gamma \vdash A)$, considering the contexts

$$C[\vdash_{\mathtt{bool}}] = C'[\lambda\{x_i\}_{i\in\mathcal{I}}.(\mathtt{if}\ [\vdash_{\mathtt{bool}}]\ \mathtt{then}\ e_2\ \mathtt{else}\ e_3)]$$
$$D_1[\vdash_B] = C'[\lambda\{x_i\}_{i\in\mathcal{I}}.(\mathtt{if}\ h_1\ \mathtt{then}\ [\vdash_B]\ \mathtt{else}\ e_3)]$$

$$D_2[\vdash_B] = C'[\lambda\,\{x_i\}_{i\in\mathcal{I}}.(\texttt{if } h_1 \texttt{ then } h_2 \texttt{ else } [\vdash_B])]$$

and $B_1 = B_2 = B_3 = A$, where again we set $\{x_i\}_{i\in\mathcal{I}} = dom(\Gamma_1) \cup dom(\Gamma_2)$, we meet the conditions $C[h_1] = D_1[e_2]$ and $D_1[h_2] = D_2[e_3]$, thus we get the chain $C[e_1] \Downarrow v \Rightarrow C[h_1] = D_1[e_2] \Downarrow u \Rightarrow D_1[h_2] = D_2[e_3] \Downarrow \nu \Rightarrow D_2[h_3] \Downarrow w$. Therefore we get the thesis from the first and last term of the chain.

$(\mathbf{c-5})$ $(e_1 \leq_{\Gamma_1, E\otimes E'} h_1 \wedge\ e_2 \leq_{\Gamma_2, x:E, y:E', A} h_2) \Rightarrow (\texttt{let } e_1 \texttt{ be } \langle x, y\rangle \texttt{ in } e_2) \leq_{\Gamma_1,\Gamma_2,A} (\texttt{let } h_1 \texttt{ be } \langle x, y\rangle \texttt{ in } h_2)$. Again using the hypothesis of context preorder for subterms one gets $\forall C[\cdot] \in \texttt{CTX}_{B_1}(\Gamma_1 \vdash E \otimes E')$, $C[e_1] \Downarrow v \Rightarrow C[h_1] \Downarrow u$ and $\forall D[\cdot] \in \texttt{CTX}_{B_2}(\Gamma_2 \vdash A)$, $D[e_2] \Downarrow \nu \Rightarrow D[h_2] \Downarrow w$.

Therefore let us take $C[\cdot] = C'[\lambda\,\{x_i\}_{i\in\mathcal{I}}.(\texttt{let } [\vdash_{E\otimes E'}] \texttt{ be } \langle x, y\rangle \texttt{ in } e_2)]$ and $D[\cdot] = C'[\lambda\,\{x_i\}_{i\in\mathcal{I}}.(\texttt{let } h_1 \texttt{ be } \langle x, y\rangle \texttt{ in } [\vdash_A])]$ which fulfill the requirement $C[h_1] = D[e_2]$. Joining the premises together we get the chain $C[e_1] \Downarrow v \Rightarrow C[h_1] = D[e_2] \Downarrow u \Rightarrow D[h_2] \Downarrow \nu$, which is the thesis.

$(\mathbf{c-6})$ $(v_1 \leq_{\Gamma_1, A} w_1 \wedge\ v_2 \leq_{\Gamma_2, E} w_2) \Rightarrow \langle v_1, v_2\rangle \leq_{\Gamma_1\Gamma_2, A\otimes E} \langle w_1, w_2\rangle$. Here, for any $C'[\cdot] \in \texttt{CTX}_B(\Gamma \vdash A \otimes E)$, we set

$$C[\cdot] = C'[\lambda\,\{x_i\}_{i\in\mathcal{I}}.\langle [\vdash_A], v_2\rangle] \in \texttt{CTX}_{A\otimes E}(\Gamma_1 \vdash A)$$
$$D[\cdot] = C'[\lambda\,\{x_i\}_{i\in\mathcal{I}}.\langle w_1, [\vdash_E]\rangle] \in \texttt{CTX}_{A\otimes E}(\Gamma_2 \vdash E)$$

such that $C[w_1] = D[v_2]$. Thus using the hypothesis of preconguence for subterms we have $C'[\lambda\,\{x_i\}_{i\in\mathcal{I}}.\langle v_1, v_2\rangle] = C[v_1] \Downarrow v \Rightarrow C[w_1] = D[v_2] \Downarrow u \Rightarrow D[w_2] = C'[\lambda\,\{x_i\}_{i\in\mathcal{I}}.\langle w_1, w_2\rangle] \Downarrow \nu$, namely the thesis. □

Lemma 2.8 (Context preorder and context equivalence behaviour with respect to contexts)**.** *Context preorder and context equivalence likewise are compatible with respect to a whatever context application to terms, namely they enjoy the properties* $\Gamma \vdash e \leq_{\Gamma,A} h : A \Rightarrow \forall C[\vdash_A] \in \texttt{CTX}_B(\Gamma \vdash A)$, $\emptyset \vdash C[e] \leq_{\emptyset,B} C[h] : B$ *and* $\Gamma \vdash e \equiv_{\Gamma,A} h : A \Rightarrow \forall C[\vdash_A] \in \texttt{CTX}_B(\Gamma \vdash A)$, $\emptyset \vdash C[e] \equiv_{\emptyset,B} C[h] : B$.

Proof. We treat the context preorder, the proof for context equivalence being analogous.

The hypothesis $\Gamma \vdash e \leq_{\Gamma,A} h : A$ entails that $\forall G[\vdash_A] \in \mathtt{CTX}_B(\Gamma \vdash A)$, $\mathbf{Obs}(G[e]) \leq \mathbf{Obs}(G[h])$, and following the definition the statement to be proved $\emptyset \vdash C[e] \leq_{\emptyset,B} C[h] : B$ is equivalent to

$$\forall D[\vdash_B] \in \mathtt{CTX}_E(\emptyset \vdash B), \mathbf{Obs}(D[C[e]]) \leq \mathbf{Obs}(D[C[h]]). \tag{2.23}$$

Nevertheless for any $D[\vdash_B]$ and $C[\vdash_A]$, the new context $G_{C,D}[\vdash_A]$ may be chosen, defined as $G_{C,D}[\vdash_A] \stackrel{\text{def}}{=} D[C[\vdash_A]]$ which, since it belongs to $\mathtt{CTX}_E(\Gamma \vdash A)$, complies with hypothesis □

2.4 Applicative Bisimilarity: Definition and Properties

Context equivalence is universally accepted as the canonical notion of equivalence of higher-order programs, being robust, and only relying on the underlying operational semantics. Proving terms *not* context equivalent is relatively easy: ending up with a single context separating the two terms suffices. On the other hand, the universal quantification over all contexts makes proofs of equivalence hard.

A variety of techniques have been proposed to overcome this problem, among them logical relations, adequate denotational models and context lemmas. As first proposed by Abramsky [1], coinductive methodologies (and the bisimulation proof method in particular) can be fruitfully employed. Abramsky's *applicative* bisimulation is based on taking argument passing as the basic interaction mechanism: what the environment can do with a λ-term is either evaluating it or passing it an argument.

Among the various approaches which can be followed to delineate the concept of (bi)simulation in the framework of a linear λ-calculus, here it has been decided to present it on the top of a labelled transition system, with the purpose to make easier to give its extension to probabilistic and quantum systems.

A *labelled transition system* (LTS in the following) is a triple $\mathscr{L} = (\mathscr{S}, \mathcal{L}, \mathcal{N})$, where $\mathscr{S}$ is a set of *states*, $\mathcal{L}$ is a set of *labels*, and $\mathcal{N}$ is a subset of $\mathscr{S} \times \mathcal{L} \times \mathscr{S}$.

If for every $s \in \mathscr{S}$ and for every $\ell \in \mathcal{L}$ there is *at most* one state $t \in \mathscr{S}$ with $(s, \ell, t) \in \mathcal{N}$, then $\mathscr{L}$ is said to be *deterministic* as it is indeed the case for ℓST_λ. The theory of bisimulation for LTSs is very well-studied [48] and forms one of the cornerstones of concurrency theory.

An applicative bisimulation relation can be thought as a bisimulation played on an LTS defined *on top of* the λ-calculus ℓST_λ, where the $\mathscr{S}$ elements are terms of ℓST_λ and the actions which label the transitions among states match the ways which the environment may operate on them. Therefore (s, ℓ, t) is permitted as an element of $\mathcal{N}$ if a suitable action $\ell \in \mathcal{L}$ exists, such that the external environment fosters the transition from s to t. The set of possible actions which the environment may accomplish on a state of ℓST_λ is shown in Figure 2.7, where the labels for every action are listed with their meaning. Within the LTS, we distinguish between *external* actions, namely those which entail an interaction of the system with the environment, and the unique *internal* action labelled by *eval*, which is the evaluation process of a program. More specifically, the LTS $\mathscr{L}_{\ell ST_\lambda}$ is defined as the triple

$$\left(\underbrace{\overline{\mathcal{T}^{\ell ST_\lambda}} \uplus \overline{\mathcal{V}^{\ell ST_\lambda}}}_{\mathscr{S}}, \underbrace{\{a_{eval}, a_{\mathtt{tt}}, a_{\mathtt{ff}}, a_{@v}, a_{\otimes g}, a_{\mathcal{Y}_A}, a_{\widehat{\mathcal{Y}_A}}\}}_{\mathcal{L}}, \mathcal{N}_{\ell ST_\lambda} \right)$$

where:

- $\overline{\mathcal{T}^{\ell ST_\lambda}}$ is the set of pairs $\cup_{A \in \mathcal{Y}} (\mathcal{T}_A^{\ell ST_\lambda} \times \{A\})$, and similarly for $\overline{\mathcal{V}^{\ell ST_\lambda}}$. Observe how any pair (v, A) appears twice as a state, once as an element of $\overline{\mathcal{T}^{\ell ST_\lambda}}$ and another one as an element of $\overline{\mathcal{V}^{\ell ST_\lambda}}$. Whenever necessary to avoid ambiguity, the second instance will be denoted as $(\widehat{v}, A)$. Similarly for the two copies of any type A one finds as labels.

- The label a_{eval} models evaluation of terms, namely the only action internal to the system, which doesn't have any effect for an external observer. Besides it is the only action which the system can perform on a term, namely on an element of the set $\mathcal{T}_A^{\ell ST_\lambda} \setminus \mathcal{V}_A^{\ell ST_\lambda}$, unless ask its type ($a_{\mathcal{Y}_A}$). The couple $a_{\mathtt{tt}}$, $a_{\mathtt{ff}}$ represents the (only) way the system can interact with a boolean constant,

Action	Name	Tern in $\mathcal{N}$
Show the value of a boolean:	$a_{\mathtt{tt}}$	$\big((\widehat{\mathtt{tt}}, \mathtt{bool}), a_{\mathtt{tt}}, (\widehat{\mathtt{tt}}, \mathtt{bool})\big)$
	$a_{\mathtt{ff}}$	$\Big((\widehat{\mathtt{ff}}, \mathtt{bool}), a_{\mathtt{ff}}, (\widehat{\mathtt{ff}}, \mathtt{bool})\Big)$
Gives an argument to a function type:	$a_{@v}$	$\Big((\widehat{\lambda\, x.e}, B \multimap A), a_{@v}, (e\{v/x\}, A)\Big)$
Substitutes a pair into an open term:	$a_{\otimes g}$	$\Big((\widehat{\langle v, u\rangle}, A \otimes B), a_{\otimes g}, (g\{v/x, u/y\}, E)\Big)$
Exhibits the type of a term:	$a_{\mathcal{Y}_A}$	$((e, A), a_{\mathcal{Y}_A}, (e, A))$
Exhibits the type of a value:	$a_{\widehat{\mathcal{Y}_A}}$	$\Big((\widehat{v}, A), a_{\widehat{\mathcal{Y}_A}}, (\widehat{v}, A)\Big)$
Evaluates the term:	a_{eval}	$\mathcal{P}_{\ell PST_\lambda}((e, A), eval, (\widehat{v}, A))$.

Figure 2.7: Possible labelled actions of the LTS in ℓPST_λ.

unless requesting for its type, which is asking to show its value. Finally the actions $a_{@v}$ and $a_{\otimes g}$ are the investigations that the system can accomplish on a function type value and a pair value respectively: their meaning is shown in Table 2.7.

- The relation $\mathcal{N}_{\ell ST_\lambda}$ contains all triples in the following forms:

$$\Big((\widehat{\lambda\, x.e}, A \multimap B), a_{@v}, (e\{v/x\}, B)\Big) \qquad \Big((\widehat{\mathtt{ff}}, \mathtt{bool}), a_{\mathtt{ff}}, (\widehat{\mathtt{ff}}, \mathtt{bool})\Big)$$

$$\Big((\widehat{\langle v, u\rangle}, A \otimes B), a_{\otimes g}, (g\{v/x, u/y\}, E)\Big) \qquad \big((\widehat{\mathtt{tt}}, \mathtt{bool}), a_{\mathtt{tt}}, (\widehat{\mathtt{tt}}, \mathtt{bool})\big)$$

$$((e,A),a_{\mathcal{Y}_A},(e,A)) \qquad \Big((\widehat{v},A),a_{\widehat{\mathcal{Y}_A}},(\widehat{v},A)\Big) \qquad ((e,A),a_{eval},(\widehat{v},A))$$

where, in the last item, we of course assume that $e \Downarrow v$.

As one can easily verify, the labelled transition system $\mathscr{L}_{\ell ST_\lambda}$ is deterministic. Besides notice that, however, both are binary relations on *states*, i.e., on elements of $\overline{\mathcal{T}^{\ell ST_\lambda}} \uplus \overline{\mathcal{V}^{\ell ST_\lambda}}$. Let us observe, however, that:

- Two pairs (e,A) and (f,B) can be put in relation only if $A = B$, because each state makes its type public through a label. For similar reasons, states in the form (v,A) and $(\widehat{u},B)$ cannot be in relation, not even if $A = B$.

- If (v,A) and (u,A) are in relation, then also $(\widehat{v},A)$ and $(\widehat{u},A)$ are in relation. Conversely, if $(\widehat{v},A)$ and $(\widehat{u},A)$ are in a (bi)simulation relation R, then $R \cup \{(v,A),(u,A)\}$ is itself a (bi)simulation.

The definition of (bi)simulation over the terms of ℓST_λ is given in the standard way, playing the (bi)simulation game on the top of the LTS; anyway we give it explicitly, for closed terms of ℓST_λ, as a family of relations indexed on the types, denoting the first one by $\mathcal{S}_A$ and the second by $\mathcal{B}_A$, where we meant that A is the type which the two terms that are on relation belong to.

It is given on the types and it distinguishes terms from values, starting from the transitions of the generic LTS and instantiating them on the labels and the states of our language $\mathcal{N}_{\ell ST_\lambda}$ which have been listed above.

- For boolean values the elements of $\mathcal{N}_{\ell ST_\lambda}$ containing the labels $a_{\mathtt{tt}}$ and $a_{\mathtt{ff}}$ are involved: therefore a relation $\mathcal{S}_{\mathtt{bool}}$ is a simulation over boolean values if $\forall v,\, w \in \mathcal{V}^{\ell ST_\lambda}_{\mathtt{bool}}$

$$\begin{aligned}&\emptyset \vdash v\,\mathcal{S}_{\mathtt{bool}}\,w : \mathtt{bool} \Rightarrow\\ &\forall \mathtt{b} \in \mathcal{V}^{\mathtt{bool}}_{\ell ST_\lambda}\ \big((\widehat{v},\mathtt{bool}),a_{\mathtt{b}},(\widehat{\mathtt{b}},\mathtt{bool})\big) \in \mathcal{N} \Rightarrow \big((\widehat{w},\mathtt{bool}),a_{\mathtt{b}},(\widehat{\mathtt{b}},\mathtt{bool})\big) \in \mathcal{N}\end{aligned} \tag{2.24}$$

- The relation of simulation between (closed) functional values is given in its applicative form, namely comparing the functions after they have been evaluated with the same value as argument. Thus a relation $\mathcal{S}_{B\multimap A}$ is a simulation on function values if $\forall\, \lambda\, x.e,\ \lambda\, x.h \in \mathcal{V}^{\ell ST_\lambda}_{B\multimap A}$,

$$\begin{aligned}\emptyset \vdash \lambda\, x.e\, \mathcal{S}_{B\multimap A}\, \lambda\, x.h : B \multimap A \Rightarrow \\ \forall v \in \mathcal{V}^{\ell ST_\lambda}_{B},\ \Big((\widehat{\lambda\, x.e}, B \multimap A), a_{@v}, (e\{v/x\}, A)\Big) \in \mathcal{N}\ \Rightarrow \\ \Big((\widehat{\lambda\, x.h}, B \multimap A), a_{@v}, (h\{v/x\}, A)\Big) \in \mathcal{N}\ \wedge\ e\{v/x\}\, \mathcal{S}_A\, h\{v/x\}. \quad (2.25)\end{aligned}$$

- For pair values the notion of simulation relies on the label $a_{\otimes g}$ of the LTS:

$$\begin{aligned}\emptyset \vdash \langle v_1, v_2\rangle\, \mathcal{S}_{A\otimes B}\, \langle u_1, u_2\rangle : A \otimes B \Rightarrow \forall g \in \mathcal{T}^{\ell ST_\lambda}_{x:A,y:B,E}, \\ \Big((\widehat{\langle v_1, v_2\rangle}, A \otimes B), a_{\otimes g}, g\{v_1/x, v_2/y\}\Big) \in \mathcal{N}\ \Rightarrow \\ \Big((\widehat{\langle u_1, u_2\rangle}, A \otimes B), a_{\otimes g}, g\{u_1/x, u_2/y\}\Big) \in \mathcal{N} \wedge\ g\{v_1/x, v_2/y\}\, \mathcal{S}_E\, g\{u_1/x, u_2/y\}. \quad (2.26)\end{aligned}$$

- With respect to terms, the label *eval* is used:

$$\begin{aligned}\emptyset \vdash e\, \mathcal{S}_A\, h : A \Rightarrow \\ \Big(((e, A), a_{eval}, (\widehat{v}, A)) \in \mathcal{N}\ \Rightarrow\ ((h, A), a_{eval}, (\widehat{w}, A)) \in \mathcal{N}\ \wedge\ \emptyset \vdash v\, \mathcal{S}_A\, w : A\Big) \quad (2.27)\end{aligned}$$

In the following some important property of (bi)simulations are shown: symbols `Sim` and `BiS` stand for the set of all simulations and bisimulation respectively, among terms which belong to $\mathcal{T}^{\ell ST_\lambda}_A$.

Lemma 2.9 (Identity relation is a bisimulation)**.** *The identity relation* $\mathbb{I}$ *is a simulation, therefore it is a bisimulation, being a symmetric relation* $\mathbb{I} \in$ `BiS`*.*

Proof. The proof follows straightforward by the definition of (bi)simulation as a family of relations indexed on types. The statement is proved in showing that $\forall e \in \mathcal{T}^{\ell ST_\lambda}_A$, $\Gamma \vdash e\, \mathcal{B}_A{}^{(i)} e : A$, for some $\mathcal{B}_A{}^{(i)} \in$ `BiS`.

$-e \in \mathcal{V}^{\ell ST_\lambda}_{\mathtt{bool}}-$ for boolean values, relying on the definition (2.9) we get the tautology

$$\forall \mathtt{b} \in \mathcal{V}^{\ell ST_\lambda}_{\mathtt{bool}} \left((\widehat{e}, \mathtt{bool}), a_{\mathtt{b}}, (\widehat{\mathtt{b}}, \mathtt{bool})\right) \in \mathcal{N} \Rightarrow \left((\widehat{e}, \mathtt{bool}), a_{\mathtt{b}}, (\widehat{\mathtt{b}}, \mathtt{bool})\right) \in \mathcal{N};$$

$-e \in \mathcal{V}^{\ell ST_\lambda}_{B \multimap A}-$ for λ–abstractions we resort (2.25) to obtain, $\forall e \in \mathcal{T}^{\ell ST_\lambda}_A$ and $\forall v \in \mathcal{V}^B$

$$\left((\widehat{\lambda\, x.e}, B \multimap A), a_{@v}, (e\{v/x\}, A)\right) \in \mathcal{N} \Rightarrow \left((\widehat{\lambda\, x.e}, B \multimap A), a_{@v}, (e\{v/x\}, A)\right) \in \mathcal{N} \ \wedge \ e\{v/x\}\, \mathbb{I}_A e\{v/x\}.$$

which is again a tautology;

$-e \in \mathcal{V}^{\ell ST_\lambda}_{A \otimes B}-$ for vector type values, we get again a tautology, being that $\langle v_1, v_2\rangle \ \mathbb{I}_{A\otimes B} \ \langle v_1, v_2\rangle$ is equivalent to $\forall g \in \mathcal{T}^{\ell ST_\lambda}_{x:A,y:B,E} \ g\{v_1/x, v_2/y\}\, \mathbb{I}_E \ g\{v_1/x, v_2/y\}$ which is trivially true;

$-e \in \mathcal{T}^{\ell ST_\lambda}_A \setminus \mathcal{V}^{\ell ST_\lambda}_A-$ in the end, for a term $e \in \mathcal{T}^{\ell ST_\lambda}_A$ we have again a tautology being $\forall v \in \mathcal{V}^A \ ((e, A), a_{eval}, (\widehat{v}, A)) \in \mathcal{N} \Rightarrow ((e, A), a_{eval}, (\widehat{v}, A)) \in \mathcal{N}$ where $v\, \mathbb{I}_A v$.

□

Lemma 2.10 (One–step reduction is a (bi)simulation)**.** *The one–step reduction relation (Figure 2.2) is a bisimulation (and, accordingly, the same holds for the evaluation relation).*

Proof. We enforce the induction hypothesis by defining a new relation on types $R_A =\!\!\rightarrow \cup\, \mathbb{I}_A$ such that $\rightarrow \subseteq R_A$ and by proving that since R_A is a bisimulation, also $\rightarrow$ enjoys the bisimulation property since it is included in R_A. By definition a pair (e, f), belongs to the relation if the condition below is respected

$$e,\, f \in \mathcal{T}^{\ell ST_\lambda}_A, \ \ (e, f) \in R_A \ \Leftrightarrow (e \rightarrow f \ \vee \ e\, \mathbb{I}_A f)\,. \tag{2.28}$$

$-e \in \mathcal{V}^{\ell ST_\lambda}_A-$ If $e = v$, then $f = e = v$ and $(e, f) \in R_A$, since $(e, f) \in \mathbb{I}_A$. Moreover the simulation property is satisfied, since

1. If $\emptyset \vdash e, f : \mathtt{bool}$ then both e and f are $\mathtt{tt}$ or $\mathtt{ff}$ and fulfill (2.24).

2. If $\emptyset \vdash e, f : B \multimap A$ then both e and f are the same λ–abstraction and (2.25) is satisfied.

3. If $\emptyset \vdash e, f : A \otimes B$ then both e and f are the same pair for previous Lemma 2.9 they are in a bisimulation relation.

$-e \in \mathcal{T}_A^{\ell ST_\lambda} \setminus \mathcal{V}_A^{\ell ST_\lambda}-$ If e is not a value, then $e \Downarrow v$ and by Lemma 2.4, there will be a sequence of one–step reduction such that $e \to f \to \ldots \to v$. Since the one step reduction is deterministic, then necessarily also $f \Downarrow v$: thus the relation

$$((e, A), a_{eval}, (\widehat{v}, A)) \in \mathcal{N} \;\Rightarrow\; ((f, A), a_{eval}, (\widehat{v}, A)) \in \mathcal{N}$$

holds, and since by definition (2.28) $v \, R_A v$, this entails that $e \, R_A f$, with $R_A \in \mathtt{Sim}$. □

Lemma 2.11 (On the composition of simulation – and bisimulation as well –)**.** *The composition of two (possibly) different simulations is a simulation itself. Namely if* $\mathcal{S}_A{}^{(1)}, \mathcal{S}_A{}^{(2)} \in \mathtt{Sim}$, $\forall e, f, g$ *such that* $e \, \mathcal{S}_A{}^{(1)} f$ *and* $f \, \mathcal{S}_A{}^{(2)} g$, *we have* $e \, \mathcal{S}_A{}^{(3)} g$, *with* $\mathcal{S}_A{}^{(3)} = \mathcal{S}_A{}^{(1)} \circ \mathcal{S}_A{}^{(2)}$ *element of* $\mathtt{Sim}$.

Proof. By inspection of the definition of simulation for different types.

$-e \in \mathcal{V}_{\ell ST_\lambda}^{\mathtt{bool}}-$ For boolean, writing the hypotheses of the double simulation relation according to the definition we get:

$$e \, \mathcal{S}_{\mathtt{bool}}{}^{(1)} f \Rightarrow \big((\widehat{e}, \mathtt{bool}), a_{\mathtt{b}}, (\widehat{\mathtt{b}}, \mathtt{bool})\big) \in \mathcal{N} \Rightarrow \Big((\widehat{f}, \mathtt{bool}), a_{\mathtt{b}}, (\widehat{\mathtt{b}}, \mathtt{bool})\Big) \in \mathcal{N} \quad \text{(hp1)}$$

$$f \, \mathcal{S}_{\mathtt{bool}}{}^{(2)} g \Rightarrow \Big((\widehat{f}, \mathtt{bool}), a_{\mathtt{b}}, (\widehat{\mathtt{b}}, \mathtt{bool})\Big) \in \mathcal{N} \Rightarrow \big((\widehat{g}, \mathtt{bool}), a_{\mathtt{b}}, (\widehat{\mathtt{b}}, \mathtt{bool})\big) \in \mathcal{N} \quad \text{(hp2)}$$

whence it easily derivable the relation

$$\big((\widehat{e}, \mathtt{bool}), a_{\mathtt{b}}, (\widehat{\mathtt{b}}, \mathtt{bool})\big) \in \mathcal{N} \Rightarrow \big((\widehat{g}, \mathtt{bool}), a_{\mathtt{b}}, (\widehat{\mathtt{b}}, \mathtt{bool})\big) \in \mathcal{N}$$

which ensures that $e(\mathcal{S}_{\mathtt{bool}}{}^{(1)} \circ \mathcal{S}_{\mathtt{bool}}{}^{(2)})g$.

$-e \in \mathcal{V}^{\ell ST_\lambda}_{B\multimap A}-$ For function values, for every tern belonging to the set $\mathcal{V}^{B\multimap A}$ such that $\lambda x.f\ \mathcal{S}_{B\multimap A}{}^{(1)}\ \lambda x.g$ and $\lambda x.g\ \mathcal{S}_{B\multimap A}{}^{(2)}\ \lambda x.h$ we have, by the definition(2.25), that $\forall v \in \mathcal{V}^A$ both the conditions

$$\begin{aligned}\Big((\widehat{\lambda x.f}, B \multimap A), a_{@_v}, (f\{v/x\}, A)\Big) \in \mathcal{N} \Rightarrow \\ \Big((\widehat{\lambda x.g}, B \multimap A), a_{@_v}, (g\{v/x\}, A)\Big) \in \mathcal{N}\ \wedge f\{v/x\}\,\mathcal{S}_A{}^{(1)} g\{v/x\} \\ \Big((\widehat{\lambda x.g}, B \multimap A), a_{@_v}, (g\{v/x\}, A)\Big) \in \mathcal{N} \Rightarrow \\ \Big((\widehat{\lambda x.h}, B \multimap A), a_{@_v}, (h\{v/x\}, A)\Big) \in \mathcal{N}\ \wedge g\{v/x\}\,\mathcal{S}_A{}^{(2)} h\{v/x\}\end{aligned}$$

hold. Therefore it follows straightly the relation

$$\begin{aligned}\Big((\widehat{\lambda x.f}, B \multimap A), a_{@_v}, (f\{v/x\}, A)\Big) \Rightarrow \Big((\widehat{\lambda x.h}, B \multimap A), a_{@_v}, (h\{v/x\}, A)\Big) \\ \wedge f\{v/x\}(\mathcal{S}_A{}^{(1)} \circ \mathcal{S}_A{}^{(2)}) h\{v/x\} \quad (2.29)\end{aligned}$$

which entails $\lambda x.f(\mathcal{S}_{B\multimap A}{}^{(1)} \circ \mathcal{S}_{B\multimap A}{}^{(2)})\lambda x.h$ with $\mathcal{S}_{B\multimap A}{}^{(1)} \circ \mathcal{S}_{B\multimap A}{}^{(2)} \in \mathtt{Sim}$.

$-e \in \mathcal{V}^{\ell ST_\lambda}_{A\otimes B}-$ For each tern of values of vector type such that $\langle v_1, v_2\rangle\,\mathcal{S}_{A\otimes B}{}^{(1)}\,\langle u_1, u_2\rangle$, and $\langle u_1, u_2\rangle\ \mathcal{S}_{A\otimes B}{}^{(2)}\ \langle \nu_1, \nu_2\rangle$ starting from the definition (2.26) one obtains the relations $\forall g \in \mathcal{T}^{\ell ST_\lambda}_{x:A,y:B,E}$, $g\{v_1/x, v_2/y\}\ \mathcal{S}_E{}^{(1)}\ g\{u_1/x, u_2/y\} \wedge g\{u_1/x, u_2/y\}\ \mathcal{S}_E{}^{(2)}$ $g\{\nu_1/x, \nu_2/y\}$, which lead to the conclusion

$$g\{v_1/x, v_2/y\}(\mathcal{S}_A{}^{(1)} \circ \mathcal{S}_A{}^{(2)}) g\{\nu_1/x, \nu_2/y\}.$$

This last relation entails that $\mathcal{S}_{A\otimes B}{}^{(3)} \in \mathtt{Sim}$.

$-e \in \mathcal{T}^{\ell ST_\lambda}_A \setminus \mathcal{V}^{\ell ST_\lambda}_A-$ Here, given three terms such that $e\,\mathcal{S}_A{}^{(1)} f$ and $f\,\mathcal{S}_A{}^{(2)} g$, recovering the definition (2.27), we get,

$$\begin{aligned}((e, A), a_{eval}, (\widehat{v}, A)) \in \mathcal{N}\ \Rightarrow\ ((f, A), a_{eval}, (\widehat{u}, A)) \in \mathcal{N}\ \wedge\ v\,\mathcal{S}_A{}^{(1)} u \\ ((f, A), a_{eval}, (\widehat{u}, A)) \in \mathcal{N}\ \Rightarrow\ ((g, A), a_{eval}, (\widehat{\nu}, A)) \in \mathcal{N}\ \wedge\ u\,\mathcal{S}_A{}^{(2)} \nu.\end{aligned}$$

From previous relation it follows

$$((e, A), a_{eval}, (\widehat{v}, A)) \in \mathcal{N}\ \Rightarrow\ ((g, A), a_{eval}, (\widehat{\nu}, A)) \in \mathcal{N}\ \wedge\ v\,\mathcal{S}_A{}^{(1)} \circ \mathcal{S}_A{}^{(2)} \nu,$$

whence $e(\mathcal{S}_A{}^{(1)} \circ \mathcal{S}_A{}^{(2)})g$, where $(\mathcal{S}_A{}^{(1)} \circ \mathcal{S}_A{}^{(2)}) = \mathcal{S}_A{}^{(3)} \in \mathtt{Sim}$ □

It should be evident at this point of the discussion, that starting by definitions (2.24)–(2.27) many relations between terms of ℓST_λ can be built: namely given a subset of $\mathcal{T}_A^{\ell ST_\lambda}$, there may generally exist multiple relations enjoying the simulation properties. Symbols `BiS` and `Sim` have been thereby adopted to denote the whole set of all possible simulations and bisimulations respectively, on $\mathcal{T}_A^{\ell ST_\lambda}$.

What about the union of every possible simulation? It is a simulation in turn, since if every element of a set of relations has the simulation property, also every union of elements of this set has the same property.

The union of *every* possible relation in `Sim` is the greatest element of `Sim` and it is called *similarity* denoting it by $\preceq_A$. Analogously let attribute the name of *bisimilarity* to the greater element in `BiS` using the symbol $\sim_A$ to denote it.

As a consequence, (bi)similarity can be seen as a relation on terms, indexed by types. Thus bisimilarity is the greatest relation among terms symmetric and featured by the properties (2.24), (2.25) and (2.26). Similarity as a relation among closed terms without type distinction is denoted with $\preceq$, as well as bisimilarity with $\sim$.

Theorem 2.2 (On the preorder induced by similarity relation)**.** *Similarity is a preorder on the set of the terms and hence bisimilarity is an equivalence relation.*

Proof. The reflexivity of similarity follows from the previously proved Lemma 2.9.

With regard to the transitivity, it has been proved with Lemma 2.11 that the composition of two possibly different simulations has again the simulation property. Being similarity the union of every simulation, it contains every simulation relation, namely it is the *greatest* one.

Given $e, f, g \in \mathcal{T}_A^{\ell ST_\lambda}$ such that $e\,\mathcal{S}_A{}^{(1)} f$ and $f\,\mathcal{S}_A{}^{(2)} g$, by definition of composition $e(\mathcal{S}_A{}^{(1)} \circ \mathcal{S}_A{}^{(2)})g$ and it has been proved with Lemma 2.11 that $(\mathcal{S}_A{}^{(1)} \circ \mathcal{S}_A{}^{(2)}) \in$ `Sim`. Moreover, by definition of similarity we have $\forall e, f, g,\ (e, f) \in \mathcal{S}_A{}^{(1)} \Rightarrow (e, f) \in \preceq_A$, $(e, g) \in \mathcal{S}_A{}^{(2)} \Rightarrow (f, g) \in \preceq_A$ and $(e, g) \in (\mathcal{S}_A{}^{(1)} \circ \mathcal{S}_A{}^{(2)}) \Rightarrow (e, g) \in \preceq_A$, which prove the transitivity of $\preceq_A$.

For the same arguments bisimulation in ℓST_λ, which is symmetric being a union of symmetric relations is an equivalence relation. □

Example 2.1. *An example of two distinct programs which can be proved bisimilar are the following:*

$$e = \lambda x.\lambda y.\lambda z.\mathtt{and}\ (xy)\ (\mathtt{or}\ z\,\mathtt{tt}); \qquad f = \lambda x.\lambda y.\lambda z.x(\mathtt{or}\ (\mathtt{and}\ z\,\mathtt{ff})\ y);$$

where and and or are the boolean function defined, in ℓST_λ, by relations (2.5). Both e and f can be given the type $(\mathtt{bool} \multimap \mathtt{bool}) \multimap \mathtt{bool} \multimap \mathtt{bool} \multimap \mathtt{bool}$ in the empty context: besides, if $\mathtt{b}_1, \mathtt{b}_2 \in \mathcal{V}^{\ell ST_\lambda}_{\mathtt{bool}}$ and $u \in \mathcal{V}^{\ell ST_\lambda}_{\mathtt{bool}\multimap\mathtt{bool}}$, both $e(u)(\mathtt{b}_1)(\mathtt{b}_2)$ and $f(u)(\mathtt{b}_1)(\mathtt{b}_2)$ may be validated to evaluate to $g\{\mathtt{b}_1/x'\}$, where u has been setted to $\lambda\, x'.g$. In fact

$$\begin{aligned} e\,((u)(\mathtt{b}_1)(\mathtt{b}_2)) \to \lambda\, y.\lambda\, z.\mathtt{and}\,(uy)(\mathtt{or}\,z\,\mathtt{tt})(\mathtt{b}_1)(\mathtt{b}_2) \to \lambda\, z.\mathtt{and}\,(u\mathtt{b}_1)(\mathtt{or}\,z\,\mathtt{tt})(\mathtt{b}_2) \to \\ \mathtt{and}\,(u\mathtt{b}_1)(\mathtt{or}\,\mathtt{b}_2\,\mathtt{tt}) \to \mathtt{and}\,g\{\mathtt{b}_1/x'\}\,\mathtt{tt} \to g\{\mathtt{b}_1/x'\}; \\ f\,((u)(\mathtt{b}_1)(\mathtt{b}_2)) \to \lambda\, y.\lambda\, z.u\mathtt{or}\,(\mathtt{and}\,z\,\mathtt{ff})\,y(\mathtt{b}_1)(\mathtt{b}_2) \to \lambda\, z.u\,(\mathtt{or}\,(\mathtt{and}\,z\,\mathtt{ff})(\mathtt{b}_1)(\mathtt{b}_2) \\ \to u\,\mathtt{or}\,(\mathtt{and}\,\mathtt{b}_2\,\mathtt{ff})\,(\mathtt{b}_1) \to u\,\mathtt{or}\,\mathtt{ff}\,\mathtt{b}_1 \to g\{\mathtt{b}_1/x'\}. \end{aligned}$$

Thus e and f can be proved to be bisimilar just by giving a preorder defined as the reflexive closure of

$$R_{e,f} = \{(e,f)\,, (e(\nu), f(\nu))\,, (e(\nu)(\mathtt{b}), f(\nu)(\mathtt{b}))\,, (e(\nu)(\mathtt{b})(\mathtt{b}'), f(\nu)(\mathtt{b})(\mathtt{b}'))\,,\} \qquad (2.30)$$

where $\nu \in \mathcal{V}^{\mathtt{bool}\multimap\mathtt{bool}}$ and $\mathtt{b}, \mathtt{b}' \in \mathcal{V}^{\mathtt{bool}}$ are generic values.

Another interesting example of terms which can be proved bisimilar are the term $e = \mathtt{if}\ f\ \mathtt{then}\ g\ \mathtt{else}\ h$ and the term ℓ obtained from e by λ-abstracting all variables which occur free in g (or, equivalently, in h), then applying the same variables to the obtained term. □

Is it that bisimilarity is sound for (i.e., included in) context equivalence? And how about the reverse inclusion? For a linear, deterministic λ-calculus like the one we are describing, both questions have already been given a positive answer [17]. In the next two sections, we will briefly sketch how the correspondence can be proved.

2.4.1 Open Extension of Applicative (Bi)similarity

In last section, two new relations among closed terms of ℓST_λ have been introduced that have been proved to be respectively a preorder and an equivalence relation (Theorem 2.2).

Nevertheless, this is only one among the requirements that a well–built relation on terms is required to fulfill, since the most desirable property for such a relation is the compatibility, which prescribes that the relation commutes with the syntactic operators of the language itself.

We exhibited the context equivalence, which fits these features, as a good tool to compare terms and programs by stating that two (or more) terms are contextually equivalent if they behave the same way – or they can be interchanged also – in whatever context.

The context here must be understood as a bigger program with an hole inside, acting as a container for smaller ones. Thus, the notion of context equivalence meets that of observational behaviour since two programs are thought to be equivalent if they behave the same when they are embedded inside whatever bigger environment.

Unfortunately, as it has been remarked, this notion of context equivalence is not easily exploitable because of the difficulty to deal with the quantification over all contexts. This is mainly the reason that the notion of bisimulation conceived in the beginnings of eighties and strongly applied in theoretical computer science with the works of Howe (1989) and Abramsky and Ong (1993) is now considered as one reliable alternative to check the equality between programs.

To show that (bi)similarity is preserved by composing terms through the syntactic constructors of the language, could prove to be a rather engaging challenge: before accomplishing such a check, we should extend the notion of similarity so that it could be possible to perform a comparison on both closed as well as open terms, being the open terms those which are defined on a non–empty set of free variables. This notion of (bi)similarity on open terms is known as *open extension* of applicative (bi)similarity.

Denoting by $\mathtt{fv}(e)$ the set of free variables of e, namely the domain of the context Γ involved in the typing judgement $\Gamma \vdash e : A$, we define the term e to be open if $\mathtt{fv}(e) \neq \emptyset$.

Given a typing context Γ such that $dom(\Gamma) = \{x_i\}_{i\in\mathcal{I}} \neq \emptyset$, a Γ–closure for Γ is nothing more than a set of suitable values $\{v_i\}_{i\in\mathcal{I}}$ whose types make possible the substitutions $\{v_i/x_i\}_{i\in\mathcal{I}}$.

A pair of open terms $\Gamma \vdash e, f : A$ which are typeable on the same context, is similar on Γ whether every Γ–closure of the couple is similar, namely if they fit the following requirement

$$\forall\, \{v_i\}_{i\in\mathcal{I}}\, \Gamma\text{–closure},\ e\{v_i/x_i\}_{i\in\mathcal{I}} \preceq_A h\{v_i/x_i\}_{i\in\mathcal{I}}. \tag{2.31}$$

If (2.31) is satisfied we will write $e \preceq_{\Gamma,A} h$. Likewise for bisimilarity.

Lemma 2.12 (Open simulations and bounded variables)**.** *(Bi)similarity is preserved under linkage of a variable, namely*

$$\lambda\, y.e \preceq_{\Gamma,B\multimap A} \lambda\, y.h \ \Leftrightarrow e \preceq_{\Gamma,y:B,A} h \tag{2.32}$$

Proof. It comes directly by the definition given for simulation on open terms, indeed by definition of open simulation applied on e and h it follows:

$$\Gamma, y : B \vdash e \preceq_A h : A \ \Rightarrow\ \forall\{v_i\}_{i\in\mathcal{I}} \in \mathcal{V}^{\ell ST_\lambda}_{\{\otimes A_i\}_{i\in\mathcal{I}}}, \forall w \in \mathcal{V}^{\ell ST_\lambda}_B \\ e\{v_i/x_i, w/y\}_{i\in\mathcal{I}} \preceq_A h\{v_i/x_i, w/y\}_{i\in\mathcal{I}}. \tag{2.33}$$

Analogously, applying the same definition to $\lambda\, y.e$ and $\lambda\, y.h$, we find

$$\Gamma \vdash \lambda\, y.e \preceq_{B\multimap A} \lambda\, y.h : B \multimap A \ \Rightarrow \\ \forall\{v_i\}_{i\in\mathcal{I}} \in \mathcal{V}^{\ell ST_\lambda}_{\{\otimes A_i\}_{i\in\mathcal{I}}}, \lambda\, y.e\{v_i/x_i\}_{i\in\mathcal{I}} \preceq_A \lambda\, y.h\{v_i/x_i\}. \tag{2.34}$$

Finally, using the definition (2.25) for closed terms of function type we get

$$\lambda\, y.e\{v_i/x_i\}_{i\in\mathcal{I}} \preceq_{B\multimap A} \lambda\, y.h\{v_i/x_i\}_{i\in\mathcal{I}} \Rightarrow \\ \forall w \in B,\ e\{v_i/x_i, w/y\}_{i\in\mathcal{I}} \preceq_A h\{v_i/x_i, w/y\}_{i\in\mathcal{I}} \tag{2.35}$$

The result is obtained comparing the right hand sides of (2.33) and (2.35). □

2.5 Similarity is a Precongruence

Now it is time to analyze more thoroughly the problem whether similarity may be a precongruence, recalling that a relation over the set $\mathcal{T}^{\ell ST_\lambda}$ is a precongruence whether it is a preorder (reflexive and transitive) and it is *compatible*, namely if it respects properties of compatibility previously stated in points (2.21a)–(2.21f).

Lemma 2.13 (Compatibility entails reflexivity)**.** *Every relation among terms which is compatible is also reflexive.*

Proof. We prove it by induction on the structure of e.

$-e = x-$ The relation $x : A \vdash x \, R \, x : A$ is true because of (2.21a)

$-e = \lambda x.f-$ Since R is compatible then by induction hypothesis it is reflexive on the smaller terms and $x : A \vdash f \, R \, f : A$. Therefore by compatibility (2.21b) we get the thesis $\emptyset \vdash \lambda x.f \, R \, \lambda x.f : B \multimap A$.

$-e = f_1 f_2-$ Here the (double) inductive hypothesis on the compatible R tells that $\emptyset \vdash f_1 \, R \, f_1 : B \multimap A$ and $\emptyset \vdash f_2 \, R \, f_2 : B$, whence by compatibility thesis comes $\emptyset \vdash f_1 f_2 \, R \, f_1 f_2 : A$.

$-e = (\texttt{if } f_1 \texttt{ then } f_2 \texttt{ else } f_3)-$ Here the induction hypothesis on the subterms of e tells us that the relation is reflexive, being compatibele, whence it comes $\emptyset \vdash f_1 \, R \, f_1 :$ $\texttt{bool}$, $\emptyset \vdash f_2 \, R \, f_2 : A$ $\emptyset \vdash f_3 \, R \, f_3 : A$, then by compatibility of R we get $e \, R \, e$.

$-e = (\texttt{let } f_1 \texttt{ be } \langle x, y \rangle \texttt{ in } f_2)-$ the structure of the proof is analogous to that of previous cases. □

Lemma 2.14. *Any compatible relation satisfies the following conditions listed below:*

$$(\mathbf{c-3l}) \qquad \Gamma \vdash e \, R \, h : B \multimap A \wedge \ \Delta \vdash f : B \ \Rightarrow \ \Gamma, \Delta \vdash ef \, R \, hf : A \tag{2.36a}$$

$$(\mathbf{c-3r}) \qquad \Gamma \vdash e : B \multimap A \wedge \ \Delta \vdash f \, R \, \ell : B \ \Rightarrow \Gamma, \Delta \vdash ef \, R \, e\ell : A \tag{2.36b}$$

$$(\mathbf{c-5l}) \qquad \Gamma \vdash e \, R \, h : E \otimes B \wedge \ \Delta, x : E, y : B \vdash f : A \ \Rightarrow$$
$$\Gamma, \Delta \vdash (\texttt{let } e \texttt{ be } \langle x, y \rangle \texttt{ in } f) \, R \, (\texttt{let } h \texttt{ be } \langle x, y \rangle \texttt{ in } f) : A \tag{2.36c}$$

$$(\mathbf{c-5r}) \qquad \Gamma \vdash e : E \otimes B \wedge \ \Delta, x : E, y : B \vdash f \, R \, \ell : A \ \Rightarrow$$

$$\Gamma, \Delta \vdash (\texttt{let}\ e\ \texttt{be}\ \langle x, y\rangle\ \texttt{in}\ f)\, R\, (\texttt{let}\ e\ \texttt{be}\ \langle x, y\rangle\ \texttt{in}\ \ell) : A \quad (2.36d)$$

Proof. Conditions (2.36a) and (2.36b) come from property (2.21c), and conditions (2.36c) and (2.36d) can be derived from (2.21e) by reflexivity. Indeed it has been proved (Lemma 2.13) that a compatible relation is also reflexive. □

Lemma 2.15. *In every transitive relation R properties (2.36a) and (2.36b) entail (2.21c). Analogously (2.36c) and (2.36d) entail (2.21e)*

Proof. Since $\forall f \in \mathcal{T}_B^{\ell ST_\lambda}\ \Gamma, \Delta \vdash e f\, R\, h f : A$ by property (2.36a) and $\forall h \in \mathcal{T}_{B\multimap A}^{\ell ST_\lambda}$ $\Gamma, \Delta \vdash h f\, R\, h \ell : A$ by property (2.36b), then (2.21c) is true by transitivity.

Similar is the proof for (2.21e). □

A natural way to prove that similarity is included in the context preorder, (and thus that bisimilarity is included in context equivalence) consists of first showing that similarity is a *precongruence*, that is to say a preorder relation which is compatible with all the operators of the language.

2.5.1 A First Failure

A direct proof that similarity is compatible could be driven by induction on the structure of e. In the following only closed terms are examined, with the purpose to later extend these results to open ones.

$-(2.21a)-$ $x : A \vdash x \preceq_A x : A$ Since x is a variable, using the open extension for similarity one gets $x \preceq_A x \Leftrightarrow \forall v \in \mathcal{V}_A^{\ell ST_\lambda}$, $x\{v/x\} \preceq_A x\{v/x\}$, which is true (Lemma 2.9).

$-(2.21b)-$ Property $e \preceq_{x:B,A} h \Rightarrow \lambda x.e \preceq_{\emptyset, B\multimap A} \lambda x.h$ is a direct consequence of Lemma 2.12.

$-(2.21d)-$ Proving this property mean to show the validity of the following statement: under the hypotheses $e \preceq_{\emptyset,\texttt{bool}} h$, $f \preceq_{\emptyset,A} \ell$, $g \preceq_{\emptyset,A} a$, it holds the property $\texttt{if}\ e\ \texttt{then}\ f\ \texttt{else}\ g \preceq_{\emptyset,A} \texttt{if}\ h\ \texttt{then}\ \ell\ \texttt{else}\ a$.

- Let $e \Downarrow v$, then by the hypothesis of similarity $e \preceq_{\emptyset,\mathtt{bool}} h$, it follows that e and h will evaluate to the same boolean constant $\mathtt{b}$. Let suppose that $\mathtt{b} = \mathtt{tt}$, then further hypothesis $f \preceq_{\emptyset,A} \ell$ ensures that $f \Downarrow u \Rightarrow \ell \Downarrow w$ with $u \preceq_A w$ and by applying rule $(if_{\mathtt{tt}} \Downarrow)$ one finds $(\mathtt{if}\ e\ \mathtt{then}\ f\ \mathtt{else}\ g) \Downarrow u$ and $(\mathtt{if}\ g\ \mathtt{then}\ h\ \mathtt{else}\ a) \Downarrow w$, therefore let us rewrite (2.27) as

$$((\mathtt{if}\ e\ \mathtt{then}\ f\ \mathtt{else}\ g, A), a_{eval}, (\widehat{u}, A)) \in \mathcal{N} \ \Rightarrow \\ ((\mathtt{if}\ h\ \mathtt{then}\ \ell\ \mathtt{else}\ a, A), a_{eval}, (\widehat{w}, A)) \in \mathcal{N} \wedge u \preceq_A w$$

  which is the thesis since condition (2.27) is matched. Similarly if $\mathtt{b} = \mathtt{ff}$ using the hypothesis $g \preceq_{\emptyset,A} a$ and the rule $(if_{\mathtt{ff}} \Downarrow)$.

- If e is divergent, then the whole term $(\mathtt{if}\ e\ \mathtt{then}\ f\ \mathtt{else}\ g)$ diverges, which ensures thesis.

$-(2.21e)-$ Here the statement (2.21e) may be splitted in two different parts: (2.36d) and (2.36c) which whether both verified allow to conclude that (2.21e) holds by Lemma 2.15.

$$\forall f \in \mathcal{T}_A^{\ell ST_\lambda},\ e \preceq_A g \Rightarrow (\mathtt{let}\ e\ \mathtt{be}\ \langle x, y\rangle\ \mathtt{in}\ f) \preceq_A (\mathtt{let}\ g\ \mathtt{be}\ \langle x, y\rangle\ \mathtt{in}\ f) \quad (2.36c)$$
$$\forall e \in \mathcal{V}_{E\otimes B}^{\ell ST_\lambda}, f \preceq_A h \ \Rightarrow (\mathtt{let}\ e\ \mathtt{be}\ \langle x, y\rangle\ \mathtt{in}\ f) \preceq_A (\mathtt{let}\ e\ \mathtt{be}\ \langle x, y\rangle\ \mathtt{in}\ h) \quad (2.36d)$$

To prove (2.36d) just recall the definition of similarity (2.27).

- Supposing that $e \Downarrow \langle u_1, u_2\rangle$, the hypothesis $\Delta, x : E, y : B \vdash f \preceq_A h : A$ gives

$$((\mathtt{let}\ e\ \mathtt{be}\ \langle x, y\rangle\ \mathtt{in}\ f, A), a_{eval}, (\widehat{v}, A)) \in \mathcal{N} \ \Rightarrow \\ ((\mathtt{let}\ e\ \mathtt{be}\ \langle x, y\rangle\ \mathtt{in}\ h, A), a_{eval}, (\widehat{w}, A)) \in \mathcal{N} \ \wedge \ v \preceq_A w, \quad (2.37)$$

  where v and w can be obtained resorting the rule $(let \Downarrow)$ which gives us $f\{u_1/x, u_2/y\} \Downarrow v$ and $h\{u_1/x, u_2/y\} \Downarrow w$. Now we know by Lemma 2.10 that $f\{u_1/x, u_2/y\} \sim_A v$ and $h\{u_1/x, u_2/y\} \sim_A w$, and by definition (2.31) that $f\{u_1/x, u_2/y\} \preceq_A h\{u_1/x, u_2/y\}$.

- If, instead, e is a divergence, this makes whole term ($\texttt{let}\ e\ \texttt{be}\ \langle x, y\rangle\ \texttt{in}\ f$) to diverge, proving the thesis.

This proves the first part of the thesis (2.36d). To prove (2.36c) one starts by the hypothesis $e \preceq_{\emptyset, E\otimes B} g$.

- Supposing $e \Downarrow \langle v_1, v_2\rangle$, the hypothesis of similarity ensures that $g \Downarrow \langle \nu_1, \nu_2\rangle$. Then one exploits the label $a_{\otimes f}$ to obtain

$$\forall f \in \mathcal{T}^{\ell ST_\lambda}_{x:E,y:B,A},\ ((\langle v_1, v_2\rangle, E\otimes B), a_{\otimes f}, (f\{v_1/x, v_2/y\}, A)) \in \mathcal{N}\ \Rightarrow$$
$$((\langle w_1, w_2\rangle, E\otimes B), a_{\otimes f}, (f\{\nu_1/x, \nu_2/y\}, A)) \in \mathcal{N}\ \wedge$$
$$f\{v_1/x, v_2/y\} \preceq_A f\{\nu_1/x, \nu_2/y\}, \quad (2.38)$$

  which proves exactly the desired property.

- Besides, likewise for property (2.36d), the divergence of e implies that also ($\texttt{let}\ e\ \texttt{be}\ \langle x, y\rangle\ \texttt{in}\ f$) will diverge, making the thesis true.

Thus property (2.21e) has been proved.

$-(2.21c)-$ To prove this property we should prove (2.36a) and (2.36b) and use Lemma 2.15. The proof of (2.36a) starts from the hypothesis $e \preceq_{\emptyset, B\multimap A} h$.

- Supposing $e \Downarrow v$ and using definition of similarity (2.27) gives:

$$((e, A), a_{eval}, (\widehat{v}, A)) \in \mathcal{N}\ \Rightarrow ((h, A), a_{eval}, (\widehat{w}, A)) \in \mathcal{N}\ \wedge\ v \preceq_{B\multimap A} w,$$

  thus, choosing $v = \lambda x.\bar{e}$ and $w = \lambda x.\bar{h}$ let us use (2.25) supposing $f \Downarrow u$, the last statement $v \preceq_{B\multimap A} w$ will be equivalent to

$$((\lambda x.\bar{e}, B \multimap A), a_{@u}, (\bar{e}\{u/x\}, A)) \in \mathcal{N} \Rightarrow$$
$$\big((\lambda x.\bar{h}, B \multimap A), a_{@u}, (\bar{h}\{u/x\}, A)\big) \in \mathcal{N} \wedge\ \bar{e}\{u/x\}\, \mathcal{S}_A\, \bar{h}\{u/x\},$$

  this proves the thesis $\forall f \in \mathcal{V}^{\ell ST_\lambda}_B$, $ef \preceq_{\emptyset, A} hf$ since the last relation is true by definition of simulation (2.25).

- Otherwise, if e is a divergence, the term ef is divergent in turn and this makes the thesis true.

This proved the first part (2.36a): now we would prove (2.36b), hence $\forall e \in \mathcal{T}^{\ell ST_\lambda}_{B\multimap A},\ f\,\mathcal{S}_B\,\ell \Rightarrow ef\,\mathcal{S}_A\,e\ell$.

Thus from hypothesis $f \preceq_{\emptyset,B} \ell$, one gets, by definition of similarity (2.27):

$$((f,B),a_{eval},(\widehat{v},B)) \in \mathcal{N} \ \Rightarrow ((\ell,B),a_{eval},(\widehat{w},B)) \in \mathcal{N} \ \wedge \ v \preceq_B w.$$

Let us notice that in the more general case $B = E' \multimap E$ is a function type. Now supposing that $e \Downarrow \lambda x.\overline{e}$, to prove (2.36b), namely $\forall e \in \mathcal{T}^{\ell ST_\lambda}_{B\multimap A},\ ef \preceq_A e\ell$, requires to show that holds the following

$$\overline{e}\{v/x\} \preceq_A \overline{e}\{w/x\}. \tag{2.39}$$

Indeed, by definition of simulation (2.25) for functions and by the relationship $ef \preceq_A e\ell,\ \forall e$, we get:

$$((ef,A),a_{eval},(\widehat{u},A)) \in \mathcal{N} \Rightarrow ((e\ell,A),a_{eval},(\widehat{\nu},A)) \in \mathcal{N} \ \wedge \ u \preceq_A \nu, \tag{2.40}$$

whence, provided that $\overline{e}\{v/x\} \Downarrow u$ and $\overline{e}\{w/x\} \Downarrow \nu$, the second condition of (2.40) is equivalent to (2.39).

Unfortunately there is no chance to prove that similarity enjoys the *subsitutivity*, namely that given $v \preceq_B v'$, $\forall \overline{e} \in \mathcal{T}^{\ell ST_\lambda}_{x:B,A}\ \overline{e}\{v/x\} \preceq_A \overline{e}\{v'/x\}$. Since the same argument can be repeated endless (v, v' can be taken as λ-abstractions in turn), we get stuck because we can not terminate the chain.

$-(2.21f)-$ In a similar way we get stuck in attempting to prove the property of compatibility for pairs – namely $v_1 \preceq_A w_1$, $v_2 \preceq_B w_2 \Rightarrow \langle v_1,v_2\rangle \preceq_{A\otimes B} \langle w_1,w_2\rangle$ – which would be valid only in case that the substitutivity was a characteristic ascribable to the similarity, entailing this way the relation $\langle v_1,v_2\rangle \preceq_{A\otimes B} \langle w_1,w_2\rangle \Rightarrow \forall g \in \mathcal{T}^{\ell ST_\lambda}_{x:A,y:B,E}\ g\{v_1/x,v_2/y\} \preceq_E g\{w_1/x,w_2/y\}$.

This leaves open the problem whether similarity to be a precongruence or no.

2.6 Howe's Lifting

While proving that $\preceq$ is a preorder is relatively easy, the naive proof of compatibility (i.e. the obvious induction) fails, because of application case. How can it be proved that similarity is a compatible and therefore a precongruence? A nice way is due to Howe [30], who proposed a powerful and reasonably robust proof based on so–called precongruence candidates. Intuitively, the structure of Howe's method is the following [44]:

1. First of all, one defines an operator $(\cdot)^H$ on typed relations, in such a way that whenever a typed relation R is a preorder, R^H is a precongruence.
2. One then proves, under the condition that R is an equivalence relation, that R is included in R^H, and that R^H is substitutive.
3. Finally, one proves that $\preceq^H$ is itself an applicative simulation. This is the so-called Key Lemma [44], definitely the most difficult of the three steps.

Points 2 and 3 together imply that $\preceq$ and $\preceq^H$ coincide. But by point 1, $\preceq^H$, thus also $\preceq$, are precongruences. In Figure 2.8, one can find the rules defining $(\cdot)^H$ when the underlying terms are those of ℓST_λ.

Lemma 2.16 (① Compatibility of R^H)**.**
If R is reflexive then R^H is compatible.

Proof. Let $\{e_n\}_{n\in\mathcal{N}}$ and $\{h_n\}_{n\in\mathcal{N}}$ denote the smaller subterms which enter in the syntax of two terms e and h. The statement which must be proved may be written as: $\forall e, h$

$$\Delta_1 \vdash e_1\, R^H h_1 : A_1 \ldots\ldots \quad \Delta_N \vdash e_N\, R^H h_N : A_N \Rightarrow\ \Delta_1 \cdots \Delta_N \vdash e\, R^H h : A \tag{2.41}$$

Using this notation it is possible to rewrite the generic Howe's relation as

$$\frac{\begin{array}{c}\Delta_1 \vdash e_1\, R^H h_1 : A_1 \\ \vdots \\ \Delta_N \vdash e_N\, R^H h_N : A_N \qquad \Delta_1 \ldots \Delta_N \vdash h\, R\, b : A\end{array}}{\Delta_1 \ldots \Delta_N \vdash e\ R^H\, b : A}\,. \tag{2.42}$$

HOWE'S RULE	NAME
$\dfrac{\emptyset \vdash c\, R\, b : A}{\emptyset \vdash c\, R^H\, b : A}$	(How_{1c})
$\dfrac{x : A \vdash x\, R\, b : A}{\emptyset \vdash x\, R^H\, b : A}$	(How_{1v})
$\dfrac{\Gamma, x : B \vdash e\, R^H\, h : A \qquad \Gamma \vdash \lambda x.h\, R\, b : B \multimap A}{\Gamma \vdash \lambda x.e\, R^H\, b : B \multimap A}$	(How_2)
$\dfrac{\Gamma \vdash e\, R^H\, h : B \multimap A \qquad \Delta \vdash f\, R^H\, \ell : B \qquad \Gamma, \Delta \vdash h\ell\, R\, b : A}{\Gamma, \Delta \vdash ef\, R^H\, b : A}$	(How_3)
$\dfrac{\begin{array}{l}\Gamma \vdash e\, R^H\, h : \mathtt{bool} \\ \Delta \vdash f\, R^H\, \ell : A \\ \Delta \vdash g\, R^H\, a : A \qquad \Gamma, \Delta \vdash (\mathtt{if}\ h\ \mathtt{then}\ \ell\ \mathtt{else}\ a)\, R\, b : A\end{array}}{\Gamma, \Delta \vdash (\mathtt{if}\ e\ \mathtt{then}\ f\ \mathtt{else}\ g)\, R^H\, b : A}$	(How_4)
$\dfrac{\begin{array}{l}\Gamma \vdash e\, R^H\, h : X \otimes Y \\ \Delta, x : X, y : Y \vdash f\, R^H\, \ell : A \qquad \Gamma, \Delta \vdash (\mathtt{let}\ h\ \mathtt{be}\ \langle x, y\rangle\ \mathtt{in}\ \ell)\, R\, b : A\end{array}}{\Gamma, \Delta \vdash (\mathtt{let}\ e\ \mathtt{be}\ \langle x, y\rangle\ \mathtt{in}\ f)\, R^H\, b : A}$	(How_5)
$\dfrac{\Gamma \vdash e\, R^H\, h : A \qquad \Delta \vdash f\, R^H\, \ell : B \qquad \Gamma, \Delta \vdash \langle h, \ell\rangle\, R\, b : A \otimes B}{\Gamma, \Delta \vdash \langle e, f\rangle\, R^H\, b : A \otimes B}$	(How_6)

Figure 2.8: Howe's lifting for the terms of ℓST_λ.

Since by hypothesis R is reflexive, in (2.42) it is possible to take $b = h$ which drives immediately to the thesis. □

Let point out as an immediate consequence of the previous Lemma 2.16, that if R is a preorder, its Howe's lifting R^H is a precongruence.

Lemma 2.17 (② Inclusion)**.** *If R is reflexive and transitive, then it is contained in R^H.*

Proof. The statement can be written as

$$\forall e, b,\ \Gamma \vdash e\ R\ b : A \Rightarrow \Gamma \vdash e\ R^H\ b : A. \tag{2.43}$$

and the property can be proved by induction on the structure of e.

▶ In the basic case, when e is a variable it is a consequence of the corresponding Howe's rule (How_{1v}), indeed $\dfrac{x : A \vdash x\, R\, b : A}{x : A \vdash x\, R^H\, b : A}$.

▶ When e is a more complex term, let us assume it is built with some constructors of the language starting by simpler terms whose set is denoted by $\{e_n\}_{n=1,\dots N}$ such that $e = \mathtt{cnstr}\,(\{e_n\}_{n\in\mathcal{N}})$, where $\mathtt{cnstr}$ stands for some syntactic constructor of the language. Similarly let us write $h = \mathtt{cnstr}\,(\{h_n\}_{n\in\mathcal{N}})$, being $\{h_n\}_{n\in\mathcal{N}}$ the subterms of h.

By Lemma 2.16, it has been proved that the reflexivity of R entails the compatibility of R^H, then R^H is reflexive too, by Lemma 2.13, being a compatible relation. Thus, considering the general Howe's rule

$$\frac{\begin{array}{l}\Delta_1 \vdash e_1\, R^H h_1 : A_1 \\ \qquad\vdots \\ \Delta_N \vdash e_N\, R^H h_N : A_N \qquad\qquad \Delta_1 \dots \Delta_N \vdash h\, R\, b : A\end{array}}{\Delta_1 \dots \Delta_N \vdash e\, R^H b : A} \tag{2.44}$$

by the reflexivity of R^H, we may write the N statements $\{\Delta_n \vdash e_n\, R^H e_n : A_n\}_{n\in\mathcal{N}}$ and use these conditions as premises of (2.44) together with $\Delta_1 \dots \Delta_N \vdash e\, R\, b : A$. Considering the last premise of (2.44), this prove the statement since $\forall e, b,\ \Gamma \vdash e\, R\, b : A \Rightarrow \Gamma \vdash e\, R^H b : A$, choosing $\Gamma = \Delta_1 \dots \Delta_N$. □

Lemma 2.18 (Pseudo transitivity of R^H)**.**
If R is transitive, then R^H enjoys the pseudo–transitivity property expressed by the following relation

$$\forall e, f, h, \big(\Delta \vdash e\ R^H\ f : A\ \wedge\ \Delta \vdash f\ R\ h : A\big)\ \Rightarrow \Delta \vdash e\ R^H\ h : A. \tag{2.45}$$

Proof. It is easy to justify this property, indeed if $e = \mathtt{cnstr}\,(\{e_i\}_{i\in\mathcal{I}})$, the first sentence in the hypothesis must be the result of the application of some Howe's rule with general form:

$$\frac{\begin{array}{l}\Delta_1 \vdash e_1\ R^H\ \ell_1 : A_1 \\ \qquad\vdots \\ \Delta_N \vdash e_N\ R^H\ \ell_N : A_N \qquad\qquad \Delta_1 \dots \Delta_N \vdash \ell\ R\ f : A\end{array}}{\underbrace{\Delta_1 \dots \Delta_N}_{\Delta} \vdash e\ R^H\ f : A} \tag{2.46}$$

here we understood the same notation previously used, where $\{e_n\}_{n=1\ldots N}$ is the set of subterms of e and likewise for ℓ.

Now using the second hypothesis of (2.45) together with the transitivity of R we get $\Delta_1 \ldots \Delta_N \vdash \ell \; R \; f : A$ and $\Delta_1 \ldots \Delta_N \vdash f \; R \; h : A \Rightarrow \Delta_1 \ldots \Delta_N \vdash \ell \; R \; h : A$, whence taking this last result as a premise for the Howe's general rule, we obtain the thesis as a consequence of the application of the rule (2.46):

$$\frac{\begin{array}{l}\Delta_1 \vdash e_1 \; R^H \, \ell_1 : A_1 \\ \qquad\vdots \\ \Delta_N \vdash e_N \; R^H \, \ell_N : A_N \qquad \Delta_1 \ldots \Delta_N \vdash \ell \; R \; h : A\end{array}}{\Delta_1 \ldots \Delta_N \vdash e \; R^H \, h : A} \tag{2.47}$$

□

Lemma 2.19 (② Substitutivity of R^H)**.** *If R is reflexive, transitive and closed under substitution, then its Howe's lifting R^H is substitutive. The property of substitutivity – which is the thesis – may be stated as*

$$\Gamma, x : B \vdash e \, R^H h : A \;\wedge\; \Delta \vdash f \, R^H \ell : B \;\Rightarrow \Gamma, \Delta \vdash e\{f/x\} \, R^H h\{\ell/x\} : A. \tag{2.48}$$

Proof. We prove it inductively, on the derivation of the generic term e.

$-e = x-$ In the basic case e is a variable which may belong or not to $dom(\Gamma)$.

In linear case $x \notin dom(\Gamma)$ and therefore e and f are the same type which is the type of x and the statement to prove becomes

$$\Gamma, x : A \vdash x \, R^H h : A \;\wedge\; \Delta \vdash f \, R^H \ell : A \;\Rightarrow \Gamma, \Delta \vdash x\{f/x\} \, R^H h\{\ell/x\} : A, \tag{2.49}$$

where the first type judgement in the hypothesis must be a consequence of the application of rule (How_{1v}) which has as premise $\Gamma, x : A \vdash x \, R \, h : A$

Now, recalling that R is closed under substitution, the previous judgement entails that $\forall \ell \in \mathcal{T}^{\ell ST_\lambda}_{\Delta,A}$, $\Gamma, \Delta \vdash x\{\ell/x\} \, R \, h\{\ell/x\} : A$, and using this last result together with the second hypothesis and the pseudo–transitivity of R^H we get

$$\Delta \vdash f \, R^H \ell : A \;\wedge\; \Gamma, \Delta \vdash \ell \, R \, h\{\ell/x\} : A \;\Rightarrow\; \Gamma, \Delta \vdash f \, R^H h\{\ell/x\} : A, \tag{2.50}$$

which is the thesis in the statement (2.49).

$-e = \mathtt{cnstr}\,(\{e_n\}_{n\in\mathcal{N}})-$ Here the hypothesis $\Gamma, x : B \vdash e\,R^{H}h : A$ in (2.48) must be a consequence of a general Howe rule as

$$\frac{\begin{array}{l}\Gamma_1 \vdash e_1\,R^{H}g_1 : A_1\\ \qquad\vdots\\ \Gamma_i, x : B \vdash e_i\,R^{H}g_i : A_i\\ \qquad\vdots\\ \Gamma_N \vdash e_N\,R^{H}g_N : A_N \qquad\qquad \Gamma_1 \ldots \Gamma_N, x : B \vdash g\,R\,h : A\end{array}}{\Gamma_1 \ldots \Gamma_N, x : B \vdash e\,R^{H}h : A} \tag{2.51}$$

where the usual notation has been followed, denoting by $\{e_k\}_{k=1,2\ldots}$, $\{g_k\}_{k=1,2\ldots}$ the subterms of e and g.

Using induction on the i-th term which, due to the linear hypothesis, is the only one which can contain the x variable we will write

$$\Gamma_i, x : B \vdash e_i\,R^{H}g_i : A_i \;\wedge\; \Delta \vdash f\,R^{H}\ell : B \;\Rightarrow \Gamma_i, \Delta \vdash e_i\{f/x\}\,R^{H}g_i\{\ell/x\} : A_i. \tag{2.52}$$

Now let us use the property of R to be closed under substitution on the last premise of (2.51)

$$\Gamma_1 \ldots \Gamma_N, x : B \vdash g\,R\,h : A \;\Rightarrow\; \forall \ell \in \mathcal{T}^{\ell ST_\lambda}_{\Delta,A},\ \Gamma_1 \ldots \Gamma_N, \Delta \vdash g\{\ell/x\}\,R\,h\{\ell/x\} : A, \tag{2.53}$$

and use (2.52) and (2.53) as premises of a general Howe's rule (2.51)

$$\frac{\begin{array}{l}\Gamma_1 \vdash e_1\,R^{H}g_1 : A_1\\ \qquad\vdots\\ \Gamma_i, \Delta \vdash e_i\{f/x\}\,R^{H}g_i\{\ell/x\} : A_i\\ \qquad\vdots\\ \Gamma_N \vdash e_N\,R^{H}g_N : A_N \qquad\qquad \Gamma_1 \ldots \Gamma_N, \Delta \vdash g\{\ell/x\}\,R\,h\{\ell/x\} : A\end{array}}{\Gamma_1 \ldots \Gamma_I, \Delta \vdash e\{f/x\}\,R^{H}g\{\ell/x\} : A}\,, \tag{2.54}$$

to get, as conclusion, the result which has to be proved. □

Lemma 2.20 (Key lemma: in ℓST_λ, $\preceq^H \subseteq \preceq$)**.**
$\preceq^H$ *is a simulation.*

Since the simulation relation is defined on types and on the top of the transition element of a LTS, we claim more properly this property distinguishing between values and terms, according to the following statements

$$\emptyset \vdash \mathtt{b} \preceq_{\mathtt{bool}}{}^H \mathtt{b}' : \mathtt{bool} \Rightarrow \mathtt{b} = \mathtt{b}' \tag{2.55a}$$

$$\emptyset \vdash \lambda x.f \preceq_{B \multimap A}{}^H \lambda x.h : B \multimap A \Rightarrow \forall v \in \mathcal{V}_B^{\ell ST_\lambda},\ \emptyset \vdash f\{v/x\} \preceq_A{}^H h\{v/x\} : A \tag{2.55b}$$

$$\emptyset \vdash \langle v_1, v_2\rangle \preceq_{A\otimes B}{}^H \langle w_1, w_2\rangle : A \otimes B \Rightarrow \forall g \in \mathcal{T}^{\ell ST_\lambda}_{x:A,y:B,E},\ g\{v_1/x, v_2/y\} \preceq_E{}^H g\{w_1/x, w_2/y\} \tag{2.55c}$$

$$\big(\emptyset \vdash e \preceq_A{}^H h : A \ \wedge\ e \Downarrow v\big) \Rightarrow \big(h \Downarrow w \ \wedge\ \emptyset \vdash v \preceq_A{}^H w : A.\big) \tag{2.55d}$$

Proof. We start by the analysis of the cases when the terms involved are values.

▶ If $e = \mathtt{b}$ and its type is $\mathtt{bool}$, the statement to prove is (2.55a). The hypothesis $\emptyset \vdash \mathtt{b} \preceq_{\mathtt{bool}}{}^H \mathtt{b}' : \mathtt{bool}$ is necessarily a consequence of rule (How_{1c}), whose unique premise is $\emptyset \vdash \mathtt{b} \preceq_{\mathtt{bool}} \mathtt{b}' : \mathtt{bool}$. Therefore with reference to the definition (2.24), supposing $\mathtt{b} = \mathtt{tt}$ it must be $\mathtt{b}' = \mathtt{tt}$, too. The same holds for $e = \mathtt{ff}$ since $\forall \mathtt{b} \in \mathcal{V}^{\mathtt{bool}}_{\ell ST_\lambda},\ \big((\widehat{e}, \mathtt{bool}), a_{\mathtt{b}}, (\widehat{\mathtt{b}}, \mathtt{bool})\big) \in \mathcal{N} \Rightarrow \Big((\widehat{\mathtt{b}'}, \mathtt{bool}), a_{\mathtt{b}}, (\widehat{\mathtt{b}}, \mathtt{bool})\Big) \in \mathcal{N}$. This proves the thesis $\mathtt{b} = \mathtt{b}'$.

▶ If $e = \lambda x.f$ then the statement of the key lemma is (2.55b). The hypothesis $\emptyset \vdash \lambda x.f \preceq_{B\multimap A}{}^H \lambda x.h : B \multimap A$ must have, as last rule (How_2) as shown below

$$\frac{x : B \vdash f \preceq_A{}^H g : A \qquad \emptyset \vdash \lambda x.g \preceq_{B\multimap A} \lambda x.h : B \multimap A}{\emptyset \vdash \lambda x.f \preceq_{B\multimap A}{}^H \lambda x.h : B \multimap A}\,. \tag{2.56}$$

The second premise of the last rule (2.56), namely $\emptyset \vdash \lambda x.g \preceq_{B\multimap A} \lambda x.h : B \multimap A$, which is a relation of similarity whose terms are arrow type value entails, by definition (2.25), that

$$\forall v \in \mathcal{V}_B^{\ell ST_\lambda},\ \Big((\widehat{\lambda x.g}, B \multimap A), a_{@v}, (g\{v/x\}, A)\Big) \in \mathcal{N} \Rightarrow \Big((\widehat{\lambda x.h}, B \multimap A), a_{@v}, (h\{v/x\}, A)\Big) \in \mathcal{N} \wedge\ g\{v/x\} \preceq_A h\{v/x\},$$

which gives, $\forall v \in \mathcal{V}_B^{\ell ST_\lambda}$, $\emptyset \vdash g\{v/x\} \preceq_A h\{v/x\} : A$. Putting together this last result with the first premise of the rule (2.56) which is $x : B \vdash f \preceq_A{}^H g : A$ and using both property of substitutivity (Lemma 2.19) and Lemma 2.18 we get the thesis:

$$\forall v \in \mathcal{V}^B,\ \big(\emptyset \vdash f\{v/x\} \preceq_A{}^H g\{v/x\} : A\ \wedge\ \emptyset \vdash g\{v/x\} \preceq_A h\{v/x\} : A\big)\ \Rightarrow \\ \Rightarrow\ \forall v \in \mathcal{V}^B, \emptyset \vdash f\{v/x\} \preceq_A{}^H h\{v/x\} : A. \quad (2.57)$$

► Let be $e = \langle v_1, v_2\rangle$ so that the property to prove is (2.55c). Since the term involved is a pair, then we are under the scope of rule (How_6) and we have, as for the first hypothesis

$$\frac{\begin{array}{ll}\emptyset \vdash v_1 \preceq_A{}^H \nu_1 : A & \\ \emptyset \vdash v_2 \preceq_B{}^H \nu_2 : B & \emptyset \vdash \langle \nu_1, \nu_2\rangle \preceq_{A\otimes B} \langle w_1, w_2\rangle : A \otimes B\end{array}}{\emptyset \vdash \langle v_1, v_2\rangle \preceq_{A\otimes B}{}^H \langle w_1, w_2\rangle : A \otimes B}. \quad (2.58)$$

Let us start from the last premise of (2.58) which is again a statement of similarity whose first term is a value, being $\langle \nu_1, \nu_2\rangle \preceq_{A\otimes B} \langle w_1, w_2\rangle$. Starting from this hypothesis and using the definition of simulation (2.26) for pairs values, we obtain the following two results

$$\forall g \in \mathcal{T}^{\ell ST_\lambda}_{x:A,y:B,E} \\ ((\langle \nu_1, \nu_2\rangle, A \otimes B), a_{\otimes g}, (g\{\nu_1/x, \nu_2/y\}, E)) \in \mathcal{N} \Rightarrow \\ ((\langle w_1, w_2\rangle, A \otimes B), a_{\otimes g}, (g\{w_1/x, w_2/y\}, E)) \in \mathcal{N}\ \wedge \\ g\{\nu_1/x, \nu_2/y\} \preceq_E g\{w_1/x, w_2/y\}. \quad (2.59)$$

Besides the first two premises of (2.58), by the substitutivity (Lemma 2.19) entail that $\forall g \in \mathcal{T}^{\ell ST_\lambda}_{x:A,y:B,E} g\{v_1/x, v_2/y\} \preceq_E{}^H g\{\nu_1/x, \nu_2/y\}$.

Now, using Lemma 2.18 we easily get the thesis since:

$$\forall g \in \mathcal{T}^{\ell ST_\lambda}_{x:A,y:B,E} \\ g\{v_1/x, v_2/y\} \preceq_E{}^H g\{\nu_1/x, \nu_2/y\}\ \wedge\ g\{\nu_1/x, \nu_2/y\} \preceq_E g\{w_1/x, w_2/y\} \Rightarrow \\ \Rightarrow g\{v_1/x, v_2/y\} \preceq_E{}^H g\{w_1/x, w_2/y\}, \quad (2.60)$$

which proves the thesis of the statement (2.55c).

► Supposing to deal with a couple of terms $\emptyset \vdash e \preceq_A{}^H h : A$ which are non values, the statement to prove is (2.55d). Here we suppose the term e as built up by a set of smaller subterms $\{e_n\}_{n\in\mathcal{N}}$ writing $e = \texttt{cnstr}\,(\{e_n\}_{n\in\mathcal{N}})$, denoting by $\texttt{cnstr}$ some generic syntactic constructor of the language.

If we look at the hypotheses of (2.55d) separately we must conclude that the first one comes from a general Howe's rule as

$$\frac{\begin{array}{l}\emptyset \vdash e_1 \preceq_{A_1}{}^H g_1 : A_1 \\ \qquad\vdots \\ \emptyset \vdash e_N \preceq_{A_N}{}^H g_N : A_N \qquad \emptyset \vdash g \preceq_A h : A\end{array}}{\emptyset \vdash e \preceq_A{}^H h : A} \tag{2.61}$$

where $\emptyset \vdash g_k : A_k$ has the above discussed meanings and stands for a typing judjement of a subterm of g.

The second hypothesis of (2.55d) must be consequence a generic big–step semantics evaluation rule which may be resumed in the following way

$$\frac{\begin{array}{l}e_1 \Downarrow u_1 \\ \quad\vdots \\ e_N \Downarrow u_N \qquad [\texttt{subst. rule}(\{u_n\}_{n\in\mathcal{N}}) \Downarrow v]\end{array}}{e \Downarrow v}\,, \tag{2.62}$$

where the substitution rule $\texttt{subst. rule}(\{u_n\}_{n\in\mathcal{N}})$ is in brackets since it appears only in the semantics rules of the constructors for terms *let* and *application*.

The proof is carried out by induction on the size of the big–step–semantics terms appearing in (2.62), but in addiction we may write down the N inductive hypotheses for the subterms of (2.61). Supposing that the N relations between the subterms has the simulation property we get the relationships

$$\left(e_n \Downarrow u_n \ \wedge\ \emptyset \vdash e_n \preceq_{A_n}{}^H g_n : A_n\right) \ \Rightarrow\ \left(g_n \Downarrow \nu_n \ \wedge\ \emptyset \vdash u_n \preceq_{A_n}{}^H \nu_n : A_n\right) \quad \forall n \in \mathcal{N}. \tag{2.63}$$

Since new N values $\{\nu_n\}_{n\in\mathcal{N}}$ have been obtained as a result of the inductive hypoth-

esis, applying the suitable evaluation rule, as in (2.62), we get

$$\frac{\begin{array}{l} g_1 \Downarrow \nu_1 \\ \quad \vdots \\ g_1 \Downarrow \nu_N \end{array} \qquad \texttt{subst. rule}(\{\nu_k\}_{k=1,2\ldots}) \Downarrow \nu}{g \Downarrow \nu}\,. \tag{2.64}$$

Taking this last result together with the last premise of (2.61) and using definition of simulation (2.27), one obtains

$$g \Downarrow \nu \ \wedge \ \emptyset \vdash g \preceq_A h : A \ \Rightarrow \ (h \Downarrow w \wedge \ \emptyset \vdash \nu \preceq_A w : A)\,, \tag{2.65}$$

which proves the first statement of the thesis (2.55d).

As for the second statement, let us just recall that ${\preceq_A}^H$ is a compatible relation and therefore, applying compatibility on (2.63) we find

$$\{\emptyset \vdash u_n {\preceq_{A_n}}^H \nu_n : A_n\}_{n\in\mathcal{N}} \ \Rightarrow \ \emptyset \vdash v {\preceq_A}^H \nu : A, \tag{2.66}$$

To complete the prove just remember the pseudo–transitivity of ${\preceq_A}^H$ which has been proved in Lemma 2.18 and let us apply it to relations (2.66) and (2.65) which give

$$\big(\emptyset \vdash v {\preceq_A}^H \nu : A \ \wedge \ \emptyset \vdash \nu \preceq_A w : A\big) \ \Rightarrow \ \emptyset \vdash v {\preceq_A}^H w : A. \tag{2.67}$$

□

Proposition 2.1. *As immediate results coming from Lemma 2.17 and from key Lemma 2.20 follow the relations* $\preceq_A = {\preceq_A}^H$ *and* $\sim_A = {\sim_A}^H$.

Proof.

$\preceq_A \subseteq {\preceq_A}^H$ by Lemma 2.17 and ${\preceq_A}^H \subseteq \preceq_A$ by Lemma 2.20, thus ${\preceq_A}^H = \preceq_A$

To show the validity of $\sim = \sim^H$, one should make use of cosimilarity, the inverse of similarity defined by the condition

$$\forall e, h \in \mathcal{T}_A^{\ell ST_\lambda},\ e \preceq_A h \ \Leftrightarrow h \preceq_A^{op} e. \tag{2.68}$$

It is easy to check from its definition on LTS that $\sim = \preceq \cap \preceq^{op}$; likewise let define ${\sim_A}^H = {\preceq_A}^H \cap (\preceq_A^{op})^H$. Since Lemma 2.17 and 2.20 entail analogues results for cosimilarity, we conclude that $(\preceq_A^{op})^H = \preceq_A^{op}$, whence it straight comes $\sim_A = {\sim_A}^H$. □

Thus $\preceq_A$ enjoys all the properties of ${\preceq_A}^H$, mainly it is therefore a compatible relation, then it is a precongruence on the set $\mathcal{T}_A^{\ell ST_\lambda}$ (similarly $\sim_A$ is a congruence on $\mathcal{T}_A^{\ell ST_\lambda}$).

2.7 Comparing Relations among Terms

An interesting question which could be asked is about the relationship between similarity and context preorder (and analogously between bisimilarity and context equivalence). To answer it, the following lemma is requested.

Lemma 2.21 (On a similarity behaviour with respect to contexts)**.** *Similarity relation is compatible with the context, namely it satisfies the condition*

$$\emptyset \vdash e \preceq_A h : A \;\Rightarrow\; \forall C \in \mathtt{CTX}_B\left(\emptyset \vdash A\right), \emptyset \vdash C[e] \preceq_B C[h] : B.$$

Proof. This property is an easy consequence of the compatibility of $\preceq_A$. It may be proved by induction on the contexts structure.

- $C[\cdot] = [\cdot] \in \mathtt{CTX}_A\left(\emptyset \vdash A\right)$ – gives the tautology $\emptyset \vdash e \preceq_A h : A \Rightarrow \emptyset \vdash e \preceq_A h : A$, obviously true.

- $C[\cdot] = \lambda x.D[\cdot] \in \mathtt{CTX}_B\left(\emptyset \vdash A\right)$ – requires to show $\emptyset \vdash \lambda x.D[e] \preceq_{E \multimap B} \lambda x.D[h] : E \multimap B$. Using induction hypothesis $x : E \vdash D[e] \preceq_B D[h] : B$, the property comes to be an obvious result of application of $(\mathbf{c-2})$.

- $C[\cdot] = D[\cdot]f \in \mathtt{CTX}_B\left(\emptyset \vdash A\right)$ – entails that one must prove $\emptyset \vdash D[e]f \preceq_B D[h]f : B$, with induction hypothesis $\emptyset \vdash D[e] \preceq_{E \multimap B} D[h] : E \multimap B$, where f has been assumed to have type E. Here we should apply the rule $(\mathbf{c-3})$, keeping in mind that $\forall f, \emptyset \vdash f \preceq_E f : E$. The proof for the class of contexts $C[\cdot] = fD[\cdot]$ is alike the previous one, employing property $(\mathbf{c-3})$.

$-C[\cdot] = (\texttt{if}\ D[\cdot]\ \texttt{then}\ f\ \texttt{else}\ g) \in \texttt{CTX}_B\,(\emptyset \vdash A)-$ Whether the context is in this form we need to prove the statement $\emptyset \vdash (\texttt{if}\ D[e]\ \texttt{then}\ f\ \texttt{else}\ g) \preceq_B (\texttt{if}\ D[h]\ \texttt{then}\ f\ \texttt{else}\ g) : B$. This is an immediate result of inductive hypothesis $\emptyset \vdash D[e] \preceq_{\texttt{bool}} D[h] : \texttt{bool}$ with relationships $\emptyset \vdash f \preceq_B f : B$ and $\emptyset \vdash g \preceq_B g : B$, provided that the property $(\mathbf{c-4})$ is granted. The proof is similar, making use of $(\mathbf{c-4})$, for linear contexts belonging to set $C[\cdot] = \texttt{if}\ f\ \texttt{then}\ D[\cdot]\ \texttt{else}\ G[\cdot]$.

$-C[\cdot] = (\texttt{let}\ D[\cdot]\ \texttt{be}\ \langle x, y\rangle\ \texttt{in}\ f)-$ and $C[\cdot] = (\texttt{let}\ f\ \texttt{be}\ \langle x, y\rangle\ \texttt{in}\ D[\cdot])$ ask the statements $\emptyset \vdash (\texttt{let}\ D[e]\ \texttt{be}\ \langle x, y\rangle\ \texttt{in}\ f) \preceq_B (\texttt{let}\ D[h]\ \texttt{be}\ \langle x, y\rangle\ \texttt{in}\ f) : B$ and $\emptyset \vdash (\texttt{let}\ f\ \texttt{be}\ \langle x, y\rangle\ \texttt{in}\ D[\cdot]) \preceq_B (\texttt{let}\ f\ \texttt{be}\ \langle x, y\rangle\ \texttt{in}\ D[h]) : B$ to be proved. We prove the first one since they are very similar and both make use of property $(\mathbf{c-5})$. Thus we should apply the induction hypothesis $\emptyset \vdash D[e] \preceq_{E\otimes E'} D[h] : E \otimes E'$ and the reflexivity of $\preceq_B$ on f, namely the relationship $x : E, y : E' \vdash f \preceq_B f : B$, exploiting the property $(\mathbf{c-5})$, to get the desired result.

$-C[\cdot] = \langle D[\cdot], v\rangle-$ and $C[\cdot] = \langle v, D[\cdot]\rangle$. Contexts classes of this type are under the domain of property $(\mathbf{c-6})$. Here the property to be shown is $\emptyset \vdash \langle V[e], v\rangle \preceq_{B\otimes E} \langle V[h], v\rangle : B \otimes E$ and the similar one $\emptyset \vdash \langle v, D[e]\rangle \preceq_{B\otimes E} \langle v, D[h]\rangle : B \otimes E$. As for the first statement, the induction hypothesis tells us that $\emptyset \vdash D[e] \preceq_B D[h] : B$, whence immediately thesis comes using property $(\mathbf{c-6})$ and reflexivity of $\preceq_E$. Similarly for the other case.

□

Theorem 2.3 (Soundness of (bi)similarity in ℓST_λ)**.** *In ℓST_λ, $\preceq$ is included in $\leq$, thus $\sim$ is included in $\equiv$.*

Proof. The statement of the theorem requires to prove the implication $\emptyset \vdash e \preceq_A h : A \Rightarrow \emptyset \vdash e \leq_A h : A$, but following the definition of context preorder, the thesis becomes $\forall C[\cdot] \in \texttt{CTX}_B\,(\emptyset \vdash A)\,,\ \mathbf{Obs}(C[e]) \leq \mathbf{Obs}(C[h])$.

By the previous Lemma 2.21, the hypothesis gives immediately the result $\forall C[\cdot] \in \mathtt{CTX}_B(\emptyset \vdash A)$, $\emptyset \vdash C[e] \preceq_B C[h] : B$, which by definition of simulation will be rewritten as

$$\begin{aligned}&\forall C \in \mathtt{CTX}_B(\Gamma \vdash A),\\ &((C[e],B), a_{eval}, (\widehat{v},B)) \in \mathcal{N} \Rightarrow \ ((C[h],B), a_{eval}, (\widehat{w},B)) \in \mathcal{N} \ \wedge \ v \preceq_B w. \end{aligned} \tag{2.69}$$

If the transition $((C[e],B), a_{eval}, (\widehat{v},B))$ doesn't occur, then $C[e]$ is divergent and $\mathbf{Obs}(C[e]) = 0$, otherwise $\mathbf{Obs}(C[e]) = 1$, but then the relation (2.69) ensures that the transition $((C[h],\ B), a_{eval},\ (\widehat{v},B))$ occurs and $\mathbf{Obs}(C[h]) = 1$, too. Therefore the hypothesis $\emptyset \vdash e \preceq_A h : A$ implies the relationship $\mathbf{Obs}(C[e]) \leq \mathbf{Obs}(C[h])$, which is the thesis. □

Theorem 2.4 (Completeness of (bi)similarity in ℓST_λ)**.** *In ℓST_λ, $\leq$ has the simulation property, thus in ℓST_λ $\equiv$ is included in $\sim$.*

Proof. Let us suppose we deal with closed terms so that we must show the truth of the statement

$$e \leq_{\emptyset,A} h \ \Rightarrow \ e \preceq_{\emptyset,A} h. \tag{2.70}$$

If otherwise, we can always reduce to this case by application of property (2.21b)

$$x : B \vdash e \leq h : A \ \Rightarrow \ \emptyset \vdash \lambda x.e \leq \lambda x.h : B \multimap A \tag{2.71}$$

Under the hypothesis of closed terms, let us show that the relation $\leq_A$ has the applicative simulation property, hence

$$v \leq_{\emptyset,\mathtt{bool}} w \Rightarrow \big(((\widehat{v},\mathtt{bool}), a_{\mathtt{b}}, (\widehat{v},\mathtt{bool})) \in \mathcal{N} \Rightarrow ((\widehat{w},\mathtt{bool}), a_{\mathtt{b}}, (\widehat{w},\mathtt{bool})) \in \mathcal{N}\big) \tag{2.72a}$$

$$\begin{aligned}\lambda x.e \leq_{\emptyset,B\multimap A} \lambda x.h \Rightarrow \Big(\forall u \in \mathcal{V}_B^{\ell ST_\lambda},\ \Big((\widehat{\lambda x.e}, B \multimap A), a_{@u}, (e\{u/x\}, A)\Big) \in \mathcal{N} \Rightarrow\\ \Big((\widehat{\lambda x.h}, B \multimap A), a_{@u}, (h\{u/x\}, A)\Big) \in \mathcal{N} \ \wedge \ e\{u/x\} \leq_{\emptyset,A} h\{u/x\}\Big)\end{aligned} \tag{2.72b}$$

$$\langle v_1, v_2\rangle \leq_{\emptyset, A\otimes B} \langle w_1, w_2\rangle \;\Rightarrow\; \forall g \in \mathcal{T}^{\ell ST_\lambda}_{x:A,y:B,E},$$
$$((\langle v_1, v_2, , \rangle A \otimes B), a_{\otimes g}, (g\{v_1/x, v_2/y\}, E)) \in \mathcal{N} \Rightarrow$$
$$\Rightarrow ((\langle w_1, w_2, , \rangle A \otimes B), a_{\otimes g}, (g\{w_1/x, w_2/y\}, E)) \in \mathcal{N}$$
$$\wedge\; g\{v_1/x, v_2/y\} \leq_{\emptyset, E} g\{w_1/x, w_2/y\} \quad (2.72c)$$

$$e \leq_{\emptyset, A} h \;\Rightarrow\; \Big(((e, A), a_{eval}, (\widehat{v}, A)) \in \mathcal{N} \Rightarrow ((h, A), a_{eval}, (\widehat{w}, A)) \in \mathcal{N} \wedge v \leq_{\emptyset, A} w\Big) \quad (2.72d)$$

If v and w are two boolean values in a preorder relation they have to be the same constant, otherwise the context $C[\cdot] = \texttt{if}\ [\cdot]\ \texttt{then tt else}\ \Omega$ could separate them, hence the relation (2.72a) follows immediately.

If $\lambda x.e \leq_{\emptyset, B\multimap A} \lambda x.h$, we must show the implication $\forall D[\cdot] \in \mathtt{CTX}_E(\emptyset \vdash A)$, $\forall u \in \mathcal{V}^{\ell ST_\lambda}_B$, $D[e\{u/x\}] \leq D[h\{u/x\}]$, starting by the premise $\forall C[\cdot] \in \mathtt{CTX}_E(\emptyset \vdash B \multimap A)$, $\mathbf{Obs}(C[\lambda x.e]) \leq \mathbf{Obs}(C[\lambda x.h])$. With this purpose let prepare the class of contexts $C_u[\cdot] = D[[\cdot]u]$.

Thus exploiting the definition of context preorder (2.19) one finds

$$\forall u,\, \forall D[\cdot] \in \mathtt{CTX}_E(\emptyset \vdash A),\, \mathbf{Obs}(D[(\lambda x.e)u]) \leq \mathbf{Obs}(D[(\lambda x.h)u]) \quad (2.73)$$

and this is enough to ensure that the transition $\Big(\widehat{(\lambda x.h}, B \multimap A), a_{@u}, (h\{u/x\}, A)\Big)$ is allowed every time that $\Big(\widehat{(\lambda x.e}, B \multimap A), a_{@u}, (e\{u/x\}, A)\Big)$ is.

Since $(\lambda x.e)u \equiv_A e\{u/x\}$ and $(\lambda x.hu) \equiv_A h\{u/x\}$, the condition (2.73) is equivalent to $\mathbf{Obs}(D[e\{u/x\}]) \leq \mathbf{Obs}(D[h\{u/x\}])$, the second statement of (2.72b), namely $e\{u/x\} \leq_{\emptyset, A} h\{u/x\}$, is also proved.

If $\langle v_1, v_2\rangle \leq_{\emptyset, A\otimes B} \langle w_1, w_2\rangle$ the thesis is $\forall g \in \mathcal{T}^{\ell ST_\lambda}_{x:A,y:B,E}$, $\forall D[\cdot] \in \mathtt{CTX}_F(\emptyset \vdash E)$, $D[g\{v_1/x, v_2/y\}] \leq D[g\{w_1/x, w_2/y\}]$, then we will preset the class of contexts

$$C_g[\cdot] = \texttt{let}\ [\cdot]\ \texttt{be}\ \langle x, y\rangle\ \texttt{in}\ D[g] \in \mathtt{CTX}_F(\emptyset \vdash E), \text{ with } g \in \mathcal{T}^{\ell ST_\lambda}_{x:A,y:B,E}.$$

Since the hypothesis tells us that $\forall C[\cdot] \in \mathtt{CTX}_F(\emptyset \vdash A \otimes B)$, $\mathbf{Obs}(C[\langle v_1, v_2\rangle]) \leq \mathbf{Obs}(C[\langle w_1, w_2\rangle])$, because of the structure of the contexts which have been chosen,

this entails that

$$\forall D[\cdot] \in \mathtt{CTX}_F\,(\emptyset \vdash E)\,\mathbf{Obs}(D[g\{v_1/x, v_2/y\}]) \leq \mathbf{Obs}(D[g\{w_1/x, w_2/y\}]),$$

which is the thesis.

If $e \leq_{\emptyset,A} h$ with $e, h \in \mathcal{T}_A^{\ell ST_\lambda} \setminus \mathcal{V}_A^{\ell ST_\lambda}$, then one must exploit Lemma 2.10, namely the relation $\Downarrow \subseteq \preceq$ which, under the hypothesis $e \Downarrow v$ and $h \Downarrow w$ ensures that $v \preceq_A e$ and $h \preceq_A w$. Thus since by Theorem 2.3 we know that $\preceq_A \subseteq \leq_A$, from hypothesis we get the chain of relations $v \leq_A e \leq_A h \leq_A w$, which brings back the proof towards one of the previous cases. □

Proposition 2.2. *In ℓST_λ bisimulation is fully–abstract with context equivalence, namely the two relations coincide.*

Proof. Is a direct consequence of Theorem 2.3 and Theorem 2.4. □

Chapter 3

Injecting Probabilistic Choice

The expressive power of ℓST_λ is rather limited, due to the presence of linearity. Nevertheless, the calculus is complete for first-order computations over the finite domain of boolean values, as discussed previously. Rather than relaxing linearity, we now modify ℓST_λ by endowing it with a form of probabilistic choice, thus obtaining a new linear λ-calculus, called ℓPST_λ, which is complete for probabilistic circuits. The transition toward the probabilistic language is formally performed by enriching ℓST_λ with a suitable choice operator denoted by $\oplus$. If $f_1, f_2 \in \mathcal{T}^A$ are terms of ℓST_λ, $f_1 \oplus f_2$ is a composite term that can behave either like f_1 or like f_2. When one component is selected the other is discarded, the choice being accomplished in a probabilistic way. The choice operators, being possibly nested, take into account the possibility to have many different evolutions paths during the calculus.

In a nondeterministic environment, a term $e = f_1 \oplus f_2$ obeys both the following reduction rules: $\dfrac{f_1 \Downarrow v_1}{f_1 \oplus f_2 \Downarrow v_1}$ and $\dfrac{f_2 \Downarrow v_2}{f_1 \oplus f_2 \Downarrow v_2}$, where it is understood that both values v_1 and v_2 are possible. Nevertheless here we adopt a *probabilistic* point of view, which is why every value must be supported by the probability which it has to appear as a result of the evaluation process. We see ℓPST_λ as an intermediate step towards ℓQST_λ, a quantum λ-calculus, where the structure of the language itself is intrinsically probabilistic, since the system follows the quantum mechanics rules.

The set of the possible terms of the language, equipped with the new operator

$\oplus$, is fully described by the following bnf form, which comes directly from (2.1):

$$v, u ::= x \mid \texttt{tt} \mid \texttt{ff} \mid \lambda x.e \mid \langle v, u\rangle$$

$$e, f, g ::= v \mid ef \mid \texttt{if } e \texttt{ then } f \texttt{ else } g \mid \langle v, u\rangle \mid \texttt{let } e \texttt{ be } \langle x, y\rangle \texttt{ in } f \mid e \oplus f \mid \Omega. \tag{3.1}$$

The set $\mathcal{Y}$ of types is the same as the one of ℓST_λ, with the new following typing rule $\dfrac{\Gamma \vdash e : A \qquad \Delta \vdash f : A}{\Gamma, \Delta \vdash e \oplus f : A}\,(tj - cho)$.

Since in a probabilistic framework we should suppose that at every step of reduction a single term evaluates to a distribution of terms, the evaluation operation is introduced as a relation $\Downarrow \subseteq \mathcal{T}^{\emptyset,A}_{\ell PST_\lambda} \times \mathscr{D}^{\ell PST_\lambda}_A$ between the sets of closed terms of type A belonging to ℓPST_λ and the one of distributions of values of type A in ℓPST_λ. The elements of $\mathscr{D}^{\ell PST_\lambda}_A$ are actually subdistributions whose support is some finite subset of the set of values $\mathcal{V}^{\ell PST_\lambda}_A$, i.e., for each such $\mathscr{E}$, we have $\mathscr{E} : \mathcal{V}^{\ell PST_\lambda}_A \to \mathbb{R}_{[0,1]}$ and $\sum_{v \in \mathcal{V}^{\ell PST_\lambda}_A} \mathscr{E}(v) \leq 1$. If $e \Downarrow \mathscr{E}$, each result of the evaluation of e comes with a probability, thus the notation $\mathscr{E} = \{v_i^{p_i}\}_{i \in \mathcal{I}}$ will often be used to denote the whole set of element of $\mathscr{E}$, each one with its probability.

Every subdistribution matches the condition $\sum_{i \in \mathcal{I}} p_i \leq 1$, where the sum is possibly lesser than 1 due to the presence of divergent paths of evaluation. For the set $\{v_i\}_{i \in \mathcal{I}}$, namely the support of $\mathscr{E}$, the symbol $\texttt{Sup}\,(\mathscr{E})$ is used. In Figure 3.1 the rules for big-step semantics in ℓPST_λ are given. If we take $\overline{\mathscr{D}^{\ell PST_\lambda}_A}$ as a symbol which denotes the space of subdistribution whose support is a subset of $\mathcal{T}^{\ell PST_\lambda}_A$, the *one–step reduction* ($\to$) and *small–step reduction* ($\Rightarrow$) operators in ℓPST_λ are binary relations $\to \subseteq \mathcal{T}^{\ell PST_\lambda}_A \times \overline{\mathscr{D}^{\ell PST_\lambda}_A}$ and $\Rightarrow \subseteq \mathcal{T}^{\ell PST_\lambda}_A \times \overline{\mathscr{D}^{\ell PST_\lambda}_A}$ which satisfy following general rules

$$\frac{}{v \Rightarrow \{v^1\}} \tag{3.2a}$$

$$\frac{e \to \left\{f_j^{q_j}\right\}_{j \in \mathcal{J}} \qquad f_j \Rightarrow \mathscr{G}_j}{e \Rightarrow \sum_{j \in \mathcal{J}} q_j \mathscr{G}_j}. \tag{3.2b}$$

Thoroughly, $\to$ is the smallest operator which fulfills the whole set of rules given in Table 3.2, while $\Rightarrow$ is the reflexive and transitive closure of $\to$. For this probabilistic

Big–step semantics rule	Name
$\dfrac{}{v \Downarrow \{v^1\}}$	$(v \Downarrow)_\wp$
$\dfrac{}{\Omega \Downarrow \emptyset}$	$(\Omega \Downarrow)_\wp$
$\dfrac{e \Downarrow \mathscr{E} \qquad f \Downarrow \mathscr{F} \qquad \left\{\ell\{u/x\} \Downarrow \mathscr{L}_{(\lambda\, x.\ell,u)}\right\}_{\lambda\, x.\ell \in \mathtt{Sup}(\mathscr{E}), u \in \mathtt{Sup}(\mathscr{F})}}{ef \Downarrow \sum_{\lambda\, x.\ell \in \mathtt{Sup}(\mathscr{E}), u \in \mathtt{Sup}(\mathscr{F})} \mathscr{E}(\lambda\, x.\ell)\mathscr{F}(u)\mathscr{L}_{(\lambda\, x.\ell,u)}}$	$(app \Downarrow)_\wp$
$\dfrac{e \Downarrow \mathscr{E} \qquad f \Downarrow \mathscr{F} \qquad g \Downarrow \mathscr{G}}{\left(\mathtt{if}\ e\ \mathtt{then}\ f\ \mathtt{else}\ g\right) \Downarrow \mathscr{E}(\mathtt{tt})\mathscr{F} + \mathscr{E}(\mathtt{ff})\mathscr{G}}$	$(if \Downarrow)_\wp$
$\dfrac{e \Downarrow \mathscr{E} \qquad f \Downarrow \mathscr{F}}{e \oplus f \Downarrow \frac{1}{2}\mathscr{E} + \frac{1}{2}\mathscr{F}}$	$(cho \Downarrow)_\wp$
$\dfrac{e \Downarrow \mathscr{E} \qquad \{f\{v_i/x, u_i/y\} \Downarrow \mathscr{F}_i\}_{\langle v_i,u_i\rangle \in \mathtt{Sup}(\mathscr{E})}}{\left(\mathtt{let}\ e\ \mathtt{be}\ \langle x, y\rangle\ \mathtt{in}\ f\right) \Downarrow \sum_{\langle v_i,u_i\rangle \in \mathtt{Sup}(\mathscr{E})} \mathscr{E}(\langle v_i, u_i\rangle)\mathscr{F}_i}$	$(let \Downarrow)_\wp$

Figure 3.1: Big–step semantics of ℓPST_λ.

language, a set of small–step operational semantics rules [38] may be provided, similarly to what has been done for ℓST_λ (see Figure 2.2). This set of rules leads a single term in a *sequence*, an element of $\overline{\mathscr{D}_A^{\ell PST_\lambda}}$ where every term occurs with the same probability: one–step operational semantics rules for ℓPST_λ are listed in Figure 3.2. More generally, in ℓPST_λ, the one–step reduction operator leads subdistribution of terms in subdistribution of terms following the rule

$$\frac{e_m \in \mathtt{Sup}\left(\mathscr{E}\right) \qquad e_m \to \{f_j^{q_j}\}_{j=1\ldots J}}{\mathscr{E} \to \mathscr{E} \setminus \{e_m^{p_m}\} \cup \{f_j^{q_j \cdot p_m}\}}\,. \tag{3.3}$$

Moreover, in ℓPST_λ, big–step reduction relation between terms and distribution of values, with the operational semantics given in Figure 3.1, enjoys the property highlighted by the following lemma.

Lemma 3.1 (Uniqueness of semantics)**.** *For each term $e \in \mathcal{T}_A^{\ell PST_\lambda}$, there is a unique distribution $\mathscr{E}$ such that $e \Downarrow \mathscr{E}$ and, $\forall v \in \mathtt{Sup}\left(\mathscr{E}\right)$, $\mid v \mid \leq \mid e \mid$*

One–step semantics rule	Name
$\dfrac{}{(\lambda\, x.e)v \to \{e\{v/x\}^1\}}$	$(app_\beta)_\wp$
$\dfrac{e \to \{f_n^{1/N}\}_{n=1\ldots N}}{eh \to \{f_n h^{1/N}\}_{n=1\ldots N}}$	$(app_L)_\wp$
$\dfrac{e \to \{f_n^{1/N}\}_{n=1\ldots N}}{ve \to \{v f_n^{1/N}\}_{n=1\ldots N}}$	$(app_R)_\wp$
$\dfrac{}{\texttt{if tt then } h \texttt{ else } \ell \to h}$	$(if - ax_{\texttt{tt}})_\wp$
$\dfrac{}{\texttt{if ff then } h \texttt{ else } \ell \to \ell}$	$(if - ax_{\texttt{ff}})_\wp$
$\dfrac{e \to \{f_n^{1/N}\}_{n=1\ldots N}}{\texttt{if } e \texttt{ then } h \texttt{ else } \ell \to \{(\texttt{if } f_n \texttt{ then } h \texttt{ else } \ell)^{1/N}\}_{n=1\ldots N}}$	$(if)_\wp$
$\dfrac{}{\texttt{let } \langle v,u\rangle \texttt{ be } \langle x,y\rangle \texttt{ in } e \to \{e\{v/x,u/y\}^1\}}$	$(let - ax)_\wp$
$\dfrac{e \to \{f_n^{1/N}\}_{n=1\ldots N}}{\texttt{let } e \texttt{ be } \langle x,y\rangle \texttt{ in } h \to \{(\texttt{let } f_n \texttt{ be } \langle x,y\rangle \texttt{ in } h)^{1/N}\}_{n=1\ldots N}}$	$(let)_\wp$
$\dfrac{}{e \oplus f \to \{e^{1/2}, f^{1/2}\}}$	$(cho - ax)_\wp$
$\dfrac{}{\Omega \to \emptyset}$	$(div)_\wp$

Figure 3.2: One-step reduction semantics rules of ℓPST_λ. Rules are given in a call–by–value leftmost reduction framework.

Proof. By structural induction of the generic term $e \in \mathcal{T}^{\ell PST_\lambda}$, examining evaluation rules.

$-e = v-$ If $e = v$, with $v \in \mathcal{V}^{\ell PST_\lambda}$ there is nothing to prove, since by the evaluation rule for values recalling the general rule (3.2a) one finds $\mathscr{E} = \{v^1\}$ which is the subdistribution whose support is a set with a unique value. Besides the condition

on the size is fulfilled being $\mid e \mid = \mid v \mid$

$-e = f_1 f_2-$ Using the induction hypothesis we find that there exist unique $\mathscr{F}_1$, $\mathscr{F}_2$ such that $f_1 \Downarrow \mathscr{F}_1$, $f_2 \Downarrow \mathscr{F}_2$ and $\forall \lambda\, x.\ell \in \mathtt{Sup}\,(\mathscr{F}_1)$, $\mid \lambda\, x.\ell \mid \leq \mid f_1 \mid$ as well as $\forall u \in \mathtt{Sup}\,(\mathscr{F}_2)$, $\mid u \mid \leq \mid f_2 \mid$. Thus, by the definition of size given in Table 2.4 and by Lemma 2.2, whereas the language is linear it holds the relation $\forall \lambda\, v.\ell \in \sup \mathscr{F}_1$, $\forall u \in \sup \mathscr{F}_2$, $\mid \ell\{u/x\} \mid < \mid \lambda\, x.\ell u \mid \leq \mid f_1 f_2 \mid$. Therefore we can use the inductive hypothesis also on $\ell\{f/x\}$ and applying $(app_{\Downarrow})_{\wp}$ we get

$$\frac{f_1 \Downarrow \mathscr{F}_1 \qquad f_2 \Downarrow \mathscr{F}_2 \qquad \left\{\ell\{u/x\} \Downarrow \mathscr{G}_{(\lambda\, x.\ell,u)}\right\}_{\lambda\, x.\ell \in \mathtt{Sup}(\mathscr{F}_1), u \in \mathtt{Sup}(\mathscr{F}_2)}}{f_1 f_2 \Downarrow \underbrace{\sum_{\lambda\, x.\ell, u} \mathscr{F}_1(\lambda\, x.\ell)\mathscr{F}_2(u)\mathscr{G}_{(\lambda\, x.\ell,u)}}_{\mathscr{E}}}, \tag{3.4}$$

thus the distribution to which e evaluates, is indeed solely determined by the formula $\mathscr{E} = \sum_{\lambda\, x.\ell \in \sup \mathscr{F}_1, u \in \sup \mathscr{F}_2} \mathscr{F}_1(\lambda\, x.\ell)\mathscr{F}_2(u)\mathscr{G}_{(\lambda\, x.\ell,u)}$.

$-e = (\mathtt{if}\ f_1\ \mathtt{then}\ f_2\ \mathtt{else}\ f_3)-$ The induction hypothesis applied on the subterms $\{f_j\}_{j \in \mathcal{J}}$, allows to state that three distributions $\{\mathscr{F}_j\}_{j=1,2,3}$ exist unequivocally such that $\{f_j \Downarrow \mathscr{F}_j\}_{j=1,2,3}$ and $\{\forall u_j \in \sup \mathscr{F}_j, \mid u_j \mid \leq \mid f_j \mid\}_{j=1,2,3}$. Therefore using inductive hypothesis on the premises of the semantic rule $(if_{\Downarrow})_{\wp}$ we get

$$\frac{f_1 \Downarrow \mathscr{F}_1 \qquad f_2 \Downarrow \mathscr{F}_2 \qquad f_3 \Downarrow \mathscr{F}_3}{\mathtt{if}\ f_1\ \mathtt{then}\ f_2\ \mathtt{else}\ f_3 \Downarrow \underbrace{\mathscr{F}_1(\mathtt{tt})\mathscr{F}_2 + \mathscr{F}_1(\mathtt{ff})\mathscr{F}_3}_{\mathscr{E}}}. \tag{3.5}$$

which gives us as distribution $\mathscr{E} = \mathscr{F}_1(\mathtt{tt})\mathscr{F}_2 + \mathscr{F}_1(\mathtt{ff})\mathscr{F}_3$, determined by the values of $\mathscr{F}_2$ and $\mathscr{F}_3$

$-e = (\mathtt{let}\ f_1\ \mathtt{be}\ \langle x, y\rangle\ \mathtt{in}\ f_2)-$ If we invoke as usual the inductive hypothesis we get that $f_1 \Downarrow \mathscr{F}_1$, with $\mathscr{F}_1$ unequivocally determined, whose values fulfill the condition $\mid \langle u, \nu\rangle \mid \leq \mid f_1 \mid \forall \langle u, \nu\rangle \in \sup \mathscr{F}_1$. Thus, by Lemma 2.2 we obtain the condition

$$\forall \langle u, \nu\rangle \in \sup \mathscr{F}_1,$$
$$\mid f_2\{u/x, \nu/y\} \mid < \mid \mathtt{let}\ \langle u, \nu\rangle\ \mathtt{be}\ \langle x, y\rangle\ \mathtt{in}\ f_2 \mid \leq \mid \mathtt{let}\ f_1\ \mathtt{be}\ \langle x, y\rangle\ \mathtt{in}\ f_2 \mid,$$

which allow to apply the inductive hypothesis to the premises of $(let_{\Downarrow})_{\wp}$ writing

$$\frac{f_1 \Downarrow \mathscr{F}_1 \qquad \left\{f_2\{u/x, \nu/y\} \Downarrow \mathscr{G}_{\langle u,\nu\rangle}\right\}_{\langle u,\nu\rangle \in \mathtt{Sup}(\mathscr{F}_1)}}{\left(\mathtt{let}\ f_1\ \mathtt{be}\ \langle x, y\rangle\ \mathtt{in}\ f_2\right) \Downarrow \underbrace{\sum_{\langle u,\nu\rangle \in \mathtt{Sup}(\mathscr{F}_1)} \mathscr{F}_1(\langle u, \nu\rangle) \cdot \mathscr{G}_{\langle u,\nu\rangle}}_{\mathscr{E}}}, \tag{3.6}$$

whence the distribution $\mathscr{E}$ such that $e \Downarrow \mathscr{E}$ is unequivocally determined.

$-e = f_1 \oplus f_2-$ Let write down the rule for choice operator $(cho \Downarrow)_{\wp}$:

$$\frac{f_1 \Downarrow \mathscr{F}_1 \qquad f_2 \Downarrow \mathscr{F}_2}{f_1 \oplus f_2 \Downarrow \frac{1}{2}(\mathscr{F}_1 + \mathscr{F}_2)}, \tag{3.7}$$

remarking that the existence and uniqueness of $\mathscr{F}_1$ and $\mathscr{F}_2$ such that $\forall u_1 \in \sup \mathscr{F}_1$, $\mid u_1 \mid \leq \mid f_1 \mid$ and $\forall u_2 \in \sup \mathscr{F}_2$, $\mid u_2 \mid \leq \mid f_2 \mid$, are determined by induction hypothesis on the subterms f_1, and f_2. It follows that $e = f_1 \oplus f_2 \ \Rightarrow e \Downarrow \frac{1}{2}(\mathscr{F}_1 + \mathscr{F}_2)$.

$-e = \langle f_1, f_2 \rangle-$ Here the distribution $\mathscr{E}$ is univocally determined by induction hypothesis, being by induction $f_1 \Downarrow \mathscr{F}_1$, $f_2 \Downarrow \mathscr{F}_2$ and $\mathscr{E} = \{\langle u_j, \nu_j \rangle^{q_j}\}_{j \in \mathcal{J}}$, where $u_j \in \mathtt{Sup}(\mathscr{F}_1)$, $\nu_j \in \mathtt{Sup}(\mathscr{F}_2)$. □

If $\Gamma \vdash e \Downarrow \mathscr{E} : A$, then the unique $\mathscr{E}$ from Lemma 3.1 is called the *semantics* of term e and is denoted simply as $[\![e]\!]$.

3.1 Probabilistic Context Preorder

Context equivalence is defined very similarly to ℓST_λ, the only difference being the underlying notion of observation, which in ℓST_λ takes the form of *convergence*, and in ℓPST_λ becomes the *probability* of convergence.

The set of possible linear contexts in ℓPST_λ is indeed obtained by the bnf form (2.17b) by simply adding the term $C[\cdot] \oplus D[\cdot]$, being therefore

$$V[\cdot] ::= [\vdash_v \cdot] \mid \lambda x.C[\cdot] \mid \langle V[\cdot], u \rangle \mid \langle u, V[\cdot] \rangle \tag{3.8}$$

$$C[\cdot] ::= [\vdash_e \cdot] \mid V[\cdot] \mid \mathtt{if}\ C[\cdot]\ \mathtt{then}\ f\ \mathtt{else}\ g \mid \mathtt{if}\ f\ \mathtt{then}\ C[\cdot]\ \mathtt{else}\ D[\cdot] \mid$$
$$fC[\cdot] \mid C[\cdot]f \mid \mathtt{let}\ f\ \mathtt{be}\ \langle x, y \rangle\ \mathtt{in}\ C[\cdot] \mid \mathtt{let}\ C[\cdot]\ \mathtt{be}\ \langle x, y \rangle\ \mathtt{in}\ f \mid C[\cdot] \oplus D[\cdot]. \tag{3.9}$$

Nevertheless, to properly give the context preorder in a linear probabilistic environment, requires to adapt the already given definition of the function **Obs** to the new probabilistic environment. Therefore here the function $\mathbf{Obs} : \mathcal{T}_{\Gamma,A}^{\ell PST_\lambda} \to \mathbb{R}$ is

defined as $\mathbf{Obs}(e) = \sum [\![e]\!]$ [1]. The definitions of contextual preorder and contextual equivalence are left unchanged with respect to deterministic ℓST_λ (2.19, 2.20), with the exception that the class of possible contexts can be built with the syntactic tree given in (3.9). We have

$$e \leq_{\Gamma,A} h \;\Leftrightarrow \forall C[\cdot] \in \mathtt{CTX}_B\left(\Gamma \vdash A\right), \mathbf{Obs}(C[e]) \leq \mathbf{Obs}(C[h]) \tag{3.10a}$$

$$e \equiv_{\Gamma,A} h \;\Leftrightarrow \forall C[\cdot] \in \mathtt{CTX}_B\left(\Gamma \vdash A\right), \mathbf{Obs}(C[e]) = \mathbf{Obs}(C[h]). \tag{3.10b}$$

It is easy to show that the probabilistic context relation is a preorder as a mere consequence of reflexivity and transitivity of $\leq$.

We shall denote by $\mathtt{CTX}_A\left(\Delta \vdash B\right)$ the collection of all possible (not necessarily ground) context such that $\emptyset \vdash C[\Delta \vdash B] : A$.

Lemma 3.2 (Probabilistic context preorder and context equivalence basic property)**.** *Probabilistic context preorder is a precongruence over $\mathcal{T}^{\ell PST_\lambda}_{\Gamma,A}$, and probabilistic context equivalence a congruence likewise.*

Proof. It is analogous to that of Lemma 2.7: only we add to the other cases the proof for property $(\mathbf{c}-\mathbf{7})$: $(e_1 \leq_{\Gamma_1,A} h_1 \wedge \;\; e_2 \leq_{\Gamma_2,A} h_2) \Rightarrow e_1 \oplus e_2 \leq_{\Gamma_1\Gamma_2,A} h_1 \oplus h_2$. For this operator we set the contexts as $C[\cdot] = C'[\lambda\{x_i\}_{i\in\mathcal{I}}.[\vdash_A] \oplus e_2] \in \mathtt{CTX}_A\left(\Gamma_1 \vdash A\right)$ and $D[\cdot] = C'[\lambda\{x_i\}_{i\in\mathcal{I}}.h_1 \oplus [\vdash_A]] \in \mathtt{CTX}_A\left(\Gamma_2 \vdash A\right)$ where where $C'[\cdot]$ is a generic context, $\{x_i\}_{i\in\mathcal{I}}$ stands for $dom(\Gamma_1) \cup dom(\Gamma_2)$. Being $C[h_1] = D[e_2]$ we get the chain $e_1 \oplus e_2 \leq_{\Gamma_1\Gamma_2,A} e_1 \oplus h_2 \leq_{\Gamma_1\Gamma_2,A} h_1 \oplus h_2$, which gives thesis by transitivity of $\leq_{\Gamma_1\Gamma_2,A}$. □

Lemma 3.3 (Probabilistic context preorder and probabilistic context equivalence behaviour with respect to contexts)**.** *Probabilistic context preorder and context equivalence are compatible with respect to whatever context application to terms, therefore*

$$\forall e, h, \forall C[\cdot] \in \mathtt{CTX}_B\left(\Gamma \vdash A\right), e \leq_{\Gamma,A} h \;\Rightarrow\; C[e] \leq_{\emptyset,B} C[h]. \tag{3.11}$$

[1] $\sum [\![e]\!]$, is a shorter form which stands for $\sum_{v\in \mathtt{Sup}\left([\![e]\!]\right)} [\![e]\!](v)$.

Proof. The proof is alike that one which has been given in deterministic case with Lemma 2.8. Hypothesis entails as a consequence $\forall D[\cdot] \in \mathtt{CTX}_B\,(\Gamma \vdash A)$, $\mathbf{Obs}(D[e]) \leq \mathbf{Obs}(D[h])$, while thesis requires that

$$\forall G[\cdot] \in \mathtt{CTX}_E\,(\emptyset \vdash B)\,,\, \mathbf{Obs}(G[C[e]]) \leq \mathbf{Obs}(G[C[h]]),$$

thus simply let choose $D[\cdot] = G[C[\cdot]] \in \mathtt{CTX}_E\,(\Gamma \vdash A)$.

□

3.1.1 Probabilistic Simulation

Would it be possible to define applicative bisimilarity for ℓPST_λ similarly to what we have done for ℓST_λ? The first obstacle towards this goal is the dynamics of ℓPST_λ, which is not deterministic but rather probabilistic, and thus cannot fit into an LTS, which traditionally describes a deterministic behaviour.

In the literature, however, various notions of probabilistic bisimulation have been introduced, and it turns out that the earliest and simplest one, due to Larsen and Skou [39], is sufficient for our purposes.

A *labelled Markov chain* (LMC in the following) is a triple $(\mathscr{S}, \mathcal{L}, \mathcal{P})$, where $\mathscr{S}$ and $\mathcal{L}$ denote a set of states and of labelled action respectively, as in the definition of a LTS, while $\mathcal{P}$ is a *transition probability matrix*, i.e., a function from $\mathscr{S} \times \mathcal{L} \times \mathscr{S}$ to $\mathbb{R}_{[0,1]}$. The set of labels for our state system is the same as the ℓST_λ LTS and it has already been discussed in Figure 2.7. Besides, to unburden formulas, here we adopt this notation: $\mathcal{P}(s, \ell, X)$, when $X \subseteq \mathscr{S}$, stands for $\sum_{t \in X} \mathcal{P}(s, \ell, t)$.

Since in a probabilistic environment $s \in \mathscr{S}$, when undergoing an action labelled ℓ will evolve with a certain probability to t, $\mathcal{P}\,(s, \ell, t)$ just expresses the probability of occurrence of this event. For every s and for every ℓ, $\mathcal{P}(s, \ell, \mathscr{S})$ respects the constraint to be equal or lesser than 1: as usual values strictly less than one correspond to the possibility of divergent systems.

Given such a LMC $\mathscr{M}$, a preorder R on $\mathscr{S}$ is said to be a *simulation* iff for every subset X of $\mathscr{S}$, it holds that

$$\mathcal{P}(s, \ell, X) \leq \mathcal{P}(t, \ell,\, R\,(X)) \tag{3.12}$$

where $R(X)$ is a subset of $\mathscr{S}$ defined be the following condition:

$$R(X) = \{s \in \mathscr{S} \mid \exists t \in X, t\, R\, s\}. \tag{3.13}$$

An equivalence relation R on $\mathscr{S}$ is said to be a *bisimulation* on $\mathscr{M}$ iff whenever $(s,t) \in R$, it holds that

$$\mathcal{P}(s, \ell, \mathtt{E}) = \mathcal{P}(t, \ell, \mathtt{E}) \tag{3.14}$$

for every equivalence class $\mathtt{E}$ of $\mathscr{S}$ modulo R.

Since the states of LMC are no more than the terms of the language, it should be remarked that the way that the environment can interact with them strongly depends on their type, thus becomes crucial the necessity to exhibit it. This is the reason that in the elements of the probability transition function, the type appears every time with both values and terms.

Implementing a labelled Markov chain (LMC), denoted by $\mathscr{M}_{\ell PST_\lambda}$, on the probabilistic language requires to choose the tern $(\mathscr{S}, \mathcal{L}, \mathcal{P})$ as shown just below

$$\mathscr{S} = \overline{\mathcal{T}^{\ell PST_\lambda}} \uplus \overline{\mathcal{V}^{\ell PST_\lambda}}, \tag{3.15a}$$

$$\mathcal{L} = \{a_{eval}, a_{\mathtt{tt}}, a_{\mathtt{ff}}, a_{@u}, a_{\otimes h}, a_{\mathcal{Y}_A}, a_{\widehat{\mathcal{Y}_A}}\}, \tag{3.15b}$$

$$\mathcal{P} = \mathcal{P}_{\ell PST_\lambda}. \tag{3.15c}$$

Let us recall that $\overline{\mathcal{T}^{\ell ST_\lambda}}$ is a set of pairs $\cup_{A \in \mathcal{Y}}(\mathcal{T}_A^{\ell ST_\lambda} \times \{A\})$, and similarly for $\overline{\mathcal{V}^{\ell ST_\lambda}}$. The notation, used in $\mathscr{L}_{\ell ST_\lambda}$, to distinguish the couple (v, A) where v appears as a term from the couple $(\widehat{v}, A)$ where v plays role of a value has been conserved identically. Beside, the function $\mathcal{P}_{\ell PST_\lambda}$ assumes the following values:

$$\mathcal{P}_{\ell PST_\lambda}\left((\widehat{\lambda\, x.e}, A \multimap B), a_{@v}, (e\{v/x\}, B)\right) = 1;$$

$$\mathcal{P}_{\ell PST_\lambda}\left((e, A), a_{eval}, (\widehat{v}, A)\right) = [\![e]\!](v);$$

$$\mathcal{P}_{\ell PST_\lambda}\left((\widehat{\langle v, u\rangle}, A \otimes B), a_{\otimes e}, (e\{v/x, u/y\}, E)\right) = 1;$$

$$\mathcal{P}_{\ell PST_\lambda}\left((\widehat{\mathtt{tt}}, \mathtt{bool}), a_{\mathtt{tt}}, (\widehat{\mathtt{tt}}, \mathtt{bool})\right) = 1; \qquad \mathcal{P}_{\ell PST_\lambda}\left((\widehat{\mathtt{ff}}, \mathtt{bool}), a_{\mathtt{ff}}, (\widehat{\mathtt{ff}}, \mathtt{bool})\right) = 1;$$

$$\mathcal{P}_{\ell PST_\lambda}\left((e, A), a_{\mathcal{Y}_A}, (e, A)\right) = 1; \qquad \mathcal{P}_{\ell PST_\lambda}\left((\widehat{v}, A), a_{\widehat{\mathcal{Y}_A}}, (\widehat{v}, A)\right) = 1,$$

and it has value 0 in all the other cases. It is easy to realize that $\mathcal{P}_{\ell PST_\lambda}$ can indeed be seen as the natural generalization of $\mathcal{N}_{\ell ST_\lambda}$: on states in the form $(\widehat{v}, A)$, the function either returns 0 or 1, while in correspondence to states like (e, A) and the label *eval*, it behaves in a genuinely probabilistic way. Probabilistic (bi)simulation, despite the endeavor required to define it, preserves all fundamental properties of its deterministic sibling.

The definition of (bi) simulation as a relation indexed on types is given considering the proper elements of transition matrix $\mathcal{P}_{\ell PST_\lambda}$, depending on whether the terms involved in the relation are values or they aren't

- For boolean values the only possible transition is a check on the value itself, therefore if the preorder $\mathcal{S}_{\mathtt{bool}}$ is a simulation over the set of boolean values:

$$\begin{aligned}&\emptyset \vdash e\,\mathcal{S}_{\mathtt{bool}}\,h : \mathtt{bool} \;\Rightarrow \forall \mathtt{b} \in \mathcal{V}_{\mathtt{bool}}^{\ell PST_\lambda}\\ &\mathcal{P}_{\ell PST_\lambda}\left((\widehat{e}, \mathtt{bool}), a_{\mathtt{b}}, (\widehat{\mathtt{b}}, \mathtt{bool})\right) \leq \mathcal{P}_{\ell PST_\lambda}\left((\widehat{h}, \mathtt{bool}), a_{\mathtt{b}}, (\widehat{\mathtt{b}}, \mathtt{bool})\right). \quad (3.16)\end{aligned}$$

- For function values, the usual definition of applicative simulation is traced out, thus if the preorder $\mathcal{S}_{B\multimap A}$ is a simulation on $\mathcal{V}_{B\multimap A}^{\ell PST_\lambda}$ then

$$\begin{aligned}&\emptyset \vdash \lambda x.e\,\mathcal{S}_{B\multimap A}\,\lambda x.h : B \multimap A \;\Rightarrow\; \forall v \in \mathcal{V}_B^{\ell PST_\lambda}, \mathcal{P}_{\ell PST_\lambda}\left((\widehat{\lambda x.e}, B \multimap A),\right.\\ &\left. a_{@v}, (e\{v/x\}, A)\right) \leq \mathcal{P}_{\ell PST_\lambda}\left((\widehat{\lambda x.h}, B \multimap A), a_{@v}, (\,\mathcal{S}_A\,(e\{v/x\}), A)\right).\end{aligned} \quad (3.17)$$

- For pairs, the definition relies on the proper transition matrix elements, being

$$\begin{aligned}&\emptyset \vdash \langle v_1, v_2\rangle\,\mathcal{S}_{A\otimes B}\,\langle u_1, u_2\rangle : A \otimes B \Rightarrow \forall g \in \mathcal{T}^{\ell ST_\lambda}_{x:A,y:B,E},\\ &\qquad \mathcal{P}_{\ell PST_\lambda}\left((\widehat{\langle v_1, v_2\rangle}, A \otimes B), a_{\otimes g}, (g\{v_1/x, v_2/y\}, E)\right) \leq\\ &\qquad \mathcal{P}_{\ell PST_\lambda}\left((\widehat{\langle u_1, u_2\rangle}, A \otimes B), a_{\otimes g}, (\,\mathcal{S}_E\,(g\{v_1/x, v_2/y\}), E)\right). \quad (3.18)\end{aligned}$$

- For terms the simulation relation is determined as a probability to evaluate to a set of values, hence its probabilistic nature is recovered:

$$\emptyset \vdash e\,\mathcal{S}_A\,h : A \;\Rightarrow\; \forall X \in \mathcal{V}^A,$$
$$\mathcal{P}_{\ell PST_\lambda}\left((e,A), a_{eval}, (X,A)\right) \leq \mathcal{P}_{\ell PST_\lambda}\left((h,A), a_{eval}, (\mathcal{S}_A\,(X), A)\right). \quad (3.19)$$

Most of the (bi)simulation properties are shared also by its probabilistic extension.

Lemma 3.4. *Every probabilistic bisimulation is also a probabilistic simulation.*

Proof. If $\mathcal{B}_A$ is a probabilistic bisimulation and $(e,h) \in \mathcal{B}_A$, then the property

$$\forall \ell, \forall\, \mathtt{E} \subseteq \mathcal{V}^A/\,\mathcal{B}_A\,,\; \mathcal{P}_{\ell PST_\lambda}\left((e,A), \ell, (\mathtt{E},A)\right) = \mathcal{P}_{\ell PST_\lambda}\left((h,A), \ell, (\mathtt{E},A)\right) \quad (3.20)$$

holds, with $\mathcal{V}^A/\,\mathcal{B}_A$ quotient set of $\mathcal{V}^A$ modulo $\mathcal{B}_A$. To show that $\mathcal{B}_A$ has the simulation property, the relation

$$\forall X \subseteq \mathcal{V}^A,\; \mathcal{P}_{\ell PST_\lambda}\left((e,A), \ell, (X,A)\right) \leq \mathcal{P}_{\ell PST_\lambda}\left((h,A), \ell, (\mathcal{B}_A\,(X), A)\right) \quad (3.21)$$

has to be proved. Let $\{\mathtt{E}_j\}_{j\in\mathcal{J}}$ be the set of equivalence classes generated by $\mathcal{B}_A$ on $\mathcal{V}^A$. If in relationship (3.20) we set $X = \mathtt{E}_j$ for some $j \in \mathcal{J}$ then the property comes immediately, being a consequence of the inclusion $=\subseteq\leq$, since $\mathcal{S}_A\,(\mathtt{E}_n) = \mathtt{E}_n$.

Otherwise let write the subset X in the form $X = \cup_{i\in\mathcal{I}} X_i$, where $X_i = X \cap \mathtt{E}_i$ and $\mathcal{I} \subseteq \mathcal{J}$; so that

$$\forall X \subseteq \mathcal{V}^A,\; \mathcal{P}_{\ell PST_\lambda}((e,A), \ell, (\cup_{i\in\mathcal{I}}(X_i), A)) = \sum_{i\in\mathcal{I}} \mathcal{P}_{\ell PST_\lambda}((h,A), \ell, (X_i, A)) \leq$$
$$\leq \sum_{i\in\mathcal{I}} \mathcal{P}_{\ell PST_\lambda}((h,A), \ell, (\mathtt{E}_i, A)) = \mathcal{P}_{\ell PST_\lambda}((h,A), \ell, (\cup_{i\in\mathcal{I}}\mathtt{E}_i, A)), \quad (3.22)$$

and the property is proved since $\mathcal{S}_A\,(X) = \mathcal{S}_A\,(\cup_{i\in\mathcal{I}}(X_i)) = \cup_{i\in\mathcal{I}}(\mathtt{E}_i)$. $\mathcal{S}^{\,op}$ is also a probabilistic simulation as a consequence of symmetric property of R and the fact, just proved, that R is a probabilistic simulation. □

Lemma 3.5. *A symmetric relation which is a probabilistic simulation is a probabilistic bisimulation.*

Proof. It has to be shown that if a relation $\widehat{\mathcal{S}_A}$ is a simulation and it enjoys the property $\forall e, h \in \mathcal{T}_A^{\ell PST_\lambda}\ (e,h) \in \widehat{\mathcal{S}_A} \Leftrightarrow (h,e) \in \widehat{\mathcal{S}_A}$, then it holds $\emptyset \vdash e\,\mathcal{B}_A\,h : A$ with $\widehat{\mathcal{S}_A} = \mathcal{B}_A$ bisimulation.

If $e, h \in \mathcal{V}_A^{\ell PST_\lambda}$, then we set $e = v$ and $h = w$ rewriting hypothesis as

$$\mathcal{P}_{\ell PST_\lambda}((\widehat{v}, A), \ell, (\widehat{u}, A)) \leq \mathcal{P}_{\ell PST_\lambda}((\widehat{w}, A), \ell, (\widehat{\nu}, A)) \ \wedge$$
$$\mathcal{P}_{\ell PST_\lambda}((\widehat{w}, A), \ell, (\widehat{\nu}, A)) \leq \mathcal{P}_{\ell PST_\lambda}((\widehat{v}, A), \ell, (\widehat{u}, A)) \quad (3.23)$$

where the label $\ell \in \mathcal{L}$ depends on the type A. Since in all these cases the relation is anyway deterministic, from (3.23) it follows immediately the equality

$$\mathcal{P}_{\ell PST_\lambda}((\widehat{v}, A), \ell, (\widehat{u}, A)) = \mathcal{P}_{\ell PST_\lambda}((\widehat{w}, A), \ell, (\widehat{\nu}, A)),$$

which proofs the thesis.

If $e, h \in \mathcal{T}_A^{\ell PST_\lambda} \setminus \mathcal{V}_A^{\ell PST_\lambda}$, then let us remark that the relation $\widehat{\mathcal{S}_A}$ being a symmetric preorder is an equivalence relation. If $\mathtt{E} \in \{\mathtt{E}_m\}_{m \in \mathcal{M}}$ is a generic equivalence class belonging to the quotient set $\mathcal{V}_A^{\ell PST_\lambda}/\widehat{\mathcal{S}_A}$ we may rewrite hypothesis as

$$\forall \mathtt{E}, \mathtt{F} \subseteq \mathcal{V}_A^{\ell ST_\lambda},\ \mathcal{P}_{\ell PST_\lambda}((e, A), a_{eval}, (\mathtt{E}, A)) \leq \mathcal{P}_{\ell PST_\lambda}\left((h, A), a_{eval}, (\widehat{\mathcal{S}_A}(\mathtt{E}), A)\right) \ \wedge$$
$$\mathcal{P}_{\ell PST_\lambda}((h, A), a_{eval}, (\mathtt{F}, A)) \leq \mathcal{P}_{\ell PST_\lambda}\left((e, A), a_{eval}, (\widehat{\mathcal{S}_A}(\mathtt{F}), A)\right). \quad (3.24)$$

Now let us recall that, by definition

$$\widehat{\mathcal{S}_A}(\mathtt{E}) = \{v \in \mathcal{V}_A^{\ell PST_\lambda} \mid \exists u \in \mathtt{E}, u\,\widehat{\mathcal{S}_A}\,v\} = \mathtt{E},$$

since v, u both belong to the same equivalence class; similarly $\widehat{\mathcal{S}_A}(\mathtt{F}) = \mathtt{F}$.

Using this result and setting, in (3.24), $\mathtt{E} = \mathtt{F}$ one finds immediately

$$\mathcal{P}_{\ell PST_\lambda}((e, A), a_{eval}, (\mathtt{E}, A)) = \mathcal{P}_{\ell PST_\lambda}((h, A), a_{eval}, (\mathtt{E}, A)),$$

which completes the proof. □

3.2 From Applicative Simulation towards Applicative Bisimilarity

Following the deterministic procedure, it should be desirable that starting from the definition of probabilistic simulation and probabilistic bisimulation, one could upgrade to the more general concepts of similarity and bisimilarity, simply taking the union of all possible simulation and bisimulation respectively.

Nevertheless the way to carry out this process in the probabilistic pattern is more complex due to the slightly different definition of simulation and bisimulation which is given in this scheme, where transitivity property is embedded in the definition itself so that it is not possible assume that the union of all possible simulations is necessarily a simulation itself, and analogously for bisimulation, as we will see just below.

Indeed, a simulation was defined as a preorder relation which enjoys the general property (3.12) and a bisimulation as an equivalence relation which enjoys the property (3.14). Hereafter, in a probabilistic environment, a relation which has the property (3.12) but not necessarily is a preorder will be referred as a *pseudo–simulation*; analogously we will call *pseudo–bisimulation* a relation which has the property (3.14) but it is not necessarily an equivalence relation.

Hence we use the symbol ${}^{[pse]}\mathtt{Sim}$ to denote the set whose elements are the probabilistic pseudo–simulation, namely the relations among the elements of the set of states $\mathscr{S}$ which have the property (3.12); similarly with ${}^{[pse]}\mathtt{BiS}$ we will denote the set of all possible probabilistic pseudo–bisimulations.

The following lemma will show that the sets defined above are closed by composition, therefore that taking two or more element of ${}^{[pse]}\mathtt{Sim}$ (${}^{[pse]}\mathtt{BiS}$ respectively) and composing them one obtains an element of ${}^{[pse]}\mathtt{Sim}$ (${}^{[pse]}\mathtt{BiS}$) in turn. It finds its analogous in Lemma 2.11, valid in the deterministic framework.

Lemma 3.6 (Pseudo-(bi)simulation set is closed under composition.)**.** $\mathcal{S}^{(1)} \in \mathtt{Sim}$ *and* $\mathcal{S}^{(2)} \in \mathtt{Sim} \Rightarrow \mathcal{S}^{(1)} \circ \mathcal{S}^{(2)} \in {}^{[pse]}\,\mathtt{Sim}$ *(and analogously for* ${}^{[pse]}\mathtt{BiS}$*).*

Proof. Exploiting the definition of composition between relation we write the hypothesis as $s\,\mathcal{S}^{(1)}t$ and $t\,\mathcal{S}^{(2)}r$, namely

$$\forall X, Y \subseteq \mathscr{S},\ \mathcal{P}(s,\ell,X) \leq \mathcal{P}\left(t,\ell,\,\mathcal{S}^{(1)}(X)\right)\ \wedge\ \mathcal{P}(t,\ell,Y) \leq \mathcal{P}\left(Y,\ell,\,\mathcal{S}^{(2)}(Y)\right), \tag{3.25}$$

recalling that, by definition $\mathcal{S}^{(i)}(X) = \{s \in \mathscr{S} \ \mid\ \exists t \in X,\, t\,\mathcal{S}^{(i)}s\}_{i=1,2}$ and likewise for $\{\mathcal{S}^{(i)}(Y)\}_{i=1,2}$.

For any X, let set $Y = \mathcal{S}^{(1)}(X)$, thus (3.25) becomes

$$\forall X \subseteq \mathscr{S},\ \mathcal{P}(s,\ell,X) \leq \mathcal{P}\left(t,\ell,\,\mathcal{S}^{(1)}(X)\right)\ \wedge \\ \mathcal{P}\left(t,\ell,\,\mathcal{S}^{(1)}(X)\right) \leq \mathcal{P}\left(r,\ell,\,\mathcal{S}^{(2)}\left(\mathcal{S}^{(1)}(X)\right)\right) \tag{3.26}$$

whence

$$\forall X \subseteq \mathscr{S},\ \mathcal{P}(s,\ell,X) \leq \mathcal{P}\left(r,\ell,\,\mathcal{S}^{(2)}\left(\mathcal{S}^{(1)}(X)\right)\right). \tag{3.27}$$

We are left to rewrite in a simpler way the set $\mathcal{S}^{(2)}\left(\mathcal{S}^{(1)}(X)\right)$. Using the definition we get

$$\mathcal{S}^{(2)}\left(\mathcal{S}^{(1)}(X)\right) = \left\{r \in \mathscr{S} \ \mid\ \exists s \in X\ \wedge\ t \in \mathcal{S}^{(1)}(X),\, s\,\mathcal{S}^{(1)}t\ \wedge\ t\,\mathcal{S}^{(2)}r\right\}, \tag{3.28}$$

namely $\mathcal{S}^{(2)}\left(\mathcal{S}^{(1)}(X)\right) \equiv \left(\mathcal{S}^{(1)} \circ \mathcal{S}^{(2)}\right)(X)$, which is properly the condition stating that $\mathcal{S}^{(1)} \circ \mathcal{S}^{(2)} \in^{[pse]} \texttt{Sim}$. □

The sets $^{[pse]}\texttt{Sim}$ and $^{[pse]}\texttt{BiS}$ seem to be the better candidates to describe the collection of all probabilistic simulation (and bisimulation respectively) although the transitivity of their elements is not ensured (but reflexivity is!). Transitivity is, indeed, a characteristic required in the definition itself of both probabilistic similarity as well as probabilistic bisimilarity.

In order to overcome this hurdle it is necessary to introduce the concept of transitive closure of a set: given a relation R – let choose it as a relation on a subset of $\mathscr{S}$ – its transitive closure R^{+} is the relation inductively defined from R by the following two rules

$$\frac{s\,R\,t}{s\,R^{+}t} \qquad (tc-1)$$

$$\frac{s\,R^{+}t \qquad t\,R^{+}r}{s\,R^{+}r} \qquad (tc-2)$$

where $s,t,r \in \mathscr{S}$; thus R^{+} is a preorder induced by R on the set $\mathscr{S}$. The transitive closure preserves fundamental properties of relation above all compatibility and closure under substitution, as the following lemmas state.

Lemma 3.7 (On the compatibility)**.** *If R is compatible then so is R^{+}.*

Proof. By induction on the structure of the terms involved in the relation, examinig the rules $(tc-1)$ and $(tc-2)$.

Given $e,h \in \mathcal{T}_A^{\ell PST_\lambda}$ which are supposed to be built with some constructor of the language by finite set of subterms $\{f_i\}_{i\in\mathcal{I}}$ and $\{\ell_i\}_{i\in\mathcal{I}}$ such that $e = \mathtt{cnstr}\,(\{f_n\}_{n\in\mathcal{N}})$ and $h = \mathtt{cnstr}\,(\{\ell_n\}_{n\in\mathcal{N}})$, the statement requires to prove that

$$\forall n \in \mathcal{N},\ \Delta_n \vdash f_n\,R^{+}\ell_n : B_n \Rightarrow \Gamma \vdash e\,R^{+}h : A. \qquad (3.29)$$

- Let suppose that for every n the set of relations appearing in (3.29) hypothesis, namely $\Delta_n \vdash f_n\,R^{+}\ell_n : B_n$ all are a consequence of the application of $(tc-1)$, then $\forall n$ the condition $\Delta_n \vdash f_n\,R\,\ell_n : B_n$ is matched. Since by hypothesis R is compatible with the rule of the language, the previous set of relations entails that $\emptyset \vdash e\,R\,h : A$. Thus applying $(tc-1)$ we get the thesis (3.29).

- Let now suppose that $\forall n \neq j$ the relations (3.29) all have $(tc-1)$ as last rule, except for a unique subterm f_j such that the condition $\Delta_j \vdash f_j\,R^{+}\ell_j : B_j$ is a consequence of $(tc-2)$. Therefore it must exist a certain g_j such that

$$\frac{\Delta_j \vdash f_j\,R^{+}g_j : B_j \qquad \Delta_j \vdash g_j\,R^{+}\ell_j : B_j}{\Delta_j \vdash f_j\,R^{+}\ell_j : B_j}\,(tc-2), \qquad (3.30)$$

Now the induction hypothesis, entailing that R^{+} is compatible *on smaller terms* since R is, can be used. Denoting by g the term built with the operator `cnstr` with subterms $f_1,\ldots g_j,\ldots f_N$ and using $(tc-1)$ for pairs belonging to the set $\mathcal{N}\setminus\{j\}$ one gets

$$\forall n \neq j, \Delta_n \vdash f_n\,R^{+}\ell_n : B_n \ \wedge\ \Delta_j \vdash g_j\,R^{+}\ell_j : B_j \Rightarrow \Gamma \vdash g\,R^{+}h : A. \qquad (3.31)$$

Moreover recalling that R is compatible, and thus reflexive – see Lemma 2.13 – we can write also by inductive hypothesis, which ensures that R^+ is compatible on smaller terms R as it is, that

$$\forall n \neq j, \Delta_n \vdash f_n \, R^+ f_n : B_n \ \wedge \ \Delta_j \vdash f_j \, R^+ g_j : B_j \Rightarrow \Gamma \vdash e \, R^+ g : A. \tag{3.32}$$

Now, applying to the conclusion of (3.31) and (3.32) the rule $(tc-2)$ we find

$$\frac{\Gamma \vdash e \, R^+ g : A \qquad \Gamma \vdash g \, R^+ h : A}{\Gamma \vdash e \, R^+ h : A} \, (tc-2). \tag{3.33}$$

- We will apply the same arguments to the cases where there are two (or more) pairs of subterms $(f_{j_1}, g_{j_1}), (f_{j_2}, g_{j_2}) \ldots\ldots$ whose relation is consequence of the application of $(tc-2)$ as last rule.
- Finally let us consider the case where all the terms have as the last rule $(tc-2)$, whence the set of relations

$$\forall n \in \mathcal{N}, \, \frac{\Delta_n \vdash f_n \, R^+ g_n : B_n \qquad \Delta_n \vdash g_n \, R^+ g_n : B_n}{\Delta_n \vdash f_n \, R^+ \ell_n : B_n} \, (tc-2) \tag{3.34}$$

whence, by induction hypothesis on the premises of (3.34), we easily get

$$\Gamma \vdash e \, R^+ g : A \ \wedge \ \Gamma \vdash g \, R^+ h : A, \tag{3.35}$$

where, of course, $g = \mathtt{cnstr}\,(\{g_n\}_{n\in\mathcal{N}})$. We find the thesis applying $(tc-2)$ with premises (3.35). □

Lemma 3.8 (If a relation is closed under value substitution, so is its transitive closure)**.** *If R is closed under substitution then so it is R^+, namely*

$$\begin{aligned} &\Big(\Gamma, x : B \vdash e \, R \, h : A \ \wedge \ v \in \mathcal{V}_B^{\ell PST_\lambda} \ \Rightarrow \ \Gamma \vdash e\{v/x\} \, R \, h\{v/x\} : A\Big) \Rightarrow \\ &\Big(\Gamma, x : B \vdash e \, R^+ h : A \ \wedge \ v \in \mathcal{V}_B^{\ell PST_\lambda} \ \Rightarrow \ \Gamma \vdash e\{v/x\} \, R^+ h\{v/x\} : A\Big) \end{aligned} \tag{3.36}$$

Proof. The proof is by induction on the derivation of the relation $\Gamma, x : B \vdash e \, R^+ h : A$. □

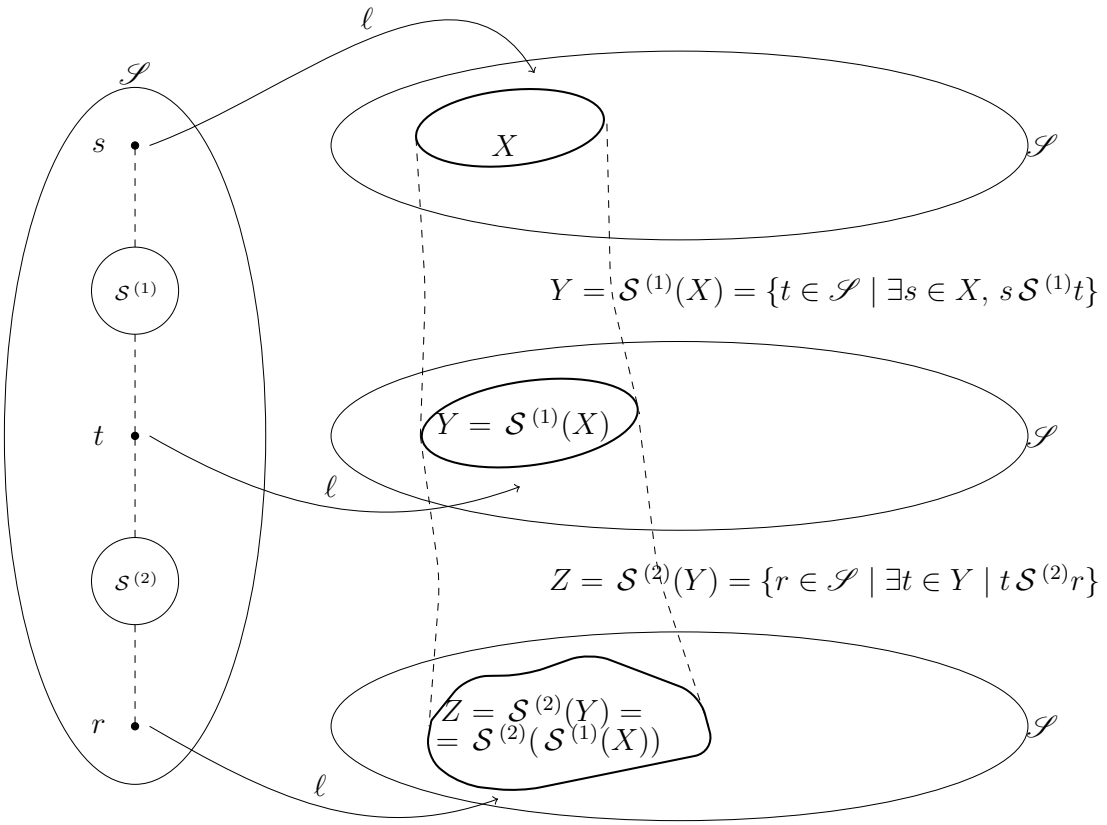

Figure 3.3: Graphical idea of two probabilistic simulations composition: in a system whose states belong to a set denoted by $\mathscr{S}$, a state $s \in \mathscr{S}$ may evolve to a whichever set of states $X \subseteq \mathscr{S}$. If $s\ \mathcal{S}^{(1)}\ t$, being $\mathcal{S}^{(1)}$ a simulation relation, then the evolution of t goes on towards a set $Y = \mathcal{S}^{(1)}(X)$. As a useful remark, let notice that since $\mathcal{S}^{(1)}$ is a simulation, it is reflexive and then $\forall X,\ X \subseteq \mathcal{S}^{(1)}(X)$. If t is, in turn related to r by mean of a different simulation $\mathcal{S}^{(2)}$, then the latter will evolve towards a set of the states $Z = \mathcal{S}^{(2)}(Y) = \mathcal{S}^{(2)}\big(\mathcal{S}^{(1)}(X)\big)$.

The following step in our path consists in enlarging the above mentioned sets of pseudo–simulations and pseudo–bisimulations in order to obtain new relations that, enjoying transitivity, are good candidates for a definition of probabilistic similarity and bisimilarity.These relations, formerly denoted as ${}^{[pse]}\texttt{Sim}$ and ${}^{[pse]}\texttt{BiS}$, have already been proved to be closed under composition: now we are going to show that

they are closed with respect to a generic union of elements.

Namely, if $R_A^{(n)}$ and $R_A^{(m)}$ both belong to ${}^{[pse]}\mathtt{Sim}$ also their union belongs to it, and the same holds for ${}^{[pse]}\mathtt{BiS}$ likewise: the elements of ${}^{[pse]}\mathtt{Sim}$ and ${}^{[pse]}\mathtt{BiS}$ indeed, are not required to be transitive relations, the main problem to extend this topics to transitive relations being that a whatever union of transitive relation is not generally a transitive relation, while the union of relations preserves the properties of reflexivity and simmetry.

As worthwhile remark, let observe that every possible relation written as $\bigcup_{n\in\mathcal{N}} \mathcal{B}_A{}^{(n)}$, with $\mathcal{B}_A{}^{(n)} \in {}^{[pse]}\ \mathtt{BiS}\ \forall n$ is reflexive and symmetric, whereas every pseudo–bisimulation is a reflexive and symmetric relation and a whatever union of reflexive and symmetric relations is in turn reflexive and symmetric. Hence $\bigcup_{n\in\mathcal{N}} \mathcal{B}_A{}^{(n)}$ has good right to belong to ${}^{[pse]}\mathtt{BiS}$.

For what has been discussed until now, the relations defined by the symmetric and transitive closures of a union of every element of ${}^{[pse]}\mathtt{Sim}$ and ${}^{[pse]}\mathtt{BiS}$ seem to be good candidates to obtain a good definition of similarity and bisimilarity.

Lemma 3.9 (Transitive union of a collection of probabilistic pseudo simulations and pseudo bisimulations)**.** *The transitive closure*

$$\mathcal{S}_A{}^{+} = \left\{ \bigcup_i R^{(i)} \;\middle|\; R^{(i)} \in {}^{[pse]}\ \mathtt{Sim} \right\}^{+}, \text{ and}$$

$$\mathcal{B}_A{}^{+} = \left\{ \bigcup_i R^{(i)} \;\middle|\; R^{(i)} \in {}^{[pse]}\ \mathtt{BiS} \right\}^{+}$$

of the union of every possible relations which belong to Sim *and* BiS *are simulation and bisimulation respectively.*

Proof. Being $\mathcal{S}_A{}^{+}$ transitive by definition, and reflexive as union of reflexive relations, it is necessarily a preorder: thus we should only show that it has the pseudo–simulation property, namely $\Gamma \vdash e\, \mathcal{S}_A{}^{+} f : A \Rightarrow$

$$\forall \ell,\, \forall X \subseteq \mathcal{T}_A^{\ell PST_\lambda},\, \mathcal{P}_{\ell PST_\lambda}\left((e,A),\ell,(X,A)\right) \leq \mathcal{P}_{\ell PST_\lambda}\left((f,A),\ell,(\mathcal{S}_A{}^{+}(X),A)\right). \tag{3.37}$$

If $(e, f) \in \mathcal{S}_A{}^{+}$, then since the relation is a union of simulation, there are $N \geq 1$ elements of $\mathtt{Sim}$ and $N-1$ intermediate terms such that $e\,\mathcal{S}_A{}^{(1)} g_1 \wedge \; g_1\,\mathcal{S}_A{}^{(2)} g_2 \wedge \cdots \wedge \; g_{N-1}\,\mathcal{S}_A{}^{(N)} f$.

Hence, the same relation between e and f can be rewritten, by definition of composition, as $e\left(\mathcal{S}_A{}^{(1)} \circ \mathcal{S}_A{}^{(2)} \circ \cdots \circ \mathcal{S}_A{}^{(N)}\right) f$. Now, using the previous Lemma 3.6, hence the property of closure under composition we find immediately

$$\left(\mathcal{S}_A{}^{(1)} \circ \mathcal{S}_A{}^{(2)} \circ \cdots \circ \mathcal{S}_A{}^{(N)}\right) \in {}^{[pse]}\,\mathtt{Sim}.$$

Analogously the relation $\mathcal{B}_A{}^{+}$, as transitive closure of the union of all possible pseudo–bisimulations, is by definition transitive but it is also reflexive and symmetric, since an arbitrary union of elements of ${}^{[pse]}\mathcal{B}_A$ which are reflexive and symmetric relations, is a reflexive and symmetric relation in turn: thus $\mathcal{B}_A{}^{+}$ is an equivalence relation on the terms space $\mathcal{T}_A^{\ell PST_\lambda}$.

Using the same procedure we evince that $\forall e, h \in \mathcal{T}_A^{\ell PST_\lambda}$ such that $\emptyset \vdash e\,\mathcal{B}_A{}^{+} h : A$, there are M elements of ${}^{[pse]}\mathtt{BiS}$, such that $e\,\mathcal{B}_A{}^{(1)} g_1 \wedge g_1\,\mathcal{B}_A{}^{(2)} g_2 \wedge \cdots \wedge g_{M-1}\,\mathcal{B}_A{}^{(M)}$, therefore one gets the result $(e, h) \in \mathcal{B}_A{}^{(1)} \circ \cdots \circ \mathcal{B}_A{}^{(M)}$, where the relation $\mathcal{B}_A{}^{(1)} \circ \mathcal{B}_A{}^{(M)}$ is an element of $\mathtt{BiS}$ by Lemma 3.6. □

Thus, let us resume the important results that have been found with Lemma 3.9 and Lemma 3.6 in the following:

Proposition 3.1. *The transitive closure $\mathcal{S}_A{}^{+}$ of the union of every possible probabilistic* pseudo*–simulation is a probabilistic simulation and the transitive closure $\mathcal{B}_A{}^{+}$ of the union of a every possible probabilistic* pseudo*–bisimulation is a probabilistic bisimulation.*

Probabilistic similarity and probabilistic bisimilarity have similar definitions.

As for probabilistic similarity one sets it as the union

$$\preceq_A = \left\{ \bigcup_i R^{(i)} \mid R^{(i)} \in \mathtt{Sim} \right\}$$

of *all* possible simulations and probabilistic bisimilarity likewise is

$$\sim_A = \left\{ \bigcup_i R^{(i)} \;\middle|\; R^{(i)} \in \mathtt{BiS} \right\},$$

understanding the meaning of the sets $\mathtt{Sim} = \{ R_A \mid R_A$ is a *probabilistic* simulation on$\mathcal{T}_A^{\ell PST_\lambda}\}$ and $\mathtt{BiS} = \{ R_A \mid R_A$ is a *probabilistic* bisimulation on$\mathcal{T}_A^{\ell PST_\lambda}\}$.

However, as already pointed out, a generic union of preorders is not necessarily a preorder, and the same holds for a generic union of equivalence relations which is not perforce an equivalence relation, since the transitivity is not saved when the union is taken.

This entails that about $\preceq_A$ and $\sim_A$, symbols which denote similarity and bisimilarity as in deterministic ℓST_λ, as a matter of fact we currently can't say yet whether $\preceq_A$ is a preorder and then a simulation itself and $\sim_A$ an equivalence relation and thus a bisimulation. This is proved by the lemma below.

Lemma 3.10 (Probabilistic similarity and bisimilarity)**.** *$\sim_A$ is an equivalence relation over $\mathcal{T}_A^{\ell PST_\lambda}$ and, likewise, $\preceq_A$ is a preorder over $\mathcal{T}_A^{\ell PST_\lambda}$.*

Proof. Indeed $\sim_A \subseteq \mathcal{B}_A{}^+$ since the second relation is the transitive closure of the first, and $\sim_A \supseteq \mathcal{B}_A{}^+$ since the second one is a bisimulation itself and, by definition $\sim_A$ contains all possible bisimulations.

Thus $\sim\ A = \mathcal{B}_A{}^+$ and $\sim$ inherits all the properties of $\mathcal{B}_A{}^+$, then it is an equivalence relation. Similarly $\preceq_A$ is a preorder on $\mathcal{T}_A^{\ell PST_\lambda}$. □

Lemma 3.11. *Probabilistic similarity $\preceq$ and co-similarity $\preceq^{op}$ satisfy to the relation $\sim = \preceq \cap \preceq^{op}$.*

Proof. The statement can be proved showing both the inclusions $\sim \subseteq (\preceq \cap \preceq^{op})$ and $(\preceq \cap \preceq^{op}) \subseteq \sim$.

- $\sim \subseteq (\preceq \cap \preceq^{op})$: for previous Lemma 3.4, which holds for every simulation, therefore for similarity too, we have $\sim \subseteq \preceq$ and $\sim \subseteq \preceq^{op}$, whence $\sim \subseteq (\preceq \cap \preceq^{op})$ comes immediately.

• $(\preceq \cap \preceq^{op}) \subseteq \sim$: the relation $\preceq \cap \preceq^{op}$ is necessarily an equivalence relation, being the symmetric intersection of two relations which are preorders by definition.

Let $\mathtt{E}$ be an element of its quotient set: since the intersection of two similarity is a similarity in turn, the following condition must hold, if $\emptyset \vdash e(\preceq_A \cap \preceq_A^{op})h : A$

$$\forall \mathtt{E} \subseteq \mathcal{V}^A, \mathcal{P}_{\ell PST_\lambda}((e, A), \ell, (\mathtt{E}, A)) \leq \mathcal{P}_{\ell PST_\lambda}((h, A), \ell, ((\preceq_A \cap \preceq_A^{op})(\mathtt{E}), A))$$

where, by definition $(\preceq_A \cap \preceq_A^{op})(\mathtt{E}) = \{h \subseteq \mathcal{T}^{\ell PST_\lambda} \mid \exists e \in \mathtt{E}, \emptyset \vdash (e \preceq_A h \wedge e \preceq_{A)}^{op} h : A\}$. Since both $\preceq_A$ and $\preceq_A^{op}$ are reflexive, $\mathtt{E} \subseteq (\preceq_A \cap \preceq_A^{op})(\mathtt{E})$; then let us define $\mathtt{E}' = (\preceq_A \cap \preceq_A^{op})(\mathtt{E}) \setminus \mathtt{E}$.

Following the definition above, an element $f \in \mathtt{E}'$ is such that $\exists e \in \mathtt{E}$, $e \preceq_A f \wedge e \preceq_A^{op} f \wedge f \notin \mathtt{E}$. Nevertheless, since as already remarked $(\preceq_A \cap \preceq_A^{op})$ is an equivalence relation, the first two conditions entail that $f \in \mathtt{E}$, indeed it is in the same equivalence class of e and the the third condition leads to a contradiction, so that necessarily $\mathtt{E}' \equiv \emptyset$ and $\mathtt{E} = (\preceq_A \cap \preceq_A^{op})(\mathtt{E})$, proving then the condition:

$$\forall \mathtt{E} \subseteq \mathcal{V}^A, \mathcal{P}_{\ell PST_\lambda}((e, A), \ell, (\mathtt{E}, A)) = \mathcal{P}_{\ell PST_\lambda}((h, A), \ell, (\mathtt{E}, A)),$$

which is the thesis. □

3.3 Probabilistic Applicative Similarity is a Precongruence

With respect to ℓST_λ, the simulation and bisimulation relations, and their largest analogous, namely similarity and bisimilarity, can be given by just instantiating the general scheme described above to the specific LMC modeling terms of ℓPST_λ and their dynamics, which has been done in definitions (3.16) – (3.19).

All these turn out to be relations on *closed* terms, but as for ℓST_λ, they can be turned into proper typed relations just by the usual extension to open terms (2.31).

The question now is: are the just introduced coinductive methodologies sound with respect to context equivalence? And is it that the proof of precongruence for similarity from Section 2.5 can be applied here? The answer is positive, but some

effort is needed [37, 12] . Above all, we are supposed to enhance the applicative similarity relation with a set of Howe's rules, which are identical to those already given for deterministic language (Figure 2.8) to which we must add a new rule for the constructor $\oplus$ which is written down just below

$$\cdot\frac{\Gamma \vdash e\, R^H h : A \qquad \Delta \vdash f\, R^H \ell : A \qquad \Gamma, \Delta \vdash h \oplus \ell\, R\, b : A}{\Gamma, \Delta \vdash e \oplus f\, R^H b : A}$$

The proofs of the properties of Howe's relation such as

- compatibility of $\preceq^H$ (Lemma 2.16)
- $\preceq \subseteq \preceq^H$ (Lemma 2.17)
- substitutivity of $\preceq^H$ (Lemma 2.19)
- pseudo–transitivity of $\preceq^H$ (Lemma 2.18)

hold identically in probabilistic and deterministic scheme (as well as in quantum one). Nevertheless, the probabilistic nature of this systems makes it harder to prove the key lemma, namely the simulation property of $\preceq^H$.

Indeed we have a double hindrance given both from the definition of probabilistic (bi)similarity which requires to extend through the symmetric and transitive closure of a relation all properties already proved and, above all, from the greater difficulty that the proof of key lemma entails in a probabilistic system.

In particular we are required to prove that the transitive closure of Howe's lifting of a general relation R, enjoys all the properties of compatibility, substitutivity and closure under substitution that R^H itself has. This will be exploited obviously with similarity.

Anyway we start by facing the problem to show that Howe's lifting of the probabilistic similarity relation is itself a probabilistic simulation.

Lemma 3.12. *Probabilistic key lemma: Howe's extension of probabilistic similarity has the probabilistic simulation relation property.*
Therefore Howe's extension of probabilistic similarity is included in similarity itself

which is, by definition, the greatest simulation. As a corollary of ${\preceq_A}^H \subseteq \preceq_A$*, we find the analogous result* ${\sim_A}^H \subseteq \sim_A$.

Proof. As it has been done in deterministic environment (relationships 3.16 –3.19), according to the definition of probabilistic simulation we split the proof distinguishing between values and terms according to the following statement:

$$\emptyset \vdash e {\preceq_{\mathtt{bool}}}^H h : \mathtt{bool} \;\Rightarrow\; \forall \mathtt{b} \in \mathcal{V}_{\mathtt{bool}}^{\ell PST_\lambda},$$
$$\mathcal{P}_{\ell PST_\lambda}\left((\widehat{e}, \mathtt{bool}), a_{\mathtt{b}}, (\widehat{\mathtt{b}}, \mathtt{bool})\right) \leq \mathcal{P}_{\ell PST_\lambda}\left((\widehat{h}, \mathtt{bool}), a_{\mathtt{b}}, ({\preceq_{\mathtt{bool}}}^H(\widehat{\mathtt{b}}), \mathtt{bool})\right) \tag{3.38a}$$

$$\emptyset \vdash \lambda x.f {\preceq_{B\multimap A}}^H \lambda x.\ell : B \multimap A \;\Rightarrow\; \forall v \in \mathcal{V}_B^{\ell PST_\lambda}, \forall X \in \mathcal{V}_A^{\ell PST_\lambda},$$
$$\mathcal{P}_{\ell PST_\lambda}\left((\widehat{\lambda x.f}, B \multimap A), a_{@v}, (X, A)\right) \leq$$
$$\leq \mathcal{P}_{\ell PST_\lambda}\left((\widehat{\lambda x.\ell}, B \multimap A), a_{@v}, ({\preceq_{B\multimap A}}^H(X), A)\right) \tag{3.38b}$$

$$\emptyset \vdash \langle v_1, v_2\rangle {\preceq_{A\otimes B}}^H \langle w_1, w_2\rangle : A \otimes B \;\Rightarrow \forall g \in \mathcal{T}_{x:A,y:B,E}^{\ell PST_\lambda}$$
$$\mathcal{P}_{\ell PST_\lambda}\left((\widehat{\langle v_1, v_2\rangle}, A \otimes B), a_{\otimes g}, (g\{v_1/x, v_2/y\}, E)\right) \leq$$
$$\leq \mathcal{P}_{\ell PST_\lambda}\left((\widehat{\langle w_1, w_2\rangle}, A \otimes B), a_{\otimes g}, \preceq_E (g\{v_1/x, v_2/y\}, E)\right) \tag{3.38c}$$

$$\left(\emptyset \vdash e {\preceq_A}^H h : A \;\wedge\; e \Downarrow \mathscr{E}\right) \;\Rightarrow\; \left(h \Downarrow \mathscr{H} \;\wedge\; \forall X \in \mathcal{V}_A^{\ell PST_\lambda},\right.$$
$$\left.\mathcal{P}_{\ell PST_\lambda}((e, A), a_{eval}, (X, A)) \leq \mathcal{P}_{\ell PST_\lambda}\left((h, A), a_{eval}, ({\preceq_A}^H(X), A)\right)\right) \tag{3.38d}$$

We have to prove the lemma for values and for terms, according to the different definition of similarity.

♢ If $\emptyset \vdash e {\preceq_{\mathtt{bool}}}^H h : \mathtt{bool}$ are boolean values we must prove the statement (3.38a). Since the relation $\emptyset \vdash e {\preceq_{\mathtt{bool}}}^H h : \mathtt{bool}$ must be a consequence of (How_{1v}), which has, as a unique premise $\emptyset \vdash e \preceq_{\mathtt{bool}} h : \mathtt{bool}$, we find the thesis as a result of the definition (3.16). Indeed, if $e \neq \mathtt{b}$ the left–hand side of

(3.38a) is zero and the inequality is obviously true; otherwise $e = \mathtt{b}$ and from $\emptyset \vdash e \preceq_{\mathtt{bool}} h : \mathtt{bool}$ it follows $h \in \preceq_{\mathtt{bool}} (\mathtt{b}) \subseteq \preceq_{\mathtt{bool}}{}^{H}(\mathtt{b})$. In this case both sides of (3.38a) are equal to one.

$\diamondsuit$ If the value is a λ–abstraction $e = \lambda\, x.f$, then we should prove the property (3.38b), originating from definition (3.17).

The hypothesis $\emptyset \vdash \lambda\, x.f \preceq_{B \multimap A}{}^{H} \lambda\, x.\ell : B \multimap A$, is an immediate consequence of Howe's rule for lambda abstractions (How_2)

$$\frac{x : B \vdash f \preceq_{A}{}^{H} g : A \qquad \emptyset \vdash \lambda\, x.g \preceq_{B \multimap A} \lambda\, x.\ell : B \multimap A}{\emptyset \vdash \lambda\, x.f \preceq_{B \multimap A}{}^{H} \lambda\, x.\ell : B \multimap A}\,, \tag{3.39}$$

Since Howe's relation is compatible, from the first premise of (3.39), it follows the relation $\lambda\, x.f \preceq_{B \multimap A}{}^{H} \lambda\, x.g$. Moreover the second premise of (3.39) entails, by definition of probabilistic similarity (3.17), the relation $\forall v \in \mathcal{V}_{B}^{\ell PST_{\lambda}}$, $\ell\{v/x\} \in \preceq_{A} (g\{v/x\})$ or, equivalently,

$$\lambda\, x.\ell \in \preceq_{B \multimap A} (\lambda\, x.g). \tag{3.40}$$

Now let us apply the induction hypothesis on the smaller terms in the premises of (3.39), entailing that

$$\lambda\, x.f \preceq_{B \multimap A}{}^{H} \lambda\, x.g \;\Rightarrow\; \lambda\, x.g \in \preceq_{B \multimap A}{}^{H} (\lambda\, x.f), \tag{3.41}$$

as it is shown in Figure 3.4. Joining the results (3.40) and (3.41) we get that $\lambda\, x.\ell \in \preceq_{B \multimap A} \left(\preceq_{B \multimap A}{}^{H} (\lambda\, x.f)\right)$. Notice that the result given in Lemma 2.17, implies that $\preceq_{B \multimap A} \subseteq \preceq_{B \multimap A}{}^{H}$, thus necessarily $\preceq_{B \multimap A} \left(\preceq_{B \multimap A}{}^{H} (\lambda\, x.f)\right) = \preceq_{B \multimap A}{}^{H} (\lambda\, x.f)$. Then we conclude that

$$\lambda\, x.\ell \in \preceq_{B \multimap A}{}^{H} (\lambda\, x.f)\,, \quad \text{hence} \quad \forall v \in \mathcal{V}_{B}^{\ell PST_{\lambda}} \ell\{v/x\} \preceq_{A}{}^{H} (f\{v/x\}) \tag{3.42}$$

and this result can be seen also a consequence of the pseudo–transitivity property of probabilistic Howe's lifting (Lemma 2.18).

Thus for any generic $X \subseteq \mathcal{V}_{A}^{\ell PST_{\lambda}}$, if $f\{v/x\} \notin X$ the inequality (3.38b) necessarily holds because the left–hand side of (3.38b) is equal to zero. Otherwise $f\{v/x\} \in X$ and by previous arguments $\lambda\, x.\ell \in \preceq_{B \multimap A}{}^{H} (\lambda\, x.f)$, whence we get $f\{v/x\} \in \preceq_{A}{}^{H} (\ell\{v/x\})$, and both sides of (3.38b) are equal to one.

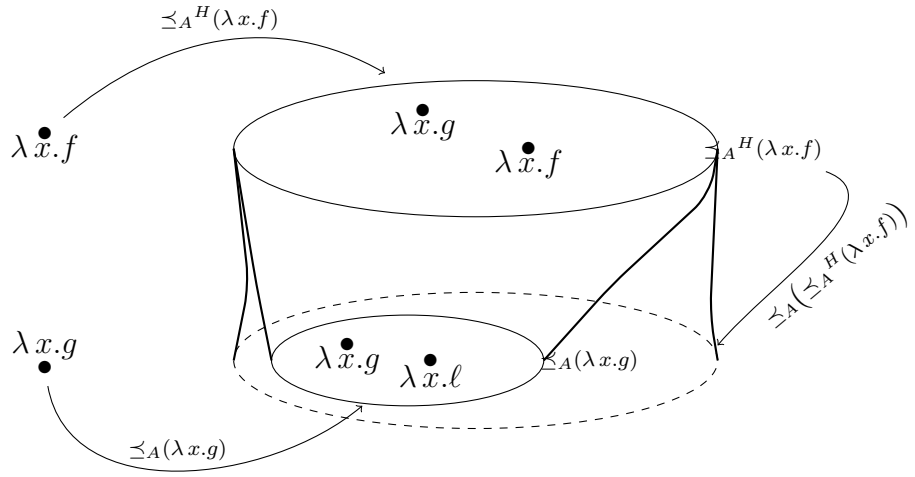

Figure 3.4: Graphical representation of the terms $\lambda x.f$, $\lambda x.g$ and $\lambda x.\ell$ involved in Howe's relation and of theirs "evolutes" under the relations $\preceq_A{}^H$ and $\preceq_A$ respectively, namely the sets $\preceq_A{}^H(\lambda x.f)$ and $\preceq_A\ (\lambda x.g)$. The cone $\preceq_A\ (\lambda x.f)$ contains $\lambda x.g$ according to the relation $\lambda x.f \preceq_A{}^H \lambda x.g$. Moreover $\lambda x.\ell \in \preceq_A\ (\lambda x.g)$, according to the relation $\lambda x.g \preceq_A \lambda x.\ell$: hence $\lambda x.\ell \in \preceq_A \big(\preceq_A{}^H(\lambda x.f)\big)$. However, since it has been proved that $\preceq_A \subseteq \preceq_A{}^H$, then $\lambda x.\ell \in \preceq_A{}^H(\lambda x.f)$.

$\diamondsuit$ We conclude the prove of the key lemma for valued terms considering the case $e = \langle v_1, v_2\rangle$, referring us to the statement (3.38c).

Here derivation tree for the hypothesis must terminate with the Howe's rule for pair, namely

$$\frac{\begin{array}{l}\emptyset \vdash v_1 \preceq_A{}^H u_1 : A \\ \emptyset \vdash v_2 \preceq_A{}^H u_2 : B \qquad \emptyset \vdash \langle u_1, u_2\rangle \preceq_{A\otimes B} \langle w_1, w_2\rangle : A \otimes B\end{array}}{\emptyset \vdash \langle v_1, v_2\rangle \preceq_{A\otimes B}{}^H \langle w_1, w_2\rangle : A \otimes B}\ (How_6). \tag{3.43}$$

By compatibility of Howe's relation, from the first two premises of (3.43) we get

$$\langle v_1, v_2\rangle \preceq_{A\otimes B}{}^H \langle u_1, u_2\rangle, \tag{3.44}$$

whence by induction hypothesis it immediately follows, $\forall g \in \mathcal{T}^{\ell PST_\lambda}_{x:A,y:B,E}$

$$\mathcal{P}_{\ell PST_\lambda}\Big((\widehat{\langle v_1, v_2\rangle}, A\otimes B), a_{\otimes g}, (g\{v_1/x, v_2/y\}, E)\Big) \leq$$
$$\leq \mathcal{P}_{\ell PST_\lambda}\Big((\widehat{\langle u_1, u_2\rangle}, A\otimes B), a_{\otimes g}, \preceq_E{}^H(g\{v_1/x, v_2/y\}, E)\Big) \quad (3.45)$$

and by definition of probabilistic similarity to the second premise of (3.43) one finds, $\forall g \in \mathcal{T}^{\ell PST_\lambda}_{x:A,y:B,E}$:

$$\mathcal{P}_{\ell PST_\lambda}\Big((\widehat{\langle u_1, u_2\rangle}, A\otimes B), a_{\otimes g}, (g\{u_1/x, u_2/y\}, E)\Big) \leq$$
$$\leq \mathcal{P}_{\ell PST_\lambda}\Big((\widehat{\langle w_1, w_2\rangle}, A\otimes B), a_{\otimes g}, \preceq_E (g\{u_1/x, u_2/y\}, E)\Big) \quad (3.46)$$

Thus, from (3.46) and from (3.45) respectively, it follwows that

$$\forall g \in \mathcal{T}^{\ell PST_\lambda}_{x:A,y:B,E}, g\{w_1/x, w_2/y\} \in \preceq_E(g\{u_1/x, u_2/y\})$$
$$\forall g \in \mathcal{T}^{\ell PST_\lambda}_{x:A,y:B,E}, g\{u_1/x, u_2/y\} \in \preceq_E{}^H(g\{v_1/x, v_2/y\}), \quad (3.47)$$

whence, since by Lemma 2.17 we know that $\preceq_E \subseteq \preceq_E{}^H$, we find $\forall g \in \mathcal{T}^{\ell PST_\lambda}_{x:A,y:B,E}$, $g\{w_1/x, w_2/y\} \in \preceq_E{}^H (g\{v_1/x, v_2/y\})$. This is the required relation since it ensure that the thesis (3.38c) is fulfilled, namely

$$\mathcal{P}_{\ell PST_\lambda}\Big((\widehat{\langle v_1, v_2\rangle}, A\otimes B), a_{\otimes g}, (g\{v_1/x, v_2/y\}, E)\Big) \leq$$
$$\leq \mathcal{P}_{\ell PST_\lambda}\Big((\widehat{\langle w_1, w_2\rangle}, A\otimes B), a_{\otimes g}, \preceq_E (g\{v_1/x, v_2/y\}, E)\Big) \quad (3.48)$$

♢ If $e = f_1 f_2$ is an application term we write hypothesis as $\emptyset \vdash f_1 f_2 \preceq_A{}^H h : A \wedge f_1 f_2 \Downarrow \mathscr{E}$ and the statement of the key lemma takes the form

$$\Big(\emptyset \vdash f_1 f_2 \preceq_A{}^H h : A \wedge e \Downarrow \mathscr{E}\Big) \Rightarrow \Big(h \Downarrow [\![h]\!] \wedge \forall W \subseteq \mathcal{V}_A^{\ell PST_\lambda},$$
$$\mathcal{P}_{\ell PST_\lambda}((f_1 f_2, A), a_{eval}, (W, A)) \leq \mathcal{P}_{\ell PST_\lambda}\big((h, A), a_{eval}, (\preceq_A{}^H(W), A)\big)\Big). \quad (3.49)$$

Lemma 3.1 and the big–step evaluation rule for applications suggest the nature of $f_1 f_2$ semantics $[\![e]\!]$, which will be denoted by $\mathscr{E}$:

$$\frac{f_1 \Downarrow \mathscr{F}_1 \qquad f_2 \Downarrow \mathscr{F}_2 \qquad b_i\{\nu_n/x\} \Downarrow \mathscr{F}_{i,n}|_{\lambda\, x.b_i \in \mathtt{Sup}(\mathscr{F}_1), \nu_n \in \mathtt{Sup}(\mathscr{F}_2)}}{f_1 f_2 \Downarrow \underbrace{\sum_{\lambda\, x.b_i \in \mathtt{Sup}(\mathscr{F}_1), \nu_n \in \mathtt{Sup}(\mathscr{F}_2)} \mathscr{F}_1(\lambda\, x.b_i)\mathscr{F}_2(\nu_n)\mathscr{F}_{i,n}}_{\mathscr{E}}}. \quad (3.50)$$

Since the hypothesis is a consequence of the Howe's rule for applications

$$\frac{\begin{array}{l}\emptyset \vdash f_2 \preceq_B{}^H g_2 : B \\ \emptyset \vdash f_1 \preceq_{B \multimap A}{}^H g_1 : B \multimap A \qquad \emptyset \vdash g_1 g_2 \preceq_A h : A\end{array}}{\emptyset \vdash f_1 f_2 \preceq_A{}^H h : A}, \tag{3.51}$$

we may apply a double inductive hypothesis to the smaller terms f_1 and f_2, obtaining

$$\begin{array}{l}\left(\emptyset \vdash f_1 \preceq_{B \multimap A}{}^H g_1 : B \multimap A \ \wedge \ f_1 \Downarrow \mathscr{F}_1\right) \ \Rightarrow \\ \left(g_1 \Downarrow \mathscr{G}_1 \ \wedge \ \forall X \subseteq \mathcal{V}^{\ell PST_\lambda}_{B \multimap A}, \mathcal{P}_{\ell PST_\lambda}\left((f_1, B \multimap A), a_{eval}, (X, B \multimap A)\right) \leq \right. \\ \qquad \left. \mathcal{P}_{\ell PST_\lambda}\left((g_1, B \multimap A), a_{eval}, (\preceq_{B \multimap A}{}^H(X), B \multimap A)\right)\right) \\ \left(\emptyset \vdash f_2 \preceq_B{}^H g_2 : B \ \wedge \ f_2 \Downarrow \mathscr{F}_2\right) \ \Rightarrow \left(g_2 \Downarrow \mathscr{G}_2 \ \wedge \forall Y \subseteq \mathcal{V}^{\ell PST_\lambda}_B, \right. \\ \left. \wedge \ \mathcal{P}_{\ell PST_\lambda}\left((f_2, B), a_{eval}, (Y, B)\right) \leq \mathcal{P}_{\ell PST_\lambda}\left((g_2, B), a_{eval}, (\preceq_B{}^H(Y), B)\right)\right)\end{array} \tag{3.52}$$

Referring to the first two premises of (3.51), let us take $g_1 \Downarrow \mathscr{G}_1$ and $g_2 \Downarrow \mathscr{G}_2$, noticing that, due to (3.52), $\mathscr{G}_1$ and $\mathscr{G}_2$ are not empty distributions unless $\mathscr{F}_1$ and $\mathscr{F}_2$ are.

The reduction rule for application unfolds us the proper form of the distribution $\mathscr{G}$ to which $g_1 g_2$ evaluates:

$$\frac{g_1 \Downarrow \mathscr{G}_1 \qquad g_2 \Downarrow \mathscr{G}_2 \qquad b_i\{\nu_j/x\} \Downarrow \mathscr{G}_{ij}|_{\lambda\, x.b_i \in \mathtt{Sup}(\mathscr{G}_1), \nu_j \in \mathtt{Sup}(\mathscr{G}_2)}}{g_1 g_2 \Downarrow \underbrace{\sum_{\lambda\, x.b_i \in \mathtt{Sup}(\mathscr{G}_1), \nu_j \in \mathtt{Sup}(\mathscr{G}_2)} \mathscr{G}_1(\lambda\, x.b_i)\mathscr{G}_2(\nu_j)\mathscr{G}_{ij}}_{\mathscr{G}}} (app \Downarrow)_{\wp}. \tag{3.53}$$

The existence of the semantics of $g_1 g_2$, together with the similarity relation (see (3.51)) between $g_1 g_2$ and h, allow to conclude that $h \Downarrow [\![h]\!]$, as the thesis requires. Now let us look at $[\![f_1 f_2]\!]$ and $[\![g_1 g_2]\!]$ using the same symbols already introduced in equations (3.50) and (3.53) with the aim to compare them. Here we are under the scope of the disentangling lemma (see [11] for further details)

which ensures for the existence of two sets of numbers $\{r_{v_i,v'}\}_{i\in\mathcal{I}}$ and $\{s_{u_j,u'}\}_{j\in\mathcal{J}}$ such that the following inequalities hold

$$\begin{array}{lll} \forall v_i,\, \mathscr{F}_1(v_i) \leq \sum_{\forall v'\in {\preceq_{B\multimap A}}^H(v_i)} r_{v_i,v'} & \wedge & \sum_{i\in\mathcal{I}} r_{v_i,v'} \leq \mathscr{G}_1(v\,') \\ \forall u_j,\, \mathscr{F}_2(u_j) \leq \sum_{\forall u'\in {\preceq_B}^H(u_j)} s_{u_j,u'} & \wedge & \sum_{j\in\mathcal{J}} s_{u_j,u'} \leq \mathscr{G}_2(u'). \end{array} \tag{3.54}$$

With reference to the induction hypothesis (3.52), let us set $v_i \in X \subseteq \mathcal{V}_{B\multimap A}^{\ell ST_\lambda}$ and $u_j \in Y \subseteq \mathcal{V}_B^{\ell ST_\lambda}$: since by definition $v' \in {\preceq_{B\multimap A}}^H(X)$ and $u' \in {\preceq_B}^H(Y)$, by substitutivity (see Lemma 2.19), supposing $v_i = \lambda x.b_i$ and $v' = \lambda x.b'$, we get

$$\forall v_i \in X,\, \forall u_j \in Y,\, \forall\, v' \in \preceq_{B\multimap A} (X)^H, \forall\, u' \in \preceq_B (Y)^H b_i\{u_j/x\} {\preceq_A}^H b'\{u'/x\},$$

and since the same relation holds also when the evaluation rules have been applied we get $\mathscr{F}_{ij} {\preceq_A}^H \mathscr{G}_{v',u'}$.

Using (3.54) and the last remarks one obtains the following inequality, which holds for any $w \in W \subseteq \mathcal{V}_A^{\ell PST_\lambda}$

$$\begin{array}{l} \mathscr{E}(w) = \sum_{v_i\in X, u_j\in Y} \mathscr{F}_1(v_i)\mathscr{F}_2(u_j)\mathscr{F}_{ij}(w) \leq \sum_{\substack{v_i\in X,\, v'\in {\preceq_{B\multimap A}}^H(v_i) \\ u_j\in Y, u'\in {\preceq_{B\multimap A}}^H(u_j)}} r_{v_i,v'} s_{u_j,u'} \cdot \\ \cdot \mathscr{G}_{v',u'}\left({\preceq_A}^H(w)\right) \leq \sum_{v'\in X', u'\in Y'} \mathscr{G}_1(v')\mathscr{G}_2(u')\mathscr{G}_{v',u'}\left({\preceq_A}^H(w)\right) \leq \\ \sum_{v'\in \mathtt{Sup}(\mathscr{G}_1), u'\in \mathtt{Sup}(\mathscr{G}_2)} \mathscr{G}_1(v')\mathscr{G}_2(u')\mathscr{G}_{v',u'}\left({\preceq_A}^H(w)\right) = \mathscr{G}({\preceq_A}^H(w)). \end{array} \tag{3.55}$$

where, to unburden the formulas the notations $X \subseteq \mathcal{V}_{B\multimap A}^{\ell PST_\lambda} \Rightarrow X' = {\preceq_{B\multimap A}}^H(X) = \bigcup_{i=1}^n {\preceq_{B\multimap A}}^H(v_i)$ and $Y \subseteq \mathcal{V}_B^{\ell PST_\lambda} \Rightarrow Y' = {\preceq_B}^H(Y) = \bigcup_{j=1}^m {\preceq_B}^H(u_j)$ have been introduced.

Finally, recalling the definitions given for $\mathscr{E}$ and $\mathscr{G}$ in (3.50) and (3.53) respectively, we can rewrite (3.55) as

$$\forall W \in \mathcal{V}_A^{\ell PST_\lambda} \qquad [\![e]\!](W) \leq \mathscr{G}\left({\preceq_A}^H(W)\right) = [\![g]\!]\left({\preceq_A}^H(W)\right). \tag{3.56}$$

Moreover, the last premise of the rule (3.51), namely the more familiar probabilistic similarity relation $\emptyset \vdash g_1 g_2 \preceq_A h : A$, denoting by $\mathscr{H}$ the semantics

of h implies that

$$\forall Z \subseteq \mathcal{V}_A^{\ell PST_\lambda},\ [\![g]\!](Z) = \mathscr{G}(Z) \leq \mathscr{H}\left(\preceq_A (Z)\right) \leq \mathscr{H}\left({\preceq_A}^H(Z)\right) = [\![h]\!]\left({\preceq_A}^H(Z)\right), \tag{3.57}$$

where for the last inequality of (3.57) the property $\preceq^H \subseteq \preceq$, which has been stated in Lemma 2.17, has used.

To complete the prove just choose the inequalities (3.56) and (3.57) setting $Z = {\preceq_A}^H(W)$ and recalling the relation $\forall W,\ {\preceq_A}^H\left({\preceq_A}^H(W)\right) = {\preceq_A}^H(W)$.

$\diamondsuit$ If $e = f_1 \oplus f_2$, $e = \texttt{if}\ f_1\ \texttt{then}\ f_2\ \texttt{else}\ f_3$, then let us write $e = \texttt{cnstr}\left(\{f_n\}_{n\in\mathcal{N}}\right)$ where `cnstr` is some syntactic constructor and $\{f_n\}_{n\in\mathcal{N}}$ are subterms of e. Hence we write the hypothesis as $\emptyset \vdash \texttt{cnstr}\left(\{f_n\}_{n\in\mathcal{N}}\right) {\preceq_A}^H h : A \ \wedge\ e \Downarrow \mathscr{E}$ and the statement which has to be proved by induction on the size of terms involved in big–step semantics rule is again (3.38d). We may refer to the thesis in (3.38d) by rewriting it in a more appropriate form

$$h \Downarrow \mathscr{H} \ \wedge\ \forall X \subseteq \mathcal{V}_A^{\ell PST_\lambda},\ \mathcal{P}_{\ell PST_\lambda}\left(\left(\texttt{cnstr}\left(\{f_n\}_{n\in\mathcal{N}}\right), A\right), a_{eval}, (X, A)\right) \leq \mathcal{P}_{\ell PST_\lambda}\left((h, A), a_{eval}, \left({\preceq_A}^H(X), A\right)\right), \tag{3.58}$$

and a suitable (big–step) semantics evaluation rule will allow us to find the proper form of $[\![e]\!]$ which will be denoted by $\mathscr{E}$

$$\frac{\{f_n\}_{n\in\mathcal{N}} \Downarrow \{\mathscr{F}_n\}_{n\in\mathcal{N}}}{\texttt{cnstr}\left(\{f_n\}_{n\in\mathcal{N}}\right) \Downarrow \mathscr{E}\left(\{\mathscr{F}_n\}_{n\in\mathcal{N}}\right)}, \tag{3.59}$$

where by writing $\mathscr{E}\left(\{\mathscr{F}_n\}_{n\in\mathcal{N}}\right)$ we understood that $\mathscr{E}$ is some function of the subterms distributions $\mathscr{E}_n$. Hereby the hypothesis must be read as a consequence of general Howe's rule, namely:

$$\frac{\begin{array}{ll}\emptyset \vdash f_1 {\preceq_{A_1}}^H g_1 : A_1 & \\ \vdots \qquad\qquad \vdots & \\ \emptyset \vdash f_N {\preceq_{A_N}}^H g_N : A_N & \qquad \emptyset \vdash \texttt{cnstr}\left(\{g_n\}_{n\in\mathcal{N}}\right) \preceq_A h : A\end{array}}{\emptyset \vdash \texttt{cnstr}\left(\{f_n\}_{n\in\mathcal{N}}\right) {\preceq_A}^H h : A}\ (How_{gen}). \tag{3.60}$$

Now from the first N premises of (3.60), N new inductive hypotheses follow, which may be written as: $\forall n \in \mathcal{N}$,

$$\emptyset \vdash f_n \preceq_{A_n}{}^H g_n : A_n \;\Rightarrow \Big(g_n \Downarrow \mathscr{G}_n \wedge \forall X_n \subseteq \mathcal{V}_A^{\ell PST_\lambda},$$
$$\mathcal{P}_{\ell PST_\lambda}\left((f_n, A_n), a_{eval}, (X_n, A_n)\right) \leq \mathcal{P}_{\ell PST_\lambda}\left((g_n, A_n), a_{eval}, (\preceq_A{}^H(X_n), A_n)\right), \tag{3.61}$$

which allow to build the distribution $\mathscr{G}$, semantics of $\texttt{cnstr}\left(\{g_n\}_{n\in\mathcal{N}}\right)$ through a suitable big–step–semantis rule, as it is shown below:

$$\frac{\{g_n\}_{n\in\mathcal{N}} \Downarrow \{\mathscr{G}_n\}_{n\in\mathcal{N}}}{\texttt{cnstr}\left(\{g_n\}_{n\in\mathcal{N}}\right) \Downarrow \mathscr{G}\left(\{\mathscr{G}_n\}_{n\in\mathcal{N}}\right)}, \tag{3.62}$$

and since $\texttt{cnstr}\left(\{g_n\}_{n\in\mathcal{N}}\right)$ has a semantics $\mathscr{G}$ and through (3.60) we see that it is related to h by a similarity relation, we must conclude that

$$h \Downarrow \mathscr{H} \;\wedge\; \forall W \subseteq \mathcal{V}_A^{\ell PST_\lambda}, [\![g]\!](W) \leq [\![h]\!](\preceq_A(W)). \tag{3.63}$$

By compatibility of $\preceq_A{}^H$, starting from first N premises of (3.60) one can deduce $\emptyset \vdash \texttt{cnstr}\left(\{f_n\}_{n\in\mathcal{N}}\right) \preceq_A{}^H \texttt{cnstr}\left(\{g_n\}_{n\in\mathcal{N}}\right) : A$, and since to this term we can apply the induction hypothesis we find

$$\emptyset \vdash \texttt{cnstr}\left(\{f_n\}_{n\in\mathcal{N}}\right) \preceq_A{}^H \texttt{cnstr}\left(\{g_n\}_{n\in\mathcal{N}}\right) : A \;\Rightarrow$$
$$\Rightarrow\; \forall X \subseteq \mathcal{V}_A^{\ell PST_\lambda}, \mathscr{E}(X) \leq \mathscr{G}(\preceq_A{}^H(X)). \tag{3.64}$$

Now let us simply rewrite the last statement making use of the semantics of the terms as

$$\forall X \subseteq \mathcal{V}_A^{\ell PST_\lambda}, [\![e]\!](X) \leq [\![g]\!](\preceq_A{}^H(X)). \tag{3.65}$$

Thus the thesis is a consequence of (3.65) and of (3.63) if for each X we set $W = \preceq_A{}^H(X)$, since using the property $\preceq_A \subseteq \preceq_A{}^H$ we obtain the result $\preceq_A (W) = \preceq_A \left(\preceq_A{}^H(X)\right) = \preceq_A{}^H(X)$, whence $\forall X \subseteq \mathcal{V}_A^{\ell PST_\lambda}, [\![e]\!](X) \leq [\![h]\!](\preceq_A{}^H(X))$.

$\diamondsuit$ Taking $e = (\texttt{let}\ f_1\ \texttt{be}\ \langle x, y\rangle\ \texttt{in}\ f_2)$ leads to the statement

$$\Big(\emptyset \vdash (\texttt{let}\ f_1\ \texttt{be}\ \langle x, y\rangle\ \texttt{in}\ f_2) \preceq_A{}^H h : A \wedge e \Downarrow \mathscr{E}\Big) \Rightarrow \Big(h \Downarrow \mathscr{H} \wedge \forall W \subseteq \mathcal{V}_A^{\ell PST_\lambda},$$
$$\mathcal{P}_{\ell PST_\lambda}\left((e, A), a_{eval}, (W, A)\right) \leq \mathcal{P}_{\ell PST_\lambda}\left((h, A), a_{eval}, (\preceq_A{}^H(W), A)\right)\Big). \tag{3.66}$$

and hypothesis comes to be a consequence of the following Howe's rule

$$\frac{\begin{array}{l} x : B, y : E \vdash f_2 \preceq_A{}^H g_2 : A \\ \emptyset \vdash f_1 \preceq_{B\otimes E}{}^H g_1 : B \otimes E \qquad\qquad \emptyset \vdash \texttt{let } g_1 \texttt{ be } \langle x, y\rangle \texttt{ in } g_2 \preceq_A h : A \end{array}}{\emptyset \vdash \texttt{let } f_1 \texttt{ be } \langle x, y\rangle \texttt{ in } f_2 \preceq_A{}^H h : A}. \tag{3.67}$$

The semantics rule for terms of this type briefs us about the form of $[\![e]\!]$ as functions their subterms semantics

$$\frac{f_1 \Downarrow \mathscr{F}_1 \qquad f_2\{v/x, u/y\} \Downarrow \mathscr{F}_{\langle v,u\rangle}\big|_{\langle v,u\rangle\in \mathtt{Sup}(\mathscr{F}_1)}}{\texttt{let } f_1 \texttt{ be } \langle x, y\rangle \texttt{ in } f_2 \Downarrow \underbrace{\sum_{\langle v,u\rangle\in \mathtt{Sup}(\mathscr{F}_1)} \mathscr{F}_1(\langle v, u\rangle)\mathscr{F}_{\langle v,u\rangle}}_{\mathscr{E}}}\,(let \Downarrow)_\wp \tag{3.68}$$

and the double induction hypothesis which stems from the first two premises of equation (3.67), introduce to the semantics of the subterms g_1 and g_2. Writing induction hypothesis for open terms such as f_2 and g_2 requires to use the definition of open extension for applicative bisimulation, as in the following:

- $$\Big(\emptyset \vdash f_1 \preceq_{B\otimes E}{}^H g_1 : B \otimes E \ \wedge f_1 \Downarrow \ \mathscr{F}_1\Big) \Rightarrow \Big(g_1 \Downarrow \mathscr{G}_1 \ \wedge \ \forall X \subseteq \mathcal{V}_{B\otimes E}^{\ell PST_\lambda} \mathcal{P}_{\ell PST_\lambda}\left((f_1, B \otimes E), a_{eval}, (X, B \otimes E)\right) \leq \mathcal{P}_{\ell PST_\lambda}\left((g_1, B \otimes E), a_{eval}, (\preceq_{B\otimes E}(X), B \otimes E)\right)\Big) \tag{3.69}$$
- $$\Big(x : B, y : E \vdash f_2 \preceq_A{}^H g_2 : A \ \wedge \ \forall \langle v, u\rangle \in \mathcal{V}_{B\otimes E}^{\ell PST_\lambda}, f_2\{v/x, u/y\} \Downarrow \mathscr{F}_{\langle v,u\rangle}\Big) \Rightarrow \Big(\forall \langle v, u\rangle \in \mathcal{V}_{B\otimes E}^{\ell PST_\lambda} g_2\{v/x, u/y\} \Downarrow \mathscr{G}_{\langle v,u\rangle} \wedge \ \forall Z \in \mathcal{V}_A^{\ell PST_\lambda} \mathcal{P}_{\ell PST_\lambda}\left((f_2\{v/x, u/y\}, A), a_{eval}, (Z, A)\right) \leq \mathcal{P}_{\ell PST_\lambda}\left((g_2\{v/x, u/y\}, A), a_{eval}, (\preceq_A{}^H(Z), A)\right)\Big) \tag{3.70}$$

whence

$$\frac{g_1 \Downarrow \mathscr{G}_1 \qquad g_2\{v/x, u/y\} \Downarrow \mathscr{G}_{\langle v,u\rangle}\big|_{\langle v,u\rangle\in \mathtt{Sup}(\mathscr{G}_1)}}{\texttt{let } g_1 \texttt{ be } \langle x, y\rangle \texttt{ in } g_2 \Downarrow \underbrace{\sum_{\langle v,u\rangle\in \mathtt{Sup}(\mathscr{G}_1)} \mathscr{G}_1(\langle v, u\rangle)\mathscr{G}_{\langle v,u\rangle}}_{\mathscr{G}}}\,(app \Downarrow)_\wp. \tag{3.71}$$

After it has been obtained how $(\texttt{let } g_1 \texttt{ be } \langle x, y\rangle \texttt{ in } g_2) \Downarrow \mathscr{G}$, starting from hypothesis of similarity supplied by the last premise of (3.67) one finds the

condition

$$h \Downarrow [\![h]\!] \ \wedge \ \forall W \in \mathcal{T}_A^{\ell PST_\lambda}[\![g]\!](W) \leq [\![h]\!](\preceq_A (W)) \tag{3.72}$$

To end the proof of this item of key Lemma requires to compare the distribution $\mathscr{E}$, $\mathscr{G}$ and $\mathscr{H}$, which appear as result in formulae (3.68) and (3.71) and to this purpose let notice that, due to the proof of substitutivity given in Lemma 2.19, we get

$$x : B, y : E \vdash f_2 \preceq_A{}^H g_2 : A \Rightarrow \ \forall \langle v, u\rangle \in \mathcal{V}_{B\otimes E}^{\ell PST_\lambda}, \ f_2\{v/x, u/y\} \preceq_A{}^H g_2\{v/x, u/y\}, \tag{3.73}$$

thus referring to the rule (3.67) and to the symbols used in (3.68), it may be derived that $\forall\langle v, u\rangle$, $\forall W$, $\mathscr{F}_{\langle v,u\rangle}(W) \leq \mathscr{G}_{\langle v,u\rangle}(\preceq_A{}^H(W))$. Therefore we will write

$$\begin{aligned}
\forall Z \subseteq \mathcal{V}_A^{\ell PST_\lambda}\mathscr{E}(Z) = &\sum_{\langle v,u\rangle \in \mathtt{Sup}(\mathscr{F}_1)} \mathscr{F}_1(\langle v, u\rangle)\mathscr{F}_{\langle v,u\rangle}(Z) \leq \\
&\leq \sum_{\langle v,u\rangle \in \preceq_{B\otimes E}{}^H(\mathtt{Sup}(\mathscr{F}_1))} \mathscr{G}_1(\langle u, w\rangle)\mathscr{G}_{\langle v,u\rangle}(\preceq_A{}^H(Z)) = \\
&\sum_{\langle v,\nu\rangle \in \{\preceq_{B\otimes E}{}^H(\mathtt{Sup}(\mathscr{F}_1))\cap \mathtt{Sup}(\mathscr{G}_1)\}} \mathscr{G}_1(\langle u, w\rangle)\mathscr{G}_{\langle v,u\rangle}(\preceq_A{}^H(Z)) \leq \\
&\leq \sum_{\langle v,u\rangle \in \mathtt{Sup}(\mathscr{G}_1)} \mathscr{G}_1(\langle u, w\rangle)\mathscr{G}_{\langle v,u\rangle}(\preceq_A{}^H(Z)) = \mathscr{G}(\preceq_A{}^H(Z)).
\end{aligned} \tag{3.74}$$

Taking (3.72) and (3.74) and setting $\forall Z$, $W = \preceq_A{}^H(Z)$ we find the thesis. Being for Lemma 2.17 $\preceq_A \subseteq \preceq_A{}^H$ we write

$$\forall Z \subseteq \mathcal{V}_A^{\ell PST_\lambda}[\![e]\!](Z) = \mathscr{E}(Z) \leq \mathscr{G}(\preceq_A{}^H(Z)) = [\![g]\!](\preceq_A{}^H(Z)) \leq [\![h]\!](\preceq_A{}^H(Z)). \tag{3.75}$$

□

3.3.1 On the transitive closure properties

Even though $\preceq_A{}^H$ is bigger (or equal) than $\preceq_A$ by lemma (2.17), being it a probabilistic relation it is not ensured to be a similarity, since a probabilistic similarity must be by definition a transitive relation.

Thus the transitive closure $(\preceq_A{}^H)^+$ should be rather considered, to be sure to really deal with the bigger probabilistic simulation.Afterward it should be shown that it fits, in turn, all the properties that $\preceq^H$ has. Many properties have already been proved somewhere in previous sections, hence the results are resumed in the following Table 3.5 where, beside to each property is featured the section where corresponding lemma appears.

PROPERTY			REFERENCE
R closed under terms substitution	$\Rightarrow$	R^+ closed for terms substitution	Lemma 3.8
R compatible	$\Rightarrow$	R^+ compatible	Lemma 3.7
R closed under terms substitution	$\Rightarrow$	R^H substitutive	Lemma 2.19
R transitive	$\Rightarrow$	R^H pseudo–transitive	Lemma 2.18
$R \subseteq R^H$			Lemma 2.17
R reflexive	$\Rightarrow$	R^H compatible	Lemma 2.16
R^H compatible	$\Rightarrow$	R^H reflexive	Lemma 2.13

Figure 3.5: Reference Table for the proved properties about Howe's lifting and about transitive closure of a relation R.

Proposition 3.2. $(\preceq_A{}^H)^+$ *is compatible.*

Proof. Since $\preceq_A$ reflexive $\overset{\text{lemma(2.16)}}{\Longrightarrow}$ $\preceq_A{}^H$ is compatible $\overset{\text{lemma(3.7)}}{\Longrightarrow}$ $(\preceq_A{}^H)^+$ is compatible. □

Proposition 3.3. $(\preceq_A{}^H)^+$ *is transitive.*

Proof. by definition of transitive closure. □

Proposition 3.4. $(\preceq_A{}^H)^+$ *is reflexive.*

Proof. Since $\preceq_A$ reflexive $\overset{\text{Lemma 2.16}}{\Longrightarrow}$ $\preceq_A{}^H$ is compatible $\overset{\text{Lemma 2.13}}{\Longrightarrow}$ $\preceq_A{}^H$ reflexive (and compatible) $\overset{\text{Lemma 3.7}}{\Longrightarrow}$ $(\preceq_A{}^H)^+$ is compatible and hence reflexive. □

Proposition 3.5. $(\preceq_A{}^H)^+$ *is a precongruence.*

Proof. This is a consequence of the previous Proposition 3.2, Proposition 3.4 and Proposition 3.3. □

Proposition 3.6. $(\preceq_A{}^H)^+$ *is closed under substitution.*

Proof. Since $\preceq_A$ is closed under substitution $\overset{\text{Lemma 2.19}}{\Longrightarrow}$ $\preceq_A{}^H$ is substitutive (and hence closed under substitution) $\overset{\text{Lemma 3.8}}{\Longrightarrow}$ $(\preceq_A{}^H)^+$ is also closed under substitution as $\preceq_A$ is. □

Proposition 3.7. $\preceq_A \subseteq (\preceq_A{}^H)^+$.

Proof. Since $\preceq_A \subseteq \preceq_A{}^H \subseteq (\preceq_A{}^H)^+$ by Lemma 2.17 and from the definition of transitive closure. □

Lemma 3.13. $\preceq_A{}^H \subseteq \preceq_A \Rightarrow (\preceq_A{}^H)^+ \subseteq \preceq_A$ *Therefore, provided that – according to the probabilistic key Lemma 3.12 – Howe's lifting has the probabilistic similarity behaviour, also its transitive closure has the same property.*

Proof. This statement has to be proved in both cases whether e, h are values or generic terms.

$e, h, \in \mathcal{V}_{\mathtt{bool}}^{\ell PST_\lambda}-$ The statement to prove is:

$$e({\preceq_{\mathtt{bool}}}^H)^+ h \;\Rightarrow\; \forall \mathtt{b} \in \{\mathtt{tt}, \mathtt{ff}\},$$
$$\mathcal{P}_{\ell PST_\lambda}\left((\widehat{e}, \mathtt{bool}), a_{\mathtt{b}}, (\widehat{\mathtt{b}}, \mathtt{bool})\right) \leq \mathcal{P}_{\ell PST_\lambda}\left((h, \mathtt{bool}), a_{\mathtt{b}}, \left(({\preceq_{\mathtt{bool}}}^H)^+(\mathtt{b}), \mathtt{bool}\right)\right). \tag{3.76}$$

If $\hat{e} \neq \mathtt{b}$ the statement (3.76) is obviously true (the left–hand side is zero), otherwise let us recall that since $e({\preceq_{\mathtt{bool}}}^H)^+ h$, then by definition $h \in ({\preceq_{\mathtt{bool}}}^H)^+(e)$ and being $e = \mathtt{b}$, we must conclude that both terms of (3.76) are equal to 1.

$e, h \in \mathcal{V}_{B \multimap A}^{\ell PST_\lambda}-$ whence the thesis

$$\lambda\, x.f({\preceq_{B\multimap A}}^H)^+ \lambda\, x.\ell \Rightarrow \left(\forall X \subseteq \mathcal{T}_A^{\ell PST_\lambda} \mathcal{P}_{\ell PST_\lambda}\left((\widehat{\lambda\, x.f}, A), a_{@v}, (X, A)\right) \leq \right.$$
$$\left. \mathcal{P}_{\ell PST_\lambda}\left((\widehat{\lambda\, x.\ell}, A), a_{@v}, \left(({\preceq_{B\multimap A}}^H)^+(X), A\right)\right)\right) \tag{3.77}$$

Given the value of $v \in \mathcal{V}_B^{\ell PST_\lambda}$, if we choose the set X in a way that $f\{v/x\} \notin X$, the inequality (3.77) is obviously true (since its left–hand size is zero), otherwise if the hypothesis $\widehat{\lambda\, x.f}({\preceq_{B\multimap A}}^H)^+\widehat{\lambda\, x.\ell}$ is a consequence of the rule $(tc-1)$, we get the relation $\widehat{\lambda\, x.f}{\preceq_{B\multimap A}}^H\widehat{\lambda\, x.\ell}$, which gives $\lambda\, x.\ell \in {\preceq_{B\multimap A}}^H(\lambda\, x.f)$, entailing the thesis, since ${\preceq_{B\multimap A}}^H \subseteq ({\preceq_{B\multimap A}}^H)^+$. Otherwise, if the hypothesis is a consequence of the rule $(tc-2)$, then for some value $\lambda\, x.g$ we must have that $\lambda\, x.f{\preceq_{B\multimap A}}^H\lambda\, x.g \wedge \lambda\, x.g{\preceq_{B\multimap A}}^H\lambda\, x.\ell$, whence $\lambda\, x.\ell \in {\preceq_{B\multimap A}}^H(\lambda\, x.g) \wedge \lambda\, x.g \in {\preceq_{B\multimap A}}^H(\lambda\, x.f)$ and these relations together ensure that $\lambda\, x.\ell \in {\preceq_{B\multimap A}}^H(\lambda\, x.f)$. This proves the inequality (3.77) because, by definition of transitive closure, ${\preceq_{B\multimap A}}^H \subseteq ({\preceq_{B\multimap A}}^H)^+$.

$e, h \in \mathcal{V}_{A\otimes B}^{\ell PST_\lambda}-$ entails the thesis:

$$\langle v_1, v_2\rangle({\preceq_{A\otimes B}}^H)^+\langle w_1, w_2\rangle \;\Rightarrow \forall g \in \mathcal{T}_{x:A,y,B,E}^{\ell PST_\lambda}$$
$$\mathcal{P}_{\ell PST_\lambda}\left((\widehat{\langle v_1, v_2\rangle}, A \otimes B), a_{\otimes g}, (g\{v_1/x, v_2/y\}, E)\right) \leq$$
$$\leq \mathcal{P}_{\ell PST_\lambda}\left((\widehat{\langle w_1, w_2\rangle}, A \otimes B), a_{\otimes g}, \left(({\preceq_E}^H)^+(g\{v_1/x, v_2/y\}), E\right)\right). \tag{3.78}$$

If the hypothesis comes from $(tc-1)$ then thesis is a consequesce of Lemma 3.12, otherwise there is a pair $\langle \nu_1, \nu_2\rangle$, such that $\langle v_1, v_2\rangle\ ({\preceq_{A\otimes B}}^H)^+\ \langle \nu_1, \nu_2\rangle$ and $\langle \nu_1, \nu_2\rangle$

$(\preceq_{A\otimes B}{}^{H})^{+}\ \langle w_1, w_2\rangle$ then, applying induction hypothesis, yields the conditions $\forall g \in \mathcal{T}^{\ell PST_\lambda}_{x:A,y,B,E}$

$$\mathcal{P}_{\ell PST_\lambda}\left((\widehat{\langle v_1, v_2\rangle}, A\otimes B), a_{\otimes g}, (g\{v_1/x, v_2/y\}, E)\right) \leq \\ \leq \mathcal{P}_{\ell PST_\lambda}\left((\widehat{\langle \nu_1, \nu_2\rangle}, A\otimes B), a_{\otimes g}, \left((\preceq_E{}^{H})^{+}(g\{v_1/x, v_2/y\}), E\right)\right) \quad (3.79)$$

$$\mathcal{P}_{\ell PST_\lambda}\left((\widehat{\langle \nu_1, \nu_2\rangle}, A\otimes B), a_{\otimes g}, (g\{\nu_1/x, \nu_2/y\}, E)\right) \leq \\ \leq \mathcal{P}_{\ell PST_\lambda}\left((\widehat{\langle w_1, w_2\rangle}, A\otimes B), a_{\otimes g}, \left((\preceq_E{}^{H})^{+}(g\{\nu_1/x, \nu_2/y\}), E\right)\right), \quad (3.80)$$

whence we can obtain the two relations $g\{\nu_1/x, \nu_2/y\} \in (\preceq_E{}^{H})^{+}(g\{v_1/x, v_2/y\})$ and $g\{w_1/x, w_2/y\} \in (\preceq_E{}^{H})^{+}(g\{\nu_1/x, \nu_2/y\})$.

Since $\forall X \in \mathcal{T}^{\ell ST_\lambda}_E$, $(\preceq_E{}^{H})^{+}\left((\preceq_E{}^{H})^{+}(X)\right) = (\preceq_E{}^{H})^{+}(X)$, the thesis follows from these last two relations.

$e, h \in \mathcal{T}^{\ell PST_\lambda}_A -$ Here as usual we enforce the induction hypothesis, twhich makes the statement to be

$$e(\preceq_A{}^{H})^{+}h \wedge e \Downarrow [\![e]\!] \Rightarrow \Big(h \Downarrow [\![h]\!] \wedge \forall X \subseteq \mathcal{V}^{\ell PST_\lambda}_A \mathcal{P}_{\ell PST_\lambda}((e, X), a_{eval}, (X, A)) \leq \\ \mathcal{P}_{\ell PST_\lambda}\left((h, X), a_{eval}, (\preceq_A{}^{H+}(X), A)\right)\Big). \quad (3.81)$$

We have therefore two cases:

- the hypothesis $e(\preceq_A{}^{H})^{+}h$ is a consequence of $(tc-1)$, whence $e \preceq_A{}^{H} h$ must hold, and as a consequence of the probabilistic key lemma (3.12) we obtain both the statements $e \Downarrow [\![e]\!]$ and

  $$\forall X \subseteq \mathcal{V}^{\ell PST_\lambda}_A, \\ \mathcal{P}_{\ell PST_\lambda}((e, A), a_{eval}, (X, A)) \leq \mathcal{P}_{\ell PST_\lambda}\left((h, A), a_{eval}, (\preceq_A{}^{H}(X), A)\right). \quad (3.82)$$

  Moreover, considering that by definition of transitive closure, $\forall X$, $\preceq_A{}^{H}(X) \subseteq (\preceq_A{}^{H})^{+}(X)$, we get immediately the thesis applying this inequality to (3.82).

- The hypothesis $e({\preceq_A}^H)^+h$ is a consequence of $(tc-2)$, then for some g we have $e({\preceq_A}^H)^+g \wedge g({\preceq_A}^H)^+h$. Thus, by inductive hypothesis applied on both terms, we write

$$\begin{aligned} g \Downarrow [\![g]\!] \ \wedge \ \forall X \subseteq \mathcal{V}_A^{\ell PST_\lambda}, \mathcal{P}_{\ell PST_\lambda}\left((e,A), a_{eval}, (X,A)\right) \leq \\ \mathcal{P}_{\ell PST_\lambda}\left((g,A), a_{eval}, (({\preceq_A}^H)^+(X),A)\right) \\ h \Downarrow [\![h]\!] \ \wedge \ \forall Y \subseteq \mathcal{V}_A^{\ell PST_\lambda}, \mathcal{P}_{\ell PST_\lambda}\left((g,A), a_{eval}, (Y,A)\right) \leq \\ \mathcal{P}_{\ell PST_\lambda}\left((h,A), a_{eval}, (({\preceq_A}^H)^+(X),A)\right) \quad (3.83) \end{aligned}$$

  then let just take $Y = {\preceq_A}^H(X)$ in the second equation (3.83)and let us recall that $({\preceq_A}^H)^+\left(({\preceq_A}^H)^+\right)(X) = ({\preceq_A}^H)^+(X)$, to get the thesis.

□

3.4 Soundness and Completeness within the Probabilistic Environment

Finally the most important feature that a relation among terms must match is to be consonant with the most classical relation of context equivalence. This means that whatever pair of term which are bisimilar must be context equivalent too.

This condition is shown by following Lemma.

Lemma 3.14 (On a probabilistic similarity behaviour with respect to contexts)**.** *Likewise in deterministic case, the probabilistic similarity relation is compatible with the context, namely it satisfies the condition*

$$\emptyset \vdash e \preceq_A h : A \ \Rightarrow \ \forall C[\cdot] \in \mathtt{CTX}_B\left(\emptyset \vdash A\right), \emptyset \vdash C[e] \preceq_B C[h] : B.$$

Proof. Based on the compatibility of applicative similarity, it was given for deterministic case – see Lemma 2.21. □

Theorem 3.1. *In ℓPST_λ, $\preceq$ is included in $\leq$, thus $\sim$ is included in $\equiv$.*

Proof. Likewise in deterministic case, one has to prove that $\emptyset \vdash e \preceq_A h : A \Rightarrow \emptyset \vdash e \leq_A h : A$, but following the definition of context preorder, the thesis becomes $\forall C[\cdot] \in \mathtt{CTX}_B(\emptyset \vdash A)$, $\mathbf{Obs}(C[e]) \leq \mathbf{Obs}(C[h])$. With respect to the deterministic case, were the analogous of the above Lemma 3.14 allows to write $\emptyset \vdash C[e] \preceq_B C[h] : B$, only the definition of similarity and that of context preorder are different.

Indeed, here the sentence $\emptyset \vdash C[e] \preceq_B C[h] : B$ is translated in the language of LMC as

$$\forall X \subseteq \mathcal{V}^B,\ \mathcal{P}_{\ell PST_\lambda}((C[e], B), a_{eval}, (X, B)) \leq \mathcal{P}_{\ell PST_\lambda}((C[h], B), a_{eval}, (\preceq_B (X), B)), \tag{3.84}$$

where the set X can be chosen so that $\mathtt{Sup}(C[e]) \subseteq X$ and $\mathtt{Sup}(C[h]) \subseteq X$.

Now it is enough to recall the meaning ascribed to these matrix elements in the probabilistic environment as well as the definition given in (3.10a) to conclude, at once $\emptyset \vdash e \preceq_A h : A \Rightarrow \forall C[\cdot] \in \mathtt{CTX}_B(\emptyset \vdash A), \emptyset \vdash C[e] \preceq_B C[h] : B \Rightarrow \mathbf{Obs}(C[e]) \leq \mathbf{Obs}(C[h])$. □

In the deterministic calculus ℓST_λ, bisimilarity not only is *included* into context equivalence, but *coincides* with it (and, analogously, similarity coincides with the context preorder). This can also be proved, e.g., by observing that in $\mathscr{L}_{\ell ST_\lambda}$, bisimilarity coincides with trace equivalence, and each linear test,namely each trace, can be implemented by a context. This result is not surprising since it has already been obtained in similar settings elsewhere [8].

But how about ℓST_λ? Actually, there is a little hope to prove full–abstraction between context equivalence and bisimilarity in a linear setting, if probabilistic choice is present. Indeed, as shown by van Breugel et al. [56], probabilistic bisimilarity can be characterized by a notion of test equivalence where tests can be conjuctive, i.e., they can be in the form $t = \langle s, p \rangle$, and t succeeds if both s and p succeed. Implementing conjuctive tests, thus, requires *copying* the tested term, which is impossible in a linear setting. Indeed, it is easy to find a counterexample to full–abstraction already in ℓPST_λ. Consider the following two terms, both of which can be given

type $\texttt{bool} \multimap \texttt{bool}$ in ℓPST_λ:

$$e = \lambda\, x.\texttt{weak}\ x\ \texttt{in}\ (\texttt{tt} \oplus \Omega) \qquad f = (\lambda\, x.\texttt{weak}\ x\ \texttt{in}\ \texttt{tt}) \oplus (\lambda\, x.\texttt{weak}\ x\ \texttt{in}\ \Omega).$$

The two terms are not bisimilar, simply because $\texttt{tt}$ and Ω are not bisimilar, and thus also $\lambda\, x.\texttt{weak}\ x\ \texttt{in}\ \texttt{tt}$ and $\lambda\, x.\texttt{weak}\ x\ \texttt{in}\ \Omega$ cannot be bisimilar. However, through trace equivalence relation, they can be proved to be context equivalent: indeed there is no way to discriminate between them by way of a linear context (see [11] for more details).

Chapter 4

Quantum Language

Although quantum computing has been historically studied at the hardware level [43], since it has been described in terms of quantum gates, neglecting flow control, in recent years an increasing consideration has been paid in deepening the knowledge of quantum languages also in terms of *flow control* [50]: in most of these models the inner logical gates, the flow control as well as the whole system with its mechanical parts are purely quantum systems which, since such they are, must be seen as superposition of many classical states. As an example, in the quantum Turing machine the tape and the position of the head itself are assumed to be superposition of several states. Nevertheless, in our analysis the quantum computation occurs through a classical program, with an ordinary set of instructions and control devices which are connected to quantum gates: this situation is usually depicted by quoting the sentence "quantum data, classical control". Linear λ-calculi with classical control and quantum data have been introduced and studied both from an operational and from a semantical point of view [51, 52].

In a quantum calculus, linearity is a necessary constraint because of the well known impossibility of copying an unknown system in a quantum, microscopical state [32]. Besides, the other important feature, which is driven by the quantum nature of the storage devices, is the need to keep track of the position of each variable inside the quantum register – denoted in the following by $\mathscr{Q}$ – which compels to give together with the term some more information, with respect to the classical case,

on the quantum variables which it depends on.

Generally speaking, a quantum system in a bound state is, mathematically, a vector of a finite–dimensional complex Hilbert space $\mathcal{H}(\{\vec{v}_n\}_{n\in\mathcal{N}})$: this entails that a quantum microscopic system is described as a linear superposition of the set $\{\vec{v}_n\}_{n\in\mathcal{N}}$ of basis vectors of the $\mathcal{H}(\{\vec{v}_n\}_{n\in\mathcal{N}})$, with complex coefficients determined by the boundary conditions. Here the set $\mathcal{N}$ is not necessarily a proper subset of the integer numbers $\mathbb{Z}$ and the squared modulus of a complex coefficient corresponding to a given basis vector in the linear combination, gives the probability that after a measurement, the system lies in this particular basis vector.

The Dirac notation became the standard in quantum mechanics because of its conciseness and versatility in representing the vectors of Hilbert's space. A generic vector is written as a ket – symbol $|\alpha\rangle$ – linear superposition of the basis kets $\{|v_n\rangle\}_{n\in\mathcal{N}}$, following the usual notation $|\alpha\rangle = \sum_{n\in\mathcal{N}} \alpha_n |v_n\rangle$ with α_n complex numbers.

The Hilbert's space of kets has a dual correspondant, consisting of all linear functionals on the kets's space whose generic element, called bra, is denoted by the symbol $\langle\alpha|$. In addition to the operations of sum and product for a number, there are two other operations defined on the elements of the Hilbert's space, namely

- the scalar product $\langle\alpha|\beta\rangle$ between two vectors $|\alpha\rangle$ and $|\beta\rangle$ of Hilbert's space, enjoing the usual property $\langle\alpha|\beta\rangle = \langle\beta|\alpha\rangle^\star$;

- the tensor product $|\alpha\rangle \otimes |\beta\rangle$ which increases the dimension of the former Hilbert's spaces to which $|\alpha\rangle$ and $|\beta\rangle$ belong.

A whichever linear operator of the Hilbert's space can always be written using vectors which belong to the Hilbert space and its dual, namely it can be put in the form $|\alpha\rangle\langle\beta|$.

4.1 On Quantum Data

The atomic unit for computation in quantum devices is the qubit, which is traditionally represented [43] as a mathematical object which may assume both the classical values $\mathtt{tt}$ and $\mathtt{ff}$. Since in quantum scheme a qubit can't be separated from the quantum register $\mathscr{Q}$ in which it is stored, we will represent, for all practical purposes, this last one as vector of the Hilbert's space, in writing, according with the Dirac's notation

$$\mathscr{Q} = \alpha_{\mathtt{tt}}|r \leftarrow \mathtt{tt}\rangle + \alpha_{\mathtt{ff}}|r \leftarrow \mathtt{ff}\rangle, \tag{4.1}$$

where $\mathscr{Q}$ is a linear superposition of the couple of basis vectors $|r \leftarrow \mathtt{tt}\rangle$ and $|r \leftarrow \mathtt{ff}\rangle$ with complex coefficients $\alpha_{\mathtt{tt}}$ and $\alpha_{\mathtt{ff}}$, and r is a quantum variable name for the qubit.

Definitionally, one can think of quantum λ-calculi as a classical one, in which ordinary – classical – terms have access to the quantum register, which models quantum data. A quantum register $\mathscr{Q}$ on the set of quantum variables $\mathcal{Q}$ is patterned through a generalisation of the equation (4.1). Thus, it is mathematically described by a an element of a finite-dimensional Hilbert space whose computational basis is the set $\mathcal{SB}(\mathcal{Q})$ of all maps from $\mathcal{Q}$ to $\{\mathtt{tt}, \mathtt{ff}\}$ which attribute to each element r_i of the quantum variables set a boolean value which is hypothetically stored in the i-th qubit of $\mathscr{Q}$.

Using the Dirac's notation, any element of this basis, a ket of Hilbert's space, takes the form

$$|r_1 \leftarrow \mathtt{b}_1, \cdots, r_N \leftarrow \mathtt{b}_N\rangle, \tag{4.2}$$

where $\mathcal{Q} = \{r_1, \ldots, r_N\}$ and $\mathtt{b}_1, \ldots, \mathtt{b}_N \in \{\mathtt{tt}, \mathtt{ff}\}$. It is worth remarking that the order of the variables in the expression above is not essential, i.e., the configurations $|r_1 \leftarrow \mathtt{b}_1, \cdots, r_N \leftarrow \mathtt{b}_N\rangle$ and $|r_{\sigma(1)} \leftarrow \mathtt{b}_{\sigma(1)}, \cdots, r_{\sigma(N)} \leftarrow \mathtt{b}_{\sigma(N)}\rangle$ correspond to the same quantum register whenever σ is a permutation.

Quantum mechanics laws describe the state of a system as a *linear superposition*

of basis vectors whence elements of this Hilbert space, called $\mathcal{H}(\mathcal{Q})$, are in the form

$$\mathscr{Q} = \sum_{\eta \in \mathcal{SB}(\mathcal{Q})} \alpha_\eta |\eta\rangle, \tag{4.3}$$

where the index of the sum belongs to $\mathcal{SB}(\mathcal{Q})$, the space of all possible maps from the quantum variables set $\mathcal{Q}$ to boolean values, which are in number of $2^{|\mathcal{Q}|}$.

The complex numbers $\alpha_\eta \in \mathbb{C}$ are the so-called *amplitudes*, and must satisfy the *normalization condition* $\sum_{\eta \in \mathcal{SB}(\mathcal{Q})} |\alpha_\eta|^2 = 1$. If $\eta \in \mathcal{SB}(\mathcal{Q})$ and r is a variable not necessarily in $\mathcal{Q}$, then $\eta\{r \leftarrow \mathtt{b}\}$ stands for the substitution which coincides with η except on r where it equals $\mathtt{b}$.

The interaction of a quantum register with the outer environment can create or destroy quantum bits increasing or decreasing the dimension of $\mathscr{Q}$. This shaping of the quantum register is mathematically described making use of some operators:

- The probability operator $\mathtt{PR}^r_{\mathtt{b}} : \mathcal{H}(\mathcal{Q}) \to \mathbb{R}_{[0,1]}$ gives the probability to obtain $\mathtt{b} \in \{\mathtt{tt}, \mathtt{ff}\}$ as a result of the measurement of $r \in \mathcal{Q}$ in the input register:

  $$\mathtt{PR}^r_{\mathtt{b}}(\mathscr{Q}) = \sum_{\eta\{r \leftarrow \mathtt{b}\}} |\alpha_\eta|^2, \tag{4.4}$$

  where the sum is over the $2^{|\mathcal{Q}|-1}$–th dimensional set of those η such that the quantum variable r has the boolean value $\mathtt{b}$.

- If $r \notin \mathcal{Q}$, then the projection operator $\mathtt{MS}^r_{\mathtt{b}} : \mathcal{H}(\mathcal{Q} \cup \{r\}) \to \mathcal{H}(\mathcal{Q})$ measures the variable r, stored in the input register, destroying the corresponding qubit. More precisely $\mathtt{MS}^r_{\mathtt{tt}}(\mathscr{Q})$ and $\mathtt{MS}^r_{\mathtt{ff}}(\mathscr{Q})$ give as a result the quantum register configuration corresponding to a measure of the variable r, when the result of the variable measurement is $\mathtt{tt}$ or $\mathtt{ff}$, respectively:

  $$\mathtt{MS}^r_{\mathtt{b}}(\mathscr{Q}) = [\mathtt{PR}^r_{\mathtt{b}}(\mathscr{Q})]^{-\frac{1}{2}} \sum_{\eta} \alpha_{\eta\{r \leftarrow \mathtt{b}\}} |\eta\rangle, \tag{4.5}$$

  where $\mathscr{Q}$ is as in (4.3). A measurement of the variable r makes the quantum register collapsing over one between the new following states, both instances of (4.5):

  $$\mathscr{Q}_{\mathtt{tt}} = [\mathtt{PR}^r_{\mathtt{tt}}(\mathscr{Q})]^{-\frac{1}{2}} \sum_{\eta} \alpha_{\eta\{r \leftarrow \mathtt{tt}\}} |\eta\rangle, \qquad \mathscr{Q}_{\mathtt{ff}} = [\mathtt{PR}^r_{\mathtt{ff}}(\mathscr{Q})]^{-\frac{1}{2}} \sum_{\eta} \alpha_{\eta\{r \leftarrow \mathtt{ff}\}} |\eta\rangle.$$

- If $r \notin \mathcal{Q}$, then the operator $\mathtt{NW}^r_{\mathtt{b}} : \mathcal{H}(\mathcal{Q}) \to \mathcal{H}(\mathcal{Q} \cup \{r\})$ creates a new qubit, accessible through the variable name r, and increases by one the dimension of the quantum register.

Qubits can not only be created and measured, but their value can also be *modified* by applying unitary operators to them. Given any such n-ary operator U, and any sequence of distinct variables $r_1, \ldots, r_n$ (where $r_i \in \mathcal{Q}$ for every $1 \leq i \leq n$), one can build a unitary operator $U_{r_1,\ldots,r_n}$ on $\mathcal{H}(\mathcal{Q})$.

In the end we note that after a measurement all the α_η must rearrange in order that the new amplitudes α' can meet again the bound $\sum_{\eta \in \mathcal{SB}(\mathcal{Q})} |\alpha'_\eta|^2 = 1$, hence they are related to the old ones by the equality $\alpha'_{\eta\{r \leftarrow \mathtt{b}\}} = \dfrac{\alpha_\eta}{\sqrt{\mathtt{PR}^r_{\mathtt{b}}(\mathcal{Q})}}$.

4.1.1 The Language

We can obtain the quantum language ℓQST_λ as an extension of basic ℓST_λ. The grammar of ℓST_λ is enhanced by expanding the set $\mathcal{T}^{\ell ST_\lambda}$ in the following way:

$$v, u ::= x \mid \mathtt{tt} \mid \mathtt{ff} \mid \lambda x.e \mid \langle v, u \rangle \mid r \tag{4.6}$$

$$e ::= v \mid ef \mid \mathtt{if}\ e\ \mathtt{then}\ f\ \mathtt{else}\ g \mid \mathtt{let}\ e\ \mathtt{be}\ \langle x, y \rangle\ \mathtt{in}\ f \mid U\,(v) \mid \mathtt{meas}_n\,(v) \mid \mathtt{new}\,(v) \mid \mathtt{cmp}\,(v, v)\,, \tag{4.7}$$

where r ranges over an infinite set of quantum variables, and U ranges over a finite set of unitary transformations (each with an arity $\mathtt{a}(U)$) and n is a natural number. Terms $\mathtt{new}\,(v)$, $\mathtt{meas}_n\,(v)$, and $U\,(v)$ enrich the language ℓST_λ: $\mathtt{new}(v)$ takes as argument a boolean constant and returns (a quantum variable pointing to) a qubit of the same value, increasing this way the dimension of the quantum register. The measurement operator $\mathtt{meas}_n\,(v)$ measures the n-th quantum bit in a quantum register, therefore decreasing its dimension. Moreover, $U(v)$ is a formal way to represent a quantum gate, namely an atomic quantum algorithm which operates on a set of variables leaving unaltered the sum of probability amplitudes in a Hilbert's space spanned by the quantum variables set itself. If n is a positive natural number, the

expression $\langle r_1, \ldots, r_n \rangle$, called a *quantum variable sequence*, is syntactic sugar for the following term:

$$\langle r_1 \rangle = r_1;$$
$$\langle r_1, \ldots, r_{n+1} \rangle = \langle \langle r_1, \ldots, r_n \rangle, r_{n+1} \rangle.$$

Quantum variable sequences are denoted with metavariables like V, W. Given a quantum variable sequence $\mathsf{V} = \langle r_1, \ldots, r_n \rangle$ and m such that $1 \leq m \leq n$, the expression V_m indicates $\langle r_1, \ldots, r_{m-1}, r_{m+1}, \ldots, r_n \rangle$. Given two quantum variable sequences $\mathsf{V} = \langle r_1, \ldots, r_n \rangle$ and $\mathsf{W} = \langle s_1, \ldots, s_m \rangle$, the expression $\mathsf{V} \cdot \mathsf{W}$ is sometime used to indicate $\langle r_1, \ldots, r_n, s_1, \ldots, s_m \rangle$. The *length* n of a quantum variable sequence $\mathsf{V} = \langle r_1, \ldots, r_n \rangle$ is denoted as $\mid \mathsf{V} \mid$. The binary operator `cmp`, takes as arguments two quantum variable sequences V, W and gives $\mathsf{V} \cdot \mathsf{W}$ as a result.

This language is similar to that presented in [29], and it differs from the quantum language introduced by Selinger and Valiron [51] in this sense: that the quantum closures syntactically allowed in this language, whose terms can be typed using the typing rules 4.1, do not generally have entangled variables, being the quantum register of a term $\texttt{cnstr}\,(\{e_n\}_{n \in \mathcal{N}})$ the tensor product of the quantum register of each subterm. The quantum entanglement is treated apart introducing the syntactic construct $\texttt{cmp}\,((, v)\, u)$, which can create sequences of qubits which are allowed for the entanglement. Specifically, in a pair of type $\texttt{qbit} \otimes \texttt{qbit}$, each one the components of the pair, can access only to its own part of the quantum register, while this doesn't happen in a term of type $\texttt{qbit}^2$.

This choice is motivated by the difficulty to correctly implement the general structure of the Howe's rules in the quantum environment if the subterms don't have unentangled subregisters.

The class of types needs to be sligthly extended with a new base type called $\texttt{qbit}^n$ valid for quantum registers, namely for quantum variables and quantum variable sequences, thus

$$\mathcal{Y} ::= \texttt{bool} \;\Big|\; \texttt{qbit}^n \;\Big|\; B \multimap A \;\Big|\; A \otimes B. \tag{4.8}$$

Type judgement quantum closure rule	Name
$\dfrac{}{\vdash [\emptyset, \mathtt{b}] : \mathtt{bool}}$	$(tj_{\mathtt{Q}} - con)$
$\dfrac{}{x : A \vdash [\emptyset, x] : A}$	$(tj_{\mathtt{Q}} - var)$
$\dfrac{\mathscr{Q} \in \mathcal{H}(\{r_1 \dots r_n\})}{\vdash [\mathscr{Q}, \langle r_1, \dots, r_n \rangle] : \mathtt{qbit}^n}$	$(tj_{\mathtt{Q}} - ser)$
$\dfrac{\Gamma, x : A \vdash [\mathscr{Q}, e] : B}{\Gamma \vdash [\mathscr{Q}, \lambda x.e] : A \multimap B}$	$(tj_{\mathtt{Q}} - abs)$
$\dfrac{\Gamma \vdash [\mathscr{Q}, e] : B \multimap A \qquad \Delta \vdash [\mathscr{U}, f] : B}{\Gamma, \Delta \vdash [\mathscr{Q} \otimes \mathscr{U}, ef] : A}$	$(tj_{\mathtt{Q}} - app)$
$\dfrac{\Gamma \vdash [\mathscr{Q}, e] : \mathtt{bool} \qquad \Delta \vdash [\mathscr{U}, f] : A \qquad \Delta \vdash [\mathscr{U}, g] : A}{\Gamma, \Delta \vdash [\mathscr{Q} \otimes \mathscr{U}, \mathtt{if}\ e\ \mathtt{then}\ f\ \mathtt{else}\ g] : A}$	$(tj_{\mathtt{Q}} - if)$
$\dfrac{\Gamma \vdash [\mathscr{Q}, v] : A \qquad \Delta \vdash [\mathscr{U}, u] : B}{\Gamma, \Delta \vdash [\mathscr{Q} \otimes \mathscr{U}, \langle v, u \rangle] : A \otimes B}$	$(tj_{\mathtt{Q}} - pai)$
$\dfrac{\Gamma \vdash [\mathscr{Q}, e] : E \otimes F \qquad \Delta, x : A, y : B \vdash [\mathscr{U}, f] : A}{\Gamma, \Delta \vdash [\mathscr{Q} \otimes \mathscr{U}, \mathtt{let}\ e\ \mathtt{be}\ \langle x, y \rangle\ \mathtt{in}\ f] : A}$	$(tj_{\mathtt{Q}} - let)$
$\dfrac{\Gamma \vdash [\mathscr{Q}, v] : \mathtt{bool}}{\Gamma \vdash [\mathscr{Q}, \mathtt{new}\,(v)] : \mathtt{qbit}}$	$(tj_{\mathtt{Q}} - new)$
$\dfrac{\Gamma \vdash [\mathscr{Q}, v] : \mathtt{qbit}^{\mathtt{a}(U)}}{\Gamma \vdash [\mathscr{Q}, U\,(v)] : \mathtt{qbit}^{\mathtt{a}(U)}}$	$(tj_{\mathtt{Q}} - uni)$
$\dfrac{\Gamma \vdash [\mathscr{Q}, v] : \mathtt{qbit}^1}{\Gamma \vdash [\mathscr{Q}, \mathtt{meas}_1\,(v)] : \mathtt{bool}}$	$(tj_{\mathtt{Q}} - mea)$
$\dfrac{\Gamma \vdash [\mathscr{Q}, v] : \mathtt{qbit}^{n+1} \qquad 1 \leq m \leq n}{\Gamma \vdash [\mathscr{Q}, \mathtt{meas}_m\,(v)] : \mathtt{qbit}^n \otimes \mathtt{bool}}$	$(tj_{\mathtt{Q}} - mea)$
$\dfrac{\Gamma \vdash [\mathscr{Q}, v] : \mathtt{qbit}^n \qquad \Delta \vdash [\mathscr{U}, u] : \mathtt{qbit}^m}{\Gamma \vdash [\mathscr{Q} \otimes \mathscr{U}, \mathtt{cmp}\,(v, u)] : \mathtt{qbit}^{n+m}}$	$(tj_{\mathtt{Q}} - cmp)$
$\dfrac{}{\vdash [\emptyset, \Omega] : A}$	$(tj_{\mathtt{Q}} - div)$

Figure 4.1: Typing rules in ℓQST_λ: the symbol $\emptyset$ denotes the empty quantum register.

Since terms only make sense, computationally, only if they are coupled with a quantum register, it is necessary to give the definition of *quantum closure* which is an element $(\mathscr{Q}, e)$ of the set $\mathcal{H}(\mathcal{Q}) \times \mathcal{T}_{\Gamma,A}^{\ell QST_\lambda}$, where $\mathcal{Q}$ is a suitable set of quantum

variables, such that $dom(\Gamma) \cap \mathcal{Q} = \emptyset$. We use the notation $[\mathscr{Q}, e]$ to denote a generic quantum closure. In Figure 4.1 the system of typing rules for $[\mathscr{Q}, e]$ within the language ℓQST_λ is given. Among the set of the quantum closures, two subsets are particulary meaningful, namely the *total* quantum closures, which fulfill the condition $\mathscr{Q} \in \mathcal{H}(\mathcal{Q})$, where $\mathcal{Q}$ is precisely the set of free quantum varables of e and the *closed* quantum closures, such that $dom(\Gamma) = \mathcal{Q} = \emptyset$.

A total and closed quantum closure is called a quantum program of ℓQST_λ.

To correctly extend Howe's techniques to the quantum environment, we need to avoid that, in the closures of the language ℓQST_λ, the quantum variables belonging to parts of the quantum register which refer to different subterms mix up, giving rise to the so called *quantum entanglement*. The only exception to this general rule occurs through the use of the special operator $\mathtt{cmp}\,(\mathsf{V}, \mathsf{W})$, which implements the operation of quantum entanglement between two quantum sequences. This strong separation inherent to the quantum registers belonging to different subterms is highlighted through the set of typing rules listed in Figure 4.1.

The semantics of ℓQST_λ is a binary relation on quantum closures: analogously to what has been made for ℓPST_λ, small step reduction operator $\rightarrow$ and the big step evaluation operator $\Downarrow$ are given as relations between the set of quantum closures – which must be correctly typed using a derivation tree based on the rules given in Figure 4.1 – and the set of quantum closures distributions. In Figures 4.2 and 4.3, we display the one–step semantics and big–step semantics for ℓQST_λ. Symbols as $\mathcal{T}_A^{[\mathtt{QC}]\,\ell QST_\lambda}$ and $\mathcal{V}_A^{[\mathtt{QC}]\,\ell QST_\lambda}$, will be employed to denote the extensions of $\mathcal{T}_A$ and $\mathcal{V}_A$ to the quantum closures set.

In ℓQST_λ, the property of *substitutivity* for the relation R implies the fullfillement of the following condition between pairs of quantum closures

$$\Gamma, x : B \vdash [\mathscr{Q}, e]\ R\ [\mathscr{W}, g] : A \wedge \Delta \vdash [\mathscr{U}, f]\ R\ [\mathscr{R}, h] : B \Rightarrow \Gamma, \Delta \vdash [\mathscr{Q} \otimes \mathscr{U}, e\{f/x\}]\ R\ [\mathscr{W} \otimes \mathscr{R}, g\{h/x\}] : A; \quad (4.9)$$

The the following Lemma 4.1 shows how to deal with the substitutions between quantum closures.

One–step semantics rule	Name
$\dfrac{}{[\mathscr{Q}, (\lambda x.e)v] \to \{[\mathscr{Q}, e\{v/x\}]^1\}}$	$(app_\beta)_{\mathtt{Q}}$
$\dfrac{[\mathscr{Q}, e] \to \{[\mathscr{Q}_i, v_i]^{p_i}\}_{i\in\mathcal{I}}}{[\mathscr{Q}\otimes\mathscr{U}, ef] \to \{[\mathscr{Q}_i\otimes\mathscr{U}, v_i f]^{p_i}\}_{i\in\mathcal{I}}}$	$(app_L)_{\mathtt{Q}}$
$\dfrac{[\mathscr{Q}, f] \to \{[\mathscr{Q}_i, \ell_i]^{p_i}\}_{i\in\mathcal{I}}}{[\mathscr{Q}\otimes\mathscr{U}, vf] \to \{[\mathscr{Q}_i\otimes\mathscr{U}, v\ell_i]^{p_i}\}_{i\in\mathcal{I}}}$	$(app_R)_{\mathtt{Q}}$
$\dfrac{}{[\mathscr{Q}, \mathtt{if\ tt\ then}\ f\ \mathtt{else}\ g] \to \{[\mathscr{Q}, f]^1\}}$	$(if - ax_{\mathtt{tt}})_{\mathtt{Q}}$
$\dfrac{}{[\mathscr{Q}, \mathtt{if\ ff\ then}\ f\ \mathtt{else}\ g] \to \{[\mathscr{Q}, g]^1\}}$	$(if - ax_{\mathtt{ff}})_{\mathtt{Q}}$
$\dfrac{[\mathscr{Q}, e] \to \{[\mathscr{Q}_i, h_i]^{p_i}\}_{i\in\mathcal{I}}}{[\mathscr{Q}\otimes\mathscr{U}, \mathtt{if}\ e\ \mathtt{then}\ f\ \mathtt{else}\ g] \to \{[\mathscr{Q}_i\otimes\mathscr{U}, \mathtt{if}\ h_i\ \mathtt{then}\ f\ \mathtt{else}\ g]^{p_i}\}_{i\in\mathcal{I}}}$	$(if)_{\mathtt{Q}}$
$\dfrac{}{[\mathscr{Q}, \mathtt{let}\ \langle v, u\rangle\ \mathtt{be}\ \langle x, y\rangle\ \mathtt{in}\ f] \to \{[\mathscr{Q}, f\{v/x, u/y\}]^1\}}$	$(let - ax)_{\mathtt{Q}}$
$\dfrac{[\mathscr{Q}, e] \to \{[\mathscr{Q}_i, h_i]^{p_i}\}_{i\in\mathcal{I}}}{[\mathscr{Q}\otimes\mathscr{U}, \mathtt{let}\ e\ \mathtt{be}\ \langle x, y\rangle\ \mathtt{in}\ g] \to \{[\mathscr{Q}_i\otimes\mathscr{U}, \mathtt{let}\ h_i\ \mathtt{be}\ \langle x, y\rangle\ \mathtt{in}\ g]^{p_i}\}_{i\in\mathcal{I}}}$	$(let)_{\mathtt{Q}}$
$\dfrac{1 \le m \le n}{[\mathscr{Q}, \mathtt{meas}_m\,(\mathsf{V})] \to \{[\mathtt{MS}^r_{\mathtt{ff}}(\mathscr{Q}), \langle \mathsf{V}_m, \mathtt{ff}\rangle]^{\mathtt{PR}^r_{\mathtt{ff}}(\mathscr{Q})}, [\mathtt{MS}^r_{\mathtt{tt}}(\mathscr{Q}), \langle \mathsf{V}_m, \mathtt{tt}\rangle]^{\mathtt{PR}^r_{\mathtt{tt}}(\mathscr{Q})}\}}$	$(mea)_{\mathtt{Q}}$
$\dfrac{}{[\mathscr{Q}, \mathtt{cmp}\,(\mathsf{V}, \mathsf{W})] \to \{[\mathscr{Q}, \mathsf{V}\cdot\mathsf{W}]^1\}}$	$(cmp)_{\mathtt{Q}}$
$\dfrac{}{[\mathscr{Q}, \mathtt{new(b)}] \to \{[\mathtt{NW}^r_{\mathtt{b}}(\mathscr{Q}), r]^1\}}$	$(new)_{\mathtt{Q}}$
$\dfrac{}{[\mathscr{Q}, U\,\langle r_1, \ldots r_n\rangle] \to \{[U_{r_1,\ldots r_n}\mathscr{Q}, \langle r_1, \ldots r_n\rangle]^1\}}$	$(uni)_{\mathtt{Q}}$
$\dfrac{}{[\mathscr{Q}, \Omega] \to \emptyset}$	$(div)_{\mathtt{Q}}$

Figure 4.2: One–step semantics of ℓQST_λ.

Lemma 4.1 (Substitutivity in ℓQST_λ)**.** *If* $[\mathscr{Q}, e] \in \mathcal{T}^{[\mathtt{QC}]\,\ell QST_\lambda}_{\Gamma, z:E, A}$ *and* $[\mathscr{U}, u] \in \mathcal{V}^{[\mathtt{QC}]\,\ell QST_\lambda}_{\Delta, E}$ *are two quantum closures, correctly typeable through the rules of Figure 4.4, then it holds that*

$$\Gamma, \Delta \vdash [\mathscr{Q}\otimes\mathscr{U}, e\{u/x\}] : A. \tag{4.10}$$

Proof. The proof is by induction on the structure of e, examining the typing rules of figure Figure 4.4.

The cases $\emptyset \vdash [\emptyset, \mathtt{b}] : \mathtt{bool}$ and $\emptyset \vdash [\mathscr{Q}, \langle r_1 \ldots, r_n \rangle] : \mathtt{qbit}^n$ are impossible since here the context lacks and there aren't free variables.

If $[\mathscr{Q}, e] = [\emptyset, x]$, then from linearity it follows that $\Gamma = \emptyset$, $A = B$ and $x = z$. Thus the relation $\Delta \vdash [\emptyset \otimes \mathscr{U}, x\{u/x\}] : A$ is true since it is equivalent to the hypothesis $\Delta \vdash [\mathscr{U}, u] : A$

If $[\mathscr{Q}, e] = [\mathscr{Q}, \lambda\, x.f]$ we must prove the assertion

$$\Gamma, z : E \vdash [\mathscr{Q}, \lambda\, x.f] : B \multimap A \wedge \vdash [\mathscr{U}, u] : E \Rightarrow \Gamma \vdash [\mathscr{Q} \otimes \mathscr{U}, \lambda\, x.f\{u/z\}] : B \multimap A. \quad (4.11)$$

Since the first hypothesis is a consequence of a $(tj - abs)_{\mathtt{Q}}$ rule of Figure 4.4 whose premise is $\Gamma, x : B, z : E \vdash [\mathscr{Q}, f] : A$, the induction hypothesis on the open term $[\mathscr{Q}, f]$ gives immediately $\Gamma, x : B \vdash [\mathscr{Q}, f\{u/z\}] : A$. Thus we can take this type judgement as the premise fo the rule $(tj - abs)_{\mathtt{Q}}$, getting the desired result.

If $[\mathscr{Q}, e] = [\mathscr{Q}, \mathtt{new}\,(v)]$, then we must prove that

$$\Gamma, z : E \vdash [\mathscr{Q}, \mathtt{new}\,(v)] : \mathtt{qbit}^1 \wedge \vdash [\mathscr{U}, u] : E \Rightarrow \Gamma \vdash [\mathscr{Q} \otimes \mathscr{U}, \mathtt{new}\,(v\{u/z\})] : \mathtt{qbit}^1. \quad (4.12)$$

Using the typing rule $(tj - new)_{\mathtt{Q}}$ in Figure 4.4, we find that the first hypothesis in (4.12) is a consequence of the premise $\Gamma \vdash [\mathscr{Q}, v] : \mathtt{bool}$, over which we can apply the induction hypothesis, which gives $\Gamma \vdash [\mathscr{Q}, v\{u/z\}] : \mathtt{bool}$. This result can be taken of premise for the rule $(tj - new)_{\mathtt{Q}}$ leading to the thesis. The cases $\emptyset \vdash [\mathscr{Q}, \mathtt{meas}\,(v)] : \mathtt{bool}$, $\emptyset \vdash [\mathscr{Q}, \mathtt{meas}_m\,(v)] : \mathtt{qbit}^n \otimes \mathtt{bool}$ and $\emptyset \vdash [\mathscr{Q}, U\,(v)] : \mathtt{qbit}^{\mathtt{a}(U)}$ are similar to $[\mathscr{Q}, \mathtt{new}\,(v)]$ and $\lambda\, x.f$.

If $[\mathscr{Q}, e] = [\mathscr{Q}, \mathtt{cnstr}\,(\{f_n\}_{n \in \mathcal{N}})]$, where $\mathtt{cnstr}$ is some binary o ternary constructor of the language, the statement to prove is

$$\Gamma, z : E \vdash [\mathscr{Q}, \mathtt{cnstr}\,(f_1 \ldots f_N)_{1 \ldots N}] : A \wedge \vdash [\mathscr{U}, u] : E \Rightarrow$$

$$\Gamma \vdash [\mathscr{Q} \otimes \mathscr{U}, \texttt{cnstr}\,((f_1 \dots f_N)\{v/z\})_{1\dots N}] : A. \quad (4.13)$$

The first typing judgement is the result of the application of a general typing rule as

$$\frac{\Gamma_1 \vdash [\mathscr{Q}_1, f_1] : A_1 \dots \Gamma_j, z : E \vdash [\mathscr{Q}_j, f_j] : A_j \dots \Gamma_N \vdash [\mathscr{Q}_N, f_N] : A_N \qquad \vdash [\mathscr{U}, u] : E}{\Gamma, z : E \vdash \left[\mathscr{Q}, \texttt{cnstr}\,(f_1 \dots f_j \dots f_N)\right] : A} \quad (4.14)$$

where $\mathscr{Q} = \mathscr{Q}_1 \otimes \dots \otimes \mathscr{Q}_N$ and, due to the linearity of ℓQST_λ, only one among the smaller terms $f_1 \dots f_N$ owns z as free variable within its typing context. Thus exploiting the induction hypothesis on this term, which is f_j, we find

$$\Gamma_j, z : E \vdash [\mathscr{Q}_j, f_j] : A_j \wedge \vdash [\mathscr{U}, u] : E \Rightarrow \Gamma_j \vdash [\mathscr{Q}_j \otimes \mathscr{U}, f_j\{u/z\}] : A_j, \quad (4.15)$$

and taking the conclusion of the implication (4.15) as premise in (4.14), gives the thesis (4.13). □

Lemma 4.2 (Subject reduction in ℓQST_λ)**.** *If a quantum closure* $\vdash [\mathscr{Q}, e] : A$ *evaluates to a distribution* $[\mathscr{Q}, e] \Downarrow \{[\mathscr{Q}_i, v_i]^{p_i}\}_{i \in \mathcal{I}}$*, then it holds,* $\forall i \in \mathcal{I}$*, the type judgement* $\vdash [\mathscr{Q}_i, v_i] : A$*.*

Proof.

If the quantum closure is $\vdash [\emptyset, \texttt{b}] : \texttt{bool}$ or $x : A \vdash [\emptyset, x] : A$ or $\vdash [\mathscr{Q}, \langle r_1 \dots r_n \rangle] : \texttt{qbit}^n$ or $\vdash [\mathscr{Q}, \lambda x.e] : B \multimap A$, then using the reduction rule $(val \Downarrow)_{\texttt{Q}}$ we obtain a distribution with a unique value, which is the quantum closure that we start from. Thus the thesis coincides with the hypothesis.

If the quantum closure is $\Gamma \vdash [\mathscr{Q}, \texttt{new}\,(v)] : \texttt{qbit}^1$ then the type of the unique value distribution follows from the structure of the function $\texttt{new}(v)$ and by the rule $(new \Downarrow)_{\texttt{Q}}$. The same remark we must do for quantum closures such as $\Gamma \vdash [\mathscr{Q}, \texttt{meas}_m\,(\mathsf{V})] : \texttt{qbit}^n \otimes \texttt{bool}$, $\Gamma \vdash [\mathscr{Q}, U\,(v)] : \texttt{qbit}^{\texttt{a}(U)}$, $\Gamma \vdash [\mathscr{Q}, \texttt{cmp}\,(v, u)] : \texttt{qbit}^{n+m}$, since the correspondent big–step reduction rules $(mea \Downarrow)_{\texttt{Q}}$, $(uni \Downarrow)_{\texttt{Q}}$ and $(cmp \Downarrow)_{\texttt{Q}}$ ensure that the type of these terms do not change during the evaluation.

Big–step semantics rule	Name
$\dfrac{}{[\mathscr{Q}, v] \Downarrow \{[\mathscr{Q}, v]^1\}}$	$(val \Downarrow)_{\mathtt{Q}}$
$\dfrac{r \text{ fresh variable}}{[\mathscr{Q}, \mathtt{new(b)}] \Downarrow \{[\mathtt{NW}^r_{\mathtt{b}}(\mathscr{Q}), r]^1\}}$	$(new \Downarrow)_{\mathtt{Q}}$
$\dfrac{}{[\mathscr{Q}, U\langle r_1 \dots r_m\rangle] \Downarrow \{[U_{r_1 \dots r_m}\mathscr{Q}, \langle r_1 \dots r_m\rangle]^1\}}$	$(uni \Downarrow)_{\mathtt{Q}}$
$\dfrac{}{[\mathscr{Q}, \mathtt{cmp}\,(\mathsf{V}, \mathsf{W})] \Downarrow \{[\mathscr{Q}, \mathsf{V} \cdot \mathsf{W}]^1\}}$	$(cmp \Downarrow)_{\mathtt{Q}}$
$\dfrac{1 \le m \le n}{[\mathscr{Q}, \mathtt{meas}_m\,(\mathsf{V})] \Downarrow \{[\mathtt{MS}^r_{\mathtt{ff}}(\mathscr{Q}), \langle \mathsf{V}_m, \mathtt{ff}\rangle]^{\mathtt{PR}^r_{\mathtt{ff}}(\mathscr{Q})}, [\mathtt{MS}^r_{\mathtt{tt}}(\mathscr{Q}), \langle \mathsf{V}_m, \mathtt{tt}\rangle]^{\mathtt{PR}^r_{\mathtt{tt}}(\mathscr{Q})}\}}$	$(mea \Downarrow)_{\mathtt{Q}}$
$\dfrac{\begin{array}{l}[\mathscr{Q}, e] \Downarrow \{[\mathscr{Q}_i, \lambda x.h_i]^{p_i}\}_{i\in\mathcal{I}} \\ [\mathscr{Q}_i \otimes \mathscr{U}, f] \Downarrow \{[\mathscr{Q}_i \otimes \mathscr{U}_j, u_j]^{q_j}\}_{j\in\mathcal{J}} \qquad [\mathscr{Q}_i \otimes \mathscr{U}_j, h_i\{u_j/x\}] \Downarrow \mathscr{E}_{i,j}\end{array}}{[\mathscr{Q} \otimes \mathscr{U}, ef] \Downarrow \sum_{i,j} p_i \cdot q_j \cdot \mathscr{E}_{i,j}}$	$(app \Downarrow)_{\mathtt{Q}}$
$\dfrac{\begin{array}{ll}[\mathscr{Q}, e] \Downarrow \{[\mathscr{Q}_{\mathtt{ff}}, \mathtt{ff}]^{p_{\mathtt{ff}}}, [\mathscr{Q}_{\mathtt{tt}}, \mathtt{tt}]^{p_{\mathtt{tt}}}\} & \begin{array}{l}[\mathscr{Q}_{\mathtt{tt}} \otimes \mathscr{U}, f] \Downarrow \{[\mathscr{U}_{\mathtt{tt}} \otimes \mathscr{U}_i, v_i]^{p_i}\}_{i\in\mathcal{I}} \\ [\mathscr{Q}_{\mathtt{ff}} \otimes \mathscr{U}, g] \Downarrow \{[\mathscr{Q}_{\mathtt{ff}} \otimes \mathscr{U}_j, v_j]^{q_j}\}_{j\in\mathcal{J}}\end{array}\end{array}}{[\mathscr{Q} \otimes \mathscr{U}, \mathtt{if}\ e\ \mathtt{then}\ f\ \mathtt{else}\ g] \Downarrow \{[\mathscr{Q}_{\mathtt{tt}} \otimes \mathscr{U}_i, v_i]^{p_{\mathtt{tt}} \cdot p_i}\}_{i\in\mathcal{I}} + \{[\mathscr{Q}_{\mathtt{tt}} \otimes \mathscr{U}_j, u_j]^{p_{\mathtt{tt}} \cdot q_j}\}_{j\in\mathcal{J}}}$	$(if \Downarrow)_{\mathtt{Q}}$
$\dfrac{[\mathscr{Q}, e] \Downarrow \{[\mathscr{Q}_i, \langle v_i, u_i\rangle]^{p_i}\}_{i\in\mathcal{I}} \qquad \forall i,\ [\mathscr{Q}_i \otimes \mathscr{U}, f\{v_i/x, u_i/y\}] \Downarrow \mathscr{E}_i}{[\mathscr{Q} \otimes \mathscr{U}, \mathtt{let}\ e\ \mathtt{be}\ \langle x, y\rangle\ \mathtt{in}\ f] \Downarrow \sum_i p_i \cdot \mathscr{E}_i}$	$(let \Downarrow)_{\mathtt{Q}}$
$\dfrac{}{[\mathscr{Q}, \Omega] \Downarrow \emptyset}$	$(div \Downarrow)_{\mathtt{Q}}$

Figure 4.3: Big–step semantics of ℓQST_λ.

If $\Gamma \vdash [\mathscr{U} \otimes \mathscr{W}, fg] : A$ and $[\mathscr{U} \otimes \mathscr{W}, fg] \Downarrow \{[\mathscr{U}_i \otimes \mathscr{W}_j, v_{i,j,n}]^{p_i \cdot q_j \cdot r_n}\}_{i\in\mathcal{I}, j\in\mathcal{J}, n\in\mathcal{N}}$ then we must show that the type judgement

$$\forall i \in \mathcal{I}, \forall j \in \mathcal{J}, \forall n \in \mathcal{N}, \Gamma \vdash [\mathscr{U}_i \otimes \mathscr{W}_j, v_{i,j,n}] : A \tag{4.16}$$

is valid. The first hypothesis is a type judgement which, for the rule $(tj - app)_{\mathtt{Q}}$ has premises

$$\Gamma \vdash [\mathscr{U}, f] : B \multimap A \ \wedge\ \emptyset \vdash [\mathscr{W}, g] : B \tag{4.17}$$

while the second hypothesis comes from the $(app \Downarrow)_{\mathtt{Q}}$ rule, which has three premises

$$[\mathscr{U}, f] \Downarrow \{[\mathscr{U}_i, \lambda x.h_i]^{p_i}\}_{i\in\mathcal{I}} \wedge \forall i,\ [\mathscr{U}_i \otimes \mathscr{W}, g] \Downarrow \{[\mathscr{U}_i \otimes \mathscr{W}_j, u_j]^{q_j}\}_{j\in\mathcal{J}} \wedge\ [\mathscr{U}_i \otimes \mathscr{W}_j, h_i\{u_j/x\}] \Downarrow \{[\mathscr{U}_i \otimes \mathscr{W}_j, v_{i,j,n}]^{r_n}\}_{n\in\mathcal{N}}. \tag{4.18}$$

The induction hypothesis on the smaller terms in relation (4.17), entails the following type judgements which hold for the distributions of (4.18):

$$\Gamma \vdash [\mathscr{U}, f] : B \multimap A \ \wedge\ [\mathscr{U}, f] \Downarrow \{[\mathscr{U}_i, \lambda\, x.h_i]^{p_i}\}_{i\in\mathcal{I}} \Rightarrow \forall i \in \mathcal{I},\, \Gamma \vdash [\mathscr{U}_i, \lambda\, x.h_i] : B \multimap A \tag{4.19a}$$

$$\emptyset \vdash [\mathscr{W}, g] : B \wedge\ , [\mathscr{U}_i \otimes \mathscr{W}, g] \Downarrow \{[\mathscr{U}_i \otimes \mathscr{W}_j, u_j]^{q_j}\}_{j\in\mathcal{J}} \Rightarrow \forall j \in \mathcal{J}, \emptyset \vdash [\mathscr{U}_i \otimes \mathscr{W}_j, u_j] : B. \tag{4.19b}$$

and since the relation (4.19a) derives from a type judgement rule $(tj - abs)_{\mathtt{Q}}$, we have

$$\forall i \in \mathcal{I},\, \Gamma, x : B \vdash [\mathscr{U}_i, h_i] : A. \tag{4.20}$$

Thus, using Lemma 4.1 on the statements (4.20) and (4.19b) one finds

$$\forall i \in \mathcal{I},\, \forall j \in \mathcal{J},\, \Gamma, x : B \vdash [\mathscr{U}_i, h_i] : A \wedge \emptyset \vdash [\mathscr{U}_i \otimes \mathscr{W}_j, u_j] : B \ \Rightarrow\ \Gamma \vdash [\mathscr{U}_i \otimes \mathscr{W}_j, h_i\{u_j/x\}] : A. \tag{4.21}$$

With this last result, applying induction hypothesis to the last term of (4.18) we get the thesis, since

$$\Gamma \vdash [\mathscr{U}_i \otimes \mathscr{W}_j, h_i\{u_j/x\}] : A \wedge [\mathscr{U}_i \otimes \mathscr{W}_j, h_i\{u_j/x\}] \Downarrow \{[\mathscr{U}_i \otimes \mathscr{W}_j, v_{i,j,n}]^{r_n}\}_{n\in\mathcal{N}} \Rightarrow \forall i \in \mathcal{I}, \forall j \in \mathcal{J}, \forall n \in \mathcal{N},\ \Gamma \vdash [\mathscr{U}_i \otimes \mathscr{W}_j, v_{i,j,n}] : A. \tag{4.22}$$

If $\Gamma \vdash [\mathscr{U} \otimes \mathscr{W}, \mathtt{if}\ f\ \mathtt{then}\ g\ \mathtt{else}\ h] : A \wedge\ [\mathscr{U} \otimes \mathscr{W}, \mathtt{if}\ f\ \mathtt{then}\ g\ \mathtt{else}\ h] \Downarrow$ $\{[\mathscr{U}_{\mathtt{tt}} \otimes \mathscr{W}_i, \nu_i]^{p_i \cdot q_{\mathtt{tt}}}, [\mathscr{U}_{\mathtt{ff}} \otimes \mathscr{W}_j, w_j]^{p_j \cdot q_{\mathtt{ff}}}\}_{i\in\mathcal{I}, j\in\mathcal{J}}$ then we should prove the validity of the type judgements $\forall i \in \mathcal{I}, \forall j \in \mathcal{J}$, $\Gamma \vdash [\mathscr{U}_{\mathtt{tt}} \otimes \mathscr{W}_i, \nu_i] : A$, $\Gamma \vdash [\mathscr{U}_{\mathtt{ff}} \otimes \mathscr{W}_j, w_j] : A$. Since the first hypothesis is a type judgement coming from the rule $(tj - if)_{\mathtt{Q}}$ which has premises

$$\Gamma_1 \vdash [\mathscr{U}, f] : \mathtt{bool} \wedge \Gamma_2 \vdash [\mathscr{W}, g] : A \wedge \Gamma_2 \vdash [\mathscr{W}, h] : A \tag{4.23}$$

while the second hypothesis in the if statement, derives from the $(if \Downarrow)_{\mathtt{Q}}$ rule, with premises

$$[\mathscr{U}, f] \Downarrow \{[\mathscr{U}_{\mathtt{tt}}, \mathtt{tt}]^{q_{\mathtt{tt}}}, [\mathscr{U}_{\mathtt{ff}}, \mathtt{ff}]^{q_{\mathtt{ff}}}\} \wedge \forall i \in \mathcal{I},\ [\mathscr{U}_{\mathtt{tt}} \otimes \mathscr{W}, g] \Downarrow \{[\mathscr{U}_{\mathtt{tt}} \otimes \mathscr{W}_i, \nu_i]^{p_j}\}_{i \in \mathcal{I}} \wedge \forall j \in \mathcal{J},\ [\mathscr{U}_{\mathtt{ff}} \otimes \mathscr{W}, h] \Downarrow \{[\mathscr{U}_{\mathtt{ff}} \otimes \mathscr{W}_j, w_j]^{p_j}\}_{j \in \mathcal{J}}. \quad (4.24)$$

Here, the induction hypothesis on the smaller terms in relation (4.23), leads directly to the following type judgements which hold for the distributions of (4.24):

$$\Gamma_1 \vdash [\mathscr{U}, f] : \mathtt{bool} \wedge [\mathscr{U}, f] \Downarrow \{[\mathscr{U}_{\mathtt{tt}}, \mathtt{tt}]^{q_{\mathtt{tt}}}, [\mathscr{U}_{\mathtt{ff}}, \mathtt{ff}]^{q_{\mathtt{ff}}}\} \Rightarrow \Gamma_1 \vdash [\mathscr{U}_{\mathtt{tt}} \otimes \mathscr{W}, \mathtt{tt}] : \mathtt{bool}, \Gamma_1 \vdash [\mathscr{U}_{\mathtt{ff}} \otimes \mathscr{W}, \mathtt{ff}] : \mathtt{bool} \quad (4.25a)$$

$$\Gamma_2 \vdash [\mathscr{W}, g] : A \wedge \forall i \in \mathcal{I}, \quad [\mathscr{U}_{\mathtt{tt}} \otimes \mathscr{W}, g] \Downarrow \{[\mathscr{U}_{\mathtt{tt}} \otimes \mathscr{W}_i, \nu_i]^{p_j}\}_{i \in \mathcal{I}} \Rightarrow \forall i \in \mathcal{I},\ \Gamma_2 \vdash [\mathscr{U}_{\mathtt{tt}} \otimes \mathscr{W}_i, \nu_i] : A \quad (4.25b)$$

$$\Gamma_2 \vdash [\mathscr{W}, h] : A \wedge \forall j \in \mathcal{J},\ [\mathscr{U}_{\mathtt{ff}} \otimes \mathscr{W}, h] \Downarrow \{[\mathscr{U}_{\mathtt{ff}} \otimes \mathscr{W}_j, w_j]^{p_j}\}_{j \in \mathcal{J}} \Rightarrow \forall j \in \mathcal{J}, \Gamma_2 \vdash [\mathscr{U}_{\mathtt{ff}} \otimes \mathscr{W}_j, w_j] : A, \quad (4.25c)$$

which is the thesis.

If $\Gamma \vdash [\mathscr{U} \otimes \mathscr{W}, \mathtt{let}\ f\ \mathtt{be}\ \langle x, y\rangle\ \mathtt{in}\ g] : A$ and $[\mathscr{U} \otimes \mathscr{W}, \mathtt{let}\ f\ \mathtt{be}\ \langle x, y\rangle\ \mathtt{in}\ g] \Downarrow \{[\mathscr{U}_i \otimes \mathscr{W}_j, v_{i,j}]^{p_i q_j}\}_{i \in \mathcal{I}, j \in \mathcal{J}}$ then we must show the goodness of the type judgement

$$\forall i \in \mathcal{I}, \forall j \in \mathcal{J}, \Gamma \vdash [\mathscr{U}_i \otimes \mathscr{W}_j, v_{i,j}] : A. \quad (4.26)$$

The first hypothesis is a type judgement which, for the rule $(tj - let)_{\mathtt{Q}}$ has premises

$$\emptyset \vdash [\mathscr{U}, f] : E \otimes F \wedge \Gamma, x : E, y : F \vdash [\mathscr{W}, g] : A \quad (4.27)$$

while the second hypothesis comes from the $(let \Downarrow)_{\mathtt{Q}}$ rule, whose premises are

$$[\mathscr{U}, f] \Downarrow \{[\mathscr{U}_i, \langle u_i, \nu_i\rangle]^{p_i}\}_{i \in \mathcal{I}} \wedge \forall i \in \mathcal{I},\ [\mathscr{U}_i \otimes \mathscr{W}, g\{u_i/x, \nu_i/y\}] \Downarrow \{[\mathscr{U}_i \otimes \mathscr{W}_j, v_{i,j}]^{q_j}\}_{j \in \mathcal{J}}. \quad (4.28)$$

The substitution Lemma 4.1 ensures on the validity of the type judgement

$$\forall i \in \mathcal{I},\ \Gamma, x : E, y : F \vdash [\mathscr{W}, g]\, A : \wedge \emptyset \vdash [\mathscr{U}_i, \langle u_i, \nu_i\rangle] : E \otimes F \ \Rightarrow \Gamma \vdash [\mathscr{U}_i \otimes \mathscr{W}, g\{u_i/x, \nu_i/y\}] : A. \quad (4.29)$$

Thus, through the induction hypothesis on the smaller terms of (4.27) which is in the form:

$$\emptyset \vdash [\mathscr{U}, f] : E \otimes F \ \wedge\ [\mathscr{U}, f] \Downarrow \{[\mathscr{U}_i, \langle u_i, \nu_i\rangle]^{p_i}\}_{i\in\mathcal{I}} \Rightarrow \forall i \in \mathcal{I},\ \emptyset \vdash [\mathscr{U}_i, \langle u_i, \nu_j\rangle] : E \otimes F \tag{4.30a}$$

$$\Gamma \vdash [\mathscr{U}_i \otimes \mathscr{W}, g\{u_i/x, \nu_i/y\}] : A \ \wedge\ [\mathscr{U}_i \otimes \mathscr{W}, g\{u_i/x, \nu_i/y\}] \Downarrow \{[\mathscr{U}_i \otimes \mathscr{W}_j, v_{i,j}]^{q_j}\}_{j\in\mathcal{J}} \Rightarrow\ \forall j \in \mathcal{J},\ \Gamma \vdash [\mathscr{U}_i \otimes \mathscr{W}_j, v_{i,j}] : A, \tag{4.30b}$$

we get the thesis. □

4.2 Quantum Context Equivalence

In supplying the notion of equivalence between quantum programs, we must consider that quantum closures are precisely what we want to compare in ℓQST_λ.

In order to give a definition of context preorder and context equivalence in a quantum language, it shall be necessary to supply each context with its own quantum register: with this purpose, we start giving, for the moment, the grammar necessary to build a *term context* $\Delta \vdash C[\Gamma \vdash A] : B$ in ℓQST_λ distinguishing, as usual, the quantum contexts which are terms, from those which are values and denoted, by $V[\cdot]$.

$$V[\cdot] ::= [\cdot] \mid \lambda x.C[\cdot] \mid \langle V[\cdot], u\rangle \mid \langle u, V[\cdot]\rangle, \tag{4.31a}$$

$$\begin{aligned} C[\cdot] ::= {} & [\cdot] \mid fC[\cdot] \mid C[\cdot]f \mid \texttt{if}\ C[\cdot]\ \texttt{then}\ f\ \texttt{else}\ g \mid \texttt{if}\ f\ \texttt{then}\ C[\cdot]\ \texttt{else}\ D[\cdot] \mid \\ & \mid \texttt{let}\ f\ \texttt{be}\ \langle x, y\rangle\ \texttt{in}\ C[\cdot] \mid \texttt{let}\ C[\cdot]\ \texttt{be}\ \langle x, y\rangle\ \texttt{in}\ f \mid \texttt{new}\,(V[\cdot]) \mid \\ & \mid \texttt{meas}_n\,(V[\cdot]) \mid U\,(V[\cdot]) \mid \texttt{cmp}\,(V[\cdot], \mathsf{V}) \mid \texttt{cmp}\,(\mathsf{V}, V[\cdot])\,. \end{aligned} \tag{4.31b}$$

Remarkably, the holes belonging to the quantum term contexts, can host both a quantum closure or a single quantum term depending on their structure, which is examined in Figure 4.4, where the typing rules for context closures are given.

A context (quantum) closure is a quantum closure $[\mathscr{U}, C[\cdot]]$ whose second component is a context: the quantum register $\mathscr{U}$ of the context closure, stores every free

quantum variable of the context $C[\cdot]$, which is the second component of the context quantum closure, recursively produced by the syntax tree (4.31a,4.31b). Sometimes the symbol $\mathbb{C}_{\mathscr{U}}[\vdash_A]$ will be used for a short form of $[\mathscr{U}, C[\vdash_A]]$: this context closure requires to be filled with a quantum closure such that $\Gamma \vdash [\mathscr{Q}, e] : A$. We will give significance to the writing $\mathbb{C}_{\mathscr{U}}[\Gamma \vdash [\mathscr{Q}, e] : A]$ imposing the equivalence

$$[\mathscr{U}, C[\Gamma \vdash [\mathscr{Q}, e] : A]] \stackrel{\text{def}}{\equiv} [\mathscr{U} \otimes \mathscr{Q}, C[e]]\,, \tag{4.32}$$

where $\otimes$ is the operator of tensor product between Hilbert's spaces of the quantum registers variables .

Since they must be employed to build a context preorder, the context closures that will be used shall be both total and closed: thoroughly, if $C[\cdot]$ is a context and $\mathcal{Q}$ is the set of its free quantum variables, then the quantum register of the context closure $[\mathscr{U}, C[\cdot]]$ is such that $\mathscr{U} \in \mathcal{H}(\mathcal{Q})$.

Similarly to what has been done with the deterministic and probabilistic languages, we will fix a symbol to identify the context closures that may be employed in the definition of quantum context equivalence and context preorder, which are those belonging to the set $\mathtt{QCTX}_B(\Gamma \vdash A)$, denoting the total context closures with type B, being by definition

$$\mathtt{QCTX}_B(\Gamma \vdash A) = \{[\mathscr{U}, C[\cdot]] \;\Big|\; \mathcal{Q} : \mathtt{qbit} \vdash C[\Gamma \vdash A] : B, \mathscr{U} \in \mathcal{H}(\mathcal{Q})\}. \tag{4.33}$$

A function $\mathbf{Obs} : \mathcal{T}_{\Gamma,A}^{[\mathtt{QC}]\,\ell QST_\lambda} \to \mathbb{R}$, is also built likewise in ℓST_λ, as the sum of probabilities that the quantum closure $[\mathscr{Q}, e] \in \mathcal{T}_{\Gamma,A}^{[\mathtt{QC}]\,\ell QST_\lambda}$ evaluates to whatever element of the set $\mathcal{V}_{\Gamma,A}^{[\mathtt{QC}]\,\ell QST_\lambda}$. Namely we define

$$\mathbf{Obs}([\mathscr{Q}, e]) = \sum_{[\mathscr{W}, v]} [\![[\mathscr{Q}, e]]\!]([\mathscr{W}, v]) = \sum [\![[\mathscr{Q}, e]]\!], \tag{4.34}$$

denoting by $[\![[\mathscr{Q}, e]]\!]$ the probability distribution corresponding to the semantics of $[\mathscr{Q}, e]$. Likewise in probabilistic case, the relation of context preorder is linked to the notion of observational behaviour of the terms involved in the relation, which can be "tested" in whatever context , thus we fix

$$[\mathscr{Q}, e] \leq_{\Gamma,A} [\mathscr{R}, h] \Rightarrow \forall\, [\mathscr{U}, C[\cdot]] \in \mathtt{QCTX}_B\,(\Gamma \vdash A)\,,$$
$$\mathbf{Obs}([\mathscr{U} \otimes \mathscr{Q}, C[e]]) \leq \mathbf{Obs}([\mathscr{U} \otimes \mathscr{R}, C[h]]), \quad (4.35)$$

where the symbols $\otimes$ stands again for the operator of *tensor product* between the Hilbert's space of the two quantum systems $\mathscr{Q}$ and $\mathscr{U}$. As in deterministic and probabilistic languages, the relation of context preorder that we have just defined is a preorder, being reflexive and transitive. Since we would like that it is a preconguence in ℓQST_λ, we must give the list of compatibility rules namely a new series of conditions similar to (2.21a–2.21f), listed for the deterministic ℓST_λ language.

$$(\mathbf{c}-\mathbf{1})_{\mathtt{Q}}\ \forall x, x : A \vdash [\emptyset, x]\ R\ [\emptyset, x] : A \qquad A \in \mathcal{Y}_{\ell QST_\lambda},$$
$$\forall r, \forall \mathscr{Q} \in \mathcal{H}(\{r\}),\ \emptyset \vdash [\mathscr{Q}, r]\ R\ [\mathscr{Q}, r] : \mathtt{qbit} \quad (4.36a)$$

$$(\mathbf{c}-\mathbf{2})_{\mathtt{Q}}\ \Gamma, x : B \vdash [\mathscr{Q}, e]\ R\ [\mathscr{R}, h] : A \Rightarrow \Gamma \vdash [\mathscr{Q}, \lambda x.e]\ R\ [\mathscr{R}, \lambda x.h] : B \multimap A \quad (4.36b)$$

$$(\mathbf{c}-\mathbf{3})_{\mathtt{Q}}\ \Gamma \vdash [\mathscr{Q}, e]\ R\ [\mathscr{W}, g] : B \multimap A \wedge \Delta \vdash [\mathscr{U}, f]\ R\ [\mathscr{R}, h] : B \Rightarrow$$
$$\Rightarrow\ \Gamma, \Delta \vdash [\mathscr{Q} \otimes \mathscr{U}, ef]\ R\ [\mathscr{W} \otimes \mathscr{R}, gh] : A \quad (4.36c)$$

$$(\mathbf{c}-\mathbf{4})_{\mathtt{Q}}$$
$$\Gamma \vdash [\mathscr{Q}, e]\ R\ [\mathscr{R}, h] : \mathtt{bool} \wedge \Delta \vdash [\mathscr{U}, f]\ R\ [\mathscr{S}, \ell] : A \wedge \Delta \vdash [\mathscr{W}, \ell]\ R\ [\mathscr{V}, a] : A \Rightarrow$$
$$\Gamma, \Delta \vdash ([\mathscr{Q} \otimes \mathscr{U} \otimes \mathscr{W}, \mathtt{if}\ e\ \mathtt{then}\ f\ \mathtt{else}\ g])\, R\, ([\mathscr{R} \otimes \mathscr{S} \otimes \mathscr{V}, \mathtt{if}\ h\ \mathtt{then}\ \ell\ \mathtt{else}\ a]) : A \quad (4.36d)$$

$$(\mathbf{c}-\mathbf{5})_{\mathtt{Q}}\ \Gamma \vdash [\mathscr{Q}, e]\ R\ [\mathscr{R}, h] : B \otimes E \wedge\ \Delta, x : B, y, E \vdash [\mathscr{U}, f]\ R\ [\mathscr{R}, h] : A \Rightarrow$$
$$\Rightarrow \Gamma, \Delta \vdash ([\mathscr{Q} \otimes \mathscr{U}, \mathtt{let}\ e\ \mathtt{be}\ \langle x, y\rangle\ \mathtt{in}\ f])\, R\, ([\mathscr{W} \otimes \mathscr{R}, \mathtt{let}\ g\ \mathtt{be}\ \langle x, y\rangle\ \mathtt{in}\ h]) : A \quad (4.36e)$$

$$(\mathbf{c}-\mathbf{6})_{\mathtt{Q}}\ \Gamma \vdash [\mathscr{Q}, v]\ R\ [\mathscr{W}, \nu] : A \wedge \Delta \vdash [\mathscr{U}, u]\ R\ [\mathscr{R}, w] : B \Rightarrow$$
$$\Rightarrow \Gamma, \Delta \vdash [\mathscr{Q} \otimes \mathscr{U}, \langle v, u\rangle]\ R\ [\mathscr{W} \otimes \mathscr{R}, \langle \nu, w\rangle] : A \otimes B \quad (4.36f)$$

$$(\mathbf{c}-\mathbf{c})_{\mathtt{Q}}\ \Gamma \vdash [\mathscr{Q}, v]\ R\ [\mathscr{W}, \nu] : A \wedge \Delta \vdash [\mathscr{U}, u]\ R\ [\mathscr{R}, w] : B \Rightarrow$$
$$\Rightarrow \Gamma, \Delta \vdash [\mathscr{Q} \otimes \mathscr{U}, \mathtt{cmp}\,(v, u)]\ R\ [\mathscr{W} \otimes \mathscr{R}, \mathtt{cmp}\,(u, w)] : \mathtt{qbit}^{n+m} \quad (4.36g)$$

$$(\mathbf{c}-\mathbf{n})_{\mathtt{Q}}\ \Gamma \vdash\ [\mathscr{Q}, v]\ R\ [\mathscr{R}, w] : \mathtt{bool} \Rightarrow \Gamma \vdash [\mathscr{Q}, \mathtt{new}\,(v)]\ R\ [\mathscr{R}, \mathtt{new}\,(w)] : \mathtt{qbit}$$

(4.36h)

$$(\mathbf{c}-\mathbf{m})_{\mathtt{Q}}\ \Gamma \vdash\ [\mathscr{Q}, v]\ R\ [\mathscr{R}, w] : \mathtt{qbit}^{n+1} \Rightarrow$$
$$\Gamma \vdash [\mathscr{Q}, \mathtt{meas}_n\,(v)]\ R\ [\mathscr{R}, \mathtt{meas}_n\,(w)] : \mathtt{qbit}^n \otimes \mathtt{bool} \tag{4.36i}$$
$$(\mathbf{c}-\mathbf{u})_{\mathtt{Q}}\ \Gamma \vdash [\mathscr{Q}, v]\ R\ [\mathscr{R}, w] :\Rightarrow\ \Gamma \vdash [\mathscr{Q}, U\,(v)]\ R\ [\mathscr{R}, U\,(w)] : \mathtt{qbit}^{\mathtt{a}(U)} \tag{4.36j}$$

Lemma 4.3 (Quantum context preorder behaviour with respect to contexts)**.** *If two quantum closures are in quantum context preorder relation, the relation is preserved whether they are embedded in a whatever context closures. This may be stated with the entailment*

$$[\mathscr{Q}, e] \leq_{\Gamma, A} [\mathscr{R}, h]\ \Rightarrow\ (\forall\,[\mathscr{U}, C[\cdot]] \in \mathtt{QCTX}_B\,(\Gamma \vdash A)\,,$$
$$[\mathscr{U} \otimes \mathscr{Q}, C[e]] \leq_B [\mathscr{U} \otimes \mathscr{R}, C[h]]) \tag{4.37}$$

Proof. The hypothesis implies the fulfillment of the condition

$$\forall\,[\mathscr{W}, D[\cdot]] \in \mathtt{QCTX}_B\,(\Gamma \vdash A)\,,\ [\mathscr{W} \otimes \mathscr{Q}, D[e]] \leq_B [\mathscr{W} \otimes \mathscr{R}, D[h]]\,, \tag{4.38}$$

while the thesis requires that

$$\forall\,[\mathscr{S}, G[\cdot]] \in \mathtt{QCTX}_E\,(\emptyset \vdash B)\,,\ [\mathscr{S} \otimes \mathscr{U} \otimes \mathscr{Q}, G[C[e]]] \leq_E [\mathscr{S} \otimes \mathscr{U} \otimes \mathscr{R}, G[C[h]]]\,, \tag{4.39}$$

is verified. But, provided to have taken $\mathscr{W} = \mathscr{S} \otimes \mathscr{U}$ and $D[\vdash_A \cdot] = G[C[\vdash_A \cdot]]$, as well $B = E$, the condition (4.39) reduces to (4.38). This proves the thesis, entailing the compatibility of the quantum context preorder relation in ℓQST_λ. □

4.2.1 Applicative Bisimilarity in ℓQST_λ

Would it be possible to have a notion of bisimilarity for ℓQST_λ? What is the underlying "Markov Chain"? It turns out that LMC as introduced in Section 3.2 are sufficient, but we need to be careful. In particular, states of the LMC are not terms, but quantum closures, of which there are in principle nondenumerably many. However, since we are only interested in quantum closures which can be obtained

Type judgement context closure rule	Name
$\dfrac{}{\Gamma \vdash_v [\emptyset, [\Gamma \vdash_v A]] : A}$	$(tjc_{\mathtt{Q}} - ax_v)$
$\dfrac{}{\Gamma \vdash_e [\emptyset, [\Gamma \vdash_e A]] : A}$	$(tjc_{\mathtt{Q}} - ax_t)$
$\dfrac{\Gamma \vdash_v [\emptyset, [\Gamma \vdash_v A]] : A}{\Gamma \vdash_e [\emptyset, [\Gamma \vdash_e A]] : A}$	$(tjc_{\mathtt{Q}} - vt)$
$\dfrac{\Gamma, x : B \vdash_e [\mathscr{Q}, C[\Theta \vdash_e E]] : A}{\Gamma \vdash_v [\mathscr{Q}, \lambda\, x.C[\Theta \vdash_e E]] : B \multimap A}$	$(tjc_{\mathtt{Q}} - abs)$
$\dfrac{\Gamma \vdash_e [\mathscr{Q}, C[\Theta \vdash_e E]] : B \multimap A \qquad \Delta \vdash [\mathscr{U}, f] : B}{\Gamma \vdash_e [\mathscr{Q} \otimes \mathscr{U}, C[\Theta \vdash_e E] f] : A}$	$(tjc_{\mathtt{Q}} - app_L)$
$\dfrac{\Gamma \vdash [\mathscr{U}, f] : B \multimap A \qquad \Delta \vdash_e [\mathscr{Q}, C[\Theta \vdash_e E]] : B}{\Gamma \vdash_e [\mathscr{Q} \otimes \mathscr{U}, f C[\Theta \vdash_e E]] : A}$	$(tjc_{\mathtt{Q}} - app_R)$
$\dfrac{\Gamma \vdash_e [\mathscr{Q}, C[\Theta \vdash_e E]] : \mathtt{bool} \qquad \Delta \vdash [\mathscr{U}, f] : A \qquad \Delta \vdash [\mathscr{U}, g] : A}{\Gamma, \Delta \vdash_e [\mathscr{Q} \otimes \mathscr{U}, \mathtt{if}\ C[\Theta \vdash_e E]\ \mathtt{then}\ f\ \mathtt{else}\ g] : A}$	$(tjc_{\mathtt{Q}} - if_L)$
$\dfrac{\Gamma \vdash [\mathscr{Q}, e] : \mathtt{bool} \quad \Delta \vdash_e [\mathscr{U}, C[\Theta \vdash_e E]] : A \quad \Delta \vdash_e [\mathscr{U}, D[\Theta \vdash_e E]] : A}{\Gamma, \Delta \vdash_e [\mathscr{Q} \otimes \mathscr{U}, \mathtt{if}\ e\ \mathtt{then}\ C[\Theta \vdash_e E]\ \mathtt{else}\ D[\Theta \vdash_e E]] : A}$	$(tjc_{\mathtt{Q}} - if_R)$
$\dfrac{\Gamma \vdash_v [\mathscr{Q}, V[\Theta \vdash_e E]] : A \qquad \Delta \vdash [\mathscr{U}, u] : B}{\Gamma, \Delta \vdash_v [\mathscr{Q} \otimes \mathscr{U}, \langle V[\Theta \vdash_e E], u\rangle] : A \otimes B}$	$(tjc_{\mathtt{Q}} - pai_L)$
$\dfrac{\Gamma \vdash [\mathscr{Q}, v] : A \qquad \Delta \vdash_v [\mathscr{U}, V[\Theta \vdash_e E]] : B}{\Gamma, \Delta \vdash_v [\mathscr{Q} \otimes \mathscr{U}, \langle v, V[\Theta \vdash_e E]\rangle] : A \otimes B}$	$(tjc_{\mathtt{Q}} - pai_R)$
$\dfrac{\Gamma \vdash_e [\mathscr{Q}, C[\Theta \vdash_e E]] : B \otimes F \qquad \Delta, x : B, y : F \vdash [\mathscr{U}, f] : A}{\Gamma, \Delta \vdash_e [\mathscr{Q} \otimes \mathscr{U}, \mathtt{let}\ C[\Theta \vdash_e E]\ \mathtt{be}\ \langle x, y\rangle\ \mathtt{in}\ f] : A}$	$(tjc_{\mathtt{Q}} - let_L)$
$\dfrac{\Gamma \vdash [\mathscr{Q}, f] : B \otimes F \qquad \Delta, x : B, y : F \vdash_e [\mathscr{U}, C[\Theta \vdash_e E]] : A}{\Gamma, \Delta \vdash_e [\mathscr{Q} \otimes \mathscr{U}, \mathtt{let}\ f\ \mathtt{be}\ \langle x, y\rangle\ \mathtt{in}\ C[\Theta \vdash_e E]] : A}$	$(tjc_{\mathtt{Q}} - let_R)$
$\dfrac{\Gamma \vdash_v [\mathscr{Q}, V[\Theta \vdash_e E]] : \mathtt{bool}}{\Gamma \vdash_v [\mathscr{Q}, \mathtt{new}\,(V[\Theta \vdash_e E])] : \mathtt{bool}}$	$(tjc_{\mathtt{Q}} - new)$
$\dfrac{\Gamma \vdash_v [\mathscr{Q}, V[\Theta \vdash_e E]] : \mathtt{qbit}^{\mathtt{a}(U)}}{\Gamma \vdash_v [\mathscr{Q}, U\,(V[\Theta \vdash_e E])] : \mathtt{qbit}^{\mathtt{a}(U)}}$	$(tjc_{\mathtt{Q}} - uni)$
$\dfrac{\Gamma \vdash_v [\mathscr{Q}, V[\Theta \vdash_e E]] : \mathtt{qbit}^1}{\Gamma \vdash_v [\mathscr{Q}, \mathtt{meas}_1\,(V[\Theta \vdash_e E])] : \mathtt{bool}}$	$(tjc_{\mathtt{Q}} - mea)$
$\dfrac{\Gamma \vdash_v [\mathscr{Q}, V[\Theta \vdash_e E]] : \mathtt{qbit}^{n+1} \qquad 1 \leq m \leq n}{\Gamma \vdash_v [\mathscr{Q}, \mathtt{meas}_m\,(V[\Theta \vdash_e E])] : \mathtt{qbit}^n \otimes \mathtt{bool}}$	$(tj_{\mathtt{Q}} - mea)$
$\dfrac{\Gamma \vdash_v [\mathscr{Q}, V[\Theta \vdash_e E]] : \mathtt{qbit}^n \qquad \Delta \vdash [\mathscr{U}, u] : \mathtt{qbit}^m}{\Gamma, \Delta \vdash_v [\mathscr{Q} \otimes \mathscr{U}, \mathtt{cmp}\,(V[\Theta \vdash_e E], u)] \mathtt{qbit}^{n+m}}$	$(tjc_{\mathtt{Q}} - cmp_L)$
$\dfrac{\Gamma \vdash [\mathscr{Q}, v] : \mathtt{qbit}^n \qquad \Delta \vdash_v [\mathscr{U}, V[\Theta \vdash_e E]] : \mathtt{qbit}^m}{\Gamma, \Delta \vdash_v [\mathscr{Q} \otimes \mathscr{U}, \mathtt{cmp}\,(v, V[\Theta \vdash_e E])] \mathtt{qbit}^{n+m}}$	$(tjc_{\mathtt{Q}} - cmp_R)$

Figure 4.4: Context typing rules for contexts closures.

(in a finite number of evaluation steps) from closures having an empty quantum register, this is not a problem: we simply take states as *those* closures, which we dub *constructible*. $\mathscr{M}_{\ell QST_\lambda}$ can be built similarly to $\mathscr{M}_{\ell PST_\lambda}$, where (constructible) quantum closures take the place of terms. Hence we set

$$\mathscr{S} = \overline{\mathcal{T}^{[\mathtt{QC}]\,\ell QST_\lambda}} \uplus \overline{\mathcal{V}^{[\mathtt{QC}]\,\ell QST_\lambda}} \tag{4.40a}$$

$$\mathcal{L} = \{a_{eval}, a_{\mathtt{tt}}, a_{\mathtt{ff}}, a_{@[\mathscr{W},\nu]}, a_{\otimes[\mathscr{W},g]}, a_{\mathcal{Y}_A}, a_{\widehat{\mathcal{Y}_A}}, a_{\mathtt{Q}[\mathscr{W},r]}\} \tag{4.40b}$$

$$\mathcal{P} = \mathcal{P}_{\ell QST_\lambda}. \tag{4.40c}$$

where $\mathcal{T}^{[\mathtt{QC}]\,\ell QST_\lambda}_{\Gamma,A} = \{[\mathscr{Q}, e] \mid e \in \mathcal{T}^{\ell QST_\lambda}_{\Gamma,A}\}$ and $\overline{\mathcal{T}^{[\mathtt{QC}]\,\ell QST_\lambda}_{\Gamma,A}}$ is the set of pairs $([\mathscr{Q}, e], A)$. Analogous meaning, just for values must be assigned to $\mathcal{V}^{[\mathtt{QC}]\,\ell QST_\lambda}_{A}$ and $\overline{\mathcal{V}^{[\mathtt{QC}]\,\ell QST_\lambda}_{A}} = \mathcal{V}^{[\mathtt{QC}]\,\ell QST_\lambda}_{A} \times \mathcal{Y}_A$. The non zero elements of the function $\mathcal{P}_{\ell QST_\lambda}$ are defined as follows:

$$\mathcal{P}_{\ell QST_\lambda}\left((\widehat{\mathtt{tt}}, \mathtt{bool}), a_{\mathtt{tt}}, (\widehat{\mathtt{tt}}, \mathtt{bool})\right) = 1;$$

$$\mathcal{P}_{\ell QST_\lambda}\left((\widehat{\mathtt{ff}}, \mathtt{bool}), a_{\mathtt{ff}}, (\widehat{\mathtt{ff}}, \mathtt{bool})\right) = 1;$$

$$\mathcal{P}_{\ell QST_\lambda}\left((\widehat{[\mathscr{Q}, \lambda x.e]}, B \multimap A), a_{@[\mathscr{W},v]}, ([\mathscr{Q} \otimes \mathscr{W}, e\{v/x\}], A)\right) = 1;$$

$$\mathcal{P}_{\ell QST_\lambda}\left((\widehat{[\mathscr{Q}, \mathsf{V}]}, \mathtt{qbit}^n), a_{\mathtt{Q}[\mathscr{W},g]}, ([\mathscr{Q} \otimes \mathscr{W}, g\{\mathsf{V}/x\}], E)\right) = 1;$$

$$\mathcal{P}_{\ell QST_\lambda}\left((\widehat{[\mathscr{Q}, \langle v, u\rangle]}, A \otimes B), a_{\otimes[\mathscr{W},g]}, ([\mathscr{Q} \otimes \mathscr{W}, g\{v/x, u/y\}], E)\right) = 1;$$

$$\mathcal{P}_{\ell QST_\lambda}\left(([\mathscr{Q}, e], A), a_{\mathcal{Y}_A}, ([\mathscr{Q}, e], A)\right) = 1;$$

$$\mathcal{P}_{\ell QST_\lambda}\left((\widehat{[\mathscr{Q}, e]}, A), a_{\widehat{\mathcal{Y}_A}}, (\widehat{[\mathscr{Q}, e]}, A)\right) = 1;$$

$$\mathcal{P}_{\ell QST_\lambda}\left(([\mathscr{Q}, e], A), a_{eval}, (\widehat{[\mathscr{U}, v]}, A)\right) = [\![[\mathscr{Q}, e]]\!]([\mathscr{U}, v]).$$

Please notice the presence of a new label for the qubits $a_{\mathtt{Q}[\mathscr{W},g]}$ which models the action of giving the qubit as an argument for the open term $[\mathscr{W}, g]$.

The simulation relation here is given on the set of the quantum closures, $\mathcal{T}^{[\mathtt{QC}]\ell QST_\lambda}_{\Gamma,A}$ using the suitable transition elements for each type and by distinguishing term by values. The full set of labelled actions for $\mathscr{M}_{\ell QST_\lambda}$ is presented in

Action	Label(s)
Show the value of a boolean:	$a_{\mathtt{tt}},\ a_{\mathtt{ff}}$
Gives a *quantum closure* as argument to a function type:	$a_{@[\mathscr{U},u]}$
Substitutes a pair into the second component of a quantum closure :	$a_{\otimes[\mathscr{U},f]}$
Substitutes the quantum variable s resident in the quantum register $\mathscr{U}$ in the open quantum closure $[\mathscr{W},g]$	$a_{\mathtt{Q}[\mathscr{W},g]}$
Exhibits the type of a value:	$a_{\widehat{\mathcal{Y}_A}}$
Exhibits the type of a term:	$a_{\mathcal{Y}_A}$
Evaluates a quantum closure $[\mathscr{Q},e]$	a_{eval}

Figure 4.5: The action allowed in $\mathscr{M}_{\ell QST_\lambda}$.

- For quantum closures belonging to the set $\mathcal{V}_{\mathtt{bool}}^{[\mathtt{QC}]\,\ell QST_\lambda}$, the quantum context is only formally involved in the definition, which is identical to the probabilistic one (3.16), provided that the following identity has been settled, that $[\emptyset,e]$ coincides with e, where $\emptyset$ is the notation for empty quantum register. $\mathcal{S}_{\mathtt{bool}}$ is a simulation for boolean quantum closures if the following condition is accomplished

$$\forall\,[\emptyset,\mathtt{b}]\in\mathcal{V}_{\mathtt{bool}}^{\ell QST_\lambda},\mathcal{P}_{\ell QST_\lambda}\left((\widehat{[\emptyset,e]},\mathtt{bool}),a_{\mathtt{b}},(\widehat{[\emptyset,\mathtt{b}]},\mathtt{bool})\right)\leq \mathcal{P}_{\ell QST_\lambda}\left((\widehat{[\emptyset,h]},\mathtt{bool}),a_{\mathtt{b}},(\widehat{[\emptyset,\mathtt{b}]},\mathtt{bool})\right). \quad (4.41)$$

- For function values the condition of simulation between quantum closures involves, as usual, the action labelled by the subsitution of a value:

$$\begin{aligned}\emptyset \vdash [\mathscr{Q}, \lambda\, x.e]\ \mathcal{S}_{B\multimap A}\ [\mathscr{R}, \lambda\, x.h] : B \multimap A \ \Rightarrow\ \forall\, [\mathscr{W}, v] \in \mathcal{V}_B^{[\mathtt{QC}]\,\ell QST_\lambda},\\ \mathcal{P}_{\ell QST_\lambda}\left(\widehat{([\mathscr{Q}, \lambda\, x.e]}, B \multimap A), a_{@[\mathscr{W},v]}, ([\mathscr{Q}\otimes\mathscr{W}, e\{v/x\}]\,, A)\right) \leq\\ \mathcal{P}_{\ell QST_\lambda}\left(\widehat{([\mathscr{R}, \lambda\, x.h]}, B \multimap A), a_{@[\mathscr{W},v]}, (\,\mathcal{S}_A\ ([\mathscr{Q}\otimes\mathscr{W}, e\{v/x\}])\,, A)\right). \quad (4.42)\end{aligned}$$

- For pairs, also in a quantum environment, the definition of simulation shall rely on the tag $a_{\otimes[\mathscr{W},g]}$ whose argument is here a quantum closure:

$$\begin{aligned}\emptyset \vdash [\mathscr{Q}, \langle v_1, v_2\rangle]\ \mathcal{S}_{A\otimes B}\ [\mathscr{R}, \langle w_1, w_2\rangle] : A\otimes B\ \Rightarrow \forall\, [\mathscr{W}, g] \in \mathcal{T}^{[\mathtt{QC}]\,\ell QST_\lambda}_{x:A,y:B;E}\\ \mathcal{P}_{\ell QST_\lambda}\left(\widehat{([\mathscr{Q}, \langle v_1, v_2\rangle}], A\otimes B), a_{\otimes[\mathscr{W},g]}, ([\mathscr{Q}\otimes\mathscr{W}, g\{v_1/x, v_2/y\}]\,, E)\right) \leq\\ \mathcal{P}_{\ell QST_\lambda}\left(([\mathscr{Q}, \widehat{\langle w_1, w_2\rangle}], A\otimes B), a_{\otimes[\mathscr{W},g]}, (\,\mathcal{S}_E\ ([\mathscr{Q}\otimes\mathscr{W}, g\{v_1/x, v_2/y\}])\,, E)\right)\end{aligned} \quad (4.43)$$

- To compare quantum variable sequences we need the elements of transition matrix labelled by $a_{\mathtt{Q}[\mathscr{W},g]}$, being g an open term of the quantum language, namely $g \in \mathcal{T}^{[\mathtt{QC}]\,x:\mathtt{qbit};E}_{\ell QST_\lambda}$:

$$\begin{aligned}\emptyset \vdash [\mathscr{Q}, \mathsf{V}]\ \mathcal{S}_{\mathtt{qbit}^n}\ [\mathscr{R}, h] : \mathtt{qbit}^n,\ \Rightarrow \forall\, [\mathscr{W}, g] \in \mathcal{T}^{[\mathtt{QC}]\,\ell QST_\lambda}_{x:\mathtt{qbit}^n;E}\\ \left(\mathcal{P}_{\ell QST_\lambda}\left(\widehat{([\mathscr{Q}, \mathsf{V}]}, \mathtt{qbit}^n), a_{\mathtt{Q}[\mathscr{W},g]}, ([\mathscr{Q}\otimes\mathscr{W}, g\{\mathsf{V}/x\}]\,, A)\right)\right.\\ \left.\leq \mathcal{P}_{\ell QST_\lambda}\left(\widehat{([\mathscr{R}, h]}, \mathtt{qbit}^n), a_{\mathtt{Q}[\mathscr{W},g]}, (\,\mathcal{S}_A\ ([\mathscr{Q}\otimes\mathscr{W}, g\{\mathsf{V}/x\}])\,, A)\right)\right)\end{aligned} \quad (4.44)$$

- For terms the definition of simulation is similar to the probabilistic case, taking into account that the domain of distributions is a subset of $\mathcal{V}_A^{[\mathtt{QC}]\,\ell QST_\lambda}$, rather than $\mathcal{V}_A^{\ell QST_\lambda}$:

$$\begin{aligned}\emptyset \vdash [\mathscr{Q}, e]\ \mathcal{S}_A\ [\mathscr{R}, h] : A \ \Rightarrow\ \forall X \in \mathcal{V}_A^{[\mathtt{QC}]\,\ell QST_\lambda},\\ \mathcal{P}_{\ell QST_\lambda}\left(([\mathscr{Q}, e]\,, A), a_{eval}, (X, A)\right) \leq \mathcal{P}_{\ell QST_\lambda}\left(([\mathscr{R}, h]\,, A), a_{eval}, (\,\mathcal{S}_A\,(X), A)\right).\end{aligned} \quad (4.45)$$

Once we have a LMC, it is easy to apply the same definitional scheme we have seen for ℓPST_λ, and obtain a notion of applicative (bi)similarity: indeed properties

proved in Lemma 3.9, Lemma 3.10 and Proposition 3.1, relying on the definition of transitive closure remain unaltered also in quantum environment, whence we must conclude that the transitive closure of the union of all possible simulations and bisimulations, being a simulation and a bisimulation in turn on the set $\mathcal{T}^{[\mathtt{QC}]\,\ell QST_\lambda}$ of quantum closures of ℓQST_λ, play the role of quantum similarity and bisimilarity respectively.

Howe's extension of the applicative (bi)simulation for ℓQST_λ is equally necessary because here too we must face the same difficulties that have been raised in deterministic and probabilistic languages, concerning the proof of compatibility for the simulation relation.

Here Howe's rules, listed in Figure 4.6, involve the quantum terms of the language as well as the deterministic ones and they are given as a relation between quantum closures. The full set of Howe's rules for ℓQST_λ in Figure 4.6, resumes in a unique instance the case of complex terms, built up with smaller subterms through a syntactic constructor, `cnstr`. In ℓQST_λ, Howe's relation enjoys the same properties of compatibility, pseudo-transitivity and substitutivity, that have been proved in deterministic case. We show these properties on the whole set of quantum closures typed with the rules provided in Figure (4.1). The property $R \subseteq R^H$ stated in Lemma 2.17 holds unchanged, being independent of the terms.

Lemma 4.4 (Compatibility of R^H in ℓQST_λ)**.**
If R is reflexive then R^H is compatible on the quantum closures of ℓQST_λ.

Proof. Starting from the reflexivity of R we want to prove the statement

$$\Gamma_1 \vdash [\mathscr{Q}_1, e_1]\ R^H\ [\mathscr{R}_1, h_1] : A_1 \ldots \Gamma_N \vdash [\mathscr{Q}_N, e_N]\ R^H\ [\mathscr{R}_N, h_N] : A_N \Rightarrow$$
$$\Gamma \vdash \left[\mathscr{Q}, \mathtt{cnstr}\left(\{e_n\}_{n\in\{1\ldots N\}}\right)\right]\ R^H\ \left[\mathscr{R}, \mathtt{cnstr}\left(\{h_n\}_{n\in\{1\ldots N\}}\right)\right] : A \quad (4.46)$$

being `cnstr` a whatever constructor of ℓQST_λ, $\mathscr{Q} = \mathscr{Q}_1 \otimes \ldots \otimes \mathscr{Q}_N$, likewise $\mathscr{R}$. Since the basic case is a tautology, being founded on Howe's rules $(How_{va1})_{\mathtt{Q}}$, $(How_{va2})_{\mathtt{Q}}$,

Howe's rule	Name
$\dfrac{\emptyset \vdash [\mathscr{Q}, \mathtt{b}] \preceq_{\mathtt{bool}} [\mathscr{R}, h] : \mathtt{bool}}{\emptyset \vdash [\mathscr{Q}, \mathtt{b}] \preceq_{\mathtt{bool}}{}^{H} [\mathscr{R}, h] : \mathtt{bool}}$	$(How_{con})_{\mathtt{Q}}$
$\dfrac{x : A \vdash [\mathscr{Q}, x] \preceq_{A} [\mathscr{R}, h] : A}{x : A \vdash [\mathscr{Q}, x] \preceq_{A}{}^{H} [\mathscr{R}, h] : A}$	$(How_{va1})_{\mathtt{Q}}$
$\dfrac{\emptyset \vdash [\mathscr{Q}, \mathsf{V}] \preceq_{\mathtt{qbit}^i} [\mathscr{R}, h] : \mathtt{qbit}^i}{\emptyset \vdash [\mathscr{Q}, \mathsf{V}] \preceq_{\mathtt{qbit}^i}{}^{H} [\mathscr{R}, h] : \mathtt{qbit}^i}$	$(How_{va2})_{\mathtt{Q}}$
$\dfrac{\Delta, x : B \vdash [\mathscr{Q}, e] \preceq_{A}{}^{H} [\mathscr{W}, g] : A \qquad \Delta \vdash [\mathscr{W}, \lambda x.g] \preceq_{B \multimap A} [\mathscr{R}, h] : B \multimap A}{\Delta \vdash [\mathscr{Q}, \lambda x.e] \preceq_{B \multimap A}{}^{H} [\mathscr{R}, \lambda x.h] : B \multimap A}$	$(How_{abs})_{\mathtt{Q}}$
$\dfrac{\Delta \vdash [\mathscr{Q}, v] \preceq_{\mathtt{bool}}{}^{H} [\mathscr{W}, \nu] : \mathtt{bool} \qquad \Delta \vdash [\mathscr{W}, \mathtt{new}\,(\nu)] \preceq_{\mathtt{qbit}^1} [\mathscr{R}, h] : \mathtt{qbit}^1}{\Delta \vdash [\mathscr{Q}, \mathtt{new}\,(v)] \preceq_{\mathtt{qbit}^1}{}^{H} [\mathscr{R}, h] : \mathtt{qbit}^1}$	$(How_{new})_{\mathtt{Q}}$
$\dfrac{\Delta \vdash [\mathscr{Q}, v] \preceq_{\mathtt{qbit}^1}{}^{H} [\mathscr{W}, \nu] : \mathtt{qbit}^1 \;\; \Delta \vdash [\mathscr{W}, \mathtt{meas}_1\,(\nu)] \preceq_{\mathtt{bool}} [\mathscr{R}, h] : \mathtt{bool}}{\Delta \vdash [\mathscr{Q}, \mathtt{meas}_1\,(v)] \preceq_{\mathtt{bool}}{}^{H} [\mathscr{R}, h] : \mathtt{bool}}$	$(How_{me1})_{\mathtt{Q}}$
$\dfrac{\Delta \vdash [\mathscr{Q}, v] \preceq_{\mathtt{qbit}^{n+1}}{}^{H} [\mathscr{W}, \nu] : \mathtt{qbit}^{n+1} \quad \Delta \vdash [\mathscr{W}, \mathtt{meas}_m\,(\nu)] \preceq [\mathscr{R}, h] : \mathtt{qbit}^n \otimes \mathtt{bool}}{\Delta \vdash [\mathscr{Q}, \mathtt{meas}_m\,(v)] \preceq_{\mathtt{bool}}{}^{H} [\mathscr{R}, h] : \mathtt{qbit}^n \otimes \mathtt{bool}}$	$(How_{men})_{\mathtt{Q}}$
$\dfrac{\Delta \vdash [\mathscr{Q}, v] \preceq_{\mathtt{qbit}^{\otimes n}}{}^{H} [\mathscr{W}, \nu] : \mathtt{qbit}^{\mathtt{a}(U)} \;\; \Delta \vdash [\mathscr{W}, U\,(\nu)] \preceq_{\mathtt{qbit}^{\mathtt{a}(U)}} [\mathscr{R}, h] : \mathtt{qbit}^{\mathtt{a}(U)}}{\Delta \vdash [\mathscr{Q}, U\,(v)] \preceq_{\mathtt{qbit}^{\otimes n}}{}^{H} [\mathscr{R}, h] : \mathtt{qbit}^{\mathtt{a}(U)}}$	$(How_{uni})_{\mathtt{Q}}$
$\dfrac{\begin{array}{l}\Delta_1 \vdash [\mathscr{Q}_1, e_1] \preceq_{A_1}{}^{H} [\mathscr{W}_1, g_1] : A_1 \\ \vdots \quad \vdots \quad \vdots \quad \vdots \\ \Delta_N \vdash [\mathscr{Q}_N, e_N] \preceq_{A_N}{}^{H} [\mathscr{W}_N, g_N] : A_N \\ \qquad \Delta_1, \ldots \Delta_N \vdash [\mathscr{W}_1 \otimes \ldots \otimes \mathscr{W}_N, \mathtt{cnstr}\,(\{g_n\}_{n \in \mathcal{N}})] \preceq_{A} [\mathscr{R}, h] : A\end{array}}{\Delta_1, \ldots, \Delta_N \vdash [\mathscr{Q}_1 \otimes \ldots \otimes \mathscr{Q}_N, \mathtt{cnstr}\,(\{e_n\}_{n \in \mathcal{N}})] \preceq_{A}{}^{H} [\mathscr{R}, h] : A}$	$(How_{gen})_{\mathtt{Q}}$

Figure 4.6: Howe's relation enhancement in quantum environment.

we analyse the rule $(How_{gen})_{\mathtt{Q}}$ which we quote below

$$\frac{\begin{array}{l}\Gamma_1 \vdash [\mathscr{Q}_1, e_1] \;\; R^{\,H} \, [\mathscr{W}_1, g_1] : A_1 \\ \quad \vdots \quad \vdots \\ \Gamma_N \vdash [\mathscr{Q}_N, e_N] \;\; R^{\,H} \, [\mathscr{W}_N, g_N] : A_N \quad \Gamma \vdash \left[\mathscr{W}, \mathtt{cnstr}\left(\{g_n\}_{n \in \{1 \ldots N}\right)\right] \;\; R \;\; [\mathscr{R}, h] : A\end{array}}{\Gamma \vdash [\mathscr{Q}, e] \;\; R^{\,H} \;\; [\mathscr{R}, h] : A}\,, \tag{4.47}$$

being as usual $\mathscr{W} = \mathscr{W}_1 \otimes \ldots \otimes \mathscr{W}_N$. Since by hypothesis R is reflexive, in (4.47) it

is possible that $\left[\mathscr{W}, \mathtt{cnstr}\left(\{g_n\}_{n\in\{1\ldots N\}}\right)\right] = [\mathscr{R}, h]$. This gives the thesis. □

Lemma 4.5 (Pseudo transitivity of R^H)**.**
If R is transitive, then R^H enjoys the pseudo–transitivity on the set of quantum, closures of ℓQST_λ, namely

$$\forall\left[\mathscr{Q}, e\right], \left[f, \mathscr{U}\right], \left[\mathscr{R}, h\right], \left(\Delta \vdash \left[\mathscr{Q}, e\right]\ R^H\ \left[\mathscr{U}, f\right] : A\ \wedge\ \Delta \vdash \left[\mathscr{U}, f\right]\ R\ \left[\mathscr{R}, h\right] : A\right)$$
$$\Rightarrow \Delta \vdash \left[\mathscr{Q}, e\right]\ R^H\ \left[\mathscr{R}, h\right] : A. \quad (4.48)$$

Proof. Considering the first hypothesis as a consequence of $(How_{gen})_{\mathtt{Q}}$ we find:

$$\frac{\begin{array}{l}\Delta_1 \vdash \left[\mathscr{Q}_1, e_1\right]\ R^H\ \left[\mathscr{W}_1, g_1\right] : A_1 \\ \quad\vdots \quad\ \vdots \\ \Delta_N \vdash \left[\mathscr{Q}_N, e_N\right]\ R^H\ \left[\mathscr{W}_N, g_N\right] : A_N\ \ \Delta \vdash \left[\mathscr{W}, \mathtt{cnstr}\left(\{g_n\}_{n\in\{1\ldots N\}}\right)\right]\ R\ \left[\mathscr{U}, f\right] : A\end{array}}{\Delta \vdash \left[\mathscr{Q}, e\right]\ R^H\ \left[\mathscr{U}, f\right] : A} \quad (4.49)$$

From the last hypothesis of the previous relation (4.49) and the second hypothesis of (4.48) by transitivity of R we get the result

$$\left(\Delta \vdash \left[\mathscr{W}, \mathtt{cnstr}\left(\{g_n\}_{n\in\{1\ldots N\}}\right)\right]\ R\ \left[\mathscr{U}, f\right] : A \wedge \Delta \vdash \left[\mathscr{U}, f\right]\ R\ \left[\mathscr{R}, h\right] : A\right) \Rightarrow$$
$$\Delta \vdash \left[\mathscr{W}, \mathtt{cnstr}\left(\{g_n\}_{n\in\{1\ldots N\}}\right)\right]\ R\ \left[\mathscr{R}, h\right] : A \quad (4.50)$$

which may be taken as last premise for Howe's general rule (4.49) to get

$$\frac{\begin{array}{l}\Delta_1 \vdash \left[\mathscr{Q}_1, e_1\right]\ R^H\ \left[\mathscr{W}_1, g_1\right] : A_1 \\ \quad\vdots \quad\ \vdots \\ \Delta_N \vdash \left[\mathscr{Q}_N, e_N\right]\ R^H\ \left[\mathscr{W}_N, g_N\right] : A_N\ \ \Delta \vdash \left[\mathscr{W}, \mathtt{cnstr}\left(\{g_n\}_{n\in\{1\ldots N\}}\right)\right]\ R\ \left[\mathscr{R}, h\right] : A\end{array}}{\Delta \vdash \left[\mathscr{Q}, e\right]\ R^H\ \left[\mathscr{R}, h\right] : A} \quad (4.51)$$

which is the thesis. □

Lemma 4.6 (Howe's relation substitutivity for quantum closures)**.** *If R is a transitive and closed under subsitution relation on the set of quantum closures $\mathcal{T}_A^{[\mathtt{QC}]\,\ell QST_\lambda}$, then R^H is* substitutive.

Proof. Let us prove the quantum substitutivity property (4.9) for Howe's lifting R^H, assuming that R is transitive and closed under substitution. The proof is by induction on the structure of the quantum closure $[\mathscr{Q}, e]$, examining Howe's rules 4.6
$-\,[\mathscr{Q}, e] = [\mathscr{Q}, x]\,-$ Here the statement is

$$\Gamma, x : B \vdash [\emptyset, x]\ R^H\ [\mathscr{W}, g] : A \wedge \Delta \vdash [\mathscr{U}, f]\ R^H\ [\mathscr{R}, h] : A \Rightarrow \Gamma, \Delta \vdash [\mathscr{U}, f]\ R^H\ [\mathscr{W} \otimes \mathscr{R}, g\{h/x\}] : A \quad (4.52)$$

where, for the the linearity of ℓQST_λ, $x \notin dom(\Gamma)$, thus necessarily $B = A$. The first hypothesis in (4.52), is necessarily a consequence of $(How_{v1})_{\mathtt{Q}}$, then it holds the relation $\Gamma \vdash [\emptyset, x]\ R\ [\mathscr{W}, g] : A$, and the closure under subsitution of R entails that $\forall\, \Delta \vdash [\mathscr{R}, h] : A$, the relation

$$\Gamma, \Delta \vdash [\mathscr{R}, h]\ R\ [\mathscr{W} \otimes \mathscr{R}, g\{h/x\}] : A. \quad (4.53)$$

Thus, the thesis follows from both the second hypothesis in (4.52) and relation (4.53), by pseudo–transitivity of Howe's relation.
$-\,[\mathscr{Q}, e] = [\mathscr{Q}, \mathtt{cnstr}\,(\{e_n\}_{n\in\mathcal{N}})]\,-$, where $\mathtt{cnstr}\,(e_1 \ldots e_N)$ is a whatever constructor of the language. This case, requires to resort to $(How_{gen})_{\mathtt{Q}}$ rule to prove the property (4.9) with $R = R^H$. Starting from the first hypothesis, namely $\Gamma, x : B \vdash [\mathscr{Q}, \mathtt{cnstr}\,(\{e_n\}_{n\in\mathcal{N}})]\ R^H\ [\mathscr{W}, g] : A$ and going back of a step in the derivation tree we obtain the following set of relations, in the linearity hypothesis

$$\frac{\begin{array}{l}\Gamma_1 \vdash [\mathscr{Q}_1, e_1]\ R^H\ [\mathscr{S}_1, \ell_1] : A_1 \\ \quad\vdots \quad\ \vdots \\ \Gamma_j, x : B \vdash [\mathscr{Q}_j, e_j]\ R^H\ [\mathscr{S}_j, \ell_j] : A_j \\ \quad\vdots \quad\ \vdots \\ \Gamma_N \vdash [\mathscr{Q}_N, e_N]\ R^H\ [\mathscr{S}_N, \ell_N] : A_N \quad \Gamma, x : B \vdash [\mathscr{S}, \mathtt{cnstr}\,(\{\ell_n\}_{n\in\mathcal{N}})]\ R\ [\mathscr{W}, g] : A\end{array}}{\Gamma, x : B \vdash [\mathscr{Q}, \mathtt{cnstr}\,(\{e_n\}_{n\in\mathcal{N}})]\ R^H\ [\mathscr{W}, g] : A} \quad (4.54)$$

where $\mathscr{Q} = \mathscr{Q}_1 \otimes \cdots \otimes \mathscr{Q}_N$ and $\mathscr{S} = \mathscr{S}_1 \otimes \cdots \otimes \mathscr{S}_N$. By linearity, the variable x must appear only once, namely in the j–th premise. Applying the induction hypothesis (of substitutivity) on the smallest j–th term, one gets the relation

$$\Gamma_j, x : B \vdash [\mathscr{Q}_j, e_j]\ R^H\ [\mathscr{S}_j, \ell_j] : A_j \wedge \Delta \vdash [\mathscr{U}, f]\ R^H\ [\mathscr{R}, h] : B \Rightarrow$$
$$\Gamma_j, \Delta \vdash [\mathscr{Q}_j \otimes \mathscr{U}, e_j\{f/x\}]\ R^H\ [\mathscr{S}_j \otimes \mathscr{R}, \ell_j\{h/x\}] : A_j \quad (4.55)$$

likewise, applying the property of closure by substitution to the last premise in (4.54), we get

$$\Gamma, x : B \vdash \left[\mathscr{S}, \texttt{cnstr}\,(\ell_1 \ldots, \ell_j \ldots \ell_N)\right]\ R\ [\mathscr{W}, g] : A \Rightarrow$$
$$\Gamma \vdash \left[\mathscr{S} \otimes \mathscr{R}, \texttt{cnstr}\,(\ell_1 \ldots, \ell_j\{h/x\} \ldots \ell_N)\right]\ R\ [\mathscr{W} \otimes \mathscr{R}, g\{h/x\}] : A. \quad (4.56)$$

Finally, to get the result and prove the general statement, we use (4.55) and (4.56) as premises of the $(How_{gen})_{\mathtt{Q}}$ rule, being.

$$\frac{\begin{array}{l}\Gamma_1 \vdash [\mathscr{Q}_1, e_1]\ R^H\ [\mathscr{S}_1, \ell_1] : A_1 \\ \vdots \quad \vdots \quad \vdots \\ \Gamma_j, \Delta \vdash [\mathscr{Q}_j \otimes \mathscr{U}, e_j\{f/x\}]\ R^H\ [\mathscr{S}_j \otimes \mathscr{R}, \ell_j\{h/x\}] : A_j \\ \vdots \quad \vdots \quad \vdots \\ \Gamma_N \vdash [\mathscr{Q}_N, e_N]\ R^H\ [\mathscr{S}_N, \ell_N] : A_N \\ \qquad \Gamma, \Delta \vdash [\mathscr{S} \otimes \mathscr{R}, \texttt{cnstr}\,(\ell_1, \ell_2 \ldots \ell\{h/x\} \ldots \ell_N)]\ R\ [\mathscr{W} \otimes \mathscr{R}, g\{h/x\}] : A\end{array}}{\Gamma, x : B \vdash [\mathscr{Q} \otimes \mathscr{U}, e\{f/x\}]\ R^H\ [\mathscr{W} \otimes \mathscr{R}, g\{h/x\}] : A} \quad (4.57)$$

which is the thesis. □

Lemma 4.7 (Quantum key lemma)**.** *Howe's extension of similarity between quantum closures – denoted by the symbol $\preceq^H$ – has the simulation property.*

Indeed we will show, for each couple of quantum closures $\emptyset \vdash [\mathscr{Q}, e] : A$, $\emptyset \vdash [\mathscr{R}, h] : A \in \mathcal{T}_A^{[\mathtt{QC}]\,\ell QST_\lambda}$ the more general property

$$\emptyset \vdash [\emptyset, e]\ {\preceq_{\texttt{bool}}}^H\ [\emptyset, h] : \texttt{bool} \Rightarrow$$
$$\forall \mathtt{b} \in \mathcal{V}_{\texttt{bool}}^{\ell PST_\lambda}, \mathcal{P}_{\ell QST_\lambda}\ (\ (\widehat{[\emptyset, e]}, \texttt{bool}), a_{\mathtt{b}}, (\widehat{[\emptyset, \mathtt{b}]}, \texttt{bool})\ ) \leq$$
$$\leq \mathcal{P}_{\ell QST_\lambda}\left((\widehat{[\emptyset, h]}, \texttt{bool}), a_{\mathtt{b}}, ({\preceq_{\texttt{bool}}}^H(\widehat{[\emptyset, \mathtt{b}]}), \texttt{bool})\right) \quad (4.58a)$$

$$\emptyset \vdash [\mathscr{Q}, \mathsf{V}] \preceq_{\mathtt{qbit}^n}{}^{H} [\mathscr{R}, h] : \mathtt{qbit}^n \Rightarrow$$
$$\forall\, [\mathscr{W}, g] \in \mathcal{T}^{[\mathtt{QC}]\,\ell QST_\lambda}_{s:\mathtt{qbit}^n, E},\ \mathcal{P}_{\ell QST_\lambda}\left(\left(\widehat{[\mathscr{Q}, \mathsf{V}]}, \mathtt{qbit}^n\right), a_{\mathtt{Q}[\mathscr{W},g]}, \left([\mathscr{Q} \otimes \mathscr{W}, g\{\mathsf{V}/s\}], E\right)\right) \leq$$
$$\leq \mathcal{P}_{\ell QST_\lambda}\left(\left(\widehat{[\mathscr{R}, h]}, \mathtt{qbit}\right), a_{\mathtt{Q}[\mathscr{W},g]}, \left(\preceq_g{}^{H}([\mathscr{Q} \otimes \mathscr{W}, g\{\mathsf{V}/s\}]), E\right)\right) \quad (4.58b)$$

$$\emptyset \vdash [\mathscr{Q}, \lambda\, x.f] \preceq_{B \multimap A}{}^{H} [\mathscr{R}, \lambda\, x.h] : B \multimap A \Rightarrow \forall\, [\mathscr{U}, v] \in \mathcal{T}^{[\mathtt{QC}]\,\ell QST_\lambda}_{B},$$
$$\mathcal{P}_{\ell QST_\lambda}\left(\left(\widehat{[\mathscr{Q}, \lambda\, x.f]}, B \multimap A\right), a_{@[\mathscr{U},v]}, \left([\mathscr{Q} \otimes \mathscr{U}, f\{v/x\}], A\right)\right) \leq$$
$$\leq \mathcal{P}_{\ell QST_\lambda}\left(\left(\widehat{[\mathscr{R}, \lambda\, x.h]}, B \multimap A\right), a_{@[\mathscr{U},v]}, \left(\preceq_A{}^{H}([\mathscr{R} \otimes \mathscr{U}, f\{v/x\}]), A\right)\right) \quad (4.58c)$$

$$\emptyset \vdash [\mathscr{Q}, \langle v_1, v_2\rangle] \preceq_{A\otimes B}{}^{H} [\mathscr{R}, \langle w_1, w_2\rangle] : A \otimes B \Rightarrow \forall\, [\mathscr{W}, g] \in \mathcal{T}^{[\mathtt{QC}]\,\ell QST_\lambda}_{x:A,y:B;E}$$
$$\mathcal{P}_{\ell QST_\lambda}\left(\left(\widehat{[\mathscr{Q}, \langle v_1, v_2\rangle}], A \otimes B\right), a_{\otimes[\mathscr{W},g]}, \left([\mathscr{Q} \otimes \mathscr{W}, g\{v_1/x, v_2/y\}], E\right)\right) \leq$$
$$\mathcal{P}_{\ell QST_\lambda}\left(\left(\widehat{[\mathscr{R}, \langle w_1, w_2\rangle}], A \otimes B\right), a_{\otimes[\mathscr{W},g]}, \left(\preceq_E{}^{H}([\mathscr{R} \otimes \mathscr{W}, g\{v_1/x, v_2/y\}]), E\right)\right) \quad (4.58d)$$

$$\emptyset \vdash [\mathscr{Q}, e] \preceq_A [\mathscr{R}, h] : A \ \wedge\ [\mathscr{Q}, e] \Downarrow \{[\mathscr{Q}_i, v_i]^{p_i}\}_{i\in\mathcal{I}} \Rightarrow$$
$$\Rightarrow \left([\mathscr{R}, h] \Downarrow \{[\mathscr{R}_m, w_m]^{q_m}\}_{m\in\mathcal{M}} \ \wedge \forall X \in \mathcal{V}^{[\mathtt{QC}]\,\ell QST_\lambda}_{A},\right.$$
$$\left.\mathcal{P}_{\ell QST_\lambda}\left(([\mathscr{Q}, e], A), a_{eval}, (X, A)\right) \leq \mathcal{P}_{\ell QST_\lambda}\left(([\mathscr{R}, h], A), a_{eval}, (\preceq_A{}^{H}(X), A)\right)\right) \quad (4.58e)$$

Proof. We carry out the proof by induction on the big–step-reduction rules of ℓQST_λ

- If $e = \mathtt{b}$ is a boolean constant, the quantum register is not involved in the definition of similarity, which is identical to that given in the probabilistic environment (3.38a). Indeed, if $\emptyset$ denote the empty quantum register, we can always set the identity $[\emptyset, \mathtt{b}] = \mathtt{b}$. To become familiar with quantum closures notation we formally write the statement as

$$\emptyset \vdash [\emptyset, e] \preceq_{\mathtt{bool}}{}^{H} [\emptyset, h] : \mathtt{bool} \Rightarrow, \forall \mathtt{b} \in \mathcal{V}^{\ell ST_\lambda}_{\mathtt{bool}}$$
$$\mathcal{P}_{\ell QST_\lambda}\left(\left(\widehat{[\emptyset, e]}, \mathtt{bool}\right), a_{\mathtt{b}}, \left(\widehat{[\emptyset, \mathtt{b}]}, \mathtt{bool}\right)\right) \leq$$

$$\mathcal{P}_{\ell QST_\lambda}\left((\widehat{[\emptyset,h]},\mathtt{bool}),a_{\mathtt{b}},(\widehat{[\emptyset,\mathtt{b}]},\mathtt{bool})\right). \quad (4.59)$$

The proof is the same as in probabilistic case (3.38a).

◇ If $e = \mathsf{V}$, the statement to prove is (4.58b). Under this condition the hypothesis $\emptyset \vdash [\mathscr{Q},\mathsf{V}] \preceq_{\mathtt{qbit}^n}{}^H [\mathscr{R},h] : \mathtt{qbit}^n$ has, as last rule, $(How_{va2})_{\mathtt{Q}}$ from Figure 4.6. Thus $\emptyset \vdash [\mathscr{Q},\mathsf{V}] \preceq_{\mathtt{qbit}^n} [\mathscr{R},h] : \mathtt{qbit}^n$ holds and, by definition of simulation for quantum variables (4.44), $\forall\, [\mathscr{W},g] \in \mathcal{T}^{[\mathtt{QC}]\,\ell QST_\lambda}_{s:\mathtt{qbit}^n,E}$, $[\mathscr{R},h] \in \preceq_E ([\mathscr{Q}\otimes\mathscr{W}, g\{\mathsf{V}/s\}])$.

Therefore, recalling that, by Lemma 2.17, $\preceq \subseteq \preceq^H$, we have

$$[\mathscr{R},h] \in \preceq_E ([\mathscr{Q}\otimes\mathscr{W}, g\{\mathsf{V}/s\}]) \subseteq \preceq_E{}^H ([\mathscr{Q}\otimes\mathscr{W}, g\{\mathsf{V}/s\}]) \quad (4.60)$$

which is the thesis.

◇ If $e = \lambda x.f$, the statement (4.58c) must be proved and hypothesis

$$\emptyset \vdash [\mathscr{Q},\lambda x.f] \preceq_{B\multimap A}{}^H [\mathscr{R},\lambda x.h] : B \multimap A \quad (4.61)$$

hypothesis must have $(How_{abs})_{\mathtt{Q}}$ as last rule – see Figure 4.6. Thus, the following premises hold

$$\begin{aligned}(abs:1)-&\quad x : B \vdash [\mathscr{Q},f] \preceq_A{}^H [\mathscr{W},g] : A\\ (abs:2)-&\quad \emptyset \vdash [\mathscr{W},\lambda x.g] \preceq_{B\multimap A} [\mathscr{R},h] : B \multimap A.\end{aligned}$$

From $(abs:1)$, by compatibility of $\preceq_{B\multimap A}{}^H$ it can be derived that $\emptyset \vdash [\mathscr{Q},\lambda x.f] \preceq_{B\multimap A}{}^H [\mathscr{W},\lambda x.g] : B \multimap A$, which has as a consequence the result $[\mathscr{W},\lambda x.g] \in \preceq_{B\multimap A}{}^H ([\mathscr{Q},\lambda x.f])$.

Furthermore from the properties of Howe's relation (Lemma 2.17), it follows that $\preceq_{B\multimap A} ([\mathscr{Q},\lambda x.f]) \subseteq \preceq_{B\multimap A}{}^H ([\mathscr{Q},\lambda x.f])$, whence the property

$$\preceq_{B\multimap A} \left(\preceq_{B\multimap A}{}^H\right) \subseteq \preceq_{B\multimap A}{}^H.$$

By latter argument, using $(abs:2)$ we find $[\mathscr{R},h] \in \preceq_{B\multimap A} ([\mathscr{W},\lambda x.g]) \subseteq \preceq_{B\multimap A}{}^H ([\mathscr{W},\lambda x.g]) \subseteq \preceq_{B\multimap A}{}^H ([\mathscr{Q},\lambda x.f])$. Since $f\{v/x\}$ is a single term, for each $X \subseteq \mathcal{T}_A^{[\mathtt{QC}]\,\ell QST_\lambda}$ there are two possibilities:

▶ $[\mathscr{Q}\otimes\mathscr{U}, f\{v/x\}] \notin X \subseteq \mathcal{T}_A^{[\mathtt{QC}]\,\ell QST_\lambda}$ that entails thesis, since the left–hand–side of (4.58c) is zero.

▶ $[\mathscr{Q}\otimes\mathscr{U}, f\{v/x\}] \in X$ whence, recalling that we showed $[\mathscr{R}, h] \in \preceq_{B\multimap A}{}^H([\mathscr{Q}, \lambda\, x.f])$, implies that both of the terms of the inequality (4.58c) are equal to one.

◇ $e = \langle v_1, v_2\rangle$ –Here we should prove the statement in form (4.58d) knowing that hypothesis

$$\emptyset \vdash [\mathscr{Q}, \langle v_1, v_2\rangle] \preceq_{A\otimes B}{}^H [\mathscr{R}, \langle w_1, w_2\rangle] : A\otimes B, \tag{4.62}$$

with $\mathscr{Q} = \mathscr{Q}_1\otimes\mathscr{Q}_2$, is a consequence of a $(How_{gen})_{\mathtt{Q}}$ rule for pairs as it is shown just below

$$\begin{aligned}(pair:1)-&\quad \emptyset \vdash [\mathscr{Q}_1, v_1] \preceq_A{}^H [\mathscr{W}_1, \nu_1] : A,\ \emptyset \vdash [\mathscr{Q}_2, v_2] \preceq_B{}^H [\mathscr{W}_2, \nu_2] : B\\ (pair:2)-&\quad \emptyset \vdash [\mathscr{W}_1\otimes\mathscr{W}_2, \langle\nu_1,\nu_2\rangle] \preceq_{A\otimes B} [\mathscr{R}, \langle w_1, w_2\rangle] : A\otimes B.\end{aligned}$$

Statement $(pair:1)$ entails, by compatibility of Howe's relation, that

$$[\mathscr{Q}_1\otimes\mathscr{Q}_2, \langle v_1, v_2\rangle] \preceq_{A\otimes B}{}^H [\mathscr{W}_1\otimes\mathscr{W}_2, \langle\nu_1,\nu_2\rangle],$$

thus, being $\mathscr{W} = \mathscr{W}_1\otimes\mathscr{W}_2$, by definition of Howe's relation, we have $\forall\,[\mathscr{U}, f] \in \mathcal{T}^{[\mathtt{QC}]\,\ell QST_\lambda}_{x:A,y:B;E}$

$$\begin{aligned}&\mathcal{P}_{\ell QST_\lambda}\left(([\mathscr{Q}, \widehat{\langle v_1, v_2\rangle}], A\otimes B), a_{\otimes[\mathscr{U},f]}, ([\mathscr{Q}\otimes\mathscr{U}, f\{v_1/x, v_2/y\}], E)\right) \leq\\ &\mathcal{P}_{\ell QST_\lambda}\left(([\mathscr{W}, \widehat{\langle\nu_1,\nu_2\rangle}], A\otimes B), a_{\otimes[\mathscr{U},f]}, (\preceq_E{}^H([\mathscr{Q}\otimes\mathscr{U}, f\{v_1/x, v_2/y\}]), E)\right).\end{aligned} \tag{4.63}$$

Moreover, using on $(pair:2)$ the definition of simulation we get $\forall\,[\mathscr{U}, f] \in \mathcal{T}^{[\mathtt{QC}]\,\ell QST_\lambda}_{x:A,y:B;E}$

$$\begin{aligned}&\mathcal{P}_{\ell QST_\lambda}\left(([\mathscr{W}, \widehat{\langle\nu_1,\nu_2\rangle}], A\otimes B), a_{\otimes[\mathscr{U},f]}, ([\mathscr{W}\otimes\mathscr{U}, f\{\nu_1/x, \nu_2/y\}], E)\right) \leq\\ &\mathcal{P}_{\ell QST_\lambda}\left(([\mathscr{R}, \widehat{\langle w_1, w_2\rangle}], A\otimes B), a_{\otimes[\mathscr{U},f]}, (\preceq_E([\mathscr{W}\otimes\mathscr{U}, f\{\nu_1/x, \nu_2/y\}]), E)\right).\end{aligned} \tag{4.64}$$

Joining the relations (4.63) and (4.64) one has, $\forall\,[\mathscr{U},f]\in\mathcal{T}^{[\mathtt{QC}]\,\ell QST_\lambda}_{x:A,y:B;E}$,

$$\begin{aligned}&[\mathscr{W}\otimes\mathscr{U},f\{\nu_1/x,\nu_2/y\}]\in\preceq_E{}^H([\mathscr{Q}\otimes\mathscr{U},f\{v_1/x,v_2/y\}])\\&[\mathscr{R}\otimes\mathscr{U},\langle w_1,w_2\rangle]\in\preceq_E([\mathscr{W}\otimes\mathscr{U},f\{\nu_1/x,\nu_2/y\}])\end{aligned}\tag{4.65}$$

whence considering the property of Howe relation $\preceq\subseteq\preceq^H$ (Lemma 2.17) can be derived that

$$[\mathscr{R}\otimes\mathscr{U},\langle w_1,w_2\rangle]\in\preceq_E{}^H([\mathscr{Q}\otimes\mathscr{U},f\{v_1/x,v_2/y\}])\,,$$

which is the thesis.

◇ If $e=\mathtt{meas}_m\,(v)$ the statement to be proved is

$$\begin{aligned}\emptyset\vdash[\mathscr{Q},\mathtt{meas}_m\,(v)]\preceq_{\mathtt{qbit}^n\otimes\mathtt{bool}}{}^H[\mathscr{R},h]:\mathtt{qbit}^n\otimes\mathtt{bool}\wedge[\mathscr{Q},\mathtt{meas}_m\,(v)]\Downarrow\mathscr{E}\;\Rightarrow\\\Big([\mathscr{R},h]\Downarrow\mathscr{H}\;\wedge\forall X\subseteq\mathcal{V}^{[\mathtt{QC}]\,\ell QST_\lambda}_{\mathtt{qbit}^n\otimes\mathtt{bool}},\\\mathcal{P}_{\ell QST_\lambda}\left(([\mathscr{Q},\mathtt{meas}_m\,(v)]\,,\mathtt{bool}),a_{eval},(X,\mathtt{bool})\right)\\\leq\mathcal{P}_{\ell QST_\lambda}\left(([\mathscr{R},h]\,,\mathtt{bool}),a_{eval},(\preceq_A{}^H(X),\mathtt{bool})\right)\Big)\end{aligned}\tag{4.66}$$

being, by the rule $(mea\Downarrow)_{\mathtt{Q}}$,

$$\mathscr{E}=\{[\mathtt{MS}^v_{\mathtt{ff}}(\mathscr{Q}),\langle\mathsf{V}_m,\mathtt{tt}\rangle]^{\mathtt{PR}^v_{\mathtt{tt}}(\mathscr{Q})},[\mathtt{MS}^v_{\mathtt{tt}}(\mathscr{Q}),\langle\mathsf{V}_m,\mathtt{ff}\rangle]^{\mathtt{PR}^v_{\mathtt{ff}}(\mathscr{Q})}\}.$$

Since the first part of the hypothesis is consequence of the rule $(How_{men})_{\mathtt{Q}}$, it stems from the premises

$$\begin{aligned}(mea:1)-&\quad v:\mathtt{qbit},\nu:\mathtt{qbit}\vdash[\mathscr{Q},v]\preceq_{\mathtt{qbit}^{n+1}}{}^H[\mathscr{W},\nu]:\mathtt{qbit}^{n+1}\\(mea:2)-&\quad\emptyset\vdash[\mathscr{W},\mathtt{meas}_m\,(\nu)]\preceq_{\mathtt{qbit}^n\otimes\mathtt{bool}}[\mathscr{R},h]:\mathtt{qbit}^n\otimes\mathtt{bool}\end{aligned}$$

From the premise $(mea:1)$ it follows, by compatibility of $\preceq_{\mathtt{qbit}}{}^H$, the relation $\emptyset\vdash[\mathscr{Q},\mathtt{meas}_m\,(v)]\preceq_{\mathtt{qbit}^n\otimes\mathtt{bool}}{}^H[\mathscr{W},\mathtt{meas}_m\,(\nu)]:\mathtt{qbit}^n\otimes\mathtt{bool}$, thus applying induction hypothesis on the the steps of big–steps semantics rule one obtains

$$[\mathscr{W},\mathtt{meas}_m\,(\nu)]\Downarrow\mathscr{G}\;\wedge\;\forall X\in\mathcal{V}^{[\mathtt{QC}]\,\ell QST_\lambda}_{\mathtt{qbit}^n\otimes\mathtt{bool}},$$

$$\mathcal{P}_{\ell QST_\lambda}\left(\left(\left[\mathscr{Q},\mathtt{meas}_m\left(v\right)\right],\mathtt{qbit}\right),a_{eval},\left(X,\mathtt{bool}\right)\right)=$$
$$=\mathtt{PR}_X^v(\mathscr{Q})\leq\mathtt{PR}^{\nu}_{\preceq_{\mathtt{qbit}^n\otimes\mathtt{bool}}{}^H(X)}(\mathscr{W})=$$
$$=\mathcal{P}_{\ell QST_\lambda}\left(\left(\left[\mathscr{W},\mathtt{meas}_m\left(\nu\right)\right],\mathtt{qbit}\right),a_{eval},\left(\preceq_{\mathtt{qbit}^n\otimes\mathtt{bool}}{}^H(X),\mathtt{bool}\right)\right),\quad(4.67)$$

where $\mathscr{G}=\{[\mathtt{MS}^v_{\mathtt{tt}}(\mathscr{Q}),\langle\mathsf{V}_m,\mathtt{tt}\rangle]^{\mathtt{PR}^v_{\mathtt{tt}}(\mathscr{Q})},[\mathtt{MS}^v_{\mathtt{ff}}(\mathscr{Q}),\langle\mathsf{V}_m,\mathtt{ff}\rangle]^{\mathtt{PR}^v_{\mathtt{ff}}(\mathscr{Q})}\}$ and $\mathtt{PR}^v_X(\mathscr{Q})$ $=\sum_{\mathtt{b}\in X}\mathtt{PR}^v_{\mathtt{b}}(\mathscr{Q})$. Whereas from $(mea:2)$ by definition of similarity we derive the inequality

$$\forall Y\in\mathcal{V}^{[\mathtt{QC}]\,\ell QST_\lambda}_{\mathtt{qbit}^n\otimes\mathtt{bool}},$$
$$\mathcal{P}_{\ell QST_\lambda}\left(\left(\left[\mathscr{W},\mathtt{meas}_m\left(\nu\right)\right],\mathtt{qbit}^n\otimes\mathtt{bool}\right),a_{eval},\left(Y,\mathtt{qbit}^n\otimes\mathtt{bool}\right)\right)=\mathtt{PR}^{\nu}_Y(\mathscr{W})\leq$$
$$\mathtt{PR}^h_{\preceq_{\mathtt{qbit}^n\otimes\mathtt{bool}}(Y)}(\mathscr{R})=\mathcal{P}_{\ell QST_\lambda}\left(\left(\left[\mathscr{R},h\right],\mathtt{qbit}\right),a_{eval},\left(\preceq_{\mathtt{bool}}\left(Y\right),\mathtt{bool}\right)\right),\quad(4.68)$$

Setting $Y=\preceq_{\mathtt{bool}}{}^H(X)$ the following chain of inequalities stems

$$\mathscr{E}(X)=\mathtt{PR}^v_X(\mathscr{Q})\leq\mathtt{PR}^{\nu}_{\preceq_{\mathtt{qbit}^n\otimes\mathtt{bool}}{}^H(X)}(\mathscr{W})\leq\mathtt{PR}^{\nu}_{\preceq_{\mathtt{qbit}^n\otimes\mathtt{bool}}\left(\preceq_{\mathtt{qbit}^n\otimes\mathtt{bool}}{}^H(X)\right)}(\mathscr{R})=$$
$$=\mathscr{H}(\preceq_{\mathtt{qbit}^n\otimes\mathtt{bool}}\left(\preceq_{\mathtt{qbit}^n\otimes\mathtt{bool}}{}^H(X)\right)).\quad(4.69)$$

If $\mathtt{Sup}\left(\mathscr{E}\right)\cap X=\emptyset$ then the inequality (4.66) is necessarily true, otherwise from (4.69) it follows that $\forall X\in\mathcal{V}^{[\mathtt{QC}]\,\ell QST_\lambda}_{\mathtt{qbit}^n\otimes\mathtt{bool}}$, $\mathscr{E}(X)\leq\mathscr{H}(\preceq_{\mathtt{qbit}^n\otimes\mathtt{bool}}{}^H(X))$, namely the thesis.

◇ If $e=\mathtt{new}\left(v\right)$ the statement of the key lemma will be

$$\emptyset\vdash\left[\mathscr{Q},\mathtt{new}\left(v\right)\right]\preceq_{\mathtt{qbit}}{}^H\left[\mathscr{R},h\right]:\mathtt{qbit}\ \wedge\ \left[\mathscr{Q},\mathtt{new}\left(v\right)\right]\Downarrow\mathscr{E}\ \Rightarrow\Bigg(\left[\mathscr{R},h\right]\Downarrow\mathscr{H}\wedge$$
$$\forall X\subseteq\mathcal{V}^{[\mathtt{QC}]\,\ell QST_\lambda}_{\mathtt{qbit}},\mathcal{P}_{\ell QST_\lambda}\left(\left(\left[\mathscr{Q},\mathtt{new}\left(v\right)\right],\mathtt{bool}\right),a_{eval},\left(X,\mathtt{bool}\right)\right)$$
$$\leq\mathcal{P}_{\ell QST_\lambda}\left(\left(\left[\mathscr{R},h\right],\mathtt{bool}\right),a_{eval},\left(\preceq_A{}^H(X),\mathtt{bool}\right)\right)\quad(4.70)$$

where, from $(new\Downarrow)_{\mathtt{Q}}$ we know the semantics of the quantum closure $[\mathscr{Q},e]$ to have the structure $\mathscr{E}=\{[\mathtt{NW}^r_v(\mathscr{Q}),r]^1\}$. Going back through the derivation tree we found that the hypotesys must come from the following premises

$$\begin{array}{ll}(new:1)- & \emptyset\vdash\left[\mathscr{Q},v\right]\preceq_{\mathtt{bool}}{}^H\left[\mathscr{W},\nu\right]:\mathtt{bool}\\ (new:2)- & \Delta\vdash\left[\mathscr{W},\mathtt{new}\left(\nu\right)\right]\preceq_{\mathtt{qbit}}\left[\mathscr{R},h\right]:\mathtt{qbit}.\end{array}\quad(4.71)$$

The hypothesis $(new:1)$, via the application of the compatibility of Howe's relation (Lemma 2.16) gives us the result

$$\emptyset \vdash [\mathscr{Q}, \mathtt{new}\,(v)] \preceq_{\mathtt{bool}}{}^{H} [\mathscr{W}, \mathtt{new}\,(\nu)] : \mathtt{bool} \tag{4.72}$$

and imposing on (4.72) the induction hypothesis we get

$$\begin{aligned}\emptyset \vdash [\mathscr{Q}, \mathtt{new}\,(v)] \preceq_{\mathtt{bool}}{}^{H} [\mathscr{W}, \mathtt{new}\,(\nu)] : \mathtt{bool} \wedge [\mathscr{Q}, \mathtt{new}\,(v)] \Downarrow \{[\mathtt{NW}_v^r(\mathscr{Q}), r]^1\} \Rightarrow \\ [\mathscr{W}, \mathtt{new}\,(\nu)] \Downarrow \mathscr{G} \ \wedge\ \forall X \in \mathcal{V}_{\mathtt{bool}}^{[\mathtt{QC}]\,\ell QST_\lambda} \mathscr{E}(X) \leq \mathscr{G}(\preceq_{\mathtt{bool}}{}^{H}(X)),\end{aligned} \tag{4.73}$$

where the semantics $\mathscr{G}$ is the one-value set $\{[\mathtt{NW}_\nu^s(\mathscr{W}), s]^1\}$. Moreover, from $(new:2)$, by the definition of similarity we may write

$$\forall Y \in \mathcal{V}_{\mathtt{qbit}}^{[\mathtt{QC}]\,\ell QST_\lambda}, \mathscr{G}(Y) \leq \mathscr{H}(\preceq_{\mathtt{qbit}}(Y)), \tag{4.74}$$

whence, if we take $Y = \preceq_{\mathtt{qbit}}{}^{H}(X)$, joining the inequalities (4.73) and (4.74) yields

$$\forall X \in \mathcal{V}_{\mathtt{qbit}}^{[\mathtt{QC}]\,\ell QST_\lambda}, \mathscr{E}(X) \leq \mathscr{G}(\preceq_{\mathtt{qbit}}{}^{H}(X)) \leq \mathscr{H}(\preceq_{\mathtt{qbit}}(\preceq_{\mathtt{qbit}}{}^{H}(X))). \tag{4.75}$$

Considering that Lemma 2.17 entails $\preceq(\preceq^H) = \preceq^H$, two possibilities may arise:

- if $[\mathtt{NW}_v^r(\mathscr{Q}), r] \notin X$, then the thesis (4.70) necessarily holds;
- if $[\mathtt{NW}_v^r(\mathscr{Q}), r] \in X$, then every the term in (4.75) is equal to 1 and the thesis (4.70) is fulfilled, likewise.

◇ If $e = U\,(v)$ we must prove the statement

$$\begin{aligned}\left(\Gamma \vdash [\mathscr{Q}, U\,(v)] \preceq_{\mathtt{qbit}^{\mathtt{a}(U)}}{}^{H} [\mathscr{R}, h] : \mathtt{qbit}^{\mathtt{a}(U)} \wedge [\mathscr{Q}, U\,(v)] \Downarrow \mathscr{E}\right) \Rightarrow \\ \left([\mathscr{R}, h] \Downarrow \mathscr{H} \ \wedge\ \forall X \in \mathcal{V}_{\mathtt{qbit}^{\mathtt{a}(U)}}^{[\mathtt{QC}]\,\ell QST_\lambda}, \mathscr{E}(X) \leq \mathscr{H}\left(\preceq_{\mathtt{qbit}^{\mathtt{a}(U)}}{}^{H}(X)\right)\right),\end{aligned} \tag{4.76}$$

where from rule $(uni \Downarrow)_{\mathtt{Q}}$ we know that $\mathscr{E}$ has the structure $\mathscr{E} = \left\{[U_v\mathscr{Q}, v]^1\right\}$. Here the (first sentence of the) thesis comes from the premises

$$\begin{aligned}(uni:1)- &\quad \Gamma \vdash [\mathscr{Q}, v] \preceq_{\mathtt{qbit}^{\mathtt{a}(U)}}{}^{H} [\mathscr{W}, \nu] : \mathtt{qbit}^{\mathtt{a}(U)} \\ (uni:2)- &\quad \emptyset \vdash [\mathscr{W}, U\,(\nu)] \preceq_{\mathtt{qbit}^{\mathtt{a}(U)}} [\mathscr{R}, h] : \mathtt{qbit}^{\mathtt{a}(U)}.\end{aligned}$$

From $(uni:1)$, applying induction hypothesis one gets the condition $\forall\,[\mathscr{U},f] \in \mathcal{T}^{[\mathtt{QC}]\,\ell QST_\lambda}_{\Gamma,x:\mathtt{qbit}^{\mathtt{a}(U)},A}\ [\mathscr{Q}\otimes\mathscr{U}, f\{v/x\}] \preceq_A{}^H [\mathscr{W}\otimes\mathscr{U}, f\{\nu/x\}]$. Given that, from (4.76) $[\mathscr{Q}, U\,(v)] \Downarrow \mathscr{E}$, taking $[\mathscr{U},f]=[\emptyset,U]$, yields

$$[\mathscr{W}, U\,(\nu)] \Downarrow \mathscr{G}\ \wedge\ \forall X \in \mathcal{V}^{[\mathtt{QC}]\,\ell QST_\lambda}_{\mathtt{qbit}^{\mathtt{a}(U)}}, \mathscr{E}(X) \leq \mathscr{G}(\preceq_{\mathtt{qbit}^{\mathtt{a}(U)}}{}^H(X)), \tag{4.77}$$

being – from evaluation rule $(uni \Downarrow)_{\mathtt{Q}}$ – $\mathscr{G} = \left\{[U_\nu \mathscr{W}, \nu]^1\right\}$. Moreover $(uni:2)$ entails, by definition of similarity, that

$$\forall Y \in \mathcal{V}^{[\mathtt{QC}]\,\ell QST_\lambda}_{\mathtt{qbit}^{\mathtt{a}(U)}}, \mathscr{G}(Y) \leq \mathscr{H}(\preceq_{\mathtt{qbit}^{\mathtt{a}(U)}}(Y)), \tag{4.78}$$

having denoted by $\mathscr{H}$ the semantics of $[\mathscr{R},h]$. Now, taking $Y = \preceq_{\mathtt{qbit}^{\mathtt{a}(U)}}{}^H(X)$ and joyning (4.77) and (4.78) gives the chain

$$\forall X \in \mathcal{V}^{[\mathtt{QC}]\,\ell QST_\lambda}_{\mathtt{qbit}^{\mathtt{a}(U)}}, \mathscr{E}(X) \leq \mathscr{G}(\preceq_{\mathtt{qbit}^{\mathtt{a}(U)}}{}^H(X)) \leq \mathscr{H}(\preceq_{\mathtt{qbit}^{\mathtt{a}(U)}}(\preceq_{\mathtt{qbit}^{\mathtt{a}(U)}}{}^H(X))), \tag{4.79}$$

which is the thesis, since by Lemma 2.17 we know that $\forall X,\ \preceq_A(\preceq_A{}^H(X)) = \preceq_A{}^H(X)$.

Precisely, two cases may occur for any X:

- $[U\mathscr{Q}, v] \notin X$, which makes (4.79) necessarily true.
- $[U\mathscr{Q}, v] \in X$ which entails all the terms in (4.79) are equal to one, and $h \Downarrow \left\{[U_h\mathscr{R}, h]^1\right\}$, with $[U_h\mathscr{R}, h] \in \preceq_{\mathtt{qbit}^{\mathtt{a}(U)}}{}^H(X)$.

◇ If $e = \mathtt{cmp}\,(v_1, v_2)$, the statement of key lemma is

$$\left(\emptyset \vdash [\mathscr{Q}, \mathtt{cmp}\,(v_1,v_2)] \preceq_{\mathtt{qbit}^{n+m}}{}^H [\mathscr{R},h] : \mathtt{qbit}^{n+m} \wedge [\mathscr{Q}, \mathtt{cmp}\,(v_1,v_2)] \Downarrow \mathscr{E}\right) \Rightarrow$$
$$\left([\mathscr{R},h] \Downarrow \mathscr{H}\ \wedge\ \forall X \in \mathcal{V}^{[\mathtt{QC}]\,\ell QST_\lambda}_{\mathtt{qbit}^{n+m}}, \mathscr{E}(X) \leq \mathscr{H}(\preceq_{\mathtt{qbit}^{n+m}}{}^H(X))\right), \tag{4.80}$$

being, for rule $(cmp \Downarrow)_{\mathtt{Q}}$, $\mathscr{E} = \{[\mathscr{Q}, v_1 \cdot v_2]^1\}$.

The first hypothesis results as an application of the (How_{gen}) and it has premises

$$\begin{aligned} (cmp:1)- \quad & \emptyset \vdash [\mathscr{Q}_1, v_1] \preceq_{\mathtt{qbit}^n}{}^H [\mathscr{W}_1, \nu_1] : \mathtt{qbit}^n, \\ & \emptyset \vdash [\mathscr{Q}_2, v_2] \preceq_{\mathtt{qbit}^m}{}^H [\mathscr{W}_2, \nu_2] : \mathtt{qbit}^m \\ (cmp:2)- \quad & \emptyset \vdash [\mathscr{W}_1 \otimes \mathscr{W}_2, \mathtt{cmp}\,(\nu_1,\nu_2)] \preceq_{\mathtt{qbit}^n} [\mathscr{R}, h] : \mathtt{qbit}^{n+m}, \end{aligned} \tag{4.81}$$

while the second hypothesis has not premises, coming from the rule $(cmp \Downarrow)_{\mathtt{Q}}$.

Starting from the hypothesis $(cmp : 1)$ we may apply the induction on the subterms, which yields the semantics of $[\mathscr{W}_1 \otimes \mathscr{W}_2, \mathtt{cmp}\,(\nu_1, \nu_2)]$ in the form $\mathscr{G} = [\mathscr{W}_1 \otimes \mathscr{W}_2, \nu_1 \cdot \nu_2]$. The same hypothesis, recalling the compatibility of Howe's lifting, gives the relationship

$$\emptyset \vdash [\mathscr{Q}_1 \otimes \mathscr{Q}_2, \mathtt{cmp}\,(v_1, v_2)] \preceq_{\mathtt{qbit}^{n+m}}{}^{H} [\mathscr{W}_1 \otimes \mathscr{W}_2, \mathtt{cmp}\,(\nu_1, \nu_2)] : \mathtt{qbit}^{n+m}. \tag{4.82}$$

Relation 4.82, still from induction hypothesis, gives

$$\forall X \in \mathcal{V}_{,}^{[\mathtt{QC}]\,\mathtt{qbit}^{n+m}} \mathscr{E}(X) \leq \mathscr{G}(\preceq_{\mathtt{qbit}^{n+m}}{}^{H}(X)). \tag{4.83}$$

From $(cmp : 2)$, by definition of similarity in quantum framework, it follows

$$\forall Y \in \mathcal{V}_{,}^{[\mathtt{QC}]\,\mathtt{qbit}^{n+m}} \mathscr{G}(Y) \leq \mathscr{H}(\preceq_{\mathtt{qbit}}\,(Y)), \tag{4.84}$$

where $[\mathscr{R}, h] \Downarrow \mathscr{H}$. Now taking $Y = \preceq_{\mathtt{qbit}^{n+m}}{}^{H}(X)$, from (4.82) and (4.84) we have the inequalities chain

$$\forall X \in \mathcal{V}_{,}^{[\mathtt{QC}]\,\mathtt{qbit}^{n+m}} \mathscr{E}(X) \leq \mathscr{G}(\preceq_{\mathtt{qbit}^{n+m}}{}^{H}(X)) \leq \mathscr{H}(\preceq_{\mathtt{qbit}^{n+m}} \left(\preceq_{\mathtt{qbit}^{n+m}}{}^{H}(X)\right)), \tag{4.85}$$

which gives the thesis (4.80), since by Lemma 2.17, $\forall X, \preceq_{\mathtt{qbit}^{n+m}} \left(\preceq_{\mathtt{qbit}^{n+m}}{}^{H}(X)\right)$ is equal to $\preceq_{\mathtt{qbit}^{n+m}}{}^{H}(X)$.

◇ If $e = \mathtt{if}\ f_1\ \mathtt{then}\ f_2\ \mathtt{else}\ f_2$, to reduce the size of the formula involved we will write e as $\mathtt{cnstr}\,(\{f_j\}_{j\in\mathcal{J}})$ where $\mathtt{cnstr}$ represents a generic syntactic constructor. Then we should prove the statement in form (4.58e), which is equivalent to show the following property

$$\begin{aligned} &\emptyset \vdash [\mathscr{Q} \otimes \mathscr{U}, \mathtt{cnstr}\,(\{f_j\}_{j\in\mathcal{J}})] \preceq_A{}^{H} [\mathscr{R}, h] : A \wedge [\mathscr{Q} \otimes \mathscr{U}, \mathtt{cnstr}\,(\{f_j\}_{j\in\mathcal{J}})] \Downarrow \mathscr{E} \\ &\Rightarrow \Big(\forall X \subseteq \mathcal{V}_A^{[\mathtt{QC}]\,\ell QST_\lambda}, \mathcal{P}_{\ell QST_\lambda}\left(([\mathscr{Q} \otimes \mathscr{U}, \mathtt{cnstr}\,(\{f_j\}_{j\in\mathcal{J}})], A), a_{eval}, (X, A)\right) \\ &\qquad \leq \mathcal{P}_{\ell QST_\lambda}\left(([\mathscr{R}, h], A), a_{eval}, (\preceq_A{}^{H}(X), A)\right) \Big) \end{aligned} \tag{4.86}$$

The semantics $\mathscr{E}$ in (4.86), is defined by the big–step reduction $(if \Downarrow)_{\mathtt{Q}}$ that we quote just below

$$\frac{[\mathscr{Q}, f_1] \Downarrow \mathscr{F}_1 \qquad [\mathscr{Q}_{\mathtt{tt}} \otimes \mathscr{U}, f_3] \Downarrow \mathscr{F}_3 \qquad [\mathscr{Q}_{\mathtt{ff}} \otimes \mathscr{U}, f_2] \Downarrow \mathscr{F}_2}{[\mathscr{Q} \otimes \mathscr{U}, \mathtt{if}\ f_1\ \mathtt{then}\ f_2\ \mathtt{else}\ f_3] \Downarrow \mathscr{F}_1(\mathtt{tt})\mathscr{F}_2 + \mathscr{F}_1(\mathtt{ff})\mathscr{F}_3},$$

where $\mathscr{F}_3 = \{[\mathscr{Q}_{\mathtt{tt}} \otimes \mathscr{U}_i, v_i]^{p_i}\}_{i\in\mathcal{I}}$, $\mathscr{F}_2 = \{[\mathscr{Q}_{\mathtt{ff}} \otimes \mathscr{U}_j, u_j]^{q_j}\}_{j\in\mathcal{J}}$ for conditional choice terms.

Referring us to Figure 4.6, we claim that the first hypothesis in the statement (4.86) must have $(How_{gen})_{\mathtt{Q}}$ as last rule. This one has a set of premises for each subterm of e plus a general rule stated in $(gen : 2)$

$$\begin{aligned}
(gen : 1)- \quad & \emptyset \vdash [\mathscr{Q}, f_1] \preceq_{\mathtt{bool}}{}^{H} [\mathscr{S}, g_1] : \mathtt{bool} \\
& \emptyset \vdash [\mathscr{U}, f_j] \preceq_{A}{}^{H} [\mathscr{W}, g_j] : A \qquad j = 2, 3 \\
(gen : 2)- \quad & \emptyset \vdash [\mathscr{S} \otimes \mathscr{W}, \mathtt{cnstr}\,(\{g_j\}_{j\in\mathcal{J}})] \preceq_{A} [\mathscr{R}, h] : A.
\end{aligned} \tag{4.87}$$

Therefore by induction hypothesis to subterms of $(gen : 1)$ we have

$$\begin{aligned}
&\left(\emptyset \vdash [\mathscr{Q}, f_1] \preceq_{\mathtt{bool}}{}^{H} [\mathscr{S}, g_1] : \mathtt{bool} \wedge [\mathscr{Q}, f_1] \Downarrow \mathscr{F}_1\right) \Rightarrow \\
&\Big([\mathscr{S}, g_1] \Downarrow \mathscr{G}_1 \wedge\ \forall X \subseteq \mathcal{V}_{\mathtt{bool}}^{[\mathtt{QC}]\,\ell QST_\lambda},\, \mathcal{P}_{\ell QST_\lambda}\left(([\mathscr{Q}, f_1], A), a_{eval}, (X, \mathtt{bool})\right) \leq \\
&\qquad\qquad \mathcal{P}_{\ell QST_\lambda}\left(([\mathscr{S}, g_1], \mathtt{bool}), a_{eval}, (\preceq_{\mathtt{bool}}{}^{H}(X), \mathtt{bool})\right)\Big)
\end{aligned} \tag{4.88}$$

$$\begin{aligned}
&\forall j \in \{2,3\},\ \left(\emptyset \vdash [\mathscr{U}, f_j] \preceq_{A}{}^{H} [\mathscr{W}, g_j] : A \wedge [\mathscr{U}, f_j] \Downarrow \mathscr{F}_j\right) \Rightarrow \\
&\Big([\mathscr{W}, g_j] \Downarrow \mathscr{G}_j \wedge\ \forall X_j \subseteq \mathcal{V}_{A}^{[\mathtt{QC}]\,\ell QST_\lambda},\, \mathcal{P}_{\ell QST_\lambda}\left(([\mathscr{U}, f_j], A), a_{eval}, (X_j, A)\right) \leq \\
&\qquad\qquad \mathcal{P}_{\ell QST_\lambda}\left(([\mathscr{W}, g_j], A), a_{eval}, (\preceq_{A}{}^{H}(X_j), A)\right)\Big)
\end{aligned} \tag{4.89}$$

As well as by induction on the size of big–step semantics, we may apply a suitable reduction rule whose premises are $[\mathscr{S}, g_1] \Downarrow \mathscr{G}_1$ and $([\mathscr{W}, g_j] \Downarrow \mathscr{G}_j)_{j\in\{2,3\}}$, to get the semantics $\mathscr{G}$ such that $[\mathscr{S} \otimes \mathscr{W}, \mathtt{cnstr}\,(\{g_j\}_{j\in\mathcal{J}})] \Downarrow \mathscr{G}$.

Now, using the definition of similarity on $(gen : 2)$, imposing $\mathscr{V} = \mathscr{S} \otimes \mathscr{W}$, lead us to the following inequality

$$(\emptyset \vdash [\mathscr{V}, \texttt{cnstr}\,(\{g_j\}_{j\in\mathcal{J}})] \preceq_A [\mathscr{R}, h] : A \ \wedge [\mathscr{V}, \texttt{cnstr}\,(\{g_j\}_{j\in\mathcal{J}})] \Downarrow \mathscr{G}) \ \Rightarrow$$
$$([\mathscr{R}, h] \Downarrow \mathscr{H} \wedge \forall Y \subseteq \mathcal{V}_A^{[\mathtt{QC}]\,\ell QST_\lambda}, \mathcal{P}_{\ell QST_\lambda}(([\mathscr{V}, \texttt{cnstr}\,(\{g_j\}_{j\in\mathcal{J}})], A), a_{eval}, (Y, A))$$
$$\leq \mathcal{P}_{\ell QST_\lambda}(([\mathscr{R}, h], A), a_{eval}, (\preceq_A (Y), A))) \quad (4.90)$$

Moreover, since the relation $\preceq^H$ is compatible, by Lemma 4.4, the premises $(gen : 1)$ entail as a further consequence that

$$\emptyset \vdash [\mathscr{Q}, e] \preceq_A{}^H [\mathscr{V}, \texttt{cnstr}\,(\{g_j\}_{j\in\mathcal{J}})] : A,$$

namely the inequality

$$\forall Z \subseteq \mathcal{V}_A^{[\mathtt{QC}]\,\ell QST_\lambda}, \mathcal{P}_{\ell QST_\lambda}(([\mathscr{Q}, e], A), a_{eval}, (Z, A)) \leq$$
$$\leq \mathcal{P}_{\ell QST_\lambda}\left(([\mathscr{W}, \texttt{cnstr}\,(\{g_j\}_{j\in\mathcal{J}})], A), a_{eval}, (\preceq_A{}^H(Z), A)\right) \quad (4.91)$$

We reach the thesis (4.86) applying the pseudo–transitivity (Lemma 4.5) to the relationships (4.91) and (4.90) setting, for each $Z \subseteq \mathcal{V}_A^{[\mathtt{QC}]\,\ell QST_\lambda}$, $Y = \preceq_A{}^H(Z)$ and recalling the relation $\preceq_A (\preceq_A{}^H(Z)) = \preceq_A{}^H(Z)$ as direct consequence of Lemma 2.17.

◇ If $e = f_1 f_2$, we list the statement as

$$\left(\emptyset \vdash [\mathscr{Q} \otimes \mathscr{U}, f_1 f_2] \preceq_A{}^H [\mathscr{R}, h] : A \wedge \ f_1 f_2 \Downarrow \mathscr{E}\right) \Rightarrow$$
$$\Bigg([\mathscr{R}, h] \Downarrow \mathscr{H} \wedge \forall X \subseteq \mathcal{V}_A^{[\mathtt{QC}]\,\ell QST_\lambda}, \mathcal{P}_{\ell QST_\lambda}(([\mathscr{Q} \otimes \mathscr{U}, f_1 f_2], A), a_{eval}, (X, A)) \leq$$
$$\leq \mathcal{P}_{\ell QST_\lambda}\left(([\mathscr{R}, h], A), a_{eval}, (\preceq_A{}^H(X), A)\right)\Bigg) \quad (4.92)$$

where

$$\mathscr{E} = \textstyle\sum_{[\mathscr{Q}_j, \lambda x.\ell_j] \in \mathtt{Sup}(\mathscr{F}_1), [\mathscr{Q}_j \otimes \mathscr{U}_n, \nu_n] \in \mathtt{Sup}(\mathscr{F}_2)} \mathscr{F}_1([\mathscr{Q}_j, \lambda x.\ell_j]) \mathscr{F}_2([\mathscr{Q}_j \otimes \mathscr{U}_n, \nu_n]) \mathscr{F}_{j,n}$$

as a result of the application of the big step reduction rule for applications in the quantum framework $(app \Downarrow)_{\mathtt{Q}}$. Recalling that first hypothesis must be consequence of a $(How_{gen})_{\mathtt{Q}}$ rule, we know that it has been derived by the

following premises

$$\begin{array}{l}(app:1)- \quad \emptyset \vdash [\mathscr{Q}, f_1] \preceq_{B\multimap A}{}^{H} [\mathscr{S}, g_1] : B \multimap A, \emptyset \vdash [\mathscr{U}, f_2] \preceq_{B}{}^{H} [\mathscr{V}, g_2] : B, \\ (app:2)- \quad \emptyset \vdash [\mathscr{S} \otimes \mathscr{V}, g_1 g_2] \preceq_{A} [\mathscr{R}, h] : A. \end{array} \tag{4.93}$$

Thus we write induction hypotheses on the subterms of $(app:1)$ in (4.93) as

- $$\begin{array}{l}\left(\emptyset \vdash [\mathscr{Q}, f_1] \preceq_{B\multimap A}{}^{H} [\mathscr{S}, g_1] : B \multimap A \ \wedge [\mathscr{Q}, f_1] \Downarrow \mathscr{F}_1\right) \Rightarrow \Big([\mathscr{S}, g_1] \Downarrow \mathscr{G}_1 \wedge \\ \forall X_A \subseteq \mathcal{V}_{B\multimap A}^{[\mathtt{QC}]\,\ell QST_\lambda}, \mathcal{P}_{\ell QST_\lambda}\left(([\mathscr{Q}, f_1], B \multimap A), a_{eval}, (X_A, B \multimap A)\right) \leq \\ \leq \mathcal{P}_{\ell QST_\lambda}\left(([\mathscr{U}, g_1], B \multimap A), a_{eval}, (\preceq_{B\multimap A}{}^{H}(X_A), B \multimap A)\right)\Big) \end{array} \tag{4.94}$$

where we set $\mathscr{F}_1 = \{[\mathscr{Q}_i, \lambda\, x.\ell_i]^{p_i}\}_{i\in\mathcal{I}}$ and $\mathscr{G}_1 = \{[\mathscr{S}_j, \lambda\, x.b_j]^{q_j}\}_{j\in\mathcal{J}}$.

- $$\begin{array}{l}\left(\emptyset \vdash [\mathscr{U}, f_2] \preceq_{B}{}^{H} [\mathscr{V}, g_2] : B \ \wedge \ [\mathscr{U}, f_2] \Downarrow \mathscr{F}_2\right) \ \Rightarrow \ \Big([\mathscr{V}, g_2] \Downarrow \mathscr{G}_2 \\ \wedge \ \forall X_B \subseteq \mathcal{V}_{B}^{[\mathtt{QC}]\,\ell QST_\lambda}, \mathcal{P}_{\ell QST_\lambda}\left(([\mathscr{U}, f_2], B), a_{eval}, (X_B, B)\right) \leq \\ \leq \mathcal{P}_{\ell QST_\lambda}\left(([\mathscr{V}, g_2], B), a_{eval}, (\preceq_{B}{}^{H}(X_B), B)\right)\Big) \end{array} \tag{4.95}$$

with $\mathscr{F}_2 = \{[\mathscr{U}_n, \nu_n]^{p_n}\}_{n\in\mathcal{N}}$ and $\mathscr{G}_2 = \{[\mathscr{V}_m, w_m]^{q_m}\}_{m\in\mathcal{M}}$.

By induction hypotheses, using the big–step semantics rule for application one builds the semantics of term $g_1 g_2$ defined as

$$\mathscr{G} = \textstyle\sum_{[\mathscr{S}_j, \lambda\, x.b_j]\in \mathtt{Sup}(\mathscr{G}_1), [\mathscr{V}_m, w_m]\in \mathtt{Sup}(\mathscr{G}_2)} \mathscr{G}_1([\mathscr{S}_j, \lambda\, x.b_j])\mathscr{G}_2([\mathscr{V}_m, w_m])\mathscr{G}_{j,m},$$

provided to have settled that $b_j\{w_m/x\} \Downarrow \mathscr{G}_{j,m}$.

With regard to relations (4.94) and (4.95), it should be remarked that

$$X_A = \{[\mathscr{Q}_i, \lambda\, x.\ell_i]\}_{i\in\mathcal{I}} \Rightarrow \preceq_{B\multimap A}{}^{H}(X_A) = \cup_{i\in\mathcal{I}}\left(\preceq_{B\multimap A}{}^{H}([\mathscr{Q}_i, \lambda\, x.\ell_i])\right) \tag{4.96}$$

$$X_B = \{[\mathscr{U}_n, \nu_j]\}_{n\in\mathcal{N}} \Rightarrow \preceq_{B}{}^{H}(X_B) = \cup_{n\in\mathcal{N}}\left(\preceq_{B}{}^{H}([\mathscr{U}_n, \nu_n])\right), \tag{4.97}$$

then from now on, we adopt the notation $X'_A = \cup_{i\in\mathcal{I}}\left(\preceq_{B\multimap A}{}^{H}([\mathscr{Q}_i, \lambda\, x.\ell_i])\right)$ and $X'_B = \cup_{n\in\mathcal{N}} \preceq_{B}{}^{H}([\mathscr{U}_n, \nu_n])$.

The disentangling lemma [11] ensures us that for every set pair X_A and X_B there are collections $\{r_1^{X_A} \dots r_N^{X_A}\}$, $\{s_1^{X_B} \dots s_N^{X_B}\}$ of real numbers depending on X_A and X_B respectively, such that

$$\begin{array}{l} \bullet \ \forall X_A \subseteq \mathcal{V}_{B\multimap A}^{[\mathtt{QC}]\,\ell QST_\lambda}, \quad \mathscr{F}_1(X_A) \leq \sum_{[\mathscr{S}_j, \lambda\, x.\ell_j] \in X'_A} r_i^{X_A} \leq \mathscr{G}_1(X'_A) \\ \bullet \ \forall X_B \subseteq \mathcal{V}_B^{[\mathtt{QC}]\,\ell QST_\lambda}, \quad \mathscr{F}_2(X_B) \leq \sum_{[\mathscr{V}_m, \nu_m] \in X'_B} s_j^{X_B^i} \leq \mathscr{G}_2(X'_B) \end{array} \tag{4.98}$$

Moreover, the induction hypothesis (4.95), is employed to compare the distributions $\mathscr{F}_{i,n}$ and $\mathscr{G}_{j,m}$ involved in turn into the definitions of $\mathscr{E}$ and $\mathscr{G}$ respectively: to this purpose, we use the substitutivity of ${\preceq_A}^H$, which has been proved with Lemma 4.6, on the smallest terms of $[\mathscr{Q} \otimes \mathscr{U}, f_1 f_2]$ and $[\mathscr{S} \otimes \mathscr{V}, g_1 g_2]$, to get the following relation, which is fulfilled $\forall\, [\mathscr{Q}_i, \lambda\, x.\ell_j] \in \mathtt{Sup}\,(\mathscr{F}_1)$, $\forall\, [\mathscr{U}_n, \nu_n] \in \mathtt{Sup}\,(\mathscr{F}_2)$, $\forall\, [\mathscr{S}_j, \lambda\, x.b_j] \in \mathtt{Sup}\,(\mathscr{G}_1)$, $\forall\, [\mathscr{V}_m, w_m] \in \mathtt{Sup}\,(\mathscr{G}_2)$:

$$\begin{array}{l} [\mathscr{Q}_i, \lambda\, x.\ell_i] \preceq_{B\multimap A}{}^H [\mathscr{S}_j, \lambda\, x.b_j] \ \wedge \ [\mathscr{U}_n, \nu_n] \preceq_B{}^H [\mathscr{V}_m, w_m] \Rightarrow \\ \Rightarrow \ [\mathscr{Q}_i \otimes \mathscr{U}_n, \ell_i\{\nu_n/x\}] \preceq_A{}^H [\mathscr{S}_j \otimes \mathscr{V}_m, b_j\{w_m/x\}] \ \Rightarrow \mathscr{F}_{i,n} \preceq_A{}^H \mathscr{G}_{j,m}. \end{array} \tag{4.99}$$

Putting together the inequalities (4.98), (4.99) we get

$$\begin{array}{l} \forall X \in \mathcal{V}_A^{[\mathtt{QC}]\,\ell QST_\lambda}, \mathscr{E}(X) = \sum_{\substack{[\mathscr{S}_j, \lambda\, x.\ell_j] \in X_A \\ [\mathscr{V}_m, w_m] \in X_B}} \mathscr{F}_1(X_A)\mathscr{F}_2(X_B)\mathscr{F}_{i,n}(X) \leq \\ \sum_{\substack{[\mathscr{S}_j, \lambda\, x.\ell_j] \in X_A \\ [\mathscr{V}_m, w_m] \in X_B}} r_j^{X_A} s_m^{X_B} \mathscr{G}_{j,m}\left({\preceq_A}^H(X)\right) \leq \mathscr{G}_1(X'_A)\mathscr{G}_2(X'_B)\mathscr{G}_{j,m}\left({\preceq_A}^H(X)\right) \\ \leq \mathscr{G}_1(\mathtt{Sup}\,(\mathscr{G}_1))\mathscr{G}_2(\mathtt{Sup}\,(\mathscr{G}_2))\mathscr{G}_{j,m}\left({\preceq_A}^H(X)\right) = \mathscr{G}\left({\preceq_A}^H(X)\right). \end{array} \tag{4.100}$$

We come back now to the premise $(app : 2)$ in (4.93); denoting by $\mathscr{H}$ the semantics of h and exploiting both of the definition of similarity and the property $\preceq_A \subseteq {\preceq_A}^H$ we must conclude that

$$\forall Y \in \mathcal{V}_A^{[\mathtt{QC}]\,\ell QST_\lambda}, \mathscr{G}(Y) \leq \mathscr{H}\left(\preceq_A (Y)\right) \leq \mathscr{H}\left({\preceq_A}^H(Y)\right). \tag{4.101}$$

Thesis comes joining this last statement (4.101) with (4.100) and setting $Y = {\preceq_A}^H(X)$, thus concluding the proof.

◇ If $e = \texttt{let}\ f_1\ \texttt{be}\ \langle x, y\rangle\ \texttt{in}\ f_2$ the statement

$$\Big(\emptyset \vdash [\mathscr{Q}\otimes\mathscr{U}, \texttt{let}\ f_1\ \texttt{be}\ \langle x,y\rangle\ \texttt{in}\ f_2] \preceq_A{}^H [\mathscr{R}, h] : A \wedge \texttt{let}\ f_1\ \texttt{be}\ \langle x,y\rangle\ \texttt{in}\ f_2 \Downarrow \mathscr{E}\Big) \Rightarrow \Big([\mathscr{R}, h] \Downarrow \mathscr{H} \wedge\ \forall X \subseteq \mathcal{V}_A^{[\mathtt{QC}]\,\ell QST_\lambda},$$
$$\mathcal{P}_{\ell QST_\lambda}\left(([\mathscr{Q}\otimes\mathscr{U}, \texttt{let}\ f_1\ \texttt{be}\ \langle x,y\rangle\ \texttt{in}\ f_2], A), a_{eval}, (X, A)\right) \leq \mathcal{P}_{\ell QST_\lambda}\left(([\mathscr{R}, h], A), a_{eval}, (\preceq_A{}^H(X), A)\right)\Big) \quad (4.102)$$

should be proved, where the semantics $\mathscr{E}$ of the term is obtained via the application of the corresponding big–step semantics rule $(let \Downarrow)_{\mathtt{Q}}$, which states

$$\frac{f_1 \Downarrow \mathscr{F}_1 = \{[\mathscr{Q}_i, \langle v_i, u_i\rangle]^{p_i}\}_{i\in\mathcal{I}} \qquad [\mathscr{Q}_i\otimes\mathscr{U}, f_2\{v_i/x, u_i/y\}] \Downarrow \mathscr{F}_i'}{[\mathscr{Q}\otimes\mathscr{U}, \texttt{let}\ f_1\ \texttt{be}\ \langle x,y\rangle\ \texttt{in}\ f_2] \Downarrow \mathscr{E} = \sum_{i\in\mathcal{I}} p_i\cdot\mathscr{F}_i'}. \quad (4.103)$$

The first statement of the hypothesis originates from the premises

$$\begin{aligned} (let:1)- \quad & \emptyset \vdash [\mathscr{Q}, f_1] \preceq_{E\otimes B}{}^H [\mathscr{S}, g_1] : E\otimes B, \\ & x : E, y : B \vdash [\mathscr{U}, f_2] \preceq_A{}^H [\mathscr{V}, g_2] : A, \\ (let:2)- \quad & \emptyset \vdash [\mathscr{S}\otimes\mathscr{V}, \texttt{let}\ g_1\ \texttt{be}\ \langle x,y\rangle\ \texttt{in}\ g_2] \preceq_A [\mathscr{R}, h] : A. \end{aligned} \quad (4.104)$$

And by induction on the dimensions of term involved in the big–step evaluation rule we find

$$\Big(\emptyset \vdash [\mathscr{Q}, f_1] \preceq_{E\otimes B}{}^H [\mathscr{S}, g_1] : E\otimes B\ \wedge [\mathscr{Q}, f_1] \Downarrow \mathscr{F}_1\Big) \Rightarrow \Big([\mathscr{S}, g_1] \Downarrow \mathscr{G}_1 \wedge$$
$$\wedge\ \forall X \subseteq \mathcal{V}_{E\otimes B}^{[\mathtt{QC}]\,\ell QST_\lambda},\ \mathcal{P}_{\ell QST_\lambda}\left(([\mathscr{Q}, f_1], E\otimes B), a_{eval}, (X, E\otimes B)\right) \leq$$
$$\leq \mathcal{P}_{\ell QST_\lambda}\left(([\mathscr{S}, g_1], E\otimes B), a_{eval}, (\preceq_{B\otimes E}{}^H(X), B\otimes E)\right)\Big), \quad (4.105)$$

with $\mathscr{F}_1 = \{[\mathscr{Q}_i, \langle v_i, u_i\rangle]^{p_i}\}_{i\in\mathcal{I}}$, $\mathscr{G}_1 = \{[\mathscr{S}_j, \langle \nu_j, w_j\rangle]^{q_j}\}_{j\in\mathcal{J}}$.

Moreover, using open extension of applicative bisimulation and induction hypothesis yields

$$x : E, y : B \vdash [\mathscr{U}, f_2] \preceq_A{}^H [\mathscr{V}, g_2] : A\ \Rightarrow \forall\, [\mathscr{W}_n, \langle v_n, u_n\rangle] \in \mathcal{V}_{E\otimes B}^{[\mathtt{QC}]\,\ell QST_\lambda},$$
$$\forall Y \in \mathcal{V}_A^{[\mathtt{QC}]\,\ell QST_\lambda} \mathcal{P}_{\ell QST_\lambda}\left(([\mathscr{Q}\otimes\mathscr{W}_n, f_2\{v_n/x, u_n/y\}], A), a_{eval}, (Y, A)\right) \leq$$

$$\leq \mathcal{P}_{\ell QST_\lambda}\left(([\mathscr{S}\otimes\mathscr{W}_n, g_2\{v_n/x, u_n/y\}], A), a_{eval}, ({\preceq_A}^H(Y), A)\right). \quad (4.106)$$

The previous induction hypothesis (4.106) uses the distribution $\mathscr{G}_1$ and the set of distributions $\{\mathscr{G}_j'\}_{j\in\mathcal{J}}$ whose generic element $\mathscr{G}_j'$ is the result of the evaluation of the quantum clusure $[\mathscr{S}\otimes\mathscr{W}_j, g_2\{v_j/x, u_j/y\}]$. This allows to write explicitly the structure of the distribution $\mathscr{G}$ to which the term $\texttt{let}\, g_1\, \texttt{be}\, \langle x,y\rangle\, \texttt{in}\, g_2$ evaluates, being $\mathscr{G} = \sum_{[\mathscr{Q}_i,\langle\nu_i,w_i\rangle]\in\texttt{Sup}(\mathscr{G}_1)} \mathscr{G}_1([\mathscr{W}_j, \langle\nu_j, w_j\rangle])\cdot\mathscr{G}_j'$ whence, starting from hypothesis of similarity contained on the second condition of (4.104) one finds

$$\Big(\emptyset\vdash[\mathscr{S}\otimes\mathscr{V}, \texttt{let}\ g_1\ \texttt{be}\ \langle x,y\rangle\ \texttt{in}\ g_2] \preceq_A [\mathscr{R}, h] : A \wedge$$
$$[\mathscr{S}\otimes\mathscr{V}, \texttt{let}\ g_1\ \texttt{be}\ \langle x,y\rangle\ \texttt{in}\ g_2] \Downarrow \mathscr{G}\Big) \Rightarrow \Big([\mathscr{R}, h]\Downarrow\mathscr{H} \wedge \forall Z\subseteq \mathcal{V}_A^{[\mathtt{QC}]\,\ell QST_\lambda},$$
$$\mathscr{G}(Z)\leq\mathscr{H}(\preceq_A(Z))\Big). \quad (4.107)$$

Now we own all the elements to compare the semantics $\mathscr{E}$, $\mathscr{F}$ and $\mathscr{G}$: indeed the first point of induction hypothesis (4.106) ensures that $\forall X\subseteq \mathcal{V}_{E\otimes B}^{[\mathtt{QC}]\,\ell QST_\lambda}$, $\mathscr{F}_1(X)\leq\mathscr{G}({\preceq_{E\otimes B}}^H(X))$ and the second points together with substitution property (Lemma 4.6) suggest us that $\forall\langle v_i,u_i\rangle\in\mathcal{V}_{E\otimes A}^{\ell PST_\lambda}$, $\forall Y\subseteq\mathcal{V}_A^{[\mathtt{QC}]\,\ell QST_\lambda}$, $\mathscr{F}_i'(Y)\leq\mathscr{G}_i'({\preceq_A}^H(Y))$, thus

$$\forall W\subseteq\mathcal{V}_A^{[\mathtt{QC}]\,\ell QST_\lambda},\ \mathscr{E}(W)=\sum_{[\mathscr{Q}_i,\langle v_i,u_i\rangle]\in X}\mathscr{F}_1([\mathscr{Q}_i,\langle v_i,u_i\rangle])\mathscr{F}_i'(W)\leq$$
$$\sum_{[\mathscr{Q}_i,\langle v_i,u_i\rangle]\in{\preceq_{E\otimes B}}^H(X)}\mathscr{G}_1([\mathscr{Q}_i,\langle v_i,u_i\rangle])\mathscr{G}_i'({\preceq_E}^H(W))\leq\mathscr{G}({\preceq_A}^H(W)). \quad (4.108)$$

Now recall the last result (4.108) $\forall W\subseteq\mathcal{V}_A^{[\mathtt{QC}]\,\ell QST_\lambda}$, $\mathscr{E}(W)\leq\mathscr{G}({\preceq_E}^H(W))$ and let us join it with (4.107), namely $\forall Z\subseteq\mathcal{V}_A^{[\mathtt{QC}]\,\ell QST_\lambda}$, $\mathscr{G}(Z)\leq\mathscr{H}(\preceq_A(Z))$ simply setting $\forall W, Z = {\preceq_A}^H(W)$, to obtain $\forall W\subseteq\mathcal{T}_A^{[\mathtt{QC}]\,\ell QST_\lambda}$, $\mathscr{E}(W)\leq\mathscr{G}({\preceq_E}^H(W))\leq\mathscr{H}\left(\preceq_A({\preceq_A}^H(W))\right)$. The thesis is proved, since we resort to the well known relation $\preceq_A\subseteq{\preceq_A}^H$ (Lemma 2.17), which entails that $\forall W$, $\preceq_A({\preceq_A}^H(W)) = {\preceq_A}^H(W)$

□

By carrying in quantum environment all the properties already stated in Proposition 3.2 and following, until Proposition 3.7, we are reassured about the reflexivity, transitivity and closure under substitution of the relation $(\preceq^H)^+$ on the set $\mathcal{T}^{[\mathtt{QC}]\,\ell QST_\lambda}$. Finally, also Lemma 3.13 is still valid since it is proved basing only on the properties of the relation $(\preceq^H)^+$, telling us that the transitive closure of Howe's extension of similarity has the simulation relation property – as we just proved that the simulation on ℓQST_λ owns it – thus it is a simulation in turn. From here it follows the

Theorem 4.1. *In ℓQST_λ, $\preceq$ is included in $\leq$, thus $\sim$ is included in $\equiv$.*

Example 4.1. *A*n interesting pair of terms which can be proved bisimilar are the following two:

$$e = \lambda\, x.\mathtt{if}\ \mathtt{meas}_1\,(x)\ \mathtt{then}\ \mathtt{ff}\ \mathtt{else}\ \mathtt{tt}; \qquad f = \lambda\, x.\mathtt{meas}_1\,(X_U(x));$$

where X_U is the unitary operator which flips the value of a qubit. From the derivation rules we get

$$\dfrac{\dfrac{\dfrac{x : \mathtt{qbit} \vdash x : \mathtt{qbit}}{x : \mathtt{qbit} \vdash \mathtt{meas}_1\,(x) : \mathtt{bool}} \quad \emptyset \vdash \mathtt{tt} : \mathtt{bool} \quad \emptyset \vdash \mathtt{ff} : \mathtt{bool}}{x : \mathtt{qbit} \vdash \mathtt{if}\ \mathtt{meas}_1\,(x)\ \mathtt{then}\ \mathtt{tt}\ \mathtt{else}\ \mathtt{ff} : \mathtt{bool}}}{\emptyset \vdash \lambda\, x.\mathtt{if}\ \mathtt{meas}_1\,(x)\ \mathtt{then}\ \mathtt{ff}\ \mathtt{else}\ \mathtt{tt} : \mathtt{qbit} \multimap \mathtt{bool}}$$

$$\dfrac{\dfrac{\dfrac{x : \mathtt{qbit} \vdash x : \mathtt{qbit}}{x : \mathtt{qbit} \vdash U\,(x) : \mathtt{qbit}}}{x : \mathtt{qbit} \vdash \mathtt{meas}_1\,(U\,(x)) : \mathtt{bool}}}{\emptyset \vdash \lambda\, x.\mathtt{meas}_1\,(U\,(x)) : \mathtt{qbit} \multimap \mathtt{bool}}$$

Whence, by definition (4.42)

$$e\,\mathcal{B}_{\mathtt{qbit}\multimap\mathtt{bool}}\,f \Rightarrow\ \forall\,[\mathscr{U}, v] \in \mathcal{V}^{[\mathtt{QC}]\,\ell QST_\lambda}_{\mathtt{qbit}}$$
$$[\mathscr{U}, \mathtt{if}\ \mathtt{meas}_1\{v/x\}\ \mathtt{then}\ \mathtt{ff}\ \mathtt{else}\ \mathtt{tt}]\ \mathcal{B}_{\mathtt{bool}}\ [X_U\mathscr{U}, \mathtt{meas}_1\{v/x\}]\,,$$

and both these terms evalutate to the same distribution, namely $\{\mathtt{tt}^{\mathtt{PR}^v_{\mathtt{ff}}(\mathscr{U})}, \mathtt{ff}^{\mathtt{PR}^v_{\mathtt{tt}}(\mathscr{U})}\}$.

□

4.3 On Full-Abstraction

Given the strong analogies between the probabilistic and quantum version of ℓST_λ, there is a little hope to recover full abstraction for bisimulation in the environment of the context equivalence for ℓQST_λ, since we didn't obtain it for ℓPST_λ. Indeed we can build a couple of terms, modelling them on the example given in ℓST_λ, which are not bisimilar for the same reason, but can be proved to be context equivalent using trace equivalence techniques. Consider, in ℓQST_λ, the terms

$$e = \texttt{if}\ \texttt{meas}_1\,(r)\ \texttt{then}\ (\lambda\, x.\texttt{tt})\ \texttt{else}\ (\lambda\, x.\Omega) \tag{4.109}$$

$$f = \lambda\, x.\texttt{if}\ \texttt{meas}_1\,(r)\ \texttt{then}\ \texttt{tt}\ \texttt{else}\ \texttt{ff} \tag{4.110}$$

equiped by the same quantum register, which can be taken, e.g., in the form

$$\mathscr{Q} = \frac{1}{\sqrt{2}}\left(|r \leftarrow \texttt{tt}\rangle + |r \leftarrow \texttt{ff}\rangle\right).$$

The terms have both type $\texttt{bool} \multimap \texttt{bool}$, and by analogy with the example that we gave in ℓST_λ, we claim that the quantum closures $[\mathscr{Q}, e]$ and $[\mathscr{Q}, f]$ are not bisimilar since they evolve under th action of the same labelled Marcov chain $\mathscr{M}_{\ell QST_\lambda}$ to a different distribution of values. On the other hand, being trace equivalent, they are also context equivalent.

Rececently, an attempt to overcome this problem and recover completeness also in the quantum language has been done, [18] giving a new notion of bisimilarity based on the distributions rather than on the terms.

What one may hope to get is full-abstraction for extensions of the considered calculi in which duplication is reintroduced, although in a controlled way. This has been recently done in a probabilistic setting by Crubillé and Dal Lago [10], and is the topic of current investigations by the authors for a quantum calculus in the style of ℓQST_λ.

4.4 Conclusions

In the literature, various attempts have been made, to endow quantum languages with a denotational semantic. Many of them exploit the game theory methods, which has been formerly implemented [3] in PCF , where the types of PCF are interpreted as games and the terms of language as game strategies.

Abramsky and Coecke [2] developed a pattern where the terms, the techniques and also the typical algorithms of quantum computing, such as the quantum teleportation protocols, are interpreted using categories theory. They focused their analysis on quantum information protocols, describing phenomena such as quantum teleportation and entanglement swapping, through the use of compact and closed categories, and formalized the superposition as well as the uncertainty intrinsically embedded in the measurement processes which characterize the systems of qubits, with a biproduct structure.

Delbecque [15, 16] developed a model closest to that we deal: based on the calculus conceived by Selinger and Valiron, he conducts its analysis on the first-order non linear languages and he presents a typed lambda calculus in which the typical structures of the quantum calculation are represented using the concepts of the game theory [13] for the probabilistic calculation. In this scheme, the quantum states and the quantum operations are represented as strategies of a game. The aim is to build a denotational semantics for the language terms, which nevertheless apparently doesn't take into account the influence of the quantum register on the behavior of the language objects.

Also Hasuo and Hoshino [29], based on the paradigm Selinger and Valiron of the quantum language with classic controls, give a denotational semantic for a quantum lambda caluculus. They use the category theory in order to equip the quantum calculus with a denotational semantics, with the purpose to foster the development of a tool for the analysis of the correctness for algorithms and protocols of quantum communications, which are often uniuntuitive. Whithin their analysis, employing the concept of monads [5] for structuring the branching, they also overcome the

hypothesis of linearity by introducing the mode !, for deletion and duplication of variables.

Another interesting, but complementary research branch focuses on the fascinating topic to implement a quantum functional language in giving a set of instruction for the creation and the manipulation of quantum data, building this way a tool to write the quantum algorithms, which have been presented untill now in term of quantum gates, as a sequence of instructions. Despite that this language [26, 27] is provided with an operational semantic, the authors have not dealt with the problem of the equivalence between terms.

Nevertheless, here we have shown a method to extend Abramsky's applicative bisimulation to a linear $\lambda-$calculi, adapting the Howe's techinque, expressly though for higher–order languages, to a simple grammar endowed with probabilistic choice and quantum data.

For the sake of simplicity, we have deliberately kept the considered calculi as simple as possible. We believe, however, that many extensions would be harmless. This includes, as an example, generalizing types to *recursive* types which, although infinitary in nature, can be dealt with very easily in a coinductive setting. Adding a form of controlled duplication requires more care, e.g. in presence of quantum data (which cannot be duplicated).

As a future perspective, we are also working for exploring another strengthened technique for comparison among terms, namely the trace equivalence, with the purpose to show that for quantum ℓQST_{λ}, trace equivalence coincides with context equivalence [19].

References

[1] S. Abramsky. The Lazy λ-Calculus. In D. Turner, editor, *Research Topics in Functional Programming*, pages 65–117. Addison Wesley, 1990.

[2] Samson Abramsky and Bob Coecke. A categorical semantics of quantum protocols. *CoRR*, quant-ph/0402130, 2004.

[3] Samson Abramsky, Radha Jagadeesan, and Pasquale Jagadeesan. Full abstraction for {PCF}. *Information and Computation*, 163(2):409 – 470, 2000.

[4] Samson Abramsky and C.-H. Luke Ong. Full abstraction in the lazy lambda calculus. *Inf. Comput.*, 105(2):159–267, 1993.

[5] Thorsten Altenkirch and Alexander S. Green. The Quantum IO Monad. In Simon Gay and Ian Mackie, editors, *Semantic Techniques in Quantum Computation*. Cambridge University Press, 2010.

[6] Ebrahim Ardeshir-Larijani, Simon J. Gay, and Rajagopal Nagarajan. Equivalence checking of quantum protocols. In *Tools and Algorithms for the Construction and Analysis of Systems - 19th International Conference, TACAS 2013, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2013, Rome, Italy, March 16-24, 2013. Proceedings*, pages 478–492, 2013.

[7] Henk Barendregt and Eric Barendsen. *Introduction to Lambda Calculus*. http://ftp.cs.ru.nl/CompMath.Found/lambda.pdf, 2000.

[8] Gavin M. Bierman. Program equivalence in a linear functional language. *J. Funct. Program.*, 10(2):167–190, 2000.

[9] Roy L. Crole. Completeness of bisimilarity for contextual equivalence in linear theories. *Electronic Journal of the IGPL*, 9(1), January 2001.

[10] Raphaëlle Crubillé and Ugo Dal Lago. On probabilistic applicative bisimulation and call-by-value -calculi. In *ESOP*, pages 209–228, 2014.

[11] Ugo Dal Lago and Alessandro Rioli. Applicative bisimulation and quantum λ-calculi (long version). Available at `http://eternal.cs.unibo.it/abqlc.pdf`, 2015.

[12] Ugo Dal Lago, Davide Sangiorgi, and Michele Alberti. On coinductive equivalences for higher-order probabilistic functional programs. In *POPL*, pages 297–308, 2014.

[13] Vincent Danos and Russell Harmer. Probabilistic game semantics. *ACM Trans. Comput. Log.*, 3(3):359–382, 2002.

[14] Timothy A. S. Davidson, Simon J. Gay, Hynek Mlnarik, Rajagopal Nagarajan, and Nick Papanikolaou. Model checking for communicating quantum processes. *IJUC*, 8(1):73–98, 2012.

[15] Yannick Delbecque. A quantum game semantics for the measurement calculus. *Electr. Notes Theor. Comput. Sci.*, 210:33–48, 2008.

[16] Yannick Delbecque. Game semantics for quantum data. *Electr. Notes Theor. Comput. Sci.*, 270(1):41–57, 2011.

[17] Yuxin Deng and Yuan Feng. Open bisimulation for quantum processes. *CoRR*, abs/1201.0416, 2012.

[18] Yuxin Deng, Yuan Feng, and Ugo Dal Lago. On coinduction and quantum lambda calculi. In *Proceedings of the 26th International Conference on Concur-*

rency Theory, volume 42 of *LIPIcs*, pages 427–440. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2015.

[19] Yuxin Deng and Yu Zhang. Program equivalence in linear contexts. *Theor. Comput. Sci.*, 585:71–90, 2015.

[20] David Deutsch. Quantum theory, the church-turing principle and the universal quantum computer. *Proceedings of the Royal Society A*, VoL 400(1818):97–117, 1985.

[21] Yuan Feng, Runyao Duan, and Mingsheng Ying. Bisimulation for quantum processes. *ACM Trans. Program. Lang. Syst.*, 34(4):17, 2012.

[22] Richard P. Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, VoL 21:467–488, 1982.

[23] Simon J. Gay and Rajagopal Nagarajan. Communicating quantum processes. In *POPL*, pages 145–157, 2005.

[24] Jean-Yves Girard. Linear logic. *Theor. Comput. Sci.*, 50:1–102, 1987.

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

[26] Alexander Green, Peter LeFanu Lumsdaine, Neil J. Ross, Peter Selinger, and Benoît Valiron. An introduction to quantum programming in Quipper. In *Proceedings of the 5th International Conference on Reversible Computation, RC 2013, Victoria, British Columbia*, volume 7948 of *Lecture Notes in Computer Science*, pages 110–124. Springer, 2013.

[27] Alexander S. Green, Peter LeFanu Lumsdaine, Neil J. Ross, Peter Selinger, and Benoît Valiron. Quipper: A scalable quantum programming language. *CoRR*, abs/1304.3390, 2013.

[28] Lov K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing*, STOC '96, pages 212–219, New York, NY, USA, 1996. ACM.

[29] Ichiro Hasuo and Naohiko Hoshino. Semantics of higher-order quantum computation via geometry of interaction. In *Proceedings of the 26th Annual IEEE Symposium on Logic in Computer Science, LICS 2011, June 21-24, 2011, Toronto, Ontario, Canada*, pages 237–246, 2011.

[30] Douglas J. Howe. Proving congruence of bisimulation in functional programming languages. *Inf. Comput.*, 124(2):103–112, 1996.

[31] Bart Jacobs. Coalgebraic walks, in quantum and turing computation. In *FOSSACS*, pages 12–26, 2011.

[32] Wootters W. K. and Zurek W. H. A single quantum cannot be cloned. *Nature*, 299:802–803, 1982.

[33] Vasileios Koutavas, Paul Blain Levy, and Eijiro Sumii. From applicative to environmental bisimulation. *Electr. Notes Theor. Comput. Sci.*, 276:215–235, 2011.

[34] Ugo Dal Lago, Andrea Masini, and Margherita Zorzi. On a measurement-free quantum lambda calculus with classical control. *Mathematical Structures in Computer Science*, 19(2):297–335, 2009.

[35] Ugo Dal Lago, Andrea Masini, and Margherita Zorzi. Confluence results for a quantum lambda calculus with measurements. *Electr. Notes Theor. Comput. Sci.*, 270(2):251–261, 2011.

[36] Ugo Dal Lago and Alessandro Rioli. Applicative bisimulation and quantum λ-calculi. In *Fundamentals of Software Engineering - 6th International Conference, FSEN 2015 Tehran, Iran, April 22-24, 2015, Revised Selected Papers*, pages 54–68, 2015.

[37] Ugo Dal Lago, Davide Sangiorgi, and Michele Alberti. On coinductive equivalences for higher-order probabilistic functional programs (long version). *CoRR*, abs/1311.1722, 2013.

[38] Ugo Dal Lago and Margherita Zorzi. Probabilistic operational semantics for the lambda calculus. *CoRR*, abs/1104.0195, 2011.

[39] Kim Guldstrand Larsen and Arne Skou. Bisimulation through probabilistic testing. *Inf. Comput.*, 94(1):1–28, 1991.

[40] Søren B. Lassen and Corin Pitcher. Similarity and bisimilarity for countable non-determinism and higher-order functions. *Electr. Notes Theor. Comput. Sci.*, 10:246–266, 1997.

[41] Robin Milner. Fully abstract models of typed λ calculi. *Theoretical Computer Science*, 4:1–22, 1977.

[42] J. Morris. *Lambda Calculus Models of Programming Languages.* PhD thesis, MIT, 1969.

[43] Michael Nielsen and Isaac Chuang. *Quantum Computation and Quantum Information.* Cambridge Universtity Press, 2010.

[44] Andrew M. Pitts. Operationally-based theories of program equivalence. In *Semantics and Logics of Computation*, pages 241–298. Cambridge University Press, 1997.

[45] G Plotkin. Lambda definability and logical relations. In *Memo SAI-RM-4, School of Artificial Intelligence.* Edinburgh, 1973.

[46] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public–key cryptosystems. *Commun. ACM*, 21(2):120–126, February 1978.

[47] Davide Sangiorgi. On the origins of bisimulation and coinduction. *ACM Trans. Program. Lang. Syst.*, 31(4):15:1–15:41, May 2009.

[48] Davide Sangiorgi. *Introduction to Bisimulation and Coinduction.* Cambridge Universtity Press, 2012.

[49] Davide Sangiorgi, Naoki Kobayashi, and Eijiro Sumii. Environmental bisimulations for higher-order languages. *ACM Trans. Program. Lang. Syst.*, 33(1):5, 2011.

[50] Peter Selinger. Towards a quantum programming language. *Mathematical Structures in Computer Science*, 14(4):527–586, 2004.

[51] Peter Selinger and Benoît Valiron. A lambda calculus for quantum computation with classical control. In *TLCA*, pages 354–368, 2005.

[52] Peter Selinger and Benoît Valiron. On a fully abstract model for a quantum linear functional language. *Electron. Notes Theor. Comput. Sci.*, 210:123–137, 2008.

[53] Peter Selinger and Benoît Valiron. Quantum lambda calculus. In Simon Gay and Ian Mackie, editors, *Semantic Techniques in Quantum Computation*, chapter 4, pages 135–172. Cambridge University Press, 2009.

[54] Zhong Shao, editor. *Programming Languages and Systems - 23rd European Symposium on Programming, ESOP 2014, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2014, Grenoble, France, April 5-13, 2014, Proceedings*, volume 8410 of *Lecture Notes in Computer Science*. Springer, 2014.

[55] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.*, 26(5):1484–1509, October 1997.

[56] Franck van Breugel, Michael W. Mislove, Joël Ouaknine, and James Worrell. Domain theory, testing and simulation for labelled markov processes. *Theor. Comput. Sci.*, 333(1-2):171–197, 2005.